Risk journals

Volume 20 Number 1
September 2016

■ **Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous-time mean–variance asset allocation under stochastic volatility**
K. Ma and P. A. Forsyth

■ **High-performance American option pricing**
Leif Andersen, Mark Lake and Dimitri Offengenden

■ **Adjusting exponential Lévy models toward the simultaneous calibration of market prices for crash cliquets**
Peter Carr, Ajay Khanna and Dilip B. Madan

■ **An exact and efficient method for computing cross-Gammas of Bermudan swaptions and cancelable swaps under the Libor market model**
Mark S. Joshi and Dan Zhu

■ **Efficient computation of exposure profiles on real-world and risk-neutral scenarios for Bermudan swaptions**
Qian Feng, Shashi Jain, Patrik Karlsson, Drona Kandhai and Cornelis W. Oosterlee

The Journal of Computational Finance

For all subscription queries, please call:
UK/Europe: +44 (0) 207 316 9300
USA: +1 646 736 1850 ROW: +852 3411 4828

Risk.net in numbers

200+
New articles & technical papers
(each month)

140,000
Users
(each month)

370,000
Page views
(each month)

6,500+ on Asset Management

6,900+ on Commodities

£21,000+ on Derivatives

19,400+ on Regulation

19,600+ on Risk Management

The Journal of Computational Finance

EDITORIAL BOARD

SUBSCRIPTIONS

The Journal of Computational Finance (Print ISSN 1460-1559 | Online ISSN 1755-2850) is published quarterly by Incisive Risk Information Limited, Haymarket House, 28–29 Haymarket, London SW1Y 4RX, UK. Subscriptions are available on an annual basis, and the rates are set out in the table below.

	UK	Europe	US
Risk.net Journals	£1945	€2795	$3095
Print	£735	€1035	$1215
Risk.net Premium	£2750	€3995	$4400

Academic discounts are available. Please enquire by using one of the contact methods below.

All prices include postage. All subscription orders, single/back issues orders, and changes of address should be sent to:

UK & Europe Office: Incisive Media (c/o CDS Global), Tower House, Sovereign Park, Market Harborough, Leicestershire, LE16 9EF, UK. Tel: 0870 787 6822 (UK), +44 (0)1858 438 421 (ROW); fax: +44 (0)1858 434958

US & Canada Office: Incisive Media, 55 Broad Street, 22nd Floor, New York, NY 10004, USA. Tel: +1 646 736 1888; fax: +1 646 390 6612

Asia & Pacific Office: Incisive Media, 20th Floor, Admiralty Centre, Tower 2, 18 Harcourt Road, Admiralty, Hong Kong. Tel: +852 3411 4888; fax: +852 3411 4811

Website: www.risk.net/journal **E-mail:** incisivehv@subscription.co.uk

The Journal of Computational Finance

GENERAL SUBMISSION GUIDELINES

Manuscripts and research papers submitted for consideration must be original work that is not simultaneously under review for publication in another journal or other publication outlets. All articles submitted for consideration should follow strict academic standards in both theoretical content and empirical results. Articles should be of interest to a broad audience of sophisticated practitioners and academics.

Submitted papers should follow *Webster's New Collegiate Dictionary* for spelling, and *The Chicago Manual of Style* for punctuation and other points of style, apart from a few minor exceptions that can be found at www.risk.net/journal. Papers should be submitted electronically via email to: journals@incisivemedia.com. Please clearly indicate which journal you are submitting to.

You must submit two versions of your paper; a single LaTeX version and a PDF file. LaTeX files need to have an explicitly coded bibliography included. All files must be clearly named and saved by author name and date of submission. All figures and tables must be included in the main PDF document and also submitted as separate editable files and be clearly numbered.

All papers should include a title page as a separate document, and the full names, affiliations and email addresses of all authors should be included. A concise and factual abstract of between 150 and 200 words is required and it should be included in the main document. Five or six keywords should be included after the abstract. Submitted papers must also include an Acknowledgements section and a Declaration of Interest section. Authors should declare any funding for the article or conflicts of interest. Citations in the text must be written as (John 1999; Paul 2003; Peter and Paul 2000) or (John *et al* 1993; Peter 2000).

The number of figures and tables included in a paper should be kept to a minimum. Figures and tables must be included in the main PDF document and also submitted as separate individual editable files. Figures will appear in color online, but will be printed in black and white. Footnotes should be used sparingly. If footnotes are required then these should be included at the end of the page and should be no more than two sentences. Appendixes will be published online as supplementary material.

Before submitting a paper, authors should consult the full author guidelines at:

http://www.risk.net/static/risk-journals-submission-guidelines

Queries may also be sent to:

The Journal of Computational Finance, Incisive Media,
Haymarket House, 28–29 Haymarket, London SW1Y 4RX, UK
Tel: +44 (0)20 7004 7531; Fax: +44 (0)20 7484 9758
E-mail: journals@incisivemedia.com

The Journal of
Computational Finance

The journal

The Journal of Computational Finance welcomes papers dealing with innovative computational techniques in the following areas.

- Numerical solutions of pricing equations: finite differences, finite elements, and spectral techniques in one and multiple dimensions.
- Simulation approaches in pricing and risk management: advances in Monte Carlo and quasi-Monte Carlo methodologies; new strategies for market factors simulation.
- Optimization techniques in hedging and risk management.
- Fundamental numerical analysis relevant to finance: effect of boundary treatments on accuracy; new discretization of time-series analysis.
- Developments in free-boundary problems in finance: alternative ways and numerical implications in American option pricing.

CONTENTS

Letter from the Editor-in-Chief vii

RESEARCH PAPERS

Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous-time mean–variance asset allocation under stochastic volatility 1
K. Ma and P. A. Forsyth

High-performance American option pricing 39
Leif Andersen, Mark Lake and Dimitri Offengenden

Adjusting exponential Lévy models toward the simultaneous calibration of market prices for crash cliquets 89
Peter Carr, Ajay Khanna and Dilip B. Madan

An exact and efficient method for computing cross-Gammas of Bermudan swaptions and cancelable swaps under the Libor market model 113
Mark S. Joshi and Dan Zhu

Efficient computation of exposure profiles on real-world and risk-neutral scenarios for Bermudan swaptions 139
Qian Feng, Shashi Jain, Patrik Karlsson, Drona Kandhai and Cornelis W. Oosterlee

Editor-in-Chief: Cornelis W. Oosterlee
Publisher: Nick Carver
Journals Manager: Dawn Hunter
Editorial Assistant: Carolyn Moclair
Marketing Manager: Ranvinder Gill
Subscription Sales Manager: Aaraa Javed
Global Head of Sales: Michael Lloyd
Global Key Account Sales Director: Michelle Godwin
Composition and copyediting: T&T Productions Ltd
Printed in UK by Printondemand-Worldwide

LETTER FROM THE EDITOR-IN-CHIEF

Cornelis W. Oosterlee
CWI – Dutch Center for Mathematics and Computer Science, Amsterdam

With this issue of *The Journal of Computational Finance*, we have something to celebrate. This is the first issue of the twentieth volume, and we are proud to have reached this milestone. Over the past nineteen volumes, developing advanced robust numerical techniques, accurate solutions and modern scientific computing in the field of financial engineering and financial risk management has become a very prominent area of research. Whereas risk management is ever present nowadays – with all its "valuation adjustments" – the pricing of financial derivatives by means of different techniques and numerical methods for optimal portfolio selection also remains important.

In this celebratory issue we present some recent, novel papers by researchers who have been frequent contributors to and strong supporters of our journal: we have an associate editor, a former editor-in-chief and the current one, and other authors who have published some of their most influential papers in our journal in recent years.

Peter Forsyth, our former editor-in-chief, is a co-author, with Kai Ma, of the first paper in the issue: "Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous-time mean–variance asset allocation under stochastic volatility". The paper deals with robust and accurate numerical solution methods for the nonlinear Hamilton–Jacobi–Bellman partial differential equation (PDE), which describes the dynamic optimal portfolio selection problem. An example of an advanced numerical PDE technique for a relevant problem is given, along with a proof of the convergence of the discrete scheme.

In the author list of the issue's second paper, "High-performance American option pricing", we find Leif Andersen, one of our associate editors and the author of a much-cited 2006 paper on the quadratic exponential Monte Carlo scheme for the Heston model. With his co-authors Mark Lake and Dimitri Offengenden, he presents a paper on a high-performance spectral collocation method for the computation of American put and call option prices. They achieve an enormous speed-up when pricing a large number of American style options, under Black–Scholes dynamics. The computational throughput of the algorithm is close to 100 000 option prices per second per CPU. The research question in this paper is often encountered in industrial banking practice.

Our third paper is by Peter Carr and Dilip Madan, who have been prominent authors in modeling and computation in financial applications for many years. In 1999 they co-authored the influential *Journal of Computational Finance* paper "Option valuation

using the fast Fourier transform". Their paper here, "Adjusting exponential Lévy models toward the simultaneous calibration of market prices for crash cliquets", is written with Ajay Khanna. Based on the insight that a variety of exponential Lévy models, when calibrated to near at-the-money option prices, typically overprice crash cliquets products, the authors propose so-called tail thinning strategies that may be employed to better connect the calibrated models to the crash cliquets prices.

The next paper in the issue, "An exact and efficient method for computing cross-Gammas of Bermudan swaptions and cancelable swaps under the Libor market model", is by Mark Joshi, who has contributed several exciting papers to our journal's success over the years, and Dan Zhu. A new simulation algorithm for computing the Hessians of Bermudan swaptions and cancelable swaps is presented, for which the resulting pathwise estimates are accurate and unbiased. A measure change, which is selected so that the variance of the likelihood ratio part is minimized at each exercise point, is performed to ensure that the first-order derivatives of the pathwise estimates of the price are continuous, resulting in an accurate and efficient simulation scheme.

The final paper in this special anniversary issue, also on Bermudan swaptions and called "Efficient computation of exposure profiles on real-world and risk-neutral scenarios for Bermudan swaptions", is by myself along with Qian Feng, Shashi Jain, Patrik Karlsson and Drona Kandhai. In the paper, real-world and risk-neutral scenarios are combined for the valuation of the exposure values of Bermudan swaptions on real-world Monte Carlo paths. Highly accurate and efficient risk management quantities like expected exposure and potential future exposure are computed, for which the risk-neutral and real-world scenarios need to be combined. Based on the stochastic grid bundling method, a robust regression-based Monte Carlo technique, it is possible to avoid nested Monte Carlo simulations.

At the time of writing this editorial, the referendum result has brought us drastic shifts in stock prices, currencies, commodities like gold, etc. The need for efficient numerical and computational techniques in risk management and in financial derivatives pricing will be ever higher. The future of computational finance is bright. While we celebrate reaching our twentieth volume, we wish you very enjoyable reading of this issue of *The Journal of Computational Finance*.

Journal of Computational Finance **20**(1), 1–37
DOI: 10.21314/JCF.2016.310

Risk journals

Research Paper

Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous-time mean–variance asset allocation under stochastic volatility

K. Ma and P. A. Forsyth

Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada; emails: k26ma@uwaterloo.ca, paforsyt@uwaterloo.ca

(Received May 19, 2015; revised July 15, 2015; accepted August 20, 2015)

ABSTRACT

We present efficient partial differential equation (PDE) methods for continuous-time mean–variance portfolio allocation problems when the underlying risky asset follows a stochastic volatility process. The standard formulation for mean–variance optimal portfolio allocation problems gives rise to a two-dimensional nonlinear Hamilton–Jacobi–Bellman (HJB) PDE. We use a wide stencil method based on a local coordinate rotation to construct a monotone scheme. Further, by using a semi-Lagrangian time-stepping method to discretize the drift term, along with an improved linear interpolation method, accurate efficient frontiers are constructed. This scheme can be shown to be convergent to the viscosity solution of the HJB equation, and the correctness of the proposed numerical framework is verified by numerical examples. We also discuss the effects on the efficient frontier of the stochastic volatility model parameters.

Keywords: mean–variance; embedding; Pareto optimal; Hamilton–Jacobi–Bellman (HJB) equation; monotone scheme; wide stencil.

Corresponding author: P. A. Forsyth

Print ISSN 1460-1559 | Online ISSN 1755-2850

1 INTRODUCTION

Consider the following prototypical asset allocation problem: an investor can choose to invest in a risk-free bond, or a risky asset, and can dynamically allocate wealth between the two assets, to achieve a predetermined criteria for the portfolio over a long time horizon. In the continuous-time mean–variance approach, risk is quantified by variance, so investors aim to maximize the expected return of their portfolios, given a risk level. Alternatively, they aim to minimize the risk level, given an expected return. As a result, mean–variance strategies are appealing due to their intuitive nature, since the results can be easily interpreted in terms of the trade-off between the risk and the expected return.

In the case where the asset follows a geometric Brownian motion (GBM), there is considerable literature on the topic (Bielecki *et al* 2005; Li and Ng 2000; Wang and Forsyth 2010; Zhou and Li 2000). The multi-period optimal strategy adopted in these papers is of pre-commitment type, which is not time consistent, as noted in Bjork and Murgoci (2010) and Basak and Chabakauri (2010). A comparison between time-consistent and pre-commitment strategies for continuous-time mean–variance optimization is given in Wang and Forsyth (2012). We note that since a time-consistent strategy can be constructed from a pre-commitment policy by adding a constraint (Wang and Forsyth 2012), the time-consistent strategy is suboptimal compared with the pre-commitment policy, ie, it is costly to enforce time consistency. In addition, it has been shown in Vigna (2014) that pre-commitment strategies can also be viewed as a target-based optimization, which involves minimizing a quadratic loss function. It is suggested in Vigna (2014) that this is intuitive, adaptable to investor preferences and also mean–variance efficient.

Most previous literature on pre-commitment mean–variance optimal asset allocation has been based on analytic techniques (Bielecki *et al* 2005; Li and Ng 2000; Nguyen and Portait 2002; Zhao and Ziemba 2000; Zhou and Li 2000). These papers have primarily employed martingale methods (Bielecki *et al* 2005; Nguyen and Portait 2002; Zhao and Ziemba 2000) or tractable auxiliary problems (Li and Ng 2000; Zhou and Li 2000). However, in general, if realistic constraints on portfolio selection are imposed, eg, no trading if insolvent and a maximum leverage constraint, then a fully numerical approach is required. As shown in Wang and Forsyth (2010), in the case where the risky asset follows a GBM, realistic portfolio constraints have a significant effect on the efficient frontier.

Another modeling deficiency in previous work on pre-commitment mean–variance optimal asset allocation is the common assumption that the risky asset follows a GBM. However, there is strong empirical evidence that asset return volatility is serially correlated, shocks to volatility are negatively correlated with asset returns and the conditional variance of asset returns is not constant over time. As a result, it is

highly desirable to describe the risky asset with a stochastic volatility model. In this case, the standard formulation of mean–variance optimal asset allocation problems gives rise to a two-dimensional nonlinear Hamilton–Jacobi–Bellman (HJB) PDE. The objective of this paper is to develop a numerical method for the pre-commitment mean–variance portfolio selection problem when the underlying risky asset follows a stochastic volatility model.

The major contributions of the paper are the following.

- We develop a fully implicit, consistent, unconditionally monotone numerical scheme for the HJB equation, which arises in the embedding formulation (Li and Ng 2000; Zhou and Li 2000) of the pre-commitment mean–variance problem under our model setup. The main difficulty in designing a discretization scheme is developing a monotone approximation of the cross-derivative term in the PDE. We use the wide stencil method (Debrabant and Jakobsen 2013; Ma and Forsyth 2014) to deal with this difficulty.

- We construct accurate efficient frontiers by using a semi-Lagrangian time-stepping method to handle the drift term and an improved method of linear interpolation at the foot of the characteristic in the semi-Lagrangian discretization. In particular, the improved interpolation method uses the exact solution value at a single point, dramatically increasing the accuracy of the numerical results. Any type of constraint can be applied to the investment policy.

- We prove that the scheme developed in this paper converges to the viscosity solution of the nonlinear HJB value equation.

- In order to trace out the efficient frontier solution of our problem, we use two techniques: the PDE method and the hybrid (PDE–Monte Carlo) method (Tse *et al* 2013). We also demonstrate that the hybrid method is superior to the PDE method.

- We carry out several numerical experiments, and illustrate the convergence of the numerical scheme as well as the effect of modeling parameters on efficient frontiers.

The remainder of this paper is organized as follows. Section 2 describes the underlying processes and the embedding framework, and gives a formulation of an associated HJB equation and a linear PDE. In Section 3, we present the discretization of the HJB equation. In Section 4, we highlight some important implementation details of the numerical method. Numerical results are presented and discussed in Section 5.

2 MATHEMATICAL FORMULATION

Suppose there are two assets in the market: one is a risk-free bond, and the other is a risky equity index. The dynamics of the risk-free bond B follows

$$\mathrm{d}B(t) = rB(t)\,\mathrm{d}t, \tag{2.1}$$

and an equity index S follows Heston's model (Heston 1993) under the real probability measure

$$\frac{\mathrm{d}S(t)}{S(t)} = (r + \xi V(t))\,\mathrm{d}t + \sqrt{V(t)}\,\mathrm{d}Z_1, \tag{2.2}$$

where the variance of the index, $V(t)$, follows a mean-reverting square-root process (Cox *et al* 1985):

$$\mathrm{d}V(t) = \kappa(\theta - V(t))\,\mathrm{d}t + \sigma\sqrt{V(t)}\,\mathrm{d}Z_2, \tag{2.3}$$

with $\mathrm{d}Z_1, \mathrm{d}Z_2$ being increments of Wiener processes. The instantaneous correlation between Z_1 and Z_2 is $\mathrm{d}Z_1\,\mathrm{d}Z_2 = \rho\,\mathrm{d}t$. The market price of volatility risk is $\xi V(t)$, which generates a risk premium proportional to $V(t)$. This assumption for the risk premium is based on Breeden's consumption-based model (Breeden 1979), and it originates from Heston (1993). Therefore, under this setup, the market is incomplete, as trading in the risky asset and the bond cannot perfectly hedge the changes in the stochastic investment opportunity set.

An investor in this market is endowed at time zero with an initial wealth of w_0, and they can continuously and dynamically alter the proportion of wealth invested in each asset. In addition, let $W(t) = S(t) + B(t)$ denote the wealth at time t, and let p denote the proportion of this wealth invested in the risky asset $S(t)$; consequently, $(1 - p)$ denotes the fraction of wealth invested in the risk-free bond $B(t)$. The allocation strategy is a function of the current state, ie, $p(\cdot)\colon (W(t), V(t), t) \to p = p(W(t), V(t), t)$. Note that in using the shorthand notations $p(\cdot)$ for the mapping, p for the value $p = p(W(t), V(t), t)$, and the dependence on the current state is implicit. From (2.1) and (2.2), we see that the investor's wealth process follows

$$\mathrm{d}W(t) = (r + p\xi V(t))W(t)\,\mathrm{d}t + p\sqrt{V}W(t)\,\mathrm{d}Z_1. \tag{2.4}$$

2.1 Efficient frontiers and embedding methods

We assume here that the investor is guided by a pre-commitment mean–variance objective based on the final wealth $W(T)$. The pre-commitment mean–variance problem and its variations have been intensively studied in the literature (Bielecki *et al* 2005; Li and Ng 2000; Nguyen and Portait 2002; Zhao and Ziemba 2000; Zhou and Li 2000). To the best of our knowledge, there is no explicit closed-form solution for the

pre-commitment mean–variance problem when the risky asset follows a stochastic volatility process along with leverage constraints.

To simplify our notation, we define $x = (w, v) = (W(t), V(t))$ for a state space. Let $E^{x,t}_{p(\cdot)}[W(T)]$ and $\mathrm{Var}^{x,t}_{p(\cdot)}[W(T)]$ denote the expectation and variance of the terminal wealth conditional on the state (x, t) and the control $p(\cdot)$. Given a risk level $\mathrm{Var}^{x,t}_{p(\cdot)}[W(T)]$, an investor desires their expected terminal wealth $E^{x,t}_{p(\cdot)}[W(T)]$ to be as large as possible. Equivalently, given an expected terminal wealth $E^{x,t}_{p(\cdot)}[W(T)]$, they wish the risk $\mathrm{Var}^{x,t}_{p(\cdot)}[W(T)]$ to be as small as possible. That is, they desire to find controls $p(\cdot)$ that generate Pareto optimal points. For notational simplicity, let $E^{x,t}_{p(\cdot)}[W(T)] = \mathcal{E}$ and $\mathrm{Var}^{x,t}_{p(\cdot)}[W(T)] = \mathcal{V}$. The problem is rigorously formulated as follows.

Define the achievable mean–variance objective set as

$$\mathcal{Y} = \{(\mathcal{V}, \mathcal{E})\colon\ p \in \mathcal{Z}\}, \tag{2.5}$$

where $\mathcal{Z}$ is the set of admissible strategies, and denote the closure of $\mathcal{Y}$ by $\bar{\mathcal{Y}}$.

Definition 2.1 A point $(\mathcal{V}, \mathcal{E}) \in \mathcal{Y}$ is Pareto mean–variance optimal if there exists no admissible strategy $\bar{p} \in \mathcal{Z}$ such that

$$\mathrm{Var}^{x,t}_{\bar{p}}\{W(T)\} \leqslant \mathcal{V}, \qquad E^{x,t}_{\bar{p}}\{W(T)\} \geqslant \mathcal{E}, \tag{2.6}$$

where at least one of the inequalities in the equation is strict. We denote by $\mathcal{P}$ the set of Pareto mean–variance optimal points. Note that $\mathcal{P} \subseteq \bar{\mathcal{Y}}$.

Although the above definition is intuitive, determining the points in $\mathcal{P}$ requires the solution of a multi-objective optimization problem, involving two conflicting criteria. A standard scalarization method can be used to combine the two criteria into an optimization problem with a single objective. In particular, for each point $(\mathcal{V}, \mathcal{E}) \in \bar{\mathcal{Y}}$, and for an arbitrary scalar $\lambda > 0$, we define the set of points $\mathcal{Y}_{P(\lambda)}$ to be

$$\mathcal{Y}_{P(\lambda)} = \Big\{(\mathcal{V}, \mathcal{E}) \in \bar{\mathcal{Y}}\colon \inf_{(\mathcal{V}_*, \mathcal{E}_*)\in\mathcal{Y}} (\lambda\mathcal{V}_* - \mathcal{E}_*)\Big\}, \tag{2.7}$$

from which a point on the efficient frontier can be derived. The set of points on the efficient frontier is then defined as

$$\mathcal{Y}_P = \bigcup_{\lambda>0} \mathcal{Y}_{P(\lambda)}. \tag{2.8}$$

Note that there is a difference between the set of all Pareto mean–variance optimal points $\mathcal{P}$ (see Definition 2.1) and the efficient frontier $\mathcal{Y}_P$ (2.8) (Tse *et al* 2014). In general,

$$\mathcal{P} \subseteq \mathcal{Y}_P,$$

but the converse may not hold if the achievable mean–variance objective set $\mathcal{Y}$ (2.5) is not convex. In this paper, we restrict our attention to constructing $\mathcal{Y}_P$ (2.8).

Due to the presence of the variance term $\mathrm{Var}^{x,t}_{p(\cdot)}[W(T)]$ in (2.7), a dynamic programming principle cannot be directly applied to solve this problem. To overcome this difficulty, we make use of the main result in Li and Ng (2000), Zhou and Li (2000) and Tse *et al* (2014), which essentially involves the embedding technique. This result is summarized in Theorem 2.3.

ASSUMPTION 2.2 *We assume that $\mathcal{Y}$ is a non-empty subset of $\{(\mathcal{V}, \mathcal{E}) \in \mathbb{R}^2 : \mathcal{V} > 0\}$ and that there exists a positive scalarization parameter $\lambda_E > 0$ such that $\mathcal{Y}_{P(\lambda_E)} \neq \emptyset$.*

THEOREM 2.3 *The embedded mean–variance objective set $\mathcal{Y}_Q$ is defined by*

$$\mathcal{Y}_Q = \bigcup_{-\infty<\gamma<\infty} \mathcal{Y}_{Q(\gamma)}, \tag{2.9}$$

where

$$\mathcal{Y}_{Q(\gamma)} = \left\{(\mathcal{V}_*, \mathcal{E}_*) \in \bar{\mathcal{Y}} : \mathcal{V}_* + \mathcal{E}_*^2 - \gamma\mathcal{E}_* = \inf_{(\mathcal{V},\mathcal{E})\in\mathcal{Y}} (\mathcal{V} + \mathcal{E}^2 - \gamma\mathcal{E})\right\}. \tag{2.10}$$

If Assumption 2.2 holds and $\lambda > \lambda_E$, then $\mathcal{Y}_{P(\lambda)} \neq \emptyset$. Assume $(\mathcal{V}_0, \mathcal{E}_0) \in \mathcal{Y}_{P(\lambda)}$. Then, if

$$\lambda\mathcal{V}_0 - \mathcal{E}_0 = \inf_{(\mathcal{V},\mathcal{E})\in\mathcal{Y}} (\lambda\mathcal{V} - \mathcal{E}), \tag{2.11}$$

then

$$\mathcal{V}_0 + \mathcal{E}_0^2 - \gamma\mathcal{E}_0 = \inf_{(\mathcal{V},\mathcal{E})\in\mathcal{Y}} (\mathcal{V} + \mathcal{E}^2 - \gamma\mathcal{E}), \quad \textit{ie, } (\mathcal{V}_0, \mathcal{E}_0) \in \mathcal{Y}_{Q(\gamma)}, \tag{2.12}$$

where $\gamma = (1/\lambda) + 2\mathcal{E}_0$. Consequently, $\mathcal{Y}_P \subseteq \mathcal{Y}_Q$.

PROOF See details in Li and Ng (2000), Zhou and Li (2000) and Dang *et al* (2016). □

Theorem 2.3 states that the mean and variance $(\mathcal{V}, \mathcal{E})$ of $W(T)$ are embedded in a scalarization optimization problem, with the objective function being $\mathcal{V} + \mathcal{E}^2 - \gamma\mathcal{E}$. Noting that

$$\begin{aligned}
&\mathcal{V} + \mathcal{E}^2 - \gamma\mathcal{E} \\
&\quad = E^{x,t}_{p(\cdot)}[W^2(T)] - (E^{x,t}_{p(\cdot)}[W(T)])^2 + (E^{x,t}_{p(\cdot)}[W(T)])^2 - \gamma E^{x,t}_{p(\cdot)}[W(T)] \\
&\quad = E^{x,t}_{p(\cdot)}[W^2(T) - \gamma W(T)] \\
&\quad = E^{x,t}_{p(\cdot)}[(W(T) - \tfrac{1}{2}\gamma)^2] + \tfrac{1}{4}\gamma^2,
\end{aligned} \tag{2.13}$$

and that we can ignore the constant $\frac{1}{4}\gamma^2$ term for the purposes of minimization, we then define the value function

$$\mathcal{U}(x,t) = \inf_{p(\cdot)\in\mathcal{Z}} E^{x,t}_{p(\cdot)}[(W(T) - \tfrac{1}{2}\gamma)^2]. \tag{2.14}$$

Theorem 2.3 implies that there exists a γ such that, for a given positive λ, a control p^* that minimizes (2.7) also minimizes (2.14). Dynamic programming can then be directly applied to (2.14) to determine the optimal control $p^*(\cdot)$.

The procedure for determining the points on the efficient frontier is as follows. For a given value of γ, the optimal strategy p^* is determined by solving for the value function problem (2.14). Once this optimal policy $p^*(\cdot)$ is known, it is then straightforward to determine a point $(\mathrm{Var}^{x,t}_{p^*(\cdot)}[W(T)], E^{x,t}_{p^*(\cdot)}[W(T)])$ on the frontier. Varying γ traces out a curve in the $(\mathcal{V},\mathcal{E})$ plane (see details in Section 4.2). Consequently, the numerical challenge is to solve for the value function (2.14). More precisely, the above procedure for constructing the efficient frontier generates points that are in the set $\mathcal{Y}_Q$. As pointed out in Tse *et al* (2014), the set $\mathcal{Y}_Q$ may contain spurious points, ie, points that are not in $\mathcal{Y}_P$. For example, when the original problem is nonconvex, spurious points can be generated. An algorithm for removing spurious points is discussed in Tse *et al* (2014). The set of points in $\mathcal{Y}_Q$ with the spurious points removed generates all points in $\mathcal{Y}_P$. Dang *et al* (2016) also discusses the convergence of finitely sampled γ to the efficient frontier.

Remark 2.4 (Range of γ) As noted in Dang *et al* (2016), a solution to (2.10) generally exists for all $\gamma \in (-\infty, +\infty)$. However, we know from the above discussion that some of these solutions may be spurious. In some cases, we can use financial reasoning to reduce the range of γ so that obvious spurious points are eliminated. We discuss this further in Section 4.2.1.

2.2 The value function problem

Following standard arguments, the value function $\mathcal{U}(w,v,\tau)$, $\tau = T - t$ (2.14) is the viscosity solution of the HJB equation

$$\begin{aligned}\mathcal{U}_\tau = \inf_{p\in\mathcal{Z}}\{&(r + p\xi v)w\mathcal{U}_w + \kappa(\theta - v)\mathcal{U}_v \\ &+ \tfrac{1}{2}(p\sqrt{v}w)^2\mathcal{U}_{ww} + p\rho\sigma\sqrt{v}w\mathcal{U}_{wv} + \tfrac{1}{2}\sigma^2 v\mathcal{U}_{vv}\},\end{aligned} \tag{2.15}$$

on the domain $(w,v,\tau) \in [0,+\infty]\times[0,+\infty]\times[0,T]$, and with the terminal condition

$$\mathcal{U}(w,v,0) = (w - \tfrac{1}{2}\gamma)^2. \tag{2.16}$$

REMARK 2.5 In one of our numerical tests, we allow p to become unbounded, which may occur when $w \to 0$ (Wang and Forsyth 2010). However, although $p \to \infty$ as $w \to 0$, we must have $(pw) \to 0$ as $w \to 0$, ie, the amount invested in the risky asset converges to zero as $w \to 0$. This is required in order to ensure that the no-bankruptcy boundary condition is satisfied (Wang and Forsyth 2010). As a result, we can then formally eliminate the problem with unbounded control by using $q = pw$ as the control, and assume q remains bounded (see details in Wang and Forsyth (2010)).

2.3 The expected wealth problem

2.3.1 The PDE formulation

Given the solution for the value function (2.14), with the optimal control $p^*(\cdot)$, we then need to determine the expected value $E^{x,t}_{p^*(\cdot)}[W(T)]$, denoted as

$$\mathcal{E}(w, v, t) = E^{x,t}_{p^*(\cdot)}[W(T)]. \tag{2.17}$$

Then, $\mathcal{E}(w, v, \tau)$, $\tau = T - t$ is given from the solution to the following linear PDE:

$$\mathcal{E}_\tau = (r + p^*\xi v)w\mathcal{E}_w + \kappa(\theta - v)\mathcal{E}_v + \tfrac{1}{2}(p^*\sqrt{v}w)^2\mathcal{E}_{ww} + p^*\rho\sigma\sqrt{v}w\mathcal{E}_{wv} + \tfrac{1}{2}\sigma^2 v\mathcal{E}_{vv}, \tag{2.18}$$

with the initial condition $\mathcal{E}(w, v, 0) = w$, where p^* is obtained from the solution of the HJB equation (2.15).

2.3.2 The hybrid (PDE–Monte Carlo) method

Alternatively, given the stored control $p^*(\cdot)$ determined from the solution of (2.15), we can directly estimate $(\mathrm{Var}^{x,t}_{p^*(\cdot)}[W(T)], E^{x,t}_{p^*(\cdot)}[W(T)])$ by using a Monte Carlo method, based on solving the stochastic differential equations (SDEs) (2.3) and (2.4). The details of the SDE discretization are given in Section 4.2. This hybrid (PDE–Monte Carlo) method was originally proposed in Tse *et al* (2013).

2.4 Allowable portfolios

In order to obtain analytical solutions, many previous papers typically make assumptions that allow for the possibility of unbounded borrowing and bankruptcy. Moreover, these models assume a bankrupt investor can still keep on trading. The ability to continue trading even though the value of an investor's wealth is negative is highly unrealistic. In this paper, we enforce the condition that the wealth value remains in the solvency regions by applying certain boundary conditions to the HJB equation (Wang and Forsyth 2008). Thus, bankruptcy is prohibited, ie,

$$w \in [0, +\infty).$$

We will also assume that there is a leverage constraint, ie, the investor must select an asset allocation satisfying

$$p = \frac{\text{risky asset value}}{\text{total wealth}} = \frac{pW(t)}{W(t)} < p_{\max},$$

which can be interpreted as the maximum leverage condition, and $p_{\max}$ is a known positive constant with typical value in $[1.0, 2.0]$. Thus, the control set

$$p \in \mathcal{Z} = [0, p_{\max}].$$

Note that when the risk premium ξ (2.2) is positive, as in our case, it is not optimal to short the risky asset, since we have only a single risky asset in our portfolio. In some circumstances, it may be optimal to short the risky asset. This will be discussed in Section 3.1.3.

3 NUMERICAL DISCRETIZATION OF THE HAMILTON–JACOBI–BELLMAN EQUATION

3.1 Localization

We will assume that the discretization is posed on a bounded domain for computational purposes. The discretization is applied to the localized finite region $(w, v) \in [0, w_{\max}] \times [0, v_{\max}]$. Asymptotic boundary conditions will be imposed at $w = w_{\max}$ and $v = v_{\max}$, which are compatible with a monotone numerical scheme.

3.1.1 The localization of V

The proper boundary on $v = 0$ needs to be specified to be compatible with the corresponding SDE (2.3), which has a unique solution (Feller 1951). If $2\kappa\theta \geq \sigma^2$, the so-called Feller condition holds, and $v = 0$ is unattainable. If the Feller condition is violated, $2\kappa\theta < \sigma^2$, then $v = 0$ is an attainable boundary but is strongly reflecting (Feller 1951). The appropriate boundary condition can be obtained by setting $v = 0$ into (2.15). That is,

$$\mathcal{U}_\tau = r w \mathcal{U}_w + \kappa\theta \mathcal{U}_v, \tag{3.1}$$

and the equation degenerates to a linear PDE. On the lower boundary $v = 0$, the variance and the risk premium vanish, according to (2.4), so that the wealth return is always the risk-free rate r. The control value p vanishes in the degenerate equation (3.1), and we can simply define $p^*(w, v = 0, t) \equiv 0$, which we need in the estimation of $(\text{Var}^{x,t}_{p^*(\cdot)}[W(T)], E^{x,t}_{p^*(\cdot)}[W(T)])$ using the Monte Carlo simulation. In this case, since the equity asset has zero volatility with drift rate r, the distinction between the equity asset and risk-free asset is meaningless.

The validity of this boundary condition is intuitively justified by the fact that the solution to the SDE for v is unique, such that the behavior of v at the boundary $v = 0$ is determined by the SDE itself, and, hence, the boundary condition is determined by setting $v = 0$ in (2.15). A formal proof that this boundary condition is correct is given in Ekström and Tysk (2010). If the boundary at $v = 0$ is attainable, then this boundary behavior serves as a boundary condition and guarantees uniqueness in the appropriate function spaces. However, if the boundary is non-attainable, then the boundary behavior is not needed to guarantee uniqueness, but it is nevertheless very useful in a numerical scheme.

On the upper boundary $v = v_{\max}$, $\mathcal{U}_v$ is set to zero. Thus, the boundary condition on $v_{\max}$ is set to

$$\mathcal{U}_\tau = \inf_{p\in\mathcal{Z}}\{(r + p\xi v)w\mathcal{U}_w + \tfrac{1}{2}(p\sqrt{v}w)^2\mathcal{U}_{ww}\}. \tag{3.2}$$

The optimal control p^* at $v = v_{\max}$ is determined by solving (3.2). This boundary condition can be justified by noting that, as $v \to \infty$, the diffusion term in the w direction in (2.15) becomes large. In addition, the initial condition (2.16) is independent of v. As a result, we expect that

$$\mathcal{U} \approx C'w + C'', \qquad v \to \infty,$$

where C' and C'' are constants, and, hence, $\mathcal{U}_v \approx 0$ at $v = v_{\max}$.

3.1.2 The localization for W

We prohibit the possibility of bankruptcy ($W(t) < 0$) by requiring $\lim_{w\to 0}(pw) = 0$ (Wang and Forsyth 2010), so, on $w = 0$, (2.15) reduces to

$$\mathcal{U}_\tau = \kappa(\theta - v)\mathcal{U}_v + \sigma^2 v\mathcal{U}_{vv}. \tag{3.3}$$

When $w \to +\infty$, we assume that the asymptotic form of the exact solution is

$$\mathcal{U}(w \to +\infty, v, \tau) = \bar{\mathcal{U}}(w) = H_2(\tau)w^2 + H_1(\tau)w + H_0(\tau); \tag{3.4}$$

we make the assumption that $p^*(w_{\max}, v, 0)$ at $w = w_{\max}$ is set to zero. That is, once the investor's wealth is very large, they prefer the risk-free asset. This can be justified from the arguments in Cui *et al* (2012) and Dang and Forsyth (2014a).

Substituting form (3.4) into PDE (2.15) and setting $p = 0$, we obtain

$$(H_0)_\tau = 0, \qquad (H_1)_\tau = rH_1, \qquad (H_2)_\tau = 2rH_2.$$

Initial conditions are determined from (2.16) and (3.4):

$$H_0(0) = \tfrac{1}{4}\gamma^2, \qquad H_1(0) = -2\gamma, \qquad H_2(0) = 1.$$

3.1.3 Alternative localization for w

$\mathcal{U}(w, v, \tau)$ is the viscosity solution of the HJB equation (2.15). Recall that the initial condition for problem (2.14) is

$$\mathcal{U}(w, v, 0) = (W(T) - \tfrac{1}{2}\gamma)^2.$$

For a fixed gamma, we define the discounted optimal embedded terminal wealth at time t, denoted by $W_{\text{opt}}(t)$, as

$$W_{\text{opt}}(t) = \tfrac{1}{2}\gamma \mathrm{e}^{-r(T-t)}. \tag{3.5}$$

It is easy to verify that $W_{\text{opt}}(t)$ is a global minimum state of the value function $\mathcal{U}(w, v, t)$. Consider the state $(W_{\text{opt}}(t), v)$, $t \in [0, T]$, and the optimal strategy $p^*(\cdot)$ such that $p^*(w, v, \mathcal{T}) \equiv 0$, $\mathcal{T} > t$. Under $p^*(\cdot)$, the wealth is all invested in the risk-free bond without further rebalancing from time t. As a result, the wealth will accumulate to $W(T) = \frac{1}{2}\gamma$ with certainty, ie, the optimal embedded terminal wealth $\frac{1}{2}\gamma$ is achievable. By (2.14), we have

$$\mathcal{U}(W_{\text{opt}}(t), v, t) = \inf_{p(\cdot)\in\mathcal{Z}} \{E^{x,t}_{p(\cdot)}[(W(T) - \tfrac{1}{2}\gamma)^2]\} = E^{x,t}_{p^*(\cdot)}[(W(T) - \tfrac{1}{2}\gamma)^2] = 0. \tag{3.6}$$

Since the value function is the expectation of a nonnegative quantity, it can never be less than zero. Hence, the exact solution for the value function problem at the special point $W_{\text{opt}}(t)$ must be zero. This result holds for both the discrete and continuous rebalancing case. For the formal proof, we refer the reader to Dang and Forsyth (2014a).

Consequently, the point $w = \frac{1}{2}\gamma \mathrm{e}^{-r\tau}$ is a Dirichlet boundary $\mathcal{U}(\frac{1}{2}\gamma \mathrm{e}^{-r\tau}, v, \tau) = 0$, and information for $w > \frac{1}{2}\gamma \mathrm{e}^{-r\tau}$ is not needed. In principle, we can restrict the domain to $0 \leqslant w \leqslant \frac{1}{2}\gamma \mathrm{e}^{-r\tau}$. However, it is computationally convenient to restrict the size of the computational domain to be $0 \leqslant w \leqslant \frac{1}{2}\gamma$, which avoids the issue of having a moving boundary, at a very small additional cost. Note that the optimal control will ensure that $\mathcal{U}(\frac{1}{2}\gamma \mathrm{e}^{-r\tau}, v, \tau) = 0$ without any need to enforce this boundary condition. This will occur, as we assume continuous rebalancing. This effect, that $W(t) \leqslant W_{\text{opt}}(t)$, is also discussed in Vigna (2014). It is interesting to note that, in the case of discrete rebalancing or jump diffusions, it is optimal to withdraw cash from the portfolio if it is ever observed that $W(t) > W_{\text{opt}}(t)$. However, Bauerle and Grether (2015) show that if the market is complete, then it is never optimal to withdraw cash from the portfolio. This is also discussed in Cui *et al* (2012) and Dang and Forsyth (2014a).

In the case of an incomplete market, such as discrete rebalancing or jump diffusions, if we do not allow the withdrawing of cash from the portfolio, then the investor has an incentive to lose money if $W(t) > W_{\text{opt}}(t)$, as pointed out in Cui *et al* (2012). In this rather perverse situation, it may be optimal to short the risky asset, so that the admissible set in this case would be $\mathcal{Z} = [p_{\min}, p_{\max}]$, with $p_{\min} < 0$.

We have verified, experimentally, that restricting the computational domain to $w \in [0, \frac{1}{2}\gamma]$ gives the same results as the domain $w \in [0, w_{\max}]$, $w_{\max} \gg \frac{1}{2}\gamma$, with asymptotic boundary condition (3.4).

REMARK 3.1 (Significance of $W(t) \leqslant W_{\text{opt}}(t)$) If we assume that initially $W(0) < W_{\text{opt}}(0)$ (otherwise, the problem is trivial if we allow cash withdrawals), then the optimal control will ensure that $W(t) \leqslant W_{\text{opt}}(t)$ for all t. Hence, continuous-time mean–variance optimization is "time consistent in efficiency" (Cui *et al* 2012). Another interpretation is that continuous-time mean–variance optimization is equivalent to minimizing the quadratic loss with respect to the wealth target $W_{\text{opt}}(T)$ (Vigna 2014).

REMARK 3.2 (Significance of $W(T) \leqslant \frac{1}{2}\gamma$) From Remark 3.1, we have trivially that $W(T) \leqslant \frac{1}{2}\gamma$; hence, from (2.14), the investor is never penalized for large gains, ie, the quadratic utility function (2.14) is always well behaved. Consequently, continuous-time mean–variance optimization is fundamentally different from its single-period counterpart.

3.2 Discretization

In the following section, we discretize (2.15) over a finite grid $N = N_1 \times N_2$ in the space (w, v). Define a set of nodes $\{w_1, w_2, \ldots, w_{N_1}\}$ in the w direction and $\{v_1, v_2, \ldots, v_{N_2}\}$ in the v direction. Denote the nth time step by $\tau^n = n\Delta\tau$, $n = 0, \ldots, N_\tau$, with $N_\tau = T/\Delta\tau$. Let $\mathcal{U}^n_{i,j}$ be the approximate solution of (2.15) at (w_i, v_j, τ^n).

It will be convenient to define

$$\Delta w_{\max} = \max_i (w_{i+1} - w_i), \qquad \Delta w_{\min} = \min_i (w_{i+1} - w_i),$$
$$\Delta v_{\max} = \max_i (v_{i+1} - v_i), \qquad \Delta v_{\min} = \min_i (v_{i+1} - v_i).$$

We assume that there is a mesh discretization parameter h such that

$$\Delta w_{\max} = C_1 h, \quad \Delta w_{\min} = C_2 h, \quad \Delta v_{\max} = C_1' h, \quad \Delta v_{\min} = C_2' h, \quad \Delta\tau = C_3 h, \tag{3.7}$$

where C_1, C_2, C_1', C_2', C_3 are constants independent of h.

In the following sections, we will give the details of the discretization for a reference node (w_i, v_j), $1 < i < N_1$, $1 < j < N_2$.

3.2.1 The wide stencil

We need a monotone discretization scheme in order to guarantee convergence to the desired viscosity solution (Barles and Souganidis 1991). We remind the reader that seemingly reasonable non-monotone discretizations can converge to the incorrect solution (Pooley *et al* 2003). Due to the cross-derivative term in (2.15), however, a classic finite difference method cannot produce such a monotone scheme. Since the control appears in the cross-derivative term, it will not be possible (in general) to determine a grid spacing or global coordinate transformation that eliminates this term. We will adopt the wide stencil method developed in Ma and Forsyth (2014) to discretize the second derivative terms. Suppose we discretize (2.15) at grid node (i, j) for a fixed control. For a fixed p, consider a virtual rotation of the local coordinate system clockwise by the angle $\eta_{i,j}$:

$$\eta_{i,j} = \tfrac{1}{2}\tan^{-1}\left(\frac{2\rho p\sigma w_i v_j}{(p\sqrt{v_j}w_i)^2 - (\sigma\sqrt{v_j})^2}\right). \tag{3.8}$$

That is, (y_1, y_2) in the transformed coordinate system is obtained by using the following matrix multiplication:

$$\begin{pmatrix} w \\ v \end{pmatrix} = \begin{pmatrix} \cos\eta_{i,j} & -\sin\eta_{i,j} \\ \sin\eta_{i,j} & \cos\eta_{i,j} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \tag{3.9}$$

We denote the rotation matrix in (3.9) as $\boldsymbol{R}_{i,j}$. This rotation operation will result in a zero correlation in the diffusion tensor of the rotated system. Under this grid rotation, the second-order terms in (2.18) are, in the transformed coordinate system (y_1, y_2),

$$a_{i,j}\frac{\partial^2 \mathcal{W}}{\partial y_1^2} + b_{i,j}\frac{\partial^2 \mathcal{W}}{\partial y_2^2}, \tag{3.10}$$

where $\mathcal{W}$ is the value function $\mathcal{W}(y_1, y_2, \tau)$ in the transformed coordinate system, and

$$\left.\begin{aligned} a_{i,j} &= (\tfrac{1}{2}(p\sqrt{v_j}w_i)^2\cos(\eta_{i,j})^2 + \rho p\sigma w_i v_j \sin(\eta_{i,j})\cos(\eta_{i,j}) \\ &\qquad + \tfrac{1}{2}(\sigma\sqrt{v_j})^2\sin(\eta_{i,j})^2), \\ b_{i,j} &= (\tfrac{1}{2}(p\sqrt{v_j}w_i)^2\sin(\eta_{i,j})^2 - \rho p\sigma w_i v_j \sin(\eta_{i,j})\cos(\eta_{i,j}) \\ &\qquad + \tfrac{1}{2}(\sigma\sqrt{v_j})^2\cos(\eta_{i,j})^2). \end{aligned}\right\} \tag{3.11}$$

The diffusion tensor in (3.10) is diagonally dominant with no off-diagonal terms; consequently, a standard finite difference discretization for the second partial derivatives results in a monotone scheme. The rotation angle $\eta_{i,j}$ depends on the grid node and the control; therefore, it is impossible to rotate the global coordinate system by

a constant angle and build a grid over the entire space (y_1, y_2). The local coordinate system rotation is only used to construct a virtual grid, which overlays the original mesh. We have to approximate the values of $\mathcal{W}$ on our virtual local grid using an interpolant $\mathcal{J}_h\mathcal{U}$ on the original mesh. To keep the numerical scheme monotone, $\mathcal{J}_h$ must be a linear interpolation operator. Moreover, to keep the numerical scheme consistent, we need to use the points on our virtual grid, whose Euclidean distances are $O(\sqrt{h})$ from the central node, where h is the mesh discretization parameter (3.7). This results in a wide stencil method, since the relative stencil length increases as the grid is refined ($\sqrt{h}/h \to +\infty$ as $h \to 0$). For more details, we refer the reader to Ma and Forsyth (2014).

Let us rewrite the HJB equation (2.15) as

$$\sup_{p\in\mathcal{Z}}\{\mathcal{U}_\tau - (r + p\xi v)w\mathcal{U}_w - \mathcal{L}^p\mathcal{U}\} = 0, \tag{3.12}$$

where the linear operator $\mathcal{L}^p$ is defined as

$$\mathcal{L}^p\mathcal{U} = \kappa(\theta - v)\mathcal{U}_v + \tfrac{1}{2}(p\sqrt{v}w)^2\mathcal{U}_{ww} + p\rho\sigma\sqrt{v}w\mathcal{U}_{wv} + \tfrac{1}{2}\sigma^2 v\mathcal{U}_{vv}. \tag{3.13}$$

The drift term $\kappa(\theta - v)\mathcal{U}_v$ in (3.13) is discretized by a standard backward or forward finite differencing discretization, depending on the sign of $\kappa(\theta - v)$. Overall, the discretized form of the linear operator $\mathcal{L}^p$ is then denoted by L_h^p:

$$\begin{aligned}
&L_h^p\mathcal{U}_{i,j}^{n+1}\\
&= 1_{\kappa(\theta-v_j)\geq 0}\frac{\kappa(\theta - v_j)}{h}\mathcal{U}_{i,j+1}^{n+1} - 1_{\kappa(\theta-v_j)<0}\frac{\kappa(\theta - v_j)}{h}\mathcal{U}_{i,j-1}^{n+1}\\
&\quad + \frac{a_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}(x_{i,j} + \sqrt{h}(\boldsymbol{R}_{i,j})_1) + \frac{a_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}(x_{i,j} - \sqrt{h}(\boldsymbol{R}_{i,j})_1)\\
&\quad + \frac{b_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}(x_{i,j} + \sqrt{h}(\boldsymbol{R}_{i,j})_2) + \frac{b_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}(x_{i,j} - \sqrt{h}(\boldsymbol{R}_{i,j})_2)\\
&\quad - \left(1_{\kappa(\theta-v_j)\geq 0}\frac{\kappa(\theta - v_j)}{h} - 1_{\kappa(\theta-v_j)<0}\frac{\kappa(\theta - v_j)}{h} + \frac{2a_{i,j}}{h} + \frac{2b_{i,j}}{h}\right)\mathcal{U}_{i,j}^{n+1},
\end{aligned} \tag{3.14}$$

where h is the discretization parameter, and the superscript p in L_h^p indicates that the discretization depends on the control p. Note that $x_{i,j} = \binom{w_i}{v_j}$, $a_{i,j}$ and $b_{i,j}$ are given in (3.11), and the presence of $\mathcal{J}_h\mathcal{U}^{n+1}(x_{i,j} \pm \sqrt{h}(\boldsymbol{R}_{i,j})_k)$, $k = 1, 2$, is due to the discretization of the second derivative terms. $(\boldsymbol{R}_{i,j})_k$ is the kth column of the rotation matrix.

3.2.2 Semi-Lagrangian time-stepping scheme

When $p \to 0$, (2.15) degenerates, with no diffusion in the w direction. As a result, we will discretize the drift term $(r + p\xi v)w\mathcal{U}_w$ in (2.15) by a semi-Lagrangian time-stepping scheme in this section. Initially introduced by Douglas and Russell (1982)

and Pironneau (1982) for atmospheric and weather numerical prediction problems, semi-Lagrangian schemes can effectively reduce the numerical problems arising from convection dominated equations.

First, we define the Lagrangian derivative $\mathrm{D}\mathcal{U}(p)/\mathrm{D}\tau$ by

$$\frac{\mathrm{D}\mathcal{U}}{\mathrm{D}\tau}(p) = \mathcal{U}_\tau - (r + p\xi v)w\mathcal{U}_w, \tag{3.15}$$

which is the rate of change of $\mathcal{U}$ along the characteristic $w = w(\tau)$ defined by the risky asset fraction p through

$$\frac{\mathrm{d}w}{\mathrm{d}\tau} = -(r + p\xi v)w. \tag{3.16}$$

We can then rewrite (3.12) as

$$\sup_{p\in\mathcal{Z}}\left\{\frac{\mathrm{D}\mathcal{U}}{\mathrm{D}\tau} - \mathcal{L}^p\mathcal{U}\right\} = 0. \tag{3.17}$$

Solving (3.16) backward in time from τ^{n+1} and τ^n for a fixed w_i^{n+1} gives the point at the foot of the characteristic

$$(w_{i*}, v_j) = (w_i \exp((r + p\xi v_j)\Delta\tau^n), v_j), \tag{3.18}$$

which, in general, is not on the PDE grid. We use the notation $\mathcal{U}^n_{i*,j}$ to denote an approximation of the value $\mathcal{U}(w_{i*}, v_j, \tau^n)$, which is obtained by linear interpolation to preserve monotonicity. The Lagrangian derivative at a reference node (i, j) is then approximated by

$$\frac{\mathrm{D}\mathcal{U}}{\mathrm{D}\tau}(p) \approx \frac{\mathcal{U}^{n+1}_{i,j} - \mathcal{U}^n_{i*,j}(p)}{\Delta\tau^n}, \tag{3.19}$$

where $\mathcal{U}^n_{i*,j}(p)$ denotes that w_{i*} depends on the control p through (3.18). For the details of the semi-Lagrangian time-stepping scheme, we refer the reader to Chen and Forsyth (2007).

Finally, by using the implicit time-stepping method, combining the expressions (3.14) and (3.19), the HJB equation (3.17) at a reference point (w_i, v_j, τ^{n+1}) is then discretized as

$$\sup_{p\in\mathcal{Z}_h}\left\{\left(\frac{\mathcal{U}^{n+1}_{i,j} - \mathcal{U}^n_{i*,j}(p)}{\Delta\tau^n}\right) - L_h^p\mathcal{U}^{n+1}_{i,j}\right\} = 0, \tag{3.20}$$

where $\mathcal{Z}_h$ is the discrete control set. There is no simple analytic expression that can be used to minimize the discrete equation (3.20), and we need to discretize the admissible control set $\mathcal{Z}$ and perform a linear search. This guarantees that we find the global maximum of (3.20), since the objective function has no known convexity properties. If the discretization step for the controls is also $O(h)$, where h is the discretization parameter, then this is a consistent approximation (Wang and Forsyth 2008).

TABLE 1 The domain definitions.

Notation	Domain
Ω	$[0, w_{\max}] \times [0, v_{\max}]$
Ω_{in}	$(0, w_{\max}) \times (0, v_{\max})$
$\Omega_{w_{\max}}$	The upper boundary $w = w_{\max}$
$\Omega_{v_{\max}}$	The upper boundary $v = v_{\max}$
$\Omega_{w_{\min}}$	The lower boundary $w = 0$
$\Omega_{v_{\min}}$	The lower boundary $v = 0$
Ω_{out}	$(w_{\max}, +\infty) \times (0, +\infty) \cup (0, +\infty) \times (v_{\max}, +\infty)$

3.3 Matrix form of the discrete equation

Our discretization is summarized as follows. The domains are defined in Table 1. For the case $(w_i, v_j) \in \Omega_{\text{in}}$, we need to use a wide stencil based on a local coordinate rotation to discretize the second derivative terms. We also need to use the semi-Lagrangian time-stepping scheme to handle the drift term $(r + p\xi v)w\mathcal{U}_w$. The HJB equation is discretized as (3.20), and the optimal p^* in this case is determined by solving (3.20). For the case $\Omega_{v_{\max}}$, the HJB equation degenerates to (3.2). In this case, the drift term is also handled by the semi-Lagrangian time-stepping scheme. With vanishing cross-derivative term, the degenerate linear operator $\mathcal{L}^p$ can be discretized by a standard finite difference method. The corresponding discretized form D_h^p is given in Section 3.3.1. The value for case $\Omega_{w_{\max}}$ is obtained by the asymptotic solution (3.4), and the optimal p^* is set to zero. At the lower boundaries $\Omega_{w_{\min}}$ and $\Omega_{v_{\min}}$, the HJB equation degenerates to a linear equation. The wide stencil and the semi-Lagrangian time-stepping scheme may require the value of the solution at a point outside the computational domain, denoted as Ω_{out}. Details on how to handle this case are given in Section 4.3. From the discretization (3.20), we can see that the measure of Ω_{out} converges to zero as $h \to 0$. Last, fully implicit time stepping is used to ensure unconditional monotonicity of our numerical scheme. Fully implicit time stepping requires the solution of highly nonlinear algebraic equations at each time step. For the applications addressed in Forsyth and Labahn (2007), an efficient method for solving the associated nonlinear algebraic systems makes use of a policy iteration scheme. We refer the reader to Huang *et al* (2012) and Forsyth and Labahn (2007) for the details of the policy iteration algorithm.

It is convenient to use a matrix form to represent the discretized equations for computational purposes. Let $\mathcal{U}_{i,j}^n$ be the approximate solution of (2.15) at (w_i, v_j, τ^n), $1 \leqslant i \leqslant N_1$, $1 \leqslant j \leqslant N_2$ and $0 \leqslant \tau^n \leqslant N_\tau$, and form the solution vector

$$\boldsymbol{U}^n = (\mathcal{U}_{1,1}^n, \mathcal{U}_{2,1}^n, \ldots, \mathcal{U}_{N_1,1}^n, \ldots, \mathcal{U}_{1,N_2}^n, \ldots, \mathcal{U}_{N_1,N_2}^n). \tag{3.21}$$

It will sometimes be convenient to use a single index when referring to an entry of the solution vector

$$\mathcal{U}_\ell^n = \mathcal{U}_{i,j}^n, \quad \ell = i + (j-1)N_1.$$

Let $N = N_1 \times N_2$. We define the $N \times N$ matrix $\boldsymbol{L}^{n+1}(\mathcal{P})$, where

$$\mathcal{P} = \{p_1, \ldots, p_N\} \tag{3.22}$$

is an indexed set of N controls, and each p_ℓ is in the set of admissible controls. $\boldsymbol{L}_{\ell,k}^{n+1}(\mathcal{P})$ is the entry on the ℓth row and kth column of the discretized matrix $\boldsymbol{L}^{n+1}(\mathcal{P})$. We also define a vector of boundary conditions $\boldsymbol{F}^{n+1}(\mathcal{P})$.

For the case $(w_i, v_j) \in \Omega_{w_{\max}}$, where the Dirichlet boundary condition (3.4) is imposed, we then have

$$\boldsymbol{F}_\ell^{n+1}(\mathcal{P}) = \bar{\mathcal{U}}(w_{\max}) \tag{3.23}$$

and

$$\boldsymbol{L}_{\ell,k}^{n+1}(\mathcal{P}) = 0, \quad k = 1, \ldots, N. \tag{3.24}$$

For the case $(w_i, v_j) \in \Omega_{v_{\min}} \cup \Omega_{w_{\min}} \cup \Omega_{v_{\max}}$, the differential operator degenerates, and the entries $\boldsymbol{L}_{\ell,k}^{n+1}(\mathcal{P})$ are constructed from the discrete linear operator D_h^p (see (3.32)). That is,

$$[\boldsymbol{L}^{n+1}(\mathcal{P})\boldsymbol{U}^{n+1}]_\ell = D_h^p \mathcal{U}_{i,j}^{n+1}. \tag{3.25}$$

For the case $(w_i, v_j) \in \Omega_{\text{in}}$, we need to use the values at the following four off-grid points $x_{i,j} \pm \sqrt{h}(\boldsymbol{R}_{i,j})_k$, $k = 1, 2$, in (3.14), and we denote those values by $\Psi_{i,j}^m$, $m = 1, 2, 3, 4$, respectively. Let (f_m, g_m) be indexes such that

$$\Psi_{i,j}^m = \begin{pmatrix} (1-\theta_w^m) w_{f_m} + \theta_w^m w_{f_m+1} \\ (1-\theta_v^m) v_{g_m} + \theta_v^m v_{g_m+1} \end{pmatrix}, \quad 0 \leqslant \theta_w^m, \theta_v^m \leqslant 1, \tag{3.26}$$

with linear interpolation weights

$$\begin{aligned} \omega_{i,j}^{f_m,g_m} &= (1-\theta_w^m)(1-\theta_v^m), & \omega_{i,j}^{f_m+1,g_m} &= \theta_w^m(1-\theta_v^m), \\ \omega_{i,j}^{f_m,g_m+1} &= (1-\theta_w^m)\theta_v^m, & \omega_{i,j}^{f_m+1,g_m+1} &= \theta_w^m\theta_v^m. \end{aligned}$$

When $\Psi_{i,j}^m \in \Omega$, using linear interpolation, values at the four points $\Psi_{i,j}^m$, $m = 1, 2, 3, 4$, are approximated as follows:

$$\mathcal{J}_h \mathcal{U}^{n+1}(\Psi_{i,j}^m) = \begin{cases} \displaystyle\sum_{\substack{d=0,1 \\ e=0,1}} \omega_{i,j}^{f_m+d,g_m+e} \mathcal{U}_{f_m+d,g_m+e}^{n+1}, & \Psi_{i,j}^m \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{3.27}$$

For linear interpolation, we have that

$$\omega_{i,j}^{f_m+d,g_m+e} \geq 0 \quad \text{and} \quad \sum_{\substack{d=0,1\\e=0,1}} \omega_{i,j}^{f_m+d,g_m+e} = 1.$$

Then, inserting (3.27) into (3.14), the entries $\boldsymbol{L}_{\ell,k}^{n+1}(\mathcal{P})$ on ℓth row are specified. When we use $\Psi_{i,j}^m \in \Omega_{\text{out}}$, we directly use its asymptotic solution $\bar{\mathcal{U}}(\Psi_{i,j}^m)$ (3.4). Thus, we need to define the vector $\boldsymbol{G}^{n+1}(\mathcal{P})$ to facilitate the construction of the matrix form in this situation when we use a point in the domain Ω_{out}:

$$\boldsymbol{G}_\ell^{n+1}(\mathcal{P}) = \begin{cases} 1_{\Psi_{i,j}^1 \in \Omega_{\text{out}}} \dfrac{a_{i,j}}{h} \bar{\mathcal{U}}(\Psi_{i,j}^1) + 1_{\Psi_{i,j}^2 \in \Omega_{\text{out}}} \dfrac{a_{i,j}}{h} \bar{\mathcal{U}}(\Psi_{i,j}^2) \\ \qquad + 1_{\Psi_{i,j}^3 \in \Omega_{\text{out}}} \dfrac{b_{i,j}}{h} \bar{\mathcal{U}}(\Psi_{i,j}^3) + 1_{\Psi_{i,j}^4 \in \Omega_{\text{out}}} \dfrac{b_{i,j}}{h} \bar{\mathcal{U}}(\Psi_{i,j}^4), & (w_i, v_j) \in \Omega_{\text{in}}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.28}$$

where $a_{i,j}$ and $b_{i,j}$ are defined in (3.11). As a result, for the case $(w_i, v_j) \in \Omega_{\text{in}}$,

$$[\boldsymbol{L}^{n+1}(\mathcal{P})\boldsymbol{U}^{n+1}]_\ell + \boldsymbol{G}_\ell^{n+1}(\mathcal{P}) = L_h^p \mathcal{U}_{i,j}^{n+1}, \tag{3.29}$$

where L_h^p is defined in (3.14).

Let $\Phi^{n+1}(\mathcal{P})$ be a linear Lagrange interpolation operator such that

$$[\Phi^{n+1}(\mathcal{P})\boldsymbol{U}]_l = \begin{cases} \mathcal{J}_h \mathcal{U}_{i^*,j}^n, & (w_{i^*}, v_j) \in \Omega, \\ \bar{\mathcal{U}}(w_{i^*})\ (3.4), & (w_{i^*}, v_j) \in \Omega_{\text{out}}, \end{cases} \tag{3.30}$$

where w_{i^*} is defined in (3.18).

The final matrix form of the discretized equations is then

$$\left.\begin{aligned} [\boldsymbol{I} - \Delta\tau^n \boldsymbol{L}^{n+1}(\hat{\mathcal{P}})]\boldsymbol{U}^{n+1} &= \Phi^{n+1}(\mathcal{P})\boldsymbol{U}^n + \Delta\tau^n \boldsymbol{G}^{n+1}(\mathcal{P}) + \boldsymbol{F}^{n+1} - \boldsymbol{F}^n, \\ \hat{p}_\ell \in \arg\min_{p \in \mathcal{Z}_h} &[\Phi^{n+1}(\mathcal{P})\boldsymbol{U}^n + \Delta\tau^n(\boldsymbol{L}^{n+1}(\mathcal{P})\boldsymbol{U}^{n+1} + \boldsymbol{G}^{n+1}(\mathcal{P}))]_\ell, \\ \ell = i + (j-1)N_1, \quad i = 2, \ldots, N_1 - 1, \quad & j = 2, \ldots, N_2, \end{aligned}\right\} \tag{3.31}$$

where $\mathcal{Z}_h$ is the discretized control set $\mathcal{Z}$.

REMARK 3.3 Note that $[\boldsymbol{I} - \Delta\tau^n \boldsymbol{L}^{n+1}(\mathcal{P})]_{\ell,k}$, $[\Phi^{n+1}(\mathcal{P})]_\ell$ and $[\boldsymbol{G}^{n+1}(\mathcal{P})]_\ell$ depend only on p_ℓ.

3.3.1 The discrete linear operator D_h^p

With vanishing cross-derivative term, the degenerate linear operator $\mathcal{L}^p$ (3.13) can be discretized by a standard finite difference method. The degenerate linear operators $\mathcal{L}^p$ in (3.1)–(3.3) are approximated as the discrete form

$$\begin{aligned} D_h^p \mathcal{U}_{i,j}^n = {} & \alpha_{i,j}^w \mathcal{U}_{i-1,j}^n + \beta_{i,j}^w \mathcal{U}_{i+1,j}^n + \alpha_{i,j}^v \mathcal{U}_{i,j-1}^n \\ & + \beta_{i,j}^v \mathcal{U}_{i,j+1}^n - (\alpha_{i,j}^w + \beta_{i,j}^w + \alpha_{i,j}^v + \beta_{i,j}^v) \mathcal{U}_{i,j}, \end{aligned} \tag{3.32}$$

where $\alpha_{i,j}^w$, $\beta_{i,j}^w$, $\alpha_{i,j}^v$ and $\beta_{i,j}^v$ are defined as follows:

$$\left.\begin{aligned} \alpha_{i,j}^w &= \frac{(\sqrt{v}\rho w_i)^2}{(w_i - w_{i-1})(w_{i+1} - w_{i-1})}, \\ \beta_{i,j}^w &= \frac{(\sqrt{v}\rho w_i)^2}{(w_{i+1} - w_i)(w_{i+1} - w_{i-1})}, \\ \alpha_{i,j}^v &= \left[\frac{(\sigma\sqrt{v_j})^2}{(v_j - v_{j-1})(v_{j+1} - v_{j-1})} + \max\left(0, -\frac{\kappa(\theta - v_j)}{v_j - v_{j-1}}\right) \right], \\ \beta_{i,j}^v &= \left[\frac{(\sigma\sqrt{v_j})^2}{(v_{j+1} - v_j)(v_{j+1} - v_{j-1})} + \max\left(0, \frac{\kappa(\theta - v_j)}{v_{j+1} - v_j}\right) \right]. \end{aligned}\right\} \tag{3.33}$$

The coefficients $\alpha_{i,j}^w$, $\beta_{i,j}^w$, $\alpha_{i,j}^v$ and $\beta_{i,j}^v$ are all nonnegative, and they are compatible with a monotone scheme. On the upper boundary $v = v_{\max}$, the coefficients α_{i,N_2}^v and $\beta_{i,N_2}^v = 0$ degenerate to zero; on the lower boundary $w = 0$, $\alpha_{1,j}^w$ and $\beta_{1,j}^w$ are set to zero. On the lower boundary $v = 0$, $\alpha_{i,1}^w = 0$, $\beta_{i,1}^w = 0$, $\alpha_{i,1}^v = 0$ and $\beta_{i,1}^w = \kappa\theta/(v_{j+1} - v_j)$, $j = 1$.

3.4 Convergence to the viscosity solution

ASSUMPTION 3.4 *If the control p is bounded, (2.15) satisfies the strong comparison property; hence, a unique continuous viscosity solution to (2.15) exists (Debrabant and Jakobsen 2013).*

Provided that the original HJB satisfies Assumption 3.4, we can show that the numerical scheme (3.31) is ℓ_∞ stable, consistent and monotone, and then the scheme converges to the unique and continuous viscosity solution (Barles and Souganidis 1991). We give a brief overview of the proof as follows.

- **Stability:** from the formation of matrix $\boldsymbol{L}$ in (3.24), (3.25) and (3.29), it is easily seen that $[\boldsymbol{I} - \Delta\tau \boldsymbol{L}^{n+1}(\mathcal{P})]$ (3.31) has positive diagonals and non-positive off-diagonals, and the ℓth row sum for the matrix is

$$\sum_k [\boldsymbol{I} - \Delta\tau \boldsymbol{L}^{n+1}(\mathcal{P})]_{\ell,k} > 0, \quad i = 1, \ldots, N_1, \ j = 1, \ldots, N_2, \tag{3.34}$$

where $\ell = i + (j-1)N_1$; hence, the matrix $[\boldsymbol{I} - \Delta\tau \boldsymbol{L}^{n+1}(\mathcal{P})]$ is diagonally dominant, and thus it is an M matrix (Varga 2009). We can then easily show that the numerical scheme is l_∞ stable by a straightforward maximum analysis, as in d'Halluin *et al* (2004).

- **Monotonicity:** to guarantee monotonicity, we use a wide stencil to discretize the second derivative terms in the discrete linear operator L_h^p (3.14) (see proof in Ma and Forsyth (2014)). Note that using linear interpolation to compute $\mathcal{U}^n_{i^*,j}$ (3.19) in the semi-Lagrangian time-stepping scheme also ensures monotonicity.

- **Consistency:** a simple Taylor series verifies consistency. As noted in Section 4.3, we may shrink the wide stencil length to avoid using points below the lower boundaries. We can use the same proof as in Ma and Forsyth (2014) to show that this treatment retains local consistency. Since we have either simple Dirichlet boundary conditions, or the PDE at the boundary is the limit from the interior, we need only use the classical definition of consistency here (see proof in Ma and Forsyth (2014)). The only case where the point $\mathcal{U}^n_{i^*,j}$ (3.19) in the semi-Lagrangian time-stepping scheme is outside the computational domain is through the upper boundary $w = w_{\max}$, where the asymptotic solution (3.4) is used. Thus, unlike the semi-Lagrangian time-stepping scheme in Chen and Forsyth (2007), we do not need the more general definition of consistency (Barles and Souganidis 1991) to handle the boundary data.

3.5 Policy iteration

Our numerical scheme requires the solution of highly nonlinear algebraic equations (3.31) at each time step. We use the policy iteration algorithm (Forsyth and Labahn 2007) to solve the associated algebraic systems. For the details of the algorithm, we refer the reader to Forsyth and Labahn (2007) and Huang *et al* (2012). Regarding the convergence of the policy iteration, since the matrix $[\boldsymbol{I} - \Delta\tau \boldsymbol{L}^{n+1}(\mathcal{P})]$ (3.31) is an M matrix and the control set $\mathcal{Z}_h$ is a finite set, it is easy to show that the policy iteration is guaranteed to converge (Forsyth and Labahn 2007).

4 IMPLEMENTATION DETAILS

4.1 Complexity

Examination of the algorithm for solving discrete equations (3.31) reveals that each time step requires the following steps.

- In order to solve the local optimization problems at each node, we perform a linear search to find the minimum for $p \in \mathcal{Z}_h$. Thus, with a total of $O(1/h^2)$

nodes, this gives a complexity $O(1/h^3)$ for solving the local optimization problems at each time step.

- We use a preconditioned Bi-CGSTAB iterative method for solving the sparse matrix at each policy iteration. The time complexity of solving the sparse M matrix is $O((1/h^2)^{5/4})$ (Saad 2003). Note that, in general, we need to reconstruct the data structure of the sparse matrix for each iteration.

Assuming that the number of policy iterations is bounded as the mesh size tends to zero, which is in fact observed in our experiments, the complexity of the time advance is thus dominated by the solution of the local optimization problems. Finally, the total complexity is $O(1/h^4)$.

4.2 The efficient frontier

In order to trace out the efficient frontier solution of problem (2.7), we proceed in the following way. Pick an arbitrary value of γ and solve problem (2.14), which determines the optimal control $p^*(\cdot)$. There are then two methods to determine the quantities of interest $(\mathrm{Var}^{x_0,0}_{p^*}[W(T)], E^{x_0,0}_{p^*}[W(T)])$, namely the PDE method and the hybrid (PDE–Monte Carlo) method. We will compare the performance of these methods in the numerical experiments.

4.2.1 The PDE method

For a fixed γ, given $\mathcal{U}(w_0, v_0, 0)$ and $\mathcal{E}(w_0, v_0, 0)$ obtained solving the corresponding equations (2.15) and (2.18) at the initial time with $W_0 = w_0$ and $V_0 = v_0$, we can then compute the corresponding pair $(\mathrm{Var}^{x_0,0}_{p^*(\cdot)}[W(T)], E^{x_0,0}_{p^*(\cdot)}[W(T)])$, where $x_0 = (w_0, v_0)$. That is,

$$
\begin{aligned}
E^{x_0,0}_{p^*(\cdot)}[W(T)] &= \mathcal{E}(w_0, x_0, 0),\\
\mathrm{Var}^{x_0,0}_{p^*(\cdot)}[W(T)] &= \mathcal{U}(w_0, v_0, 0) - \gamma\mathcal{E}(w_0, x_0, 0) - \tfrac{1}{4}\gamma^2 - \mathcal{E}(w_0, v_0, 0)^2,
\end{aligned}
$$

which gives us a single candidate point $\mathcal{Y}_{\mathcal{Q}(\gamma)}$. Repeating this for many values of γ gives us a set of candidate points.

We are effectively using the parameter γ to trace out the efficient frontier. From Theorem 2.3, we have that $\gamma = (1/\lambda) + 2\mathcal{E}_0$. If $\lambda \to \infty$, the investor is infinitely risk averse, and invests only the risk-free bond; hence, in this case, the smallest possible value of γ is

$$
\gamma_{\min} = 2w_0 \exp(rT). \tag{4.1}
$$

In practice, the interesting part of the efficient frontier is in the range $\gamma \in [\gamma_{\min}, 10\gamma_{\min}]$. Finally, the efficient frontier is constructed from the upper left convex hull of $\mathcal{Y}_{\mathcal{Q}}$ (Tse *et al* 2014) to remove spurious points. In our case, however,

it turns out that all the points are on the efficient frontier, and there are no spurious points, if $\gamma \geq \gamma_{\min}$.

4.2.2 The hybrid (PDE–Monte Carlo) discretization

In the hybrid method, given the stored optimal control $p^*(\cdot)$ from solving the HJB PDE (2.15), $(\mathrm{Var}^{x_0,0}_{p^*(\cdot)}[W(T)], \mathrm{Var}^{x_0,0}_{p^*(\cdot)}[W(T)])$ are then estimated by Monte Carlo simulations. We use the Euler scheme to generate the Monte Carlo simulation paths of the wealth (2.4), and an implicit Milstein scheme to generate the Monte Carlo simulation paths of the variance process (2.3). Starting with $W_0 = w_0$ and $V_0 = v_0$, the Euler scheme for the wealth process (2.4) is

$$W_{t+\Delta t} = W_t \exp((r + p^*\xi V_t - 0.5(p^*\sqrt{V_t})^2)\Delta t + p^*\sqrt{V_t \Delta t}\phi_1), \tag{4.2}$$

and the implicit Milstein scheme of the variance process (2.3) (Kahl and Jäckel 2006) is

$$V_{t+\Delta t} = \frac{V_t + \kappa\theta\Delta t + \sigma\sqrt{V_t\Delta t}\phi_2 + \frac{1}{4}\sigma^2\Delta t(\phi_2^2 - 1)}{1 + \kappa\Delta t}, \tag{4.3}$$

where ϕ_1 and ϕ_2 are standard normal variables with correlation ρ. Note that this discretization scheme will result in strictly positive paths for the variance process if $4\kappa\theta > \sigma^2$ (Kahl and Jäckel 2006). For the cases where this bound does not hold, it will be necessary to modify (4.3) to prevent problems with the computation of $\sqrt{V_t}$. For instance, whenever V_t drops below zero, we could use the Euler discretization

$$V_{t+\Delta t} = V_t + \kappa(\theta - V_t^+)\Delta t + \sigma\sqrt{V_t^+}\sqrt{\Delta t}\phi_2, \tag{4.4}$$

where $V_t^+ = \max(0, V_t)$. Lord *et al* (2010) reviews a number of similar remedies to get around the problem of when V_t becomes negative and concludes that the simple fix (4.4) works best.

4.3 Outside the computational domain

To make the numerical scheme consistent in a wide stencil method (Section 3.2.1), the stencil length needs to be increased to use the points beyond the nearest neighbors of the original grid. Therefore, when solving the PDE in a bounded region, the numerical discretization may require points outside the computational domain. When a candidate point we use is outside the computational region at the upper boundaries, we can directly use its asymptotic solution (3.4). For a point outside the upper boundary $w = w_{\max}$, the asymptotic solution is specified by (3.4). For a point outside the upper boundary $v = v_{\max}$, by the implication of the boundary condition $\mathcal{U}_v = 0$ on $v = v_{\max}$, we have

$$\mathcal{U}(w, v, \tau) = \mathcal{U}(w, v_{\max}, \tau), \quad v > v_{\max}. \tag{4.5}$$

However, we have to take special care when we may use a point below the lower boundaries $w = 0$ or $v = 0$, because (2.15) is defined over $[0, \infty] \times [0, \infty]$. The possibility of using points below the lower boundaries only occurs when the node (i, j) falls in a possible region close to the lower boundaries

$$[h, \sqrt{h}] \times (0, w_{\max}] \cup (0, v_{\max}] \times [h, \sqrt{h}],$$

as discussed in Ma and Forsyth (2014). We use the algorithm proposed in Ma and Forsyth (2014) so that only information within the computational domain is used. That is, when one of the four candidate points $x_{i,j} \pm \sqrt{h}(\boldsymbol{R}_{i,j})_k, k = 1, 2$, (3.14) is below the lower boundaries, we then shrink its corresponding distance (from the reference node (i, j)) to h, instead of the original distance $\sqrt{h}$. This simple treatment ensures that all data required is within the domain of the HJB equation. It is straightforward to show that this discretization is consistent (Ma and Forsyth 2014).

In addition, due to the semi-Lagrangian time stepping (Section 3.2.2), we may need to evaluate the value of an off-grid point $(w_{i*} = w_i \mathrm{e}^{(r-p\xi v_j)\Delta\tau^n}, v_j)$ (3.18). This point may be outside the computational domain through the upper boundary $w = w_{\max}$ (the only possibility). When this situation occurs, the asymptotic solution (3.4) is used.

4.4 An improved linear interpolation scheme

When solving the value function problem (2.15) or the expected value problem (2.18) on a computational grid, it is required to evaluate $\mathcal{U}(\cdot)$ and $\mathcal{E}(\cdot)$, respectively, at points other than a node of the computational grid. This is especially important when using semi-Lagrangian time stepping. Hence, interpolation must be used. As mentioned earlier, to preserve the monotonicity of the numerical schemes, linear interpolation for an off-grid node is used in our implementation. Dang and Forsyth (2014b) introduce a special linear interpolation scheme applied along the w direction to significantly improve the accuracy of the interpolation in a two-dimensional impulse control problem. We modify this algorithm in our problem setup.

We then take advantage of the results in Section 3.1.3 to improve the accuracy of the linear interpolation. Assume that we want to proceed from time step τ^n to τ^{n+1}, and that we want to compute $\mathcal{U}(\bar{w}, v_j, \tau^n)$, where $\bar{w}$ is neither a grid point in the w direction nor the special value $W_{\text{opt}}(T - \tau^n)$, and W_{opt} is defined in (3.5). Further, assume that $w_k < \bar{w} < w_{k+1}$ for some grid points w_k and w_{k+1}. For presentation purposes, let $w_{\text{special}} = W_{\text{opt}}(T - \tau^n)$ and $\mathcal{U}_{\text{special}} = 0$. An improved linear interpolation scheme along the w direction for computing $\mathcal{U}(\bar{w}, v_j, \tau^n)$ is shown in Algorithm 1. Note that the interpolation along the v direction is a plain

Algorithm 1 Improved linear interpolation scheme along the w direction for the function value problem.

1: **if** $w_{\text{special}} < w_k$ or $w_{\text{special}} > w_{k+1}$ **then**
2: set $w_{\text{left}} = w_k$, $\mathcal{U}_{\text{left}} = \mathcal{U}^n_{k,j}$, $w_{\text{right}} = w_{k+1}$ and $\mathcal{U}^n_{\text{right}} = \mathcal{U}^n_{k+1,j}$
3: **else**
4: **if** $w_{\text{special}} < \bar{w}$ **then**
5: set $w_{\text{left}} = w_{\text{special}}$, $\mathcal{U}_{\text{left}} = \mathcal{U}_{\text{special}}$, $w_{\text{right}} = w_{k+1}$ and $\mathcal{U}^n_{\text{right}} = \mathcal{U}^n_{k+1,j}$
6: **else**
7: set $w_{\text{left}} = w_k$, $\mathcal{U}_{\text{left}} = \mathcal{U}^n_{k,j}$, $w_{\text{right}} = w_{\text{special}}$ and $\mathcal{U}^n_{\text{right}} = \mathcal{U}_{\text{special}}$
8: **end if**
9: **end if**
10: Apply linear interpolation to $(w_{\text{left}}, \mathcal{U}_{\text{left}})$ and $(w_{\text{right}}, \mathcal{U}_{\text{right}})$ to compute $\mathcal{U}(\bar{w}, v_j, \tau^n)$

linear interpolation; thus, we only illustrate the interpolation algorithm in the w direction.

Following the same line of reasoning used for the function value problem, we have that

$$\mathcal{E}(v, W_{\text{opt}}(t), t) = \tfrac{1}{2}\gamma.$$

By using this result, a similar method to Algorithm 1 can be used to improve the accuracy of linear interpolation when computing the expected value $\mathcal{E}(\bar{w}, v_j, \tau^n)$.

REMARK 4.1 For the discretization of the expected value problem (2.18), we still use semi-Lagrangian time stepping to handle the drift term $(r + p^*\xi v)w\mathcal{E}_w$. Since it may be necessary to evaluate $\mathcal{E}^n_{i^*,j}$ at points other than a node of the computational grid, we need to use linear interpolation.

5 NUMERICAL EXPERIMENTS

In this section, we present the numerical results of the solution of (2.15) applied to the continuous-time mean–variance portfolio allocation problem. In our problem, the risky asset (2.2) follows the Heston model. The parameter values of the Heston model used in our numerical experiments are taken from Aït-Sahalia and Kimmel (2007) and based on empirical calibration from the Standard & Poor's 500 (S&P 500) and VIX index data sets during 1990 to 2004 (under the real probability measure). Table 2 lists the Heston model parameters, and Table 3 lists the parameters of the mean–variance portfolio allocation problem.

TABLE 2 Parameter values in the Heston model.

κ	θ	σ	ρ	ξ
5.07	0.0457	0.48	−0.767	1.605

TABLE 3 Input parameters for the mean–variance portfolio allocation problem.

Investment horizon T	10
Risk-free rate r	0.03
Leverage constraint $p_{\max}$	2
Initial wealth w_0	100
Initial variance v_0	0.0457

TABLE 4 Grid and time step refinement levels used during numerical tests.

Refinement	Time steps	W nodes	V nodes	$\mathcal{Z}_h$ nodes
0	160	112	57	8
1	320	223	113	15
2	640	445	225	29
3	1280	889	449	57

On each refinement, a new grid point is placed halfway between all old grid points, and the number of time steps is doubled. A constant time step size is used. $w_{\max} = 6 \times 10^6$ and $v_{\max} = 3.0$. The number of finitely sampled γ points is 50. Note that increasing $w_{\max}$ by an order of magnitude and doubling $v_{\max}$ results in no change to the points on the efficient frontier to five digits. Increasing the number of γ points did not result in any appreciable change to efficient frontier (no spurious points in this case).

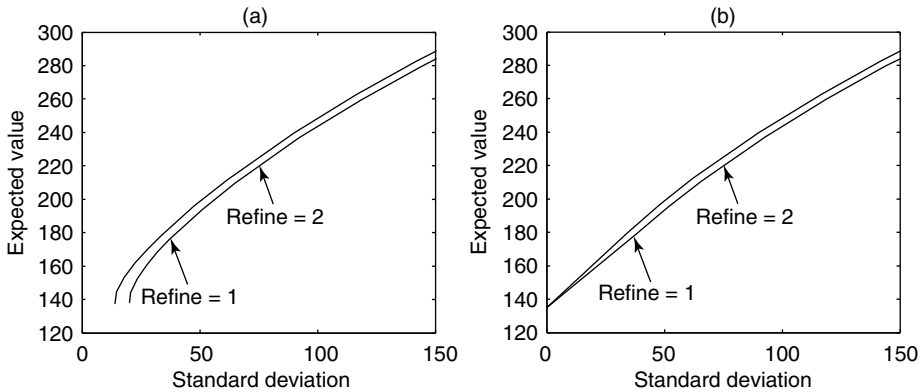
For all the experiments, unless otherwise noted, the details of the grid, the control set and time step refinement levels used are given in Table 4.

5.1 Effects of the improved interpolation scheme for the PDE method

In this subsection, we discuss the effects on numerical results of the linear interpolation scheme described in Section 4.4. We plot expected values against standard deviation, since both variables have the same units. Figure 1(a) illustrates the numerical efficient frontiers obtained using standard linear interpolation. It is clear that the results are very inaccurate for small standard deviations. It appears that the numerical methods were not able to construct the known point on the exact efficient frontier

$$(\mathrm{Var}^{x,t}_{p^*(\cdot)}[W(T)],\, E^{x,t}_{p^*(\cdot)}[W(T)]) = (0, w_0 \mathrm{e}^{rT}) \approx (0, 134.9859).$$

FIGURE 1 Close-up of efficient frontier for small standard deviations.



(a) No special interpolation. (b) Special interpolation.

This trivial case corresponds to the case where $\gamma = \gamma_{\min}$ (4.1), and the investor invests only in the risk-free bond and not in the risky asset. However, as shown in Figure 1(a), in this special case, the standard deviation obtained by the numerical scheme using standard linear interpolation is far from the exact solution.

Figure 1(b) shows the numerical efficient frontiers obtained with the improved linear interpolation scheme, where Algorithm 1 is utilized. It is obvious that the numerical efficient frontiers obtained with the improved linear interpolation scheme are more reasonable, especially for the small standard deviation region. In particular, the special point at which the variance is zero is now approximated accurately. This result illustrates the importance of using the optimal embedded terminal wealth $W_{\text{opt}}(t)$ and its function value for linear interpolation in constructing accurate numerical efficient frontiers. In all our numerical experiments in the following, the improved linear interpolation scheme is used.

5.2 Convergence analysis

In this section, we illustrate the convergence of our numerical scheme and compare the performance of two methods, namely the PDE method (Section 4.2.1) and the hybrid method (4.2.2), for constructing the mean–variance frontier under our model setup.

Figure 2 shows that the mean standard deviation efficient frontiers computed by both the PDE method and the hybrid method converge to the same frontier as the computational grid is refined. Our numerical results demonstrate that the hybrid frontiers in general converge faster to the limit results than the pure PDE solutions. This same

TABLE 5 The convergence table for the PDE method.

Refine	Mean	Change	Ratio	SD	Change	Ratio
0	207.1434			71.3924		
1	210.4694	3.3260		65.5090	−5.88336	
2	212.1957	1.7263	1.92	62.0862	−3.42288	1.72
3	213.1481	0.95238	1.81	60.4738	−1.61237	2.12

Small standard deviation (SD) case with $\gamma = 540$.

TABLE 6 The convergence table for the hybrid method.

Refine	Mean	Change	Ratio	SD	Change	Ratio
0	212.2993			56.6128		
1	213.2077	0.908		57.7652	1.152	
2	213.7573	0.550	1.65	58.2987	0.534	2.16
3	213.9903	0.233	2.36	58.5253	0.227	2.35

Small standard deviation (SD) case with $\gamma = 540$.

TABLE 7 The convergence table for the PDE method.

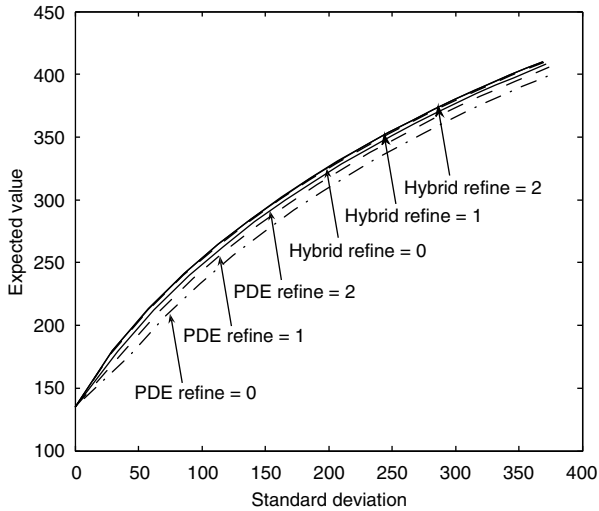
Refine	Mean	Change	Ratio	SD	Change	Ratio
0	320.5139			217.0009		
1	325.5443	5.030		212.1886	−4.812	
2	328.2670	2.723	1.85	209.8434	−2.345	2.05
3	329.8172	1.550	1.76	208.9045	−0.939	2.50

Large standard deviation (SD) case with $\gamma = 1350$.

phenomenon was observed in Tse *et al* (2013). As shown in Figure 2, the frontiers obtained by the hybrid method are almost identical for refinement levels 1 and 2. Note that for both methods the optimal control is always computed by solving the HJB PDEs.

The same time steps are used in both the PDE method and Monte Carlo simulations for each refinement level. For example, the frontiers labeled with "Refine $= 1$" for both methods in Figure 2 use the time steps specified as "Refinement 1" in Table 4. To achieve small sampling error in Monte Carlo simulations, 10^6 simulations are performed for the numerical experiments. The standard error in Figure 2 can then be estimated. For example, consider a point on the frontier with a large standard deviation value that is about 350. For the expected value of $W(T)$, the sample error is approximately $350/\sqrt{10^6} \approx 0.35$, which could be negligible in Figure 2.

FIGURE 2 Convergence of frontiers in the PDE method and the hybrid method.



The frontiers labeled with "PDE" are obtained from the PDE method (Section 4.2.1). The frontiers labeled with "Hybrid" (Section 4.2.2) are obtained from a Monte Carlo simulation that uses the optimal controls determined by solving the HJB equation (2.15).

We will verify our conclusion by examining several specific points on these efficient frontiers in Figure 2. Tables 5 and 6 show computed means and standard deviations for different refinement levels when $\gamma = 540$. The numerical results indicate first-order convergence is achieved for both the PDE method and the hybrid method. In this case, our numerical results demonstrate that the hybrid frontiers converge faster to the limit results than the PDE solutions. Tables 7 and 8 show computed means and standard deviations for different refinement levels when $\gamma = 1350$. The numerical results indicate first-order convergence is achieved for the PDE method. In this case, our numerical results also demonstrate that the hybrid frontiers converge faster to the limit results than the PDE solutions. However, the convergence ratio for the hybrid method is erratic. As we noted before, in this case, the sample error for the estimate of the mean value is about $0.2 \simeq 200/\sqrt{10^6}$, which makes the convergence ratio estimates in Table 8 unreliable. To decrease the sample error to, for example, 0.02, the number of simulation paths would have to increase to 100×10^6, which is unaffordable in terms of the computational cost. Note that in the case $\gamma = 540$, with the small standard deviation, the sample error for the mean is about $0.05 \simeq 50/\sqrt{10^6}$.

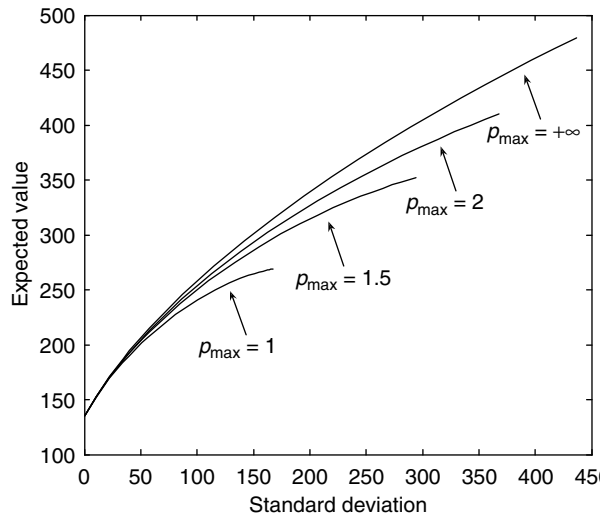
REMARK 5.1 (Efficiency of the hybrid method) We remind the reader that for both the hybrid and PDE methods, the same (computed) control is used. The more rapid

TABLE 8 The convergence table for the hybrid method.

Refine	Mean	Change	Ratio	SD	Change	Ratio
0	329.4411			206.0875		
1	330.5172	1.076		206.8351	0.748	
2	330.7066	0.189	5.68	207.1958	0.361	2.07
3	331.2820	0.575	0.33	207.3707	0.175	2.06

Large standard deviation (SD) case with $\gamma = 1350$.

FIGURE 3 Sensitivity analysis of the efficient frontiers with respect to different leverage constraints $p_{\max}$.



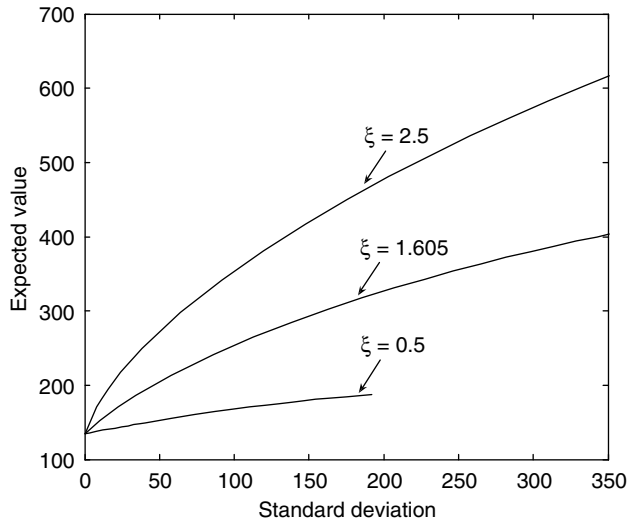
The Heston parameters and remaining model parameters are given in Tables 2 and 3. The hybrid method with discretization level 2 is used.

convergence of the hybrid method is simply due to a more accurate estimate of the expected quantities (with a known control). This result is somewhat counterintuitive, since it suggests that a low accuracy control can be used to generate high accuracy expected values. We also observe this from the fact that a fairly coarse discretization of the admissible set $\mathcal{Z}_h$ generates fairly accurate solutions.

5.3 Sensitivity of efficient frontiers

In this subsection, we show some numerical sensitivity analysis for the major market parameters, namely the leverage constraints $p_{\max}$, the market risk ξ, the mean

FIGURE 4 Sensitivity analysis of the efficient frontiers with respect to different risk premium factor ξ values.

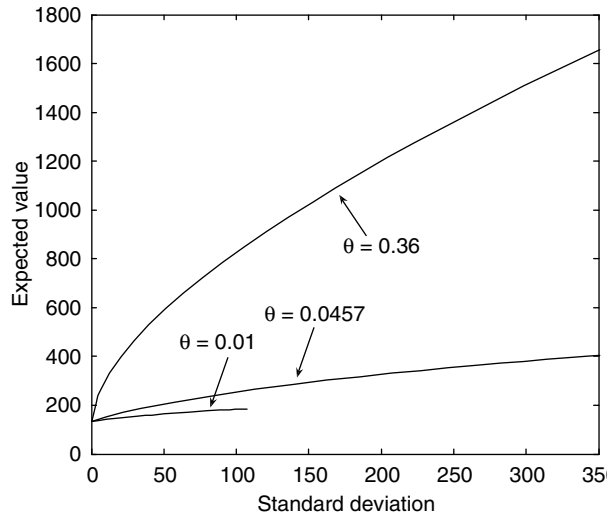


The remaining Heston parameters and model parameters are given in Tables 2 and 3. The hybrid method with discretization level 2 is used.

reversion level for the variance θ, the volatility of the variance σ, the correlation ρ between the risky asset and the variance, and the mean reversion speed κ. In our numerical tests, the corresponding frontiers are generated as the market parameter of interest changes, and the values of the remaining parameters are fixed and listed in Tables 2 and 3. We use the hybrid method with discretization level 2.

As observed in Figure 3, with $p_{\max} = \{1, 1.5, 2, +\infty\}$, we can see that larger values of the leverage constraints $p_{\max}$ result in much more dominant efficient frontiers. From Figure 4, with $\xi = \{0.5, 1.605, 2.5\}$, we can see that larger values of ξ result in much more dominant efficient frontiers. The maximal standard deviation point ($\gamma = +\infty$) on the efficient frontier with $\xi = 0.5$ is only about 191, which is much smaller than those with larger ξ values. From Figure 5, $\theta = \{0.01, 0.0457, 0.36\}$, we can see that larger values of the mean reversion level θ for the variance result in much more dominant efficient frontiers. The maximal standard deviation point ($\gamma = +\infty$) on the efficient frontier with $\theta = 0.01$ is only about 108, which is much smaller than those with larger θ values. From Figure 6, $\sigma = \{0.2, 0.48, 0.7\}$, we can see that larger values of the volatility of the variance σ result in slightly more dominant, efficient frontiers in general. In particular, these

FIGURE 5 Sensitivity analysis of the efficient frontiers with respect to different mean reversion level θ values.



The remaining Heston parameters and model parameters are given in Tables 2 and 3. The hybrid method with discretization level 2 is used.

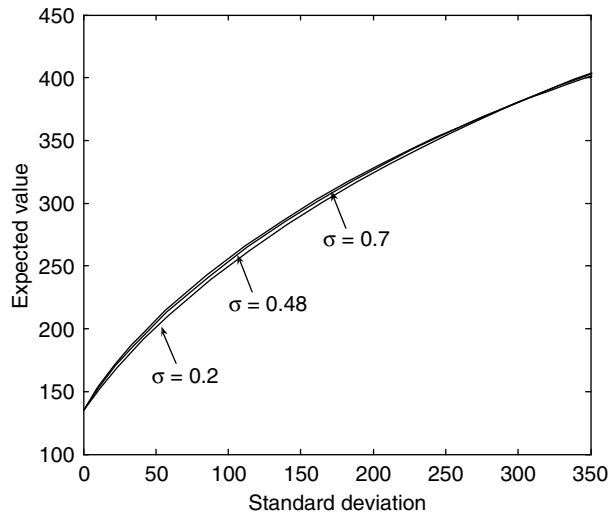
efficient frontiers in the large standard deviation region with different σ values are almost identical.

However, from Figure 7, with $\rho = \{-0.767, -0.3, 0\}$, we can see that an increase in the correlation ρ produces frontiers with a slightly smaller expected value for a given standard deviation. These efficient frontiers in the large standard deviation region with different ρ values are almost identical. The effect of the κ values on the efficient frontiers is more complex. From Figure 8, $\kappa = \{1, 5.07, 20\}$, in the small standard deviation region, an increase in κ produces frontiers with a smaller expected value for a given standard deviation. However, when the standard deviation increases to about 230, the larger values of κ gradually result in more significant dominant efficient frontiers.

5.4 Comparison between constant volatility and stochastic volatility cases

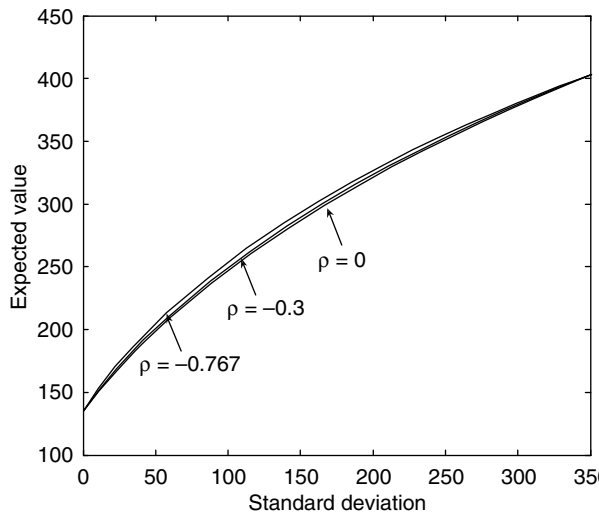
In this paper, the risky asset follows the stochastic volatility model ((2.2) and (2.3)). In this section, we will compare the constant volatility and stochastic volatility cases in terms of mean–variance efficiency for the continuous-time pre-commitment mean–variance problem. With a constant volatility, the risky asset is governed by the

FIGURE 6 Sensitivity analysis of the efficient frontiers with respect to different σ values.



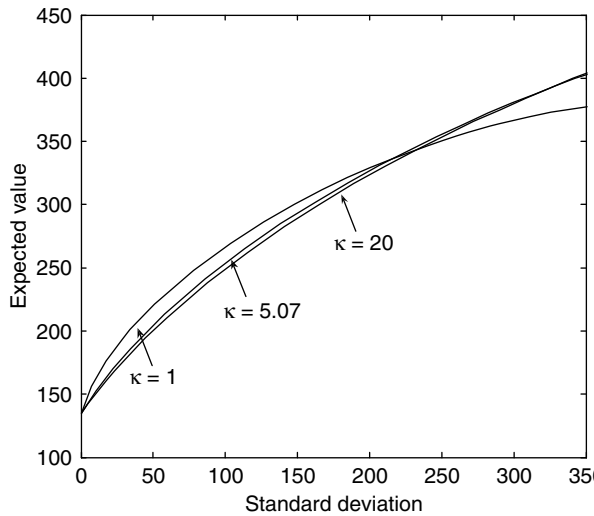
The remaining Heston parameters and model parameters are given in Tables 2 and 3. The hybrid method with discretization level 2 is used.

FIGURE 7 Sensitivity analysis of the efficient frontiers with respect to different ρ values.



The remaining Heston parameters and model parameters are given in Tables 2 and 3. The hybrid method with discretization level 2 is used.

FIGURE 8 Sensitivity analysis of the efficient frontiers with respect to different κ values.



The Heston parameters and remaining model parameters are given in Tables 2 and 3. The hybrid method with discretization level 2 is used.

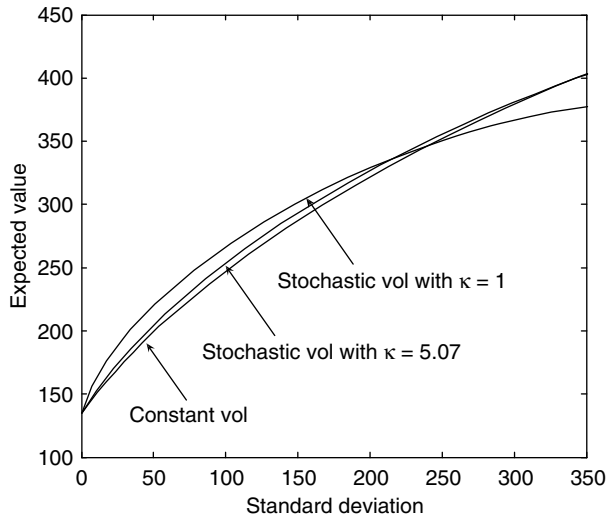
following GBM process:

$$\frac{\mathrm{d}S}{S} = (r + \mu)\,\mathrm{d}t + \sigma_S\,\mathrm{d}Z_s. \tag{5.1}$$

To compare this with the stochastic volatility case in Table 2, the constant volatility σ_S is set to $\sqrt{\theta} \approx 0.2138$, and the risky return over the risk-free rate μ is set to $\xi\sigma_S^2 = 0.0733485$, which has the same mean premium of the volatility risk as the stochastic volatility model (2.2). This then corresponds to the case where the variance $V(t)$ in (2.2) is fixed to the mean reversion level θ. The remaining mean–variance problem parameters are the same as those listed in Table 3.

Figure 9 illustrates the fact that the efficient frontiers produced by using the stochastic volatility slightly dominate the curve produced by the constant volatility model. With the Heston model's parameters in Table 2, we may conclude that the efficient frontier produced by the constant volatility is a good approximation of the frontier generated by the stochastic volatility model. From Figure 9, however, we see that if the mean reversion speed κ is set to a small value, eg, 1, in the stochastic volatility case, the efficient frontiers computed using a constant volatility model will be considerably different from those computed using the stochastic volatility model. The quantity $1/\kappa$ is measured in years and is related to the time over which a volatility shock dissipates. Specifically, the half-life of a volatility shock is $(\ln 2)/\kappa$.

FIGURE 9 Efficient frontier comparison between constant volatility and stochastic volatility cases.



For the stochastic volatility cases, $\kappa = 1, 5.07$; the remaining stochastic volatility parameters are given in Table 2. The GBM parameters are given in Section 5.4.

Finally, using the portfolio allocation strategy that is precomputed and stored from the constant volatility case, we then carry out a Monte Carlo simulation where the risky asset follows the stochastic volatility model. We then compare the results using this approximate control, with the optimal control computed using the full stochastic volatility model. From Table 9, we can see that the mean–variance pairs computed using the optimal strategy are very close to the strategy computed using the GBM approximation. Based on several tests, a good heuristic guideline is that if $\kappa T > 40$, then the GBM control is a good approximation to the exact optimal control.

6 CONCLUSION

In this paper, we develop an efficient fully numerical PDE approach for the pre-commitment continuous-time mean–variance asset allocation problem when the risky asset follows a stochastic volatility model. We use the wide stencil method (Ma and Forsyth 2014) to overcome the main difficulty in designing a monotone approximation. We show that our numerical scheme is monotone, consistent and ℓ_∞-stable. Hence, the numerical solution is guaranteed to converge to the unique viscosity solution of the corresponding HJB PDE, assuming that the HJB PDE satisfies a strong

TABLE 9 Given a γ, the optimal portfolio allocation strategy is computed and stored assuming a control process, which is either GBM or stochastic volatility (SV).

		$\gamma = 540$		$\gamma = 1350$	
Control process	**Price process**	**Mean**	**SD**	**Mean**	**SD**
GBM	GBM	209.50	59.68	330.09	213.01
GBM	SV	212.68	58.42	329.13	207.23
SV	SV	213.99	58.53	331.28	207.37

The mean–variance pairs are estimated by Monte Carlo simulation, using the stored controls, assuming that the actual price process follows either GBM or stochastic volatility. For the stochastic volatility case, the parameters are given in Table 2. For the GBM case, the variance is fixed to the mean value of the stochastic volatility case. SD denotes standard deviation.

comparison property. Further, using semi-Lagrangian time stepping to handle the drift term, along with an improved method of linear interpolation, allows us to compute accurate efficient frontiers. When tracing out the efficient frontier solution of our problem, we demonstrate that the hybrid (PDE–Monte Carlo) method (Tse *et al* 2013) converges faster than the pure PDE method. Similar results are observed in Tse *et al* (2013). Finally, if the mean reversion time $1/\kappa$ is small compared with the investment horizon T, then a constant volatility GBM approximation to the stochastic volatility process gives a very good approximation to the optimal strategy.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper. This work was supported by the Bank of Nova Scotia and the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Aït-Sahalia, Y., and Kimmel, R. (2007). Maximum likelihood estimation of stochastic volatility models. *Journal of Financial Economics* **83**(2), 413–452 (http://doi.org/b3c4dg).

Barles, G., and Souganidis, P. E. (1991). Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis* **4**(3), 271–283.

Basak, S., and Chabakauri, G. (2010). Dynamic mean–variance asset allocation. *Review of Financial Studies* **23**(8), 2970–3016 (http://doi.org/cqnc3g).

Bauerle, N., and Grether, S. (2015). Complete markets do not allow free cash flow streams. *Mathematical Methods of Operations Research* **81**, 137–145 (http://doi.org/bkpx).

Bielecki, T. R., Jin, H., Pliska, S. R., and Zhou, X. Y. (2005). Continuous-time mean–variance portfolio selection with bankruptcy prohibition. *Mathematical Finance* **15**(2), 213–244 (http://doi.org/cj3svf).

Bjork, T., and Murgoci, A. (2010). A general theory of Markovian time inconsistent stochastic control problems. SSRN Working Paper (http://doi.org/fzcfp2).

Breeden, D. T. (1979). An intertemporal asset pricing model with stochastic consumption and investment opportunities. *Journal of Financial Economics* **7**(3), 265–296 (http://doi.org/c8t9tr).

Chen, Z. L., and Forsyth, P. A. (2007). A semi-Lagrangian approach for natural gas storage valuation and optimal operation. *SIAM Journal on Scientific Computing* **30**(1), 339–368 (http://doi.org/bqnf83).

Cox, J. C., Ingersoll, J., Jonathan, E., and Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society* **53**(2), 385–407 (http://doi.org/cbb2pm).

Cui, X., Li, D., Wang, S., and Zhu, S. (2012). Better than dynamic mean–variance: time inconsistency and free cash flow stream. *Mathematical Finance* **22**(2), 346–378 (http://doi.org/dvr3gq).

Dang, D. M., and Forsyth, P. A. (2014a). Better than pre-commitment mean–variance portfolio allocation strategies: a semi-self-financing Hamilton–Jacobi–Bellman equation approach. *European Journal on Operational Research* **132**, 271–302 (http://doi.org/bkpk).

Dang, D. M., and Forsyth, P. A. (2014b). Continuous time mean–variance optimal portfolio allocation under jump diffusion: a numerical impulse control approach. *Numerical Methods for Partial Differential Equations* **30**(2), 664–698 (http://doi.org/bkpz).

Dang, D. M., Forsyth, P. A., and Li, Y. (2016). Convergence of the embedded mean–variance optimal points with discrete sampling. *Numerische Mathematik* **132**(2), 271–302 (http://doi.org/bkp2).

Debrabant, K., and Jakobsen, E. (2013). Semi-Lagrangian schemes for linear and fully non-linear diffusion equations. *Mathematics of Computation* **82**(283), 1433–1462 (http://doi.org/bkp3).

d'Halluin, Y., Forsyth, P. A., and Labahn, G. (2004). A penalty method for American options with jump diffusion processes. *Numerische Mathematik* **97**(2), 321–352 (http://doi.org/d5sn5v).

Douglas, J. J., and Russell, T. F. (1982). Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM Journal on Numerical Analysis* **19**(5), 871–885 (http://doi.org/dzghv5).

Ekström, E., and Tysk, J. (2010). The Black–Scholes equation in stochastic volatility models. *Journal of Mathematical Analysis and Applications* **368**(2), 498–507 (http://doi.org/drbjzd).

Feller, W. (1951). Two singular diffusion problems. *Annals of Mathematics* **54**(1), 173–182 (http://doi.org/bvf5m4).

Forsyth, P. A., and Labahn, G. (2007). Numerical methods for controlled Hamilton–Jacobi–Bellman PDEs in finance. *The Journal of Computational Finance* **11**(2), 1–44 (http://doi.org/bkp4).

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* **6**(2), 327–343 (http://doi.org/fg525s).

Huang, Y., Forsyth, P. A., and Labahn, G. (2012). Combined fixed point and policy iteration for HJB equations in finance. *SIAM Journal on Numerical Analysis* **50**(4), 1849–1860 (http://doi.org/bkp5).

Kahl, C., and Jäckel, P. (2006). Fast strong approximation Monte Carlo schemes for stochastic volatility models. *Quantitative Finance* **6**(6), 513–536 (http://doi.org/cqtf4z).

Li, D., and Ng, W.-L. (2000). Optimal dynamic portfolio selection: multiperiod mean–variance formulation. *Mathematical Finance* **10**(3), 387–406 (http://doi.org/dt4h2f).

Lord, R., Koekkoek, R., and Dijk, D. V. (2010). A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance* **10**(2), 177–194 (http://doi.org/bjjn3n).

Ma, K., and Forsyth, P. A. (2014). An unconditionally monotone numerical scheme for the two factor uncertain volatility model. *IMA Journal on Numerical Analysis* (forthcoming).

Nguyen, P., and Portait, R. (2002). Dynamic asset allocation with mean variance preferences and a solvency constraint. *Journal of Economic Dynamics and Control* **26**(1), 11–32 (http://doi.org/dgkfgc).

Pironneau, O. (1982). On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik* **38**(3), 309–332 (http://doi.org/dz4dc9).

Pooley, D. M., Forsyth, P. A., and Vetzal, K. R. (2003). Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis* **23**(2), 241–267 (http://doi.org/bq6t6z).

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM (http://doi.org/bq25g6).

Tse, S. T., Forsyth, P. A., Kennedy, J. S., and Windcliff, H. (2013). Comparison between the mean–variance optimal and the mean-quadratic-variation optimal trading strategies. *Applied Mathematical Finance* **20**(5), 415–449 (http://doi.org/bkp6).

Tse, S. T., Forsyth, P. A., and Li, Y. (2014). Preservation of scalarization optimal points in the embedding technique for continuous time mean variance optimization. *SIAM Journal on Control and Optimization* **52**(3), 1527–1546 (http://doi.org/bkp7).

Varga, R. S. (2009). *Matrix Iterative Analysis*, Volume 27. Springer.

Vigna, E. (2014). On efficiency of mean–variance based portfolio selection in defined contribution pension schemes. *Quantitative Finance* **14**(2), 237–258 (http://doi.org/bkp8).

Wang, J., and Forsyth, P. A. (2008). Maximal use of central differencing for Hamilton–Jacobi–Bellman PDEs in finance. *SIAM Journal on Numerical Analysis* **46**(3), 1580–1601 (http://doi.org/ffw87k).

Wang, J., and Forsyth, P. A. (2010). Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous time mean variance asset allocation. *Journal of Economic Dynamics and Control* **34**(2), 207–230 (http://doi.org/d2s8c8).

Wang, J., and Forsyth, P. A. (2012). Comparison of mean variance like strategies for optimal asset allocation problems. *International Journal of Theoretical and Applied Finance* **15**(2), 1250014 (http://doi.org/bhwk).

Zhao, Y., and Ziemba, W. T. (2000). Mean–variance versus expected utility in dynamic investment analysis. Working Paper, University of British Columbia.

Zhou, X. Y., and Li, D. (2000). Continuous-time mean–variance portfolio selection: a stochastic LQ framework. *Applied Mathematics and Optimization* **42**(1), 19–33 (http://doi.org/b6r9zv).

Journal of Computational Finance **20**(1), 39–87
DOI: 10.21314/JCF.2016.312

Risk journals

Research Paper

High-performance American option pricing

Leif Andersen, Mark Lake and Dimitri Offengenden

Bank of America Merrill Lynch, One Bryant Park, New York, NY 10036, USA;
emails: leif.andersen@baml.com, mark.lake@bankofamerica.com,
daoffengenden@gmail.com

(Received January 12, 2015; revised August 10, 2015; accepted September 15, 2015)

ABSTRACT

We develop a new high-performance spectral collocation method for the computation of American put and call option prices. The proposed algorithm involves a carefully posed Jacobi–Newton iteration for the optimal exercise boundary, aided by Gauss–Legendre quadrature and Chebyshev polynomial interpolation on a certain transformation of the boundary. The resulting scheme is straightforward to implement and converges at a speed several orders of magnitude faster than existing approaches. Computational effort depends on required accuracy; at precision levels similar to, say, those computed by a finite-difference grid with several hundred steps, the computational throughput of the algorithm in the Black–Scholes model is typically close to 100 000 option prices per second per CPU. For benchmarking purposes, Black–Scholes American option prices can generally be computed to ten or eleven significant digits in less than one-tenth of a second.

Keywords: American options; integral equations; high-speed collocation methods; fixed-point iterations; optimal exercise.

Corresponding author: L. Andersen

Print ISSN 1460-1559 | Online ISSN 1755-2850

1 INTRODUCTION

American put and call options trade on a large number of exchanges worldwide and cover many different asset classes, such as foreign exchange (FX), commodities and equities. The work undertaken in this paper was originally motivated by the practical problem of computing real-time risk for large portfolios of such options, especially the popular type of contract in which the exercise value references a futures contract.[1] Such applications often require thousands, if not tens of thousands, of option value computations per "tick" of the clock, so at their core necessarily lies a very fast method for the computation of American option premiums.

Although more sophisticated models have been proposed over the years, the current market practices for the pricing and relative-value analysis of listed American options still revolve almost exclusively around the standard Black–Scholes model.[2] Robust and well tested, this model has proven its mettle over decades, and it still offers sufficient flexibility for market participants to quote, position themselves and manage risk effectively. Indeed, as is the case for European options, quotation standards for exchange-traded American options are commonly based on Black–Scholes implied volatility.

To calculate American option prices (or implied volatility, for a known option price) in the Black–Scholes model, it is, unfortunately, necessary to rely on numerical methods, as no true closed-form American option price exists. For speed reasons, it is not uncommon to apply semi-analytical approximation methods for American options, such as those in the works of Barone-Adesi and Whaley (1987), Bunch and Johnson (2000), Ju and Zhong (1999) and many more.[3] However, while typically fast, these methods by their very nature involve inaccuracies that, even for the best algorithms, can be highly significant. Moreover, there is generally no possibility of using these approximations to make an intelligent, and application-specific, trade-off between speed and accuracy. That is, there is no way one can systematically increase the precision of a computed price by spending more computational resources on the problem.

---

[1] We are here primarily interested in the "true" American futures options traded in the United States (CME, CBOT) and Asia (SGC, TSE). The "American" options traded in Europe (LIFFE, EUREX, NLX) are equipped with a margining mechanism that effectively simplifies them to European options.

[2] Many over-the-counter American put and call options are, in fact, also priced with the Black–Scholes models, often with special-purpose grids for the "American" implied volatility. This, for instance, is common practice in equity markets, even for relatively long-dated puts and calls.

[3] A more complete list of papers on American option approximation methods can be found in Section 4.5.

There are, of course, many well-known methods that can produce American option prices to very high precision. A partial list includes binomial trees (see Cox *et al* 1979), "accelerated" binomial trees (see, for example, Joshi 2009; Leisen and Reimer 1996), finite-difference methods (see, for example, Brennan and Schwartz 1978; Forsyth and Vetzal 2002), the method of lines (Carr and Faguet 1994), least-squares Monte Carlo methods (see, for example, Andersen and Broadie 2004; Longstaff and Schwartz 2001) and numerical integral equation methods (Subrahmanyam and Yu (1993) and many others). Of those, Monte Carlo methods are no doubt the most flexible but also the slowest and noisiest; they are therefore of little relevance to our specific application, where speed and accuracy, rather than flexibility, is at a premium. Of the remaining methods, it is fair to say that the integral equation methods have not fared particularly well in head-to-head comparisons (see, for example, AitSahlia and Carr 1997) and are reputed to be slow – but this is nevertheless the approach that we shall pursue here.[4] We are motivated by excellent results in a fixed income setting (see Andersen 2007; Andreasen 2007) as well as by recent papers (see, for example, Cortazar *et al* 2013; Kim *et al* 2013) that show promising speed-accuracy performance for the integral equation approach.

In this paper, we refine earlier work in a number of ways in order to achieve levels of speed and accuracy far better than previous methods. Our approach is based on a spectral collocation method on a carefully transformed integral equation for the optimal exercise boundary. The boundary is found by a modified Jacobi–Newton function iteration, starting from an approximate guess. When run on a single run-of-the-mill 2GHz CPU, the numerical scheme in this paper will, without much optimization, produce accurate pricing precision at the level of a finite-difference grid solver with a few hundred steps (or a binomial tree with thousands of steps) at the rate of around 100 000 option prices per second, per CPU. If our goal is extreme precision, rather than speed, one hundredth of second of computation time with our approach can produce prices that are beyond the practical reach of a finite-difference grid, even when run with hundreds of thousands (or even millions) of steps.

The rest of this paper is organized as follows. In Section 2, we state process and payout assumptions and outline a series of basic results for American put options and their optimal exercise boundary. In particular, we list a series of equivalent integral equations for the optimal exercise boundary. Section 3 surveys known numerical methods for the boundary computations and provides a general analysis of iterative methods for the exercise boundary. In Section 4, this analysis is supplemented by a series of relevant theoretical results for the shape and asymptotics of the exercise boundary. Section 4 also discusses certain boundary approximation schemes, some

[4] For instance, in Chen and Chadam (2007) the integral equation method advocated by the authors takes in the order of minutes to solve on a Sun Sparc server.

of which are new. Armed with the material of Sections 3 and 4, we proceed to develop in detail our new numerical scheme in Section 5. Numerical tests of the scheme are provided in Section 6, and Section 7 outlines a variety of extensions of the method to different option payouts and to different stochastic processes for the underlying asset. Finally, Section 8 concludes the paper.

2 MODEL SETUP

2.1 Process definition

Consider a financial asset exhibiting constant lognormal volatility at an annualized rate of σ. Let r be a constant risk-free interest rate, and let $\beta(t) = \exp(rt)$ be the rolling money market numeraire. In the risk-neutral measure $\mathbb{Q}$ induced by β, let the asset process be given by a geometric Brownian motion stochastic differential equation (GBM SDE) of the form

$$\mathrm{d}S(t)/S(t) = \mu\,\mathrm{d}t + \sigma\,\mathrm{d}W(t), \tag{2.1}$$

where $W(t)$ is a $\mathbb{Q}$-Brownian motion. The constant drift μ in (2.1) is asset specific and given by arbitrage considerations. For instance, if S is a stock paying dividends at a continuous rate of q, we have $\mu = r - q$. Alternatively, if S represents a futures (or forward) price, we have $\mu = 0$. The case $\mu = 0$ is of particular practical importance, and it originally motivated most of the work in this paper, but for generality we work with the case $\mu = r - q$ going forward. Extensions to time-dependent parameters are possible, and these are discussed in Section 7.

2.2 The American put and its price formula

Within the setting of Section 2.1, we shall focus on a K-strike American put option, paying $(K - S(\nu))^+$ if exercised at time $\nu \in [0, T]$, with T being the terminal maturity of the put. Note that American call option prices can always be inferred from put prices through put–call symmetry (see, for example, McDonald and Schroder 1998). If not previously exercised, the time t no-arbitrage value of the put is given by

$$p(t) = \sup_{\nu \in [t,T]} \mathbb{E}_t(\mathrm{e}^{-r(\nu - t)}(K - S(\nu))^+), \tag{2.2}$$

where $\mathbb{E}_t(\cdot)$ denotes the time t expectation in $\mathbb{Q}$, and the supremum is taken over all stopping times on $[t, T]$. For the T-maturity American put on an asset following (2.1), it is known that the decision to exercise is characterized by a deterministic exercise boundary $S_T^*(t)$, in the sense that the optimal exercise policy ν^* (as seen at time 0) may be written

$$\nu^* = \inf(t \in [0, T]\colon S(t) \leqslant S_T^*(t)). \tag{2.3}$$

We emphasize that the exercise boundary $S_T^*(t)$ for the American put is indexed by option maturity T, as the decision to exercise at time t obviously depends on how much time is left before the option matures. For the constant-parameter process (2.1), however, it is easily seen that $S_T^*(t) = B(T-t)$ for some (time-reversed) boundary function $B\colon \mathbb{R} \to \mathbb{R}$ satisfying $B(0) = K$ and (see Section 4.1) $B(0+) = K\min(1, r/q)$.

By well-known arguments (see Kim (1990), Jacka (1991) and Jamshidian (1992), among several others), the exercise boundary, if known, may be used to compute the American put price through an integral equation. Specifically, if we let $V(T-t, S)$ be the time t price of the T-maturity American put when $S(t) = S$, then, for $\tau = T - t$,[5]

$$V(\tau, S) = v(\tau, S) + \int_0^\tau rK\mathrm{e}^{-r(\tau-u)}\Phi(-d_-(\tau-u, S/B(u)))\,\mathrm{d}u - \int_0^\tau qS\mathrm{e}^{-q(\tau-u)}\Phi(-d_+(\tau-u, S/B(u)))\,\mathrm{d}u, \quad (2.4)$$

where $v(\tau, S)$ is the European put option price, $\Phi(\cdot)$ is the cumulative Gaussian distribution function and

$$d_\pm(\tau, z) \triangleq \frac{\ln z + (r-q)\tau \pm \frac{1}{2}\sigma^2\tau}{\sigma\sqrt{\tau}}.$$

In (2.4), by the standard Black–Scholes result, we have

$$v(\tau, S) = \mathrm{e}^{-r\tau}K\Phi(-d_-(\tau, S/K)) - S\mathrm{e}^{-q\tau}\Phi(-d_+(\tau, S/K)). \quad (2.5)$$

Let us briefly discuss the intuition behind (2.4). First, transforming the integration variable back to calendar time, we notice that (with $\tau - u = s - t$, or $u = T - s$):

$$V(\tau, S) = v(\tau, S) + \int_t^T \mathbb{E}[\mathrm{e}^{-r(s-t)}(rK - qS(s))1_{S(s)<S_T^*(s)}\,\mathrm{d}s \mid S(t) = S]. \quad (2.6)$$

The term $(rK - qS(s))1_{S(s)\leqslant S_T^*(s)}\,\mathrm{d}s$ (which is always nonnegative) is the "carry" associated with the early exercise right of the American option. It represents the cashflow on the time interval $[s, s+\mathrm{d}s]$ that the American option holder would require to give up on their early exercise rights. Specifically, we recognize $rK\,\mathrm{d}s$ as the interest rate payment on the strike (long), and $-qS(s)\,\mathrm{d}s$ as the dividend payment on the stock (short). Integrating the present value of this stream of cashflows yields the American exercise premium. We note that (2.6) is more general than (2.4) and continues to hold for more complex Markovian dynamics for S than (2.1); we shall revisit this in Section 7.

[5] Such that $p(t) = V(T-t, S(t))$.

2.3 Integral equations for the boundary B

In order to apply (2.4), a methodology to construct the function $B(\tau)$ is needed. For this purpose, first notice that, for $S > B(\tau)$, $V(\tau, S)$ will satisfy the (time-reversed) Black–Scholes partial differential equation (PDE)

$$V_\tau - (r-q)SV_S - \tfrac{1}{2}S^2\sigma^2 V_{SS} + rV = 0, \qquad V(0,S) = (K-S)^+, \tag{2.7}$$

subject to the value match condition

$$V(\tau, B(\tau)) = K - B(\tau) \tag{2.8}$$

and the smooth pasting condition

$$V_S(\tau, B(\tau)) = -1. \tag{2.9}$$

It is worth noting that the fundamental conditions (2.8) and (2.9) may be combined with each other and with (2.7) to reveal new relations for the behavior of the American put at the exercise boundary. For instance, differentiating (2.8) with respect to τ and then using (2.9) yields

$$\frac{\mathrm{d}}{\mathrm{d}\tau}V(\tau, B(\tau)) = -\frac{\mathrm{d}}{\mathrm{d}\tau}B(\tau) = V_S(\tau, B(\tau))\frac{\mathrm{d}}{\mathrm{d}\tau}B(\tau).$$

On the other hand, by the chain rule,

$$\frac{\mathrm{d}}{\mathrm{d}\tau}V(\tau, B(\tau)) = V_\tau(\tau, B(\tau)) + V_S(\tau, B(\tau))\frac{\mathrm{d}}{\mathrm{d}\tau}B(\tau),$$

and it follows that

$$V_\tau(\tau, B(\tau)) = 0, \tag{2.10}$$

a result we believe was first presented in Bunch and Johnson (2000). Inserting (2.8), (2.9) and (2.10) into (2.7) then yields, in the limit $S \downarrow B(\tau)$,

$$V_{SS}(\tau, B(\tau)) = \frac{2(rK - qB(\tau))}{B(\tau)^2\sigma^2}. \tag{2.11}$$

Combining any of the four relations (2.8), (2.9), (2.10) and (2.11) with the basic expression (2.4) will yield a different integral equation for $B(\tau)$.[6] The most common

[6] By forming additional derivatives (including cross-derivatives such as $\partial^2 V/\partial S\partial\tau$), many other integral equations are obviously possible. We may also "blend" different integral equations with a set of mixing weights.

representation in the financial literature uses (2.8) to write[7]

$$K - B(\tau) = v(\tau, B(\tau)) + \int_0^\tau rK\mathrm{e}^{-r(\tau-u)}\Phi(-d_-(\tau-u, B(\tau)/B(u)))\,\mathrm{d}u - \int_0^\tau qB(\tau)\mathrm{e}^{-q(\tau-u)}\Phi(-d_+(\tau-u, B(\tau)/B(u)))\,\mathrm{d}u, \tag{2.12}$$

an integral equation for $B(\tau)$ that resembles, but is more complicated than, a nonlinear Volterra equation.[8] Using the fact that $\Phi(-x) = 1 - \Phi(x)$ and

$$x\int_0^\tau \mathrm{e}^{-x(\tau-u)}\,\mathrm{d}u = 1 - \mathrm{e}^{-x\tau}, \tag{2.13}$$

we can rewrite (2.12) in a form that is better suited for our purposes

$$B(\tau)\mathrm{e}^{-q\tau}\left\{\Phi(d_+(\tau, B(\tau)/K)) + q\int_0^\tau \mathrm{e}^{qu}\Phi(d_+(\tau-u, B(\tau)/B(u)))\,\mathrm{d}u\right\} = K\mathrm{e}^{-r\tau}\left\{\Phi(d_-(\tau, B(\tau)/K)) + r\int_0^\tau \mathrm{e}^{ru}\Phi(d_-(\tau-u, B(\tau)/B(u)))\,\mathrm{d}u\right\}. \tag{2.14}$$

Differentiating (2.4) with respect to S and inserting the resulting equality into (2.9) yields, after multiplication with $-B(\tau)$ and usage of (2.13), the alternative equation

$$B(\tau)\mathrm{e}^{-q\tau}\Phi(d_+(\tau, B(\tau)/K)) + B(\tau)\mathrm{e}^{-q\tau}q\int_0^\tau \mathrm{e}^{qu}\left(\Phi\left(d_+\left(\tau-u, \frac{B(\tau)}{B(u)}\right)\right) + \frac{\phi(d_+(\tau-u, B(\tau)/B(u)))}{\sigma\sqrt{\tau-u}}\right)\mathrm{d}u = rK\int_0^\tau \mathrm{e}^{ru}\frac{\phi(d_-(\tau-u, B(\tau)/B(u)))}{\sigma\sqrt{\tau-u}}\,\mathrm{d}u, \tag{2.15}$$

where $\phi(x) = (2\pi)^{-1/2}\mathrm{e}^{-x^2/2}$ is the Gaussian density. Unlike (2.12), this equation lacks symmetry between integral and nonintegral terms, but we may restore this by using

$$\frac{K\mathrm{e}^{-r\tau}}{\sigma\sqrt{\tau}}\phi(d_-(\tau, B(\tau)/K)) = \frac{B(\tau)\mathrm{e}^{-q\tau}}{\sigma\sqrt{\tau}}\phi(d_+(\tau, B(\tau)/K)),$$

[7] Exceptions include Kim *et al* (2013) and Hou *et al* (2000), who appear to use (2.9) and (2.11), respectively, although the technique used in both papers to develop the integral equation is less straightforward than the one used here. The condition (2.10) is used by Chadam and Chen (2003) both in technical work and to create an approximate ODE-based iteration for the boundary.

[8] In a regular nonlinear Volterra equation, the integration kernel on the right-hand side of the equation does not depend on $B(\tau)$.

which leads to

$$
\begin{aligned}
B(\tau)\mathrm{e}^{-q\tau}\Bigg\{ & \Phi(d_+(\tau, B(\tau)/K)) + \frac{\phi(d_+(\tau, B(\tau)/K))}{\sigma\sqrt{\tau}} \\
& + q\int_0^\tau \mathrm{e}^{qu}\Bigg(\Phi(d_+(\tau-u, B(\tau)/B(u))) \\
& \qquad + \frac{\phi(d_+(\tau-u, B(\tau)/B(u)))}{\sigma\sqrt{\tau-u}}\Bigg)\mathrm{d}u\Bigg\} \\
= K\mathrm{e}^{-r\tau}\Bigg\{ & \frac{\phi(d_-(\tau, B(\tau)/K))}{\sigma\sqrt{\tau}} + r\int_0^\tau \mathrm{e}^{ru}\frac{\phi(d_-(\tau-u, B(\tau)/B(u)))}{\sigma\sqrt{\tau-u}}\mathrm{d}u\Bigg\}.
\end{aligned}
\tag{2.16}
$$

3 NUMERICAL SCHEMES FOR THE EXERCISE BOUNDARY

3.1 First approach to numerical solution

The various integral equations in Section 2.3 above have no simple analytical solution, and they generally need to be solved by numerical methods. The most straightforward technique (see AitSahlia and Lai (2001), Kallast and Kivinukk (2003), Kim (1990), Yue-Kuen (1998) and Subrahmanyam and Yu (1993), to name a few) attacks (2.14) with a slight adaptation of direct quadrature methods for Volterra equations (as presented, for instance, in Press (1992, Chapter 18.2)). This approach uses fixed numerical quadrature weights (eg, trapezoid) on a discrete grid $\{\tau_i\}_{i=0}^n$, and it constructs $B(\tau_i)$ sequentially starting from $B(\tau_0) = K\min(1, r/q)$ at $\tau_0 = 0+$. Given $B(\tau_0), B(\tau_1), \ldots, B(\tau_{i-1})$, satisfying (2.14) at $\tau = \tau_i$ will involve a nonlinear root search for $B(\tau_i)$, easily handled by the one-dimensional Newton's method, say. Once the exercise boundary function has been constructed on the grid $\{\tau_i\}_{i=0}^n$, the entire exercise boundary function might be approximated through interpolation (eg, a spline, as in Cortazar *et al* (2013)).

Due to the the presence of τ and $B(\tau)$ in the integration kernels, when computing the integrals needed to establish $B(\tau_i)$ at some point τ_i, the numerical quadrature scheme must be applied to the entire interval $[0, \tau_i]$, rather than just $[\tau_{i-1}, \tau_i]$. As a consequence, the effort of direct quadrature is typically $\mathcal{O}(mn^2)$, where n is the number of discretization points and m is the average number of root-search iterations required to establish $B(\tau_i)$. The convergence of such methods is algebraic and rather modest compared with modern methods for integral equations.[9]

[9] A numerical method has algebraic convergence if its error decreases as $1/n^p$, with n being the number of discretization points and p some constant (say, 2 for a second-order method). A method with spectral convergence, however, has an error that decreases as $1/n^n$, ie, exponentially.

3.2 Second approach to numerical solution

Equations of the type (2.14)–(2.16) may all be rearranged to the form

$$B(\tau) = K\mathrm{e}^{-(r-q)\tau}\frac{N(\tau, B)}{D(\tau, B)}, \tag{3.1}$$

where N, D are functionals depending on $B(u)$, $u \leqslant \tau$. For instance, for the case (2.16) we can introduce operators[10]

$$\mathcal{K}_1(\tau) = \int_0^\tau \mathrm{e}^{qu}\Phi(d_+(\tau - u, B(\tau)/B(u)))\,\mathrm{d}u, \tag{3.2}$$

$$\mathcal{K}_2(\tau) = \int_0^\tau \frac{\mathrm{e}^{qu}}{\sigma\sqrt{\tau - u}}\phi(d_+(\tau - u, B(\tau)/B(u)))\,\mathrm{d}u, \tag{3.3}$$

$$\mathcal{K}_3(\tau) = \int_0^\tau \frac{\mathrm{e}^{ru}}{\sigma\sqrt{\tau - u}}\phi(d_-(\tau - u, B(\tau)/B(u)))\,\mathrm{d}u, \tag{3.4}$$

such that (say)

$$N(\tau, B) = \frac{\phi(d_-(\tau, B(\tau)/K))}{\sigma\sqrt{\tau}} + r\mathcal{K}_3(\tau), \tag{3.5}$$

$$D(\tau, B) = \frac{\phi(d_+(\tau, B(\tau)/K))}{\sigma\sqrt{\tau}} + \Phi(d_+(\tau, B(\tau)/K)) + q(\mathcal{K}_1(\tau) + \mathcal{K}_2(\tau)). \tag{3.6}$$

For ease of reference, we denote (3.1), with N, D set as in (3.5) and (3.6), as fixed point system A, abbreviated as FP-A.

If we instead start from (2.12), say, we have

$$N(\tau, B) = \Phi(d_-(\tau, B(\tau)/K)) + r\int_0^\tau \mathrm{e}^{ru}\Phi(d_-(\tau - u, B(\tau)/B(u)))\,\mathrm{d}u, \tag{3.7}$$

$$D(\tau, B) = \Phi(d_+(\tau, B(\tau)/K)) + q\int_0^\tau \mathrm{e}^{qu}\Phi(d_+(\tau - u, B(\tau)/B(u)))\,\mathrm{d}u. \tag{3.8}$$

Equation (3.1), with N, D set as in (3.7) and (3.8), is denoted fixed point system B, or FP-B for short.

[10] Operators $\mathcal{K}_1$ and $\mathcal{K}_2$ might be combined. Splitting them serves to highlight the fact that $\mathcal{K}_2$ has a singular kernel, due to the factor $(\tau - u)^{-1/2}$.

Fixed point system A was used in the recent paper by Kim *et al* (2013) for the case $q = 0$ to devise a fixed point iteration on an equidistant grid $\{\tau_i\}_{i=1}^n$, where, in the jth iteration,[11]

$$B^{(j)}(\tau_i) = K\mathrm{e}^{-(r-q)\tau_i}\frac{N(\tau_i, B^{(j-1)})}{D(\tau_i, B^{(j-1)})}, \quad i = 1, \ldots, n. \tag{3.9}$$

Kim *et al* (2013) initialize (3.9) at a flat initial guess of $B^{(0)}(\tau) = K\min(1, r/q)$ for all τ, with about six to ten fixed point iterations needed for convergence. Using polynomial interpolation and adaptive Gauss–Kronrod quadrature to evaluate the integrals $\mathcal{K}_1$, $\mathcal{K}_1$ and $\mathcal{K}_3$ in N and D, they record reasonable numerical efficiency, roughly fifty to 100 and five to ten times faster than the binomial tree and finite-difference methods, respectively. If l is the (average) number of Gauss–Kronrod quadrature points used, the computational complexity of the algorithm is $\mathcal{O}(lmn)$, excluding the polynomial interpolation.

We note that the n equations in (3.9) are independent of each other, which allows for straightforward parallelization of the algorithm across multiple processing units.[12] This observation is used in Cortazar *et al* (2013), where a variant of the method in Kim *et al* (2013) has been implemented, with predictably good performance, on a multi-core computer. Besides emphasizing the parallelization aspect of fixed point iterations, Cortazar *et al* (2013) modify the algorithm in Kim *et al* (2013) in a number of ways. They ultimately recommend that (3.9) be initialized at an approximation in Barone-Adesi and Whaley (1987) and, rather surprisingly, that all integrals be evaluated by trapezoid integration on an equidistant maturity grid. Perhaps most significantly, Cortazar *et al* (2013) conclude that the integrals in (3.5) and (3.6) are less well behaved, and result in slower performance, than those in (3.7) and (3.8).[13]

[11] Kim *et al* (2013) list, but never test, expressions for the case $q > 0$. These expressions contain a number of typos, which we have corrected here.

[12] Low-level parallelization of option pricing algorithms – especially relatively fast ones – is often of limited practical value, inasmuch as most banks do not price a single option at a time. Rather, they price entire trading books, often involving 1000s of options, at many input parameter settings. Should one have multiple cores available for computation, a more straightforward parallelization strategy would be to have each core assigned to a different option and/or a different input configuration, rather than having multiple cores dealing with a single option.

[13] As demonstrated in Section 6, we have not been able to detect this. We speculate that Cortazar *et al* (2013) may have run into difficulties with the $(\tau - u)^{-1/2}$ singularities in (3.5) and (3.6). See Section 5 for how to handle this.

3.3 Analysis of fixed point iteration schemes

Iteration schemes based on (3.1) are obviously not unique: we have not only multiple equivalent expressions for the boundary, but also multiple ways to arrange each of these expressions into the form (3.1). For instance, if we let h denote some function or functional, it is clear that (3.1) remains valid (subject to some regularity on h) if we add $\mathrm{e}^{(r-q)\tau}B(\tau)/K \cdot h$ to N and h to D, ie, we write

$$B(\tau) = K\mathrm{e}^{-(r-q)\tau}\frac{N(\tau, B) + \mathrm{e}^{(r-q)\tau}B(\tau)/Kh(\tau, B)}{D(\tau, B) + h(\tau, B)}. \tag{3.10}$$

Alternatively, we may use a more traditional relaxation-type formulation, where we write

$$B(\tau) = K\mathrm{e}^{-(r-q)\tau}\frac{N(\tau, B)}{D(\tau, B)}(1 - h(\tau, B)) + h(\tau, B)B(\tau) \tag{3.11}$$

for some function(al) h. Despite their formal equivalence, it should be obvious that not all expressions for $B(\tau)$ are equally suited for embedding in a fixed point iteration; in fact, some seemingly reasonable formulations might fail to define a proper contraction mapping and will not converge.

Establishing whether a particular equation for the boundary is a candidate for fixed point iteration is essentially a question of how sensitive the right-hand side of (3.1) is to perturbations in the exercise boundary: the lower, the better. With this in mind, cursory examination of the various expressions in Section 2.3 shows that the two most promising candidates for (3.1) are the two (fixed point systems A and B) already considered in Section 3.2.[14]

Characterizing perturbation sensitivity is most easily done through the Gateaux derivative formalism, listed below for our two candidate equations. The proof relies only on simple algebraic manipulations and is omitted.

LEMMA 3.1 *Set $f \triangleq K\mathrm{e}^{-(r-q)\tau}N/D$ in (3.1), and consider a perturbation of the exercise boundary around its optimal location of the proportional form*

$$\ln B(\tau) \to \ln B(\tau) + \omega g(\tau), \quad \omega \in \mathbb{R},$$

where $g\colon \mathbb{R} \to \mathbb{R}$ is a given scalar function. Then, for both fixed point system A (in (3.5) and (3.6)) and fixed point system B (in (3.7) and (3.8)),

$$\left.\frac{\partial f}{\partial \omega}\right|_{\omega=0} = \frac{K\mathrm{e}^{-(r-q)\tau}}{D(\tau, B)}\epsilon, \tag{3.12}$$

[14] As pointed out in Section 2.3, there are a large number of candidate equations available to us, and we do not claim to have performed an exhaustive search.

where

$$\epsilon = \int_0^\tau \mathrm{e}^{ru}\left(r - q\frac{B(u)}{K}\right)\psi(\tau-u, B(\tau)/B(u)) \times \frac{\phi(d_-(\tau-u, B(\tau)/B(u)))}{\sigma\sqrt{\tau-u}}(g(\tau)-g(u))\,\mathrm{d}u.$$

For system B, we have $\psi = -1$*, and for system A,*

$$\psi(\tau-u, B(\tau)/B(u)) = \frac{d_-(\tau-u, B(\tau)/B(u))}{\sigma\sqrt{\tau-u}}. \tag{3.13}$$

COROLLARY 3.2 *For flat proportional shifts of the exercise boundary away from its optimal location, ie, when* $g(\tau)$ *is a constant, then, for both fixed point systems A and B,*

$$\left.\frac{\partial f}{\partial \omega}\right|_{\omega=0} = 0.$$

While Corollary 3.2 is limited to flat proportional shifts, the insensitivity to such shifts nevertheless suggests that a fixed point iteration on both fixed point systems A and B should generally be effective. The computational efforts involved in evaluating N/D are more or less identical for the two systems, and Corollary 3.2 does not settle which of the two representations is more favorable in numerical work. If we focus on robustness toward nonparallel proportional shifts, (3.12) suggests that a relevant metric is

$$\max_{u\leqslant\tau}\left\|\frac{\psi(\tau-u, B(\tau)/B(u))}{D(\tau, B)}\right\|,$$

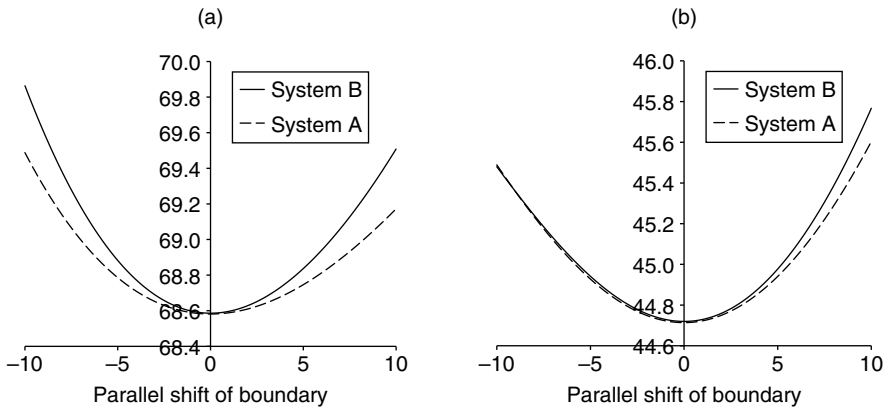
in the sense that the method with the smaller value might be less affected by nonparallel proportional perturbations.

Another metric of interest is the convexity of the function f, eg, as measured by $\partial^2 f/\partial\omega^2$; for fast convergence, we want the absolute value of this derivative to be as small as possible.[15] A few calculations show that, for parallel shifts in log-space (ie, $g(x)$ is a constant),

$$\left.\frac{\partial^2 f}{\partial \omega^2}\right|_{\omega=0} = -\frac{K\mathrm{e}^{-(r-q)\tau}}{B(\tau)^2}\frac{\phi(d_-(\tau, B(\tau)/K))}{\sigma\sqrt{\tau}}\frac{\psi(\tau, B(\tau)/K)}{D(\tau, B)},$$

where ψ is given in Lemma 3.1. Comparing this with (3.13), it follows that the quantity ψ/D determines both the robustness and convergence speed of the fixed point system. Empirically, one finds that fixed point system A normally produces smaller values of $|\psi/D|$ than system B, except for the case in which $r \gg q$ and σ is small. Figures 1 and 2 show a few illustrative graphs; note that the convexity of the graphs is generally lower for system A, except for the left panel in Figure 2, where r is large relative to q.

FIGURE 1 The right-hand side of a fixed point system versus perturbations of boundary: case $r \leqslant q$.



The functional $f = Ke^{-(r-q)\tau}N/D$ in (3.1), graphed against parallel proportional perturbations to the exercise boundary from its optimal level. (a) $r = q = 5\%$, $\sigma = 25\%$, $K = 130$, $\tau =$ five years. (b) $r = 2.5\%$, $q = 5\%$, $\sigma = 25\%$, $K = 130$, $\tau =$ five years.

In summary, our investigations so far suggest that system A, despite the conclusions of Cortazar *et al* (2013), seems more promising overall than system B, except possibly for the case $r \gg q$, where system A may, if iterated on directly, be less stable than system B. We examine these predictions empirically in Section 6.

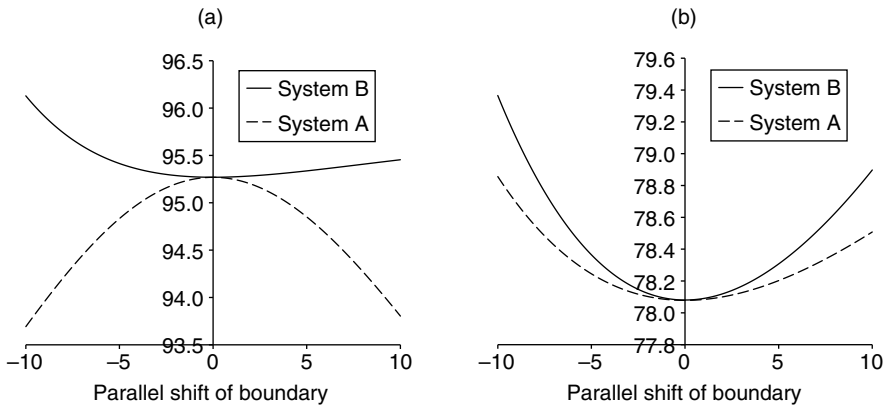
3.4 Roadmap for the design of an efficient numerical scheme

Excluding interpolation cost, the computational effort of existing methods designed to attack (3.1) is, as discussed, $\mathcal{O}(lmn)$, where l is the number of nodes used in the evaluation of integrals, m is the number of iterations used in the fixed point iteration and n is the number of points on the τ-grid at which the fixed point iteration is solved. Our goal in this paper is to design an efficient spectral collocation method for the numerical solution of (3.1), where we aim to keep each of the constants l, m and n as small as possible. The strategy for this involves multiple steps, but roughly speaking our roadmap is as follows.

(1) Formulate the fixed point iteration system (3.1) to be rapidly converging and stable, through relaxation (eg, (3.11)) and a Jacobi–Newton iteration strategy.

[15] Recall that a (scalar) fixed point iteration $x = f(x)$ has convergence order q if $f'(x_p) = f''(x_p) = \cdots = f^{(q-1)}(x_p) = 0$, where x_p is the fixed point.

FIGURE 2 The right-hand side of a fixed point system versus perturbations of boundary: case $r > q$.



The functional $f = K\mathrm{e}^{-(r-q)\tau}N/D$ in (3.1), graphed against parallel proportional perturbations to the exercise boundary from its optimal level. (a) $r = 10\%$, $q = 0\%$, $\sigma = 10\%$, $K = 100$, $\tau =$ three years. (b) $r = 5\%$, $q = 2.5\%$, $\sigma = 25\%$, $K = 130$, $\tau =$ five years.

(2) Start the fixed point iteration at a tight, and computationally efficient, first guess for B.

(3) Establish a strategy to set up a sparse fixed point grid in τ-space.

(4) Choose a smooth function space and a suitable variable transformation to interpolate B on the the τ-grid.

(5) Choose an efficient variable transformation and quadrature scheme to numerically compute the integrals in (3.1).

Our proposed scheme is discussed in detail in Section 5, but it hinges on additional results for the asymptotics of the exercise boundary for small τ. We cover the required results in Section 4 below.

4 BOUNDARY PROPERTIES AND APPROXIMATIONS

4.1 Short-expiration asymptotic behavior of B

To better understand the shape of the the function $B(\tau)$, it is useful to first consider known asymptotic results for $B(\tau)$ for large and small values of τ. Starting with the small-τ limit, relevant results have been developed by numerous authors, including Barles *et al* (1995), Chen and Chadam (2007), Evans *et al* (2002) and, most recently,

Zhang and Li (2010). First, we notice that while $B(0) = K$ always, a small carry argument (similar to the one that led to (2.6)) demonstrates that $\lim_{\tau\downarrow 0} B(\tau) = X$, where

$$X = \begin{cases} K, & r \geqslant q, \\ K(r/q), & r < q. \end{cases} \tag{4.1}$$

When $r < q$, the boundary is therefore discontinuous at $\tau = 0$. The limit-behavior of $B(\tau)$ as τ approaches 0 can be further characterized through asymptotic expansions (see, for example, Zhang and Li 2010). For our purposes, leading-order terms suffice:

$$\frac{B(\tau)}{X} \sim \frac{B_s(\tau)}{X} = \begin{cases} \exp(-\sqrt{-k_1\tau\ln(k_2\tau)}), & r = q, \\ \exp(-\sqrt{-\frac{1}{2}k_1\tau\ln(k_3\tau)}), & r > q, \\ \exp(-k_4\sqrt{\tau}), & r < q, \end{cases} \tag{4.2}$$

where

$$k_1 = 2\sigma^2, \qquad k_2 = 4\sqrt{\pi}r, \qquad k_3 = 8\pi\left(\frac{r-q}{\sigma}\right)^2, \qquad k_4 \approx \sigma\sqrt{2}\times 0.451723.$$

We note that the square root short-expiration boundary $B_s(\tau)$ for $r < q$ is quite different from those of $r \geqslant q$, which involve logarithms. We examine this in more detail later.

4.2 Long-expiration asymptotic behavior of B

For $\tau \to \infty$, the exercise boundary straightens out (from above) to a flat strike-dependent level B_{inf}, which may be solved for easily by standard barrier pricing methods, as in Merton (1973). The result is

$$B_{\text{inf}} = K\frac{\theta_-}{\theta_- - 1}, \tag{4.3}$$

where

$$\theta_\pm = \alpha \pm \sqrt{\beta}, \quad \alpha = \frac{1}{2} - \frac{r-q}{\sigma^2}, \ \beta = \alpha^2 + 2\sigma^{-2}r.$$

The decay toward B_{inf} for large τ is not trivial, and only fairly recently has there been progress in characterizing the long-term asymptotics. Cook (2009) and Ahn *et al* (2009) proposed similar asymptotic results, which were recently extended and sharpened by Chen *et al* (2011). The Chen *et al* (2011) result is

$$\ln B(\tau) \sim \ln B_{\text{inf}} + \gamma(\sigma^2\tau/2)^{-3/2}\mathrm{e}^{-\beta\tau}, \quad \tau \to \infty, \tag{4.4}$$

where γ is an unknown constant. The presence of an unknown constant γ is shared by the expressions in Cook (2009) and Ahn *et al* (2009) and limits the practical usefulness of the results in many respects.

4.3 Boundary shape

The following result was shown in Chen *et al* (2013) (see also Bayraktar and Xing 2009).

THEOREM 4.1 *The American put exercise boundary $B(\tau)$ is infinitely differentiable (C^∞) on $\tau \in (0, \infty)$. When $r \geq q$ or $q \gg r$, the boundary $B(\tau)$ is convex for all $\tau > 0$. However, when $q > r$, and $q - r \ll 1$, then $B(\tau)$ is not uniformly convex in τ. In particular, if $\varepsilon = \ln(q/r)$ is positive and sufficiently small, then there exists a $\hat{\tau}$ for which $\mathrm{d}^2 B(\hat{\tau})/\mathrm{d}\tau^2 < 0$, where*

$$0 < \hat{\tau} \leq \frac{\varepsilon}{3\sigma^2 |\ln(\varepsilon)|}.$$

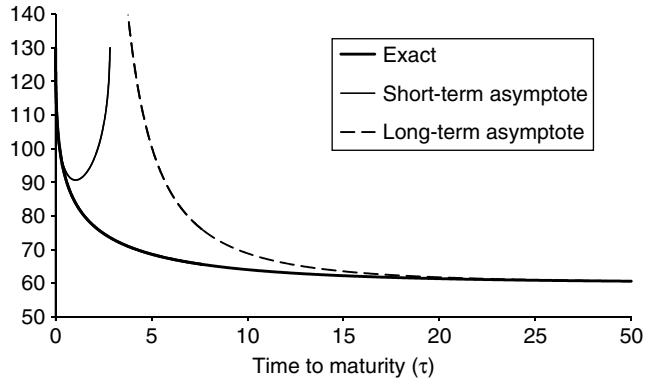
We may take two things away from this theorem. First, the exercise boundary is smooth, except at the origin $\tau = 0$. Second, the boundary is generally convex except for the case in which q is slightly smaller than r. For this case, there will be a small region of nonconvexity close to $\tau = 0$. As we show in a later example (see Figure 4), the violation of convexity can negatively impact the precision of short-time expansions.

4.4 Numerical examples

To get a feel for the various asymptotic expansions above, consider first the case in which $r = q = 5\%$, $K = 130$ and $\sigma = 25\%$. Figure 3 shows the behavior of the short- and long-term approximation for B in (4.2) and (4.4), respectively, along with a high-accuracy estimate of the true exercise boundary. In estimating the constant γ in (4.4), we followed Cook (2009) and match (4.4) to the true asymptote at $\sigma^2\tau/2 = 1$. It is quite obvious that both asymptotic expressions have limited ranges of applicability: the long-term expansion has very poor precision for maturities less than several decades, and the short-term expansion starts to show marked inaccuracies after less than a year (and eventually outright ceases to exist after $\tau = 2.82$ years). Of particular interest in Figure 3 is the behavior around the origin, where the exercise boundary undergoes rapid change and, in the limit of $\tau \downarrow 0$, exhibits unbounded derivatives of all orders.

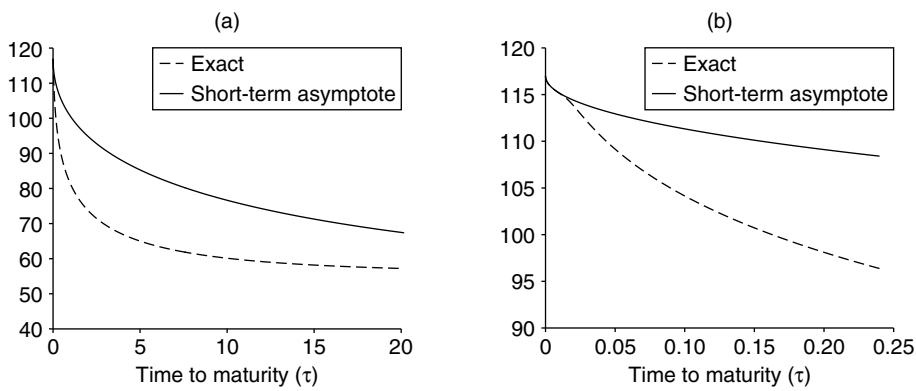
The case $r > q$ generally behaves similarly way to the case $r = q$, with the limit $r \downarrow q$ moving smoothly to the case $r = q$. For the case $r < q$, however, the short-term asymptotic behavior changes abruptly as r goes below q, with the $\sqrt{\tau \ln \tau}$ behavior being replaced by a simpler $\sqrt{\tau}$ asymptote. While the $\sqrt{\tau}$ asymptote is "classical" (see Wilmott *et al* 1995) and often useful for $r \ll q$, it is not robust for r close to q, due to the breakdown of convexity in a region close to the origin (see Theorem 4.1). Figure 4 illustrates this phenomenon, especially part (b), where the behavior around the origin has been emphasized. It is notable that virtually all papers on short-term expansions fail to recognize the deficiency apparent in this figure.

FIGURE 3 Exercise boundary asymptotes.



Various asymptotes of $B(\tau)$ for the case $r = q = 5\%$, $\sigma = 25\%$, $K = 130$. The "short-term asymptote" and "long-term asymptote" graphs are computed from (4.2) and (4.4), respectively. The "exact" boundary is computed numerically by the method outlined in Section 5.

FIGURE 4 Exercise boundary asymptote.



Short-term asymptote of $B(\tau)$ for the case $r = 4.5\%$, $q = 5\%$, $\sigma = 25\%$, $K = 130$. (a) and (b) are identical, except for the range on the x-axis. The "short-term asymptote" graph is computed from (4.2) and the "exact" boundary is computed numerically by the method outlined in Section 5.

4.5 Approximations to the exercise boundary

Dating back to the 1980s, there is a remarkable volume of literature on methods to approximate the early exercise boundary. For a sample of results, see, among others, Barone-Adesi and Elliott (1991), Barone-Adesi and Whaley (1987), Bjerksund

and Stensland (1993a), Broadie and Detemple (1996), Bunch and Johnson (2000), Carr (1998), Chadam and Chen (2003), Cook (2009), Frontczak (2013), Huang and Subrahmanyam (1996), Johnson (1983), Ju (1998), Ju and Zhong (1999), Lee and Paxson (2003), Li (2008, 2009), MacMillan (1986), Zhu (2006) and Zhu and He (2007). We should note that not all of these support general values for μ in (2.1), as many older methods rely on the (classical) zero-dividend assumption, $\mu = r$.

Based on the recent comparisons in Li (2009) and Frontczak (2013), the most precise of the currently published methods appears to be the QD^+ algorithm of Li (2009), a method built on ideas in Ju and Zhong (1999). Like the majority of published methods, QD^+ involves implicit definitions of the boundary that, at each point τ, must be resolved by iterations and a root-search algorithm.

4.6 Approximation based on asymptotics

For later purposes, we briefly want to develop an approximation where the boundary is entirely analytical, ie, it does not involve a root search for each τ at which the boundary is needed. We do this by specifying the boundary as an analytical function, which is meant to reasonably interpolate the long- and short-time asymptotic behavior listed in Section 2 above. For concreteness, we first focus on the case $r \geqslant q$.

Judging from Figure 3, it appears reasonable to first focus on the challenging small-τ regime. To get around the fact that (4.2) ceases to exist for large τ, let us consider replacing the logarithmic term with a different function, eg,

$$x^*(\tau) \sim -\sqrt{\tau\alpha(\tau)}$$

for some function $\alpha\colon \mathbb{R}^+ \to \mathbb{R}^+$. To mimic the behavior of $x^*(\tau)$ around the origin, we require that $\alpha(\tau)$ explode in the limit $\tau \downarrow 0$, at a rate strictly slower than τ^{-1}. Additionally, we wish for the derivative of $\tau\alpha(\tau)$ to become unbounded for $\tau \downarrow 0$.

Combining the requirements above, with a nod toward simplicity and ease of computing, one possible ansatz is to set

$$\alpha(\tau) = a\tau^b, \quad a > 0,\ b \in (-1, 0), \tag{4.5}$$

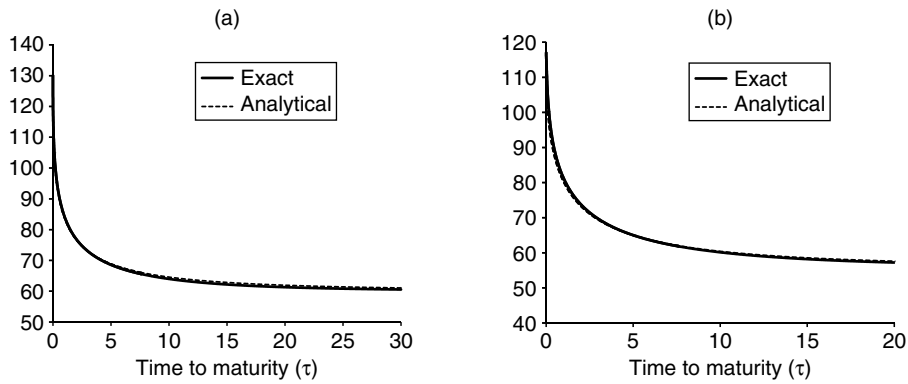
for constants a, b to be determined. To also ensure reasonable long-term asymptotic behavior, more precisely, we here assume that

$$B(\tau) \approx \tilde{B}(\tau) \triangleq B_{\text{inf}} + (K - B_{\text{inf}})\text{e}^{-\sqrt{a\tau^{b+1}}}, \tag{4.6}$$

where B_{inf} is given in (4.3).

There are numerous ways of determining the "best" values of the constants a and b, but what matters here is that a and b in practice can always be set such that (4.6) is remarkably close to the true exercise boundary for all values of τ. In fact, this is

FIGURE 5 Exercise boundary approximation.



Approximation of $B(\tau)$ using (4.6). (a) $r = q = 5\%$, $\sigma = 25\%$, $K = 130$. (b) $r = 4.5\%$, $q = 5\%$, $\sigma = 25\%$, $K = 130$. The "analytical" graphs use (4.6) with $a = 1.3$ and $b = -0.15$ (part (a)), and $a = 0.85$ and $b = -0.1$ (part (b)). The "exact" boundary is computed numerically by the method outlined in Section 5.

true for all configurations of r and q, not just $r \geq q$; see Figure 5 for some examples. In virtually all cases, the best-fitting value of b is slightly negative, even for the case $r < q$.[16]

5 A NEW NUMERICAL SCHEME

Having now laid a sufficient foundation, we return to the roadmap in Section 3.4 and begin the concrete development of an efficient numerical scheme for the boundary equation (3.1). For reasons already discussed, we focus on fixed point system A ((3.5) and (3.6)), but our algorithm is easily extended to fixed point system B ((3.7) and (3.8)).

Fundamentally, we wish to use a collocation scheme, ie, we solve (3.1) on a discrete set of collocation nodes $\{\tau_i\}_{i=1}^n$ only and construct the full function $B(\tau)$ by interpolation. While there are many possibilities for this interpolation (eg, splines), we here rely on polynomial approximation; the resulting scheme can therefore also be cast as a projection method on the space of n-dimensional polynomials in τ.[17]

[16] While (4.2) suggests that $b = 0$ for the case $r < q$, we have already seen in Figure 4 that this can be misleading.

[17] There are numerous "myths" about the supposed instability of polynomial approximation, especially when the polynomial is of high order. As discussed in Trefethen (2012), such instabilities are associated with poor choices of interpolation grids (eg, equidistant grids) and computational methods, rather than with the fundamental precision of polynomial approximation.

Background material on collocation-type projection methods for integral equations – a very active research area in numerical mathematics – can be found in Atkinson (1992), Brunner (2004), Elnagar and Kazemi (1996) and Tang *et al* (2008).

In the practical computation of the boundary, specific algorithms are required for the computation of integral operators and for the numerical solution of (3.1). In addition, we need to make efficient choices for the parameter space in which we operate, especially with regard to integration variable transformations and boundary interpolations. We outline and justify our choices in the following sections.

5.1 Fixed point iteration scheme

Let us first investigate how we can use the relaxation formulation (3.11) to construct a (dampened) Jacobi–Newton iteration for the fixed point system (3.1). Dropping some arguments, we write

$$B(\tau) = (1 - h(\tau))\, f(\tau, B) + h(\tau) B(\tau), \tag{5.1}$$

where

$$f(\tau, B) = K^*(\tau) \frac{N(\tau, B)}{D(\tau, B)}, \tag{5.2}$$

with $K^*(\tau) \triangleq K \mathrm{e}^{-(r-q)\tau}$. For a diagonal Jacobi iteration scheme, we are interested in measuring the sensitivity of the right-hand side of (3.11) with respect to moves in $B(\tau)$.[18] Treating f as a functional on an arbitrary function Q, we get

$$f'(\tau, Q) \triangleq \frac{\partial f}{\partial Q(\tau)} = K^*(\tau) \left(\frac{N'(\tau, Q)}{D(\tau, Q)} - \frac{D'(\tau, Q) N(\tau, Q)}{D(\tau, Q)^2} \right), \tag{5.3}$$

where N' and D' indicate partial derivatives with respect to $B(\tau)$. For instance, for fixed point system A (with N and D given in (3.5) and (3.6)), we have

$$\begin{aligned} N'(\tau, Q) = {} & -d_-(\tau, Q(\tau)/K) \frac{\phi(d_-(\tau, Q(\tau)/Q(u)))}{Q(\tau)\sigma^2\tau} \\ & - r \int_0^\tau \frac{\mathrm{e}^{ru} d_-(\tau, Q(\tau)/Q(u))}{Q(\tau)\sigma^2(\tau - u)} \phi(d_-(\tau - u, Q(\tau)/Q(u)))\, \mathrm{d}u \end{aligned} \tag{5.4}$$

and

$$\begin{aligned} D'(\tau, Q) = {} & -\frac{K^*(\tau)}{Q(\tau)} d_-(\tau, Q(\tau)/K) \frac{\phi(d_-(\tau, Q(\tau)/K))}{Q(\tau)\sigma^2\tau} \\ & - q \frac{K^*(\tau)}{Q(\tau)} \int_0^\tau \frac{Q(u)}{K} \frac{\mathrm{e}^{ru} d_-(\tau - u, Q(\tau)/Q(u))}{\sigma^2(\tau - u)} \phi(d_-(\tau - u, Q(\tau)/Q(u)))\, \mathrm{d}u. \end{aligned} \tag{5.5}$$

[18] Note that we do not consider the sensitivity to $B(u)$, $u \neq \tau$, which defines our scheme to be of the Jacobi type.

In an iterative scheme for B, at iteration j we now wish to locally cancel out first-order sensitivity by setting h such that the derivative $(1-h)f'$ equals $-h$. This suggests an iteration in which

$$B^{(j)}(\tau) = (1 - h^{(j-1)}(\tau)) f(\tau, B^{(j-1)}) + h^{(j-1)}(\tau) B^{(j-1)}(\tau),$$

with

$$h^{(j-1)}(\tau) = \frac{f'(\tau, B^{(j-1)})}{f'(\tau, B^{(j-1)}) - 1}.$$

Rearranging, we arrive at the Jacobi–Newton scheme we will rely on:

$$B^{(j)}(\tau) = B^{(j-1)}(\tau) + \eta \frac{B^{(j-1)}(\tau) - f(\tau, B^{(j-1)})}{f'(\tau, B^{(j-1)}) - 1}. \tag{5.6}$$

We note that a naive (Richardson) fixed point scheme is recovered if we set $\eta = 1$ and $f'(\tau, B^{(j-1)}) = 0$.

Equation (5.6) relies on the computation of two new integrals in order to evaluate $f'(\tau, B^{(j-1)})$. While this can often be done rapidly (several terms in the integrands are shared with other integrals), to the extent that we are mainly concerned with near-proportional shifts (see Lemma 3.1), it may be sufficient to ignore the integral terms in (5.4) and (5.5). Doing so generally results in a performant scheme, and it is the one that we shall use in most of our numerical tests.

5.2 Time variable transformation

Our fundamental iteration algorithm (5.6) will be applied on a discrete set of n collocation nodes $\{\tau_i\}_{i=1}^n$, to be established later. In the evaluation of the necessary integrals, the presence of $(\tau - u)^{-1/2}$ in the integrals $\mathcal{K}_2(\tau)$ and $\mathcal{K}_3(\tau)$ (see (3.3) and (3.4)) of fixed point system A needs to be considered first. While there are numerous methods in the literature for handling weakly singular kernels (see, for example, Brunner 1984, 1985), we here deal with the singularity analytically through the variable transformation

$$z = \sqrt{\tau - u}, \tag{5.7}$$

which, as $\mathrm{d}z = -\frac{1}{2}(\tau - u)^{-1/2}\,\mathrm{d}u$, removes the kernel singularity of $\mathcal{K}_2(\tau)$ and $\mathcal{K}_3(\tau)$. Using the z variable, the integration region is $[0, \sqrt{\tau}]$ (rather than $[0, \tau]$). As we later want to apply high-precision quadrature to approximate the integrals, it is convenient to additionally introduce a transformation to normalize the integrals to the standard quadrature interval $[-1, 1]$ by writing

$$y = -1 + 2\frac{z}{\sqrt{\tau}} = -1 + 2\frac{\sqrt{\tau - u}}{\sqrt{\tau}}. \tag{5.8}$$

Applying this transformation to all three integrals $\mathcal{K}_1(\tau), \ldots, \mathcal{K}_3(\tau)$, we get

$$\mathcal{K}_1(\tau) = \frac{\mathrm{e}^{q\tau}}{2}\tau\int_{-1}^{1}\mathrm{e}^{-(q/4)\tau(1+y)^2}(y+1)\Phi\left(d_+\left(\frac{\tau(1+y)^2}{4}, \frac{B(\tau)}{B(\tau-\tau(1+y)^2/4)}\right)\right)\mathrm{d}y, \tag{5.9}$$

$$\mathcal{K}_2(\tau) = \mathrm{e}^{q\tau}\sqrt{\tau}\int_{-1}^{1}\frac{\mathrm{e}^{-(q/4)\tau(1+y)^2}}{\sigma}\phi\left(d_+\left(\frac{\tau(1+y)^2}{4}, \frac{B(\tau)}{B(\tau-\tau(1+y)^2/4)}\right)\right)\mathrm{d}y, \tag{5.10}$$

$$\mathcal{K}_3(\tau) = \mathrm{e}^{r\tau}\sqrt{\tau}\int_{-1}^{1}\frac{\mathrm{e}^{-(r/4)\tau(1+y)^2}}{\sigma}\phi\left(d_-\left(\frac{\tau(1+y)^2}{4}, \frac{B(\tau)}{B(\tau-\tau(1+y)^2/4)}\right)\right)\mathrm{d}y. \tag{5.11}$$

We note that the transformation (5.7) is not strictly necessary for $\mathcal{K}_1(\tau)$, but it is generally easier to apply it to all integrals. For the integrals in fixed point system B, the transformation is not needed (but does little harm if applied).

5.3 Numerical integration scheme

Assume for a moment that we can evaluate the function $B(\tau-\tau(1+y)^2/4)$ for all y on $(-1, 1)$. To compute (5.9)–(5.11), a numerical integration scheme is required to approximate the integrals. As our integrand is smooth (see Theorem 4.1), a number of quadrature rules are possible here, such as Gaussian and tanh–sinh quadrature rules. Common for such methods is that, for each τ-indexed integrand $c(y;\tau)$ in (5.9)–(5.11), one may write an l-point scheme

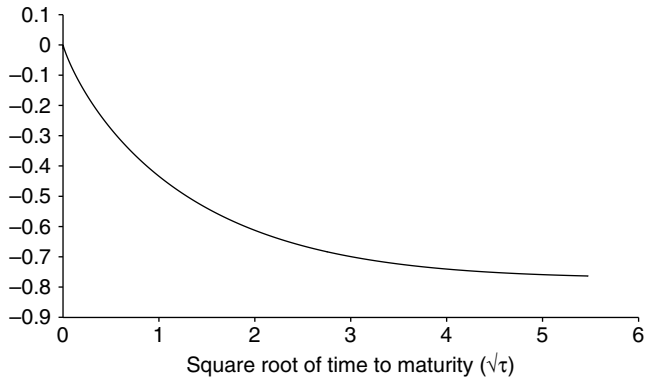
$$\int_{-1}^{1}c(y;\tau)\,\mathrm{d}y \approx \sum_{k=1}^{l}w_k c(y_k;\tau), \tag{5.12}$$

where the weights w_k and the nodes y_k are specific to the chosen quadrature rule.

5.4 Interpolation and collocation time grid

As discussed earlier, we intend to apply nth order polynomial interpolation to uncover B between collocation points. While we would like to keep the value of n as low as possible, it is clear from numerical results in Section 4.5 that the function $B(\tau)$ is not necessarily well characterized by a low-dimensional polynomial in τ, especially close to the origin, where derivatives diverge. As mentioned earlier, empirical examination

FIGURE 6 Function G.



The function G in (5.14) as a function of $\sqrt{\tau}$. Model settings are $r = q = 5\%$, $\sigma = 25\%$, $K = 130$.

of the power b in (4.5) shows that b is virtually always small (and nonpositive), so one would expect the function $\ln(B(\tau))^2$ to be quite close to a straight line for small τ.[19] Writing our interpolation scheme on $\ln(B(\tau))^2$ in τ-space works well, as does using the transformation

$$G(\sqrt{\tau}) = \ln(B(\tau)/X), \qquad X = K\min(1, r/q). \tag{5.13}$$

As confirmed by Figure 6, the function G is generally much better behaved than B itself, and it is a good candidate for polynomial interpolation. As it turns out, the combination of the two transformations, ie,

$$H(\sqrt{\tau}) = G(\sqrt{\tau})^2 = \ln(B(\tau)/X)^2, \tag{5.14}$$

is even better suited for our purposes (and still easy to compute and invert), so we shall use this for our later numerical experiments.

To establish an interpolation grid $\{x_i\}_{i=0}^n$ for $H = H(x)$, an equidistant grid in x (or in x^2, for that matter) should not be used. Such a grid is prone to instabilities (the well-known Runge phenomenon), and it rarely produces a competitive polynomial approximation to the underlying function. So, instead, we use Chebyshev nodes of the second kind,

$$x_i = \frac{\sqrt{\tau_{\max}}}{2}(1 + z_i), \quad z_i = \cos\left(\frac{\mathrm{i}\pi}{n}\right), \quad i = 0, \ldots, n, \tag{5.15}$$

[19] Of course, $\ln(B(\tau))^{2/(1-b)}$ would be even closer to a straight line, but working with noninteger powers in interpolation is typically not worth the additional computational cost.

where $\tau_{\max}$ is the longest maturity for which we shall need to recover the exercise boundary. The z_i are here just the extrema of the Chebyshev polynomial $T_n(z) = \cos(n \cos^{-1} z)$. It is known (see, for example, Berrut and Trefethen 2004) that the Chebyshev node placement eliminates the Runge phenomenon. In addition, it is frequently near-optimal in the sense that the resulting polynomial interpolant is close to the minimax polynomial, ie, the nth-order polynomial that has the smallest maximum deviation to the true function on $[0, \sqrt{\tau_{\max}}]$.

To concretely carry out interpolation of H between the Chebyshev nodes, let us normalize arguments to $[-1, 1]$ by writing $H(x) = q(z)$, $x = \sqrt{\tau_{\max}}(1 + z)/2$. It is known that the Chebyshev interpolant, q_{C}, to q can be expressed as

$$q_{\mathrm{C}}(z) = \sum_{k=0}^{n}{}'' a_k T_k(z), \quad a_k = \frac{2}{n} \sum_{i=0}^{n}{}'' q_i \cos \frac{ik\pi}{n}, \tag{5.16}$$

where $q_i = q(z_i) = H(x_i)$ and the $''$ indicates that the first and last terms in the sum are to be halved. It can readily be verified that this relation ensures that $q_{\mathrm{C}}(z_i) = q_i$, as desired. For an arbitrary $-1 < z < 1$, $q_{\mathrm{C}}(z)$ can now be evaluated efficiently and stably using the Clenshaw algorithm:[20]

$$\begin{aligned} b_n(z) &= \tfrac{1}{2} a_n, \qquad b_{n+1}(z) = 0, \\ b_k(z) &= a_k + 2z b_{k+1}(z) - b_{k+2}(z), \quad k = n-1, \ldots, 0, \\ q_{\mathrm{C}}(z) &= \tfrac{1}{2}[b_0(z) - b_2(z)]. \end{aligned} \tag{5.17}$$

The grid $\{x_i\}_{i=1}^n$ establishes our interpolation nodes and defines our collocation grid $\{\tau_i\}_{i=1}^n$, ie, the discrete points in τ-space at which (5.6) is run. Specifically, we have

$$\tau_i = x_i^2, \quad i = 1, \ldots, n, \tag{5.18}$$

and $B(\tau_0) = B(0+) = K \min(1, r/q)$.

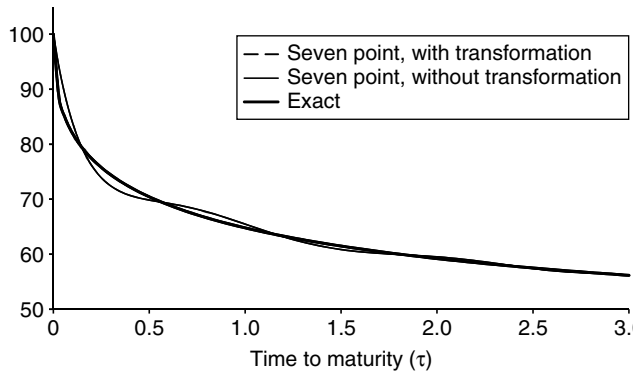
Finally, for illustration, Figure 7 shows the exercise boundary interpolants from a naive low-dimensional collocation scheme applied directly on $\ln B$, compared with the one above applied to H. As one might have guessed from Figure 6, the latter avoids ringing and is far more accurate overall.

5.5 Complete algorithm, computational effort and convergence

Assume now that l, m and n are given, as is the desired maximum horizon $\tau_{\max}$. Let us summarize the complete algorithm for computing the optimal exercise boundary.

[20] This relation originates from the Chebyshev polynomial recurrence $T_n(z) = 2zT_{n-1}(z) - T_{n-2}(z)$.

FIGURE 7 Three-year exercise boundary.



Seven-point interpolated boundaries, with and without H-transformation. Model settings are $r = q = 5\%$, $\sigma = 25\%$, $K = 100$. The seven-point boundary with H-transformation cannot be distinguished from the exact boundary at the resolution of the figure.

(1) Compute the Chebyshev nodes from (5.15). This establishes the collocation grid $\{\tau_i\}_{i=0}^n$, by (5.18).

(2) Compute or look up the quadrature nodes y_k and weights w_k, for $k = 1, \ldots, l$.

(3) Use one of the approximate methods (eg, QD$^+$) referenced in Section 4.5 to establish an initial guess for B on the n points $\{\tau_i\}_{i=1}^n$.[21] Let the guess be denoted $B^{(0)}(\tau_i)$, $i = 0, \ldots, n$, with $B^{(0)}(\tau_0) = B(0+) = K \min(1, r/q)$.

(4) For $j = 1$ to $j = m$, execute steps (5)–(9) below.

(5) Set $H(\sqrt{\tau}) = \ln(B^{(j-1)}(\tau)/X)^2$ and initialize the Chebyshev interpolation in Section 5.4 by computing the a_k in (5.16).

(6) For each τ_i, $i = 1, \ldots, n$, use the Clenshaw algorithm (5.17) to establish (through q_C and H)

$$B^{(j-1)}(\tau_i - \tau_i(1 + y_k)^2/4), \quad k = 1, \ldots, l.$$

(7) Use numerical quadrature (similar to (5.12)) to compute the integrals necessary to establish $N(\tau_i, B^{(j-1)})$ and $D(\tau_i, B^{(j-1)})$. Compute $f(\tau_i, B^{(j-1)})$, $i = 1, \ldots, n$.

[21] In the actual implementation, it is more efficient to work on H (5.14) than on B directly.

(8) Let $f'(\tau_i, B^{(j-1)})$ be defined as in (5.3). For $i = 1, \ldots, n$, the following hold.

   (a) For a full Jacobi–Newton step, compute $N'(\tau_i, B^{(j-1)})$ and $D'(\tau_i, B^{(j-1)})$ per (5.4) and (5.5); the integral terms will be done by numerical integration, as in step (7). Compute f' by (5.3).

   (b) For a partial Jacobi–Newton step, omit the integrals in (5.4) and (5.5). Compute f' by (5.3).

   (c) For an ordinary (Richardson) fixed point iteration, set $f' = 0$.

(9) Compute $B^{(j)}(\tau_i)$, $i = 1, \ldots, n$, from (5.6).

(10) We are done. The optimal exercise boundary at $\{\tau_i\}_{i=0}^n$ is approximated by $B(\tau_i) \approx B^{(m)}(\tau_i)$. For values of τ between nodes in $\{\tau_i\}_{i=0}^n$, the Chebyshev interpolant of $B^{(m)}$ (with a_k computed in step (5)) is used to approximate $B(\tau)$.

Upon completion of the algorithm, we are left with a continuous representation of $B(\tau)$ on the interval $[0, \tau_{\max}]$. This boundary may be cached for later use, or it may be turned into put option prices at various maturities inside $[0, \tau_{\max}]$ by usage of (2.4). The integrals in (2.4) can be computed straightforwardly by any high-performance quadrature rule.

Let us briefly consider the computational effort of our scheme. Focusing first on the interpolation effort, for each boundary iteration, nl interpolations are required from (5.17), each involving an operation count of $\mathcal{O}(n)$. In addition, the a_k must be computed once per boundary iteration and requires $\mathcal{O}(n^2)$ operations.[22] Note that there are only n distinct cosine values in the whole computation, or half that if we use $\cos(\mathrm{i}\pi/n) = -\cos((n-\mathrm{i})\pi)/n$; these values may be computed once and stored for lookup, so the computation of the a_k is fast. For the total algorithm in steps 1 to 10, we conclude that we have an interpolation cost of

$$C_{\text{interp}} = \mathcal{O}(mn^2) + \mathcal{O}(lmn^2) = \mathcal{O}(lmn^2). \tag{5.19}$$

The additional cost of performing the boundary iteration involves a series of preprocessing steps (steps 1–3), the operation count of which is $\mathcal{O}(l) + \mathcal{O}(n)$ and normally negligible. Computation of integrals by quadrature (step (7)) dominates the in-loop cost and involves an operation cost of $\mathcal{O}(\ln)$ per boundary iteration. The total cost of numerical integration is therefore

$$C_{\text{integral}} = \mathcal{O}(lmn). \tag{5.20}$$

[22] It is possible to reduce this to $\mathcal{O}(n \log n)$ by using the fast cosine transform, but there is little reason to do this, as the cost of the Clenshaw recurrence dominates (see (5.19)).

Comparison of (5.19) and (5.20) shows that the interpolation cost nominally dominates the integration cost for sufficiently large n. At realistic levels of n, however, one typically finds that (5.20) is larger than (5.19), due to the simplicity of the interpolation operations. Nevertheless, the form of (5.19) shows that the polynomial order n is perhaps more critical than the parameters l and m, and one should aim to keep it as low as possible (which we here do, in part, by executing the interpolation on the smooth function H in (5.14)).

Finally, a note on the expected convergence of our method. Provided that (a) we use a spectral quadrature scheme (eg, Gaussian quadrature) and (b) the integrands are sufficiently smooth, then theoretically our algorithm should converge to the American option price at an exponential rate as l and n are increased. In practice, spectral collocation methods may or may not attain this theoretical ideal, but we would nevertheless expect the method to be highly competitive against, say, grid-based methods with algebraic convergence (see Footnote 9 on p. 46). This is no different from the basic application of, say, Gaussian quadrature in the numerical integration of smooth functions: the actual convergence order may or may not be at the peak theoretical limit, but it is virtually always much better than simple integration schemes (eg, trapezoid). We empirically examine the convergence properties in Section 6.

5.6 Notes to algorithm

Let us briefly provide a few comments and suggestions to the algorithm outlined in steps 1 to 10 above.

- It is not uncommon in the literature to use the Chebyshev nodes (step (1)) both for interpolation and for numerical integration, enforcing $l = n$. The resulting quadrature rule is known as Clenshaw–Curtis quadrature (see Trefethen 2008). For flexibility and performance, however, our algorithm separates interpolation and integration, allowing for different node count and node placement; the computational overhead associated with this extension is minimal. Also, our tests found Gauss–Legendre quadrature (say) always outperformed Clenshaw–Curtis quadrature, although the differences between the methods were often relatively minor (which is consistent with the findings in Trefethen (2008)).

- In addition to Gauss–Legendre quadrature, we experimented with several integration methods, including Gauss–Kronrod quadrature. While other schemes may offer benefits when embedded in an adaptive quadrature algorithm, for exogenously specified l the simplicity and strong performance of the Gauss–Legendre quadrature makes it our default choice, especially for moderate values of l: say 10 to 15 or less. For high-precision option price estimates with large

l (as in the tests in Section 6.1.1), we recommend using the tanh–sinh quadrature rule. While tanh–sinh quadrature is uncommon in finance applications, the method is robust and has strong convergence properties for integrands with singularities (see, for example, Bailey *et al* 2005).

- Our algorithm is designed to work with exogenously fixed values of l, m and n. In an industrial application, it is likely that one would upgrade the algorithm to work with stopping criteria based on exogenously specified tolerances. Adaptive integration rules would be useful for this. For our purposes, in this paper – which focuses on testing and documentation – it is more relevant to work with fixed l, m and n.

- In step (8), one has to choose between various flavors of Jacobi–Newton iteration and ordinary (Richardson) fixed point iteration. As there is some overhead associated with using Jacobi–Newton iteration, a reasonable strategy is often to use Jacobi–Newton stepping for the first one or two iterations, and then, when the boundary is close to the optimum value, switch to ordinary fixed point iteration. Given the result of Corollary 3.2, we expect the convergence of ordinary fixed point iteration to be rapid, once we are close to the root.

- If one has an accurate estimate for the convergence rate of American option prices as a function of (say) h, it is often possible to use Richardson extrapolation to accelerate convergence.[23] For a spectral method (such as ours), the applicability of these techniques remains an open question. As we shall see, our algorithm converges so fast that additional acceleration techniques are rarely, if ever, needed in practice.

- For the final conversion of the exercise boundary into option prices via (2.4), we recommend using the same integration rule used inside our collocations scheme, applied on the transformation (5.7). We also recommend using a relatively large number of integration nodes, p, especially if l and n are themselves large. A good rule of thumb is to use at least $p = l$ or even $p = 2l$ integration nodes for the final option value integral; the overhead of doing so is typically small relative to the cost of finding the exercise boundary in the first place.

6 NUMERICAL TESTS

In this section, we conduct a variety of tests of the algorithm in Section 5.5. Our emphasis is on the futures case $r = q$, covered in Section 6.1, but we also list results

[23] That said, many market participants are generally concerned about the robustness of extrapolation methods, as they can sometimes produce poor sensitivities. Also, a small error in the estimation of convergence order can lead the extrapolation results astray.

for other cases in Section 6.2. Most of our tests involve computing the optimal exercise boundary B at various configurations of the basic scheme, typically by varying the parameters n (the number of collocation nodes), m (the number of iterations) and l (the number of integration nodes). As the boundary itself is normally of less importance than American put option prices, the reported precision metric is mostly absolute and relative price errors of option prices.

Concretely, the method in Section 5 was coded in C++ and compiled with Visual C++ 2013 on a 2GHz PC. The default setup was as follows.

- QD^+ is used to establish the first guess for B at the n collocation nodes (step (3)).
- For the first iteration, we use a partial Jacobi–Newton step (option (b) in step (8)). For subsequent iterations, we use ordinary fixed point iteration (option (c) in step (8)).
- l, m and n are specified exogenously, and no stopping criteria are used in the algorithm. We often set $l \approx 2n$.
- The number of integration nodes p used for computing the option price from a given boundary (through (2.4)) is also specified exogenously; we often set $p \approx 2l$.
- Unless otherwise specified, for benchmark "exact" option values we use the fixed point method with $(l, m, n) = (131, 16, 64)$, $p = 131$ and tanh–sinh quadrature. We justify this choice in Section 6.1.1.

We emphasize that our tests do not cache the computed exercise boundary for use with multiple options; rather, we recompute the boundary for each option (thereby setting $\tau_{\max}$ equal to the option maturity). For a truly optimized algorithm, implementation of a cache mechanism is a distinct possibility and could potentially lead to significant performance gains. However, systems overly reliant on caches may have difficulty with time-dependent parameters, an extension considered in Section 7.

6.1 Case $r = q$

6.1.1 PDEs and high-precision baseline values

For our later speed tests against the literature, we shall need to establish high-precision benchmark values that we can consider the "exact" American option values. While it is common in the literature to use a high-dimensional (eg, 10 000 steps) binomial tree for this purpose, the resulting benchmarks are far too imprecise for our purposes here. Instead, we shall use our own algorithm (fixed point system A).

We initially considered using alternative methods for establishing benchmark values, but our tests showed that no other algorithm could produce sufficiently precise

TABLE 1 Estimated one-year American premium for $K = S = 100$.

Method	American premium	Relative error	CPU seconds
PDE 50	0.102249534103	4.6E−02	N/A
PDE 100	0.105487101741	1.4E−02	2.9E−03
PDE 500	0.106872668265	7.5E−04	9.6E−03
PDE 1k	0.106927949843	2.3E−04	3.1E−02
PDE 5k	0.106950984474	1.6E−05	7.3E−01
PDE 10k	0.106952141410	5.3E−06	2.9E+00
PDE 50k	0.106952659976	4.0E−07	7.6E+01
PDE 250k	0.106952688738	1.3E−07	2.3E+03
FP-A (65,8,32)	0.106952702747	N/A	3.0E−03

Model settings were $r = q = 5\%$ and $\sigma = 0.25$. The "PDE" method is a Crank–Nicolson finite-difference grid with an equal number of asset steps and time steps, set as indicated in the table. The "FP-A" method is fixed point system A, with $(l, m, n) = (65, 8, 32)$ and $p = 101$ option price quadrature nodes; the integration scheme was tanh–sinh quadrature. Relative errors of the PDE method are measured against the FP-A premium.

benchmark values quickly enough for the thousands of option tests we wanted to run. To show a typical result of our tests, consider using a production-quality Crank–Nicolson finite-difference method to establish the benchmark value for a one-year American put option. Table 1 lists the American put premium for various grid sizes.[24] It also shows that even a comparatively modest precision setting for our method will produce an option value that coincides to seven significant digits, with the price computed by a $250\,000 \times 250\,000$ step (!) finite-difference grid. However, while our method completes its calculation in about 0.003 seconds, the finite-difference grid takes about 40 minutes to run: a relative speed difference in the order of one million.

It can be verified that the finite-difference grid solver used in Table 1 has a convergence order of around 2 for a moderate number of steps, but the convergence ultimately tapers off for large grids. In fact, adding more steps in the finite-difference grid does not lead to meaningful precision improvements after one reaches about $100\,000 \times 100\,000$ nodes; instead, rounding errors cause results to oscillate randomly up and down at about the seventh or eighth digit after the floating point. The FP-A method, on the other hand, displays no such behavior, and the ten or eleven first digits after the floating point rapidly stabilize (see Table 2).

For those that are curious about whether the finite-difference grid perhaps becomes more competitive at lower precision requirements, we shall show more comparisons later on; but, basically, the answer is no. For instance, for the example above, the

[24] That is, the difference between the American and European put prices.

TABLE 2 Estimated one-year American premium for $K = S = 100$.

(l, m, n)	p	American premium	Relative error	CPU seconds
(5,1,4)	15	0.106783919132	1.5E−03	9.9E−06
(7,2,5)	20	0.106934846948	1.7E−04	2.3E−05
(11,2,5)	31	0.106939863585	1.2E−04	3.1E−05
(15,2,6)	41	0.106954833468	2.0E−05	4.5E−05
(15,3,7)	41	0.106952928838	2.1E−06	7.1E−05
(25,4,9)	51	0.106952731254	2.7E−07	1.7E−04
(25,5,12)	61	0.106952704598	1.7E−08	2.9E−04
(25,6,15)	61	0.106952703049	2.8E−09	4.6E−04
(35,8,16)	81	0.106952702764	1.6E−10	8.4E−04
(51,8,24)	101	0.106952702748	1.5E−11	2.1E−03
(65,8,32)	101	0.106952702747	8.5E−13	3.0E−03

Model settings were $r = q = 5\%$ and $\sigma = 0.25$. All numbers were computed using fixed point system A, with (l, m, n) and p as given in the table. Relative errors are measured against the American premium computed with $(l, m, n) = (201, 16, 64)$ and $p = 201$. Results for $(l, m, n) = (5, 1, 4)$ and $(l, m, n) = (7, 2, 5)$ were computed with Gauss–Legendre quadrature; all other results were computed with tanh–sinh quadrature.

FP-A method with $(l, m, n) = (7, 2, 5)$ is five times more precise than the 500×500 finite-difference grid, but it runs more than 400 times faster than the finite-difference grid.

6.1.2 Some literature comparisons

To expand the performance of our scheme against competing methods, we start off with a warm-up example on one of the more challenging option examples in Cortazar *et al* (2013), namely a three-year option with $r = q = 4\%$, $K = 100$ and $\sigma = 0.2$. Results from a variety of algorithms are compared against this benchmark in Table 3.

At the given discretization levels, it is clear from the table that the most accurate method is our fixed point system A (FP-A) method, which is one or two orders of magnitude more precise than the FIK-F(400), Bin-BS and FP-B methods, which (roughly) tie for second place. More importantly, it is clear that the FP-A and FP-B methods are both much more computationally efficient than the FIK-F(400) and Bin-BS methods. For instance, according to Cortazar *et al* (2013), the average number of iterations used to establish the boundary is about five or more, hence the complexity of the FIK-F(400) method (excluding spline interpolation) is, as explained in Section 3.1, $\mathcal{O}(5 \times 400 \times 400) = \mathcal{O}(800, 000)$. In contrast, the FP-A and FP-B methods have complexity (excluding polynomial interpolation) $\mathcal{O}(3 \times 10 \times 7) = \mathcal{O}(210)$, or about three to four orders of magnitude less than

TABLE 3 Estimated three-year put option price values for $K = 100$ and S as listed in the table.

Spot S	True price	Bin 1000	Bin-BS 15 000	FIK-F 60	FIK-F 400	KJK 32	FP-A (10,3,7)	FP-B (10,3,7)
80	23.22834	23.22864	23.22837	23.22921	23.2284	23.22855	23.22834	23.22823
		3.1E−04	*3.1E−05*	*8.7E−04*	*6.1E−05*	*2.1E−04*	*2.4E−06*	*1.1E−04*
100	12.60521	12.60282	12.60529	12.60592	12.60526	12.60548	12.60521	12.60517
		2.4E−03	*7.8E−05*	*7.1E−04*	*4.8E−05*	*2.7E−04*	*2.2E−06*	*3.9E−05*
120	6.482425	6.48423	6.48247	6.48289	6.48246	6.48268	6.482425	6.482414
		1.8E−03	*4.5E−05*	*4.7E−04*	*3.5E−05*	*2.6E−04*	*1.5E−07*	*1.0E−05*

Model settings were $r = q = 4\%$ and $\sigma = 0.2$. First row: name of method; second row: number of steps used in algorithm. "Bin" is a regular binomial tree with 1000 time steps; "Bin-BS" is an accelerated binomial tree with 15 000 time steps; "FIK-F" is the recommended method in Cortazar *et al* (2013), with the number of integration nodes set to 60 or 400; "KJK" is the method in Kim *et al* (2013) with thirty-two integration nodes; "FP-A" and "FP-B" are the methods of this paper applied to fixed point systems A and B, respectively, $(l, m, n) = (10, 3, 7)$ and $p = 25$ (Gauss–Legendre quadrature). Italicized numbers are absolute price errors compared with a high-precision "exact" estimate (from FP-A with $(l, m, n) = (131, 16, 64)$). The "Bin-BS", "FIK-F" and "KJK" numbers were taken from Cortazar *et al* (2013, Table 1).

FIK-F(400). (The Bin-BS method has complexity $\mathcal{O}(15,000^2)$ and is not competitive from an efficiency perspective.)

Note that clean timing comparisons of FP-A/B against FIK-F could not be accomplished due to several issues, including the fact that Cortazar *et al* (2013) parallelized their algorithms on Matlab, whereas we used a C++ algorithm on a single CPU.[25] Nevertheless, it appears (from Cortazar *et al* (2013, Table 4)) that the FIK-F(400) routine prices about five options per second. As we shall see later, in Table 6 the FP-A/B algorithm is about three to four orders of magnitude faster than this, at comparable or better accuracy levels.

In Table 4, we extend the analysis to all twelve options in Cortazar *et al* (2013, Tables 2 and 3). Rather than reporting individual pricing errors, we here just report the root mean square (RMS) and relative RMS (RRMS) errors across all option prices. For reference, we also include results from a finite-difference grid solver.

The conclusions here are similar to those above: at the chosen discretization levels, the FP-A method is the most accurate, with the FP-B and 1000-step PDE solver tied for second place. Once again, the computational complexity of the FP-A/B method is far less than its competitors.

6.1.3 Timing and convergence

To expand on our timing and convergence results, we now move away from individual option tests and instead focus on a large set of option payouts and model parameters. This set is found by generating all possible combinations of the parameter ranges in Table 5.

In Table 6, we show select timing and error statistics for FP-A on the set of options in Table 5; note that we extended the error reporting to now also include maximum absolute error (MAE) and maximum relative error (MRE). For brevity, we omitted the FP-B numbers, which were, on average, about five to ten times less precise and had similar computation times. For better intuition about the timing numbers, Table 7 lists errors and computation times for a Crank–Nicolson finite-difference grid method, a popular method used in many banks.

Tables 6 and 7 confirm our earlier observation: the FP-A method outperforms the finite-difference solver at all levels and measures of precision, with the outperformance growing rapidly as a function of required precision. For instance, at a (fairly crude) RMSE precision tolerance of approximately 0.5×10^{-3}, the FP-A method is around 400 to 500 times faster than a PDE solver; but at a (high) precision tolerance of

[25] While Cortazar *et al* (2013) state on p. 22 that their method is "fast because iterations are performed in parallel", it is hard to figure out precisely which of their timing results relied on parallelization and which ones did not. Additionally, it is not stated how many cores were used for those results that relied on parallel processing.

TABLE 4 Estimated half- and three-year put option price errors (RMS and RRMS) for $K = 100$ and S equal to 80, 100 or 120.

	PDE 100	PDE 1000	Bin 1000	Bin 10 000	FIK-F 60	FIK-F 400	KJK 32	FP-A (10,3,7)	FP-B (10,3,7)
RMS	2.9E−03	4.6E−05	2.7E−04	3.0E−04	9.3E−04	6.5E−05	2.8E−03	6.8E−06	7.4E−05
RRMS	1.5E−04	2.7E−06	8.1E−04	8.8E−05	2.7E−04	1.9E−05	8.3E−04	2.5E−07	2.6E−06

Model settings were $r = q = 4\%$ and σ either 0.2 or 0.5. The "PDE" method is a Crank–Nicolson finite-difference grid with an equal number of asset steps and time steps, set as indicated in the table. The first and second row naming conventions and settings are otherwise identical to those used in Table 3. The "FIK-F" and "KJK" numbers were extracted from Cortazar *et al* (2013, Tables 2 and 3).

TABLE 5 Model and contract parameter ranges for timing and precision tests in Tables 6 and 7.

Parameter	Range
$r = q$	{2%,4%,6%,8%,10%}
S	{25,50,80,90,100,110,120,150,175,200}
T	{1/12,0.25,0.5,0.75,1,2,3}
σ	{0.1,0.2,0.3,0.4,0.5,0.6}

In all cases $K = 100$.

TABLE 6 American put errors and timing results for fixed point system A ("FP-A"), using various values of l, m, n and p, and covering the parameter ranges in Table 5.

	(l, m, n); p						
	(3,1,2); 5	**(5,1,4); 8**	**(7,2,6); 15**	**(12,3,8); 25**	**(20,4,10); 50**	**(33,5,12); 65**	**(41,6,16); 81**
RMSE	7.2E−03	3.6E−04	3.3E−05	4.2E−06	3.3E−07	6.6E−08	7.5E−09
RRMSE	4.2E−04	1.4E−05	2.7E−06	3.6E−07	2.3E−08	1.8E−09	1.8E−10
MAE	6.2E−02	3.2E−03	4.1E−04	3.7E−05	2.7E−06	1.5E−06	1.6E−07
MRE	2.7E−03	8.7E−05	2.3E−05	2.8E−06	1.8E−07	2.9E−08	3.3E−09
Opt/Sec	192 307	96 154	37 594	13 928	5734	2732	1399

The last two columns used tanh–sinh quadrature; all other columns used Gauss–Legendre quadrature. The "opt/sec" row lists average calculation times in option prices per second. As in Cortazar *et al* (2013), we removed very low option prices from the computations; the minimum price threshold was set to 0.5. This left a total of 1675 options in the test set.

10^{-6}, the FP-A method is about 25 000 times faster than the finite-difference grid. We note in passing that if one is happy with the 10^{-3} RMSE threshold, our (only casually optimized) FP-A routine running on a middle-of-the-road PC has a throughput in the order of 100 000 option prices per second.

6.2 General case

We now turn to the general case, where $r = q$ is no longer necessarily true. Unless $r \gg q$, it is still the case that the FP-A method is more accurate than FP-B, by roughly one order of magnitude. However, as we hinted in Section 3.3, the FP-A method has a tendency to generate oscillations – and sometimes outright blow up – for strongly drift-dominated dynamics, ie, when $r - q$ is large relative to σ. This behavior is somewhat similar to, say, that of many finite-difference methods (see, for example, Tavella and Randall 2000). An earlier version of this paper discussed how to eliminate the FP-A oscillations by introducing dampening factors into the fixed point iterations, but the introduction of such remedies sometimes renders the FP-A method

TABLE 7 American put errors and timing results for a Crank–Nicolson finite-difference grid method, covering the parameter ranges in Table 5.

	PDE steps				
	250	**500**	**1000**	**5000**	**10 000**
RMSE	4.4E−04	1.2E−04	3.5E−05	2.5E−06	8.6E−07
RRMSE	3.3E−05	1.0E−05	3.4E−06	2.7E−07	9.3E−08
MAE	4.2E−03	9.8E−04	2.1E−04	8.6E−06	2.9E−06
MRE	1.1E−04	3.8E−05	1.3E−05	1.2E−06	4.4E−07
Opt/Sec	270	109	32	1.4	0.35

The numbers of time and asset steps are set equal at the values listed in the table. As in Table 6, options with prices less than 0.5 were removed from the set.

TABLE 8 Model and contract parameter ranges for timing and precision tests in Tables 9 and 10.

Parameter	**Range**
r	{2%,4%,6%,8%,10%}
q	{0%,4%,8%,12%}
S	{25,50,80,90,100,110,120,150,175,200}
T	{1/12,0.25,0.5,0.75,1}
σ	{0.1,0.2,0.3,0.4,0.5,0.6}

In all cases, $K = 100$.

slower (and more complex) than the FP-B method. A safe and robust recommendation is instead to always use the FP-B method for the nonfutures case, ie, whenever $r \neq q$. While one can obviously do better through intelligent branching between FP-A and FP-B, say, the speed and robustness of the FP-B method most likely will make such complications unnecessary in practice. For simplicity, all tests below are therefore based on the FP-B method.

6.2.1 Bulk test

In our first test, we first consider a bulk setup similar to the one in Section 6.1.3. For this, we use the parameter ranges described in Table 8. We report errors and timing results for method FP-B in Table 9. For comparison, Table 10 lists results for a binomial tree and a finite-difference grid.

As evidenced by the tables, our earlier conclusions carry over to the case $r \neq q$: for a given level of precision, the fixed point method is several orders of magnitude faster than binomial trees and finite-difference grids.

TABLE 9 Various error measures on American put prices for fixed point method B (FP-B), using various values of l, m, n and p.

	(l,m,n); p				
	(5,2,4); 8	**(8,4,6); 15**	**(21,6,10); 41**	**(25,8,12); 51**	**(31,16,16); 61**
RMSE	7.5E−04	1.2E−04	1.9E−06	3.6E−07	4.5E−08
RRMSE	1.3E−04	2.5E−05	3.4E−07	7.7E−08	4.3E−09
MAE	5.0E−03	6.9E−04	4.2E−05	1.1E−05	1.2E−06
MRE	1.2E−03	2.3E−04	1.1E−05	3.6E−06	1.3E−07
Opt/Sec	68 027	20 001	3650	2058	641

The numbers in the first two columns used Gauss–Legendre quadrature; the remaining numbers used tanh–sinh quadrature. Parameter ranges were as in Table 8, but with all option prices less than 0.5 removed from the data set. This left a total of 4495 options in the test set.

6.2.2 *Literature comparison: fast trees*

For low-to-medium precision levels, there is evidence that binomial trees can be effectively "tuned" to the problem of American option pricing. For instance, Joshi (2009) and Chen and Joshi (2010) recently presented binomial tree results that, by some, are apparently considered the current benchmarks for high-efficiency American option pricing in the Black–Scholes model.[26] Joshi (2009) ranks 220 different binomial tree methods, including Leisen–Reimer and Tian trees, as well as a multitude of extrapolation and smoothing techniques. Chen and Joshi (2010) refines the best-performing method further, for an additional 50% performance gain, to arrive at what the authors characterize as "the most efficient known numerical method to value American put options in the Black–Scholes model at the important accuracy levels of 10^{-3} to 10^{-4}". Table 11 below compares our fixed point method B to the timing results in Chen and Joshi (2010).[27]

As the table demonstrates, for the RMS precision range of $10^{-3} - 10^{-4}$, to which the trees in Chen and Joshi (2010) were specifically tuned, the FP-B method is here on average about 120 times more efficient than the best results in Chen and Joshi (2010). Of course, should one desire tighter tolerances than in Chen and Joshi (2010), then the speed advantage of the FP-B method would become even more pronounced, as we experienced in Section 6.1.1. For very crude tolerances (eg, 10^{-2}), the QD^+

[26] We thank a referee for pointing this out to us.

[27] Joshi (2009) and Chen and Joshi (2010) use a 3GHz Pentium 4 processor, which for a low-memory single-threaded analytics application should be comparable to our 2GHz Xeon processor.

TABLE 10 Various error measures on American put prices for a binomial tree ("Bin") and for a Crank–Nicolson finite-difference grid ("PDE").

	Method						
	Bin		PDE				
	100 steps	1000 steps	250 steps	500 steps	1000 steps	5000 steps	10 000 steps
RMSE	1.4E−02	1.4E−03	4.2E−04	1.2E−04	3.7E−05	3.1E−06	1.2E−06
RRMSE	3.5E−03	2.9E−04	6.7E−05	2.1E−05	6.9E−06	6.6E−07	2.7E−07
MAE	5.9E−02	6.0E−03	2.1E−03	6.3E−04	1.5E−05	1.3E−05	5.3E−06
MRE	3.5E−02	3.3E−03	5.7E−04	1.6E−04	4.7E−05	4.3E−06	1.8E−06
Opt/Sec	8718	603	270	104	32	1.37	0.34

The finite-difference grid is set to have an equal number of time and asset steps. The set of options was identical to that used in Table 9.

TABLE 11 American put pricing speed, in options per second, for a given maximal RMS error target.

RMS target	Best tree	FP-B
1.00E−03	725.3	51 813
5.00E−04	350.0	35 714
1.00E−04	60.3	11 655

The test set consists of 2000 American put options, generated randomly by the prescription in Chen and Joshi (2010); note that $q = 0$ for all options. The "best tree" results were the fastest results reported in Chen and Joshi (2010, Figure 9); the FP-B results were computed using fixed point method B with Gauss–Legendre quadrature. The (l, m, n, p) settings used were (5,2,5,11), (6,3,5,15) and (13,4,7,31); the resulting RMS errors were 9.2E−4, 4.0E−4 and 8.6E−5, respectively.

first boundary guess is typically sufficient (ie, $m = 0$), which of course outperforms any tree algorithm by a substantial margin.

7 EXTENSIONS

While our numerical scheme was developed for American puts (and calls, through put–call symmetry) on an underlying following a constant-parameter lognormal process, certain extensions to both the payout and the process are possible. For instance, Andersen (2007) and Andreasen (2007) (among others) discuss applications of integral equation to American and Bermudan swaptions in Gaussian or near-Gaussian interest rate models. We discuss a few other extensions here.

7.1 American exchange options

An American exchange option on two underlyings, S_1 and S_2, has the payout

$$p(v) = (c_1 S_1(v) - c_2 S_2(v))^+,$$

where c_1 and c_2 are nonzero constants, and $v \in [0, T]$ is the time at which the option contract gets exercised by its holder. If we assume that S_1 and S_2 are correlated GBM processes with constant volatilities, drifts and correlation, then a simple measure change allows one (see Bjerksund and Stensland 1993b) to recast the American exchange option as an ordinary American put option, which may then readily be priced by the methods in Section 5.

7.2 Time-dependent parameters

Consider now again the regular American put, but assume that r, q and σ in (2.1) are no longer constants but instead smooth, well-behaved deterministic functions of time:

$$\mathrm{d}S(t)/S(t) = (r(t) - q(t))\,\mathrm{d}t + \sigma(t)\,\mathrm{d}W(t). \tag{7.1}$$

For a T-maturity American put, there will, as in (2.3), exist an optimal T-indexed critical stock price function $S_T^*(t)$, below which the option should be exercised. With time-dependent parameters, we may no longer represent the critical stock price as a boundary function B depending solely on $T-t$. However, we can always tag the boundary function with a specific maturity index T, writing $B_T(\tau) = S_T^*(t)$, with $\tau = T-t$. As this boundary function now only applies to a single maturity T, it is less natural to work in reversed time than was the case for time-homogeneous parameters; nevertheless, for consistency and comparison with earlier results, we will work with B_T, rather than with S_T^*. Note that introducing time dependency invalidates the usage of a single exercise boundary function for all maturities. This is rarely of consequence in practical applications, where it is common to compute the exercise boundary anew for each option, rather than use a cached "universal" boundary for all maturities (see the discussion at the beginning of Section 6).

For the process (7.1), we may rely on (2.6) to establish the put value. To simplify notation, define

$$
\begin{aligned}
P(t,T) &= \exp\left(-\int_t^T r(u)\,\mathrm{d}u\right),\\
Q(t,T) &= \exp\left(-\int_t^T q(u)\,\mathrm{d}u\right),\\
\Sigma(t,T) &= \int_t^T \sigma(u)^2\,\mathrm{d}u,
\end{aligned}
$$

which allows us to write, for the T-maturity put price with $\tau = T-t$,

$$
\begin{aligned}
&V_T(\tau,S)\\
&\quad = v_T(\tau,S)\\
&\qquad + \int_t^T \mathbb{E}(P(t,s)(r(s)K - q(s)S(s))1_{S(s)<B_T(T-s)} \mid S(t)=S)\,\mathrm{d}s\\
&\quad = v_T(\tau,S)\\
&\qquad + \int_0^\tau P(T-\tau,T-u)Kr(T-u)\Phi(-d_-(\tau-u,S/B_T(u),T-\tau))\,\mathrm{d}u\\
&\qquad - \int_0^\tau Q(T-\tau,T-u)Sq(T-u)\Phi(-d_+(\tau-u,S/B_T(u),T-\tau))\,\mathrm{d}u.
\end{aligned}
\tag{7.2}
$$

Here, both the American put price (V_T) and the European put price (v_T) are subindexed with maturity T to reflect the time dependence in the process parameters. Also, it is

necessary to redefine d_+ and d_- to have an extra time argument:

$$d_\pm(\delta, z, t) = \frac{\ln z + \int_t^{t+\delta} (r(u) - q(u))\,\mathrm{d}u \pm \frac{1}{2}\int_t^{t+\delta} \sigma(u)^2\,\mathrm{d}u}{\sqrt{\int_t^{t+\delta} \sigma(u)^2\,\mathrm{d}u}}$$
$$= \frac{\ln(zQ(t, t+\delta)/P(t, t+\delta)) \pm \frac{1}{2}\Sigma(t, t+\delta)}{\sqrt{\Sigma(t, t+\delta)}},$$

with the European put price now being

$$v_T(\tau, S) = P(t,T)K\Phi(-d_-(\tau, S/K, T-\tau)) - Q(t,T)S\Phi(-d_+(\tau, S/K, T-\tau)).$$

Using (7.2) at $S = B_T(\tau)$, we get a boundary representation similar to (2.14). Further rearranging (as in (2.12)), we get the time-dependent version of fixed point system B, (3.7) and (3.8):

$$B_T(\tau) = K\frac{P(T-\tau, T)}{Q(T-\tau, T)}\frac{N_T(\tau, B_T)}{D_T(\tau, B_T)}, \tag{7.3}$$

with

$$N_T(\tau, B_T) = \Phi(d_-(\tau, B_T(\tau)/K, T-\tau)) + \int_0^\tau r(T-u)\frac{P(T-\tau, T-u)}{P(T-\tau, T)}\Phi(d_-(\tau-u, B_T(\tau)/B_T(u), T-\tau))\,\mathrm{d}u, \tag{7.4}$$

$$D_T(\tau, B_T) = \Phi(d_+(\tau, B_T(\tau)/K, T-\tau)) + \int_0^\tau q(T-u)\frac{Q(T-\tau, T-u)}{Q(T-\tau, T)}\Phi(d_+(\tau-u, B_T(\tau)/B_T(u), T-\tau))\,\mathrm{d}u. \tag{7.5}$$

Not surprisingly, (7.4) and (7.5) generalize (3.7) and (3.8) by, essentially, replacing the integration kernel terms $\sigma^2(\tau-u)$, $r\mathrm{e}^{ru}$ and $q\mathrm{e}^{qu}$ with

$$\Sigma(T-\tau, T-u), \qquad r(T-u)\frac{P(T-\tau, T-u)}{P(T-\tau, T)}, \qquad q(T-u)\frac{Q(T-\tau, T-u)}{Q(T-\tau, T)}, \tag{7.6}$$

respectively.

The substitutions in (7.6) carry over directly to fixed point system A. For brevity, we omit the obvious expressions.

Numerical computation of the boundary B_T from (7.3)–(7.5) (or a version for fixed point system A) can be done by a straightforward adjustment of the spectral collocation

method in Section 5, essentially by modification of the integration kernels as indicated above. We trust that the reader can complete the scheme, but offer several remarks.

First, when making an initial guess for $B_T(\tau)$, it is useful to compute approximating constant values of r, q and σ for usage in either QD$^+$ or the analytical approximation scheme in Section 4.5. For instance, when pricing the option at time $t = 0$, we might use parameter averages computed as

$$\bar{x} = \frac{1}{\bar{T}} \int_0^{\bar{T}} x(u)\,\mathrm{d}u, \quad x = r, q, \sigma^2, \tag{7.7}$$

where the dimensioning time horizon $\bar{T}$ could, for instance, be set to T or to some fraction of T (eg, $T/2$).

Second, with the inclusion of additional time-dependent terms in the integrals, it is likely that one, for a given level of precision, needs more collocation nodes (n) and integration steps (m) than in the constant-parameter setup.[28] How much additional work is required would depend on how strongly r, q and σ depend on time, so, while brute force is always an option, a sophisticated algorithm would likely be adaptive in its choice of n and m, with a user-specified stopping criterion. It should be noted that some quadrature and interpolation rules are better suited for adaptive algorithms than others, in the sense that these schemes make adding incremental nodes to an existing grid straightforward. The Chebyshev spacing (5.15) is well suited for an adaptive algorithm, but the Gauss–Legendre quadrature method is not; instead, one could use Clenshaw–Curtis or Gauss–Kronrod quadrature. As mentioned earlier, the former method conveniently uses Chebyshev spacing and, despite some theoretical drawbacks, appears to offer a performance in practice that is nearly as good as Gauss–Legendre quadrature.

Finally, as the initial guess based on (7.7) is likely to be less accurate than in the constant-parameter case, one would expect to need more iterations l to converge to an accurate solution of the collocation iteration. Again, it might be useful to introduce a user-specified stopping criterion that stops the iterations when improvements are below a certain absolute or relative error threshold.

We note that one extreme case of time dependence occurs if dividends (q) are paid discretely, rather than continuously in time. For this case, however, the boundary properties change materially, and the algorithm outlined in this paper requires substantial modifications – we refer to Andersen *et al* (2014).

[28] This effect will obviously affect all numerical routines, not just the one in this paper.

7.3 Other extensions

So far, we have only dealt with lognormal processes, but (2.6) applies to any sufficiently regular SDE for S. In practice, however, we would need substantial analytical tractability in order to use the scheme.[29] For instance, if r and q are deterministic, we need closed-form expressions – or at least fast and accurate approximations – for the two quantities ($s > t$):

$$\mathbb{E}(1_{S(s)<S_T^*(s)} \mid S(t) = S), \qquad \mathbb{E}(S(s)1_{S(s)<S_T^*(s)} \mid S(t) = S). \tag{7.8}$$

Aside from the lognormal case, specific models where this is possible include the CEV process in Cox (1975):

$$\mathrm{d}S(t) = \mu S(t)\,\mathrm{d}t + \sigma S(t)^p\,\mathrm{d}W(t),$$

or the displaced diffusion process

$$\mathrm{d}S(t) = \mu S(t)\,\mathrm{d}t + \sigma(bS(t) + (1-b)S(0))\,\mathrm{d}W(t).$$

The development of collocation schemes for these processes (and for their extensions to time-dependent parameters) closely follows the steps in Sections 5 and 7.2.

Extensions to S-processes that include jumps are also possible, although the decomposition (2.4) of the American option price will now contain a new term originating from the fact that S may move between the exercise and continuation regions in discrete jumps that cross over the boundary (see Pham (1997) and Gukhal (2001) for the details). Numerical methods for the jump-diffusion case are still in relative infancy, and application of the methods in Section 5 is an interesting area for future research.

Finally, let us touch upon the extensions to stochastic volatility models, such as the Heston process in Heston (1993). For this type of process, the exercise boundary B will depend on the state of a stochastic volatility driver ϑ as well as on time to maturity, a significant increase in both dimension and complexity. Several authors (for example, Ziogas and Chiarella 2005) have attempted to simplify the problem by simplifying this dependency

$$\ln B(\tau, \vartheta) \approx b_0(\tau) + \vartheta b_1(\tau).$$

With this ansatz, a computationally efficient decomposition similar to (2.4) is possible, although the necessary terms in (7.8) will have to be computed by potentially slow

[29] For general SDEs, a finite-difference method would most likely be the method of choice.

inverse Fourier methods.[30] The competitiveness of the resulting scheme against, say, modern finite-difference methods is, in our opinion, still an open question.

8 CONCLUSION

In this paper, we have introduced a new algorithm for very fast computation of American put and call option prices in a lognormal (Black–Scholes) diffusion setting. The method proposed here is based on an integral equation formulation for the fixed boundary, an approach that has recently seen renewed interest after promising results in Kim *et al* (2013) and Cortazar *et al* (2013). Combining modern methods for integral equations with a set of carefully designed transformations of the boundary, we are able to speed up computational efficiency by several orders of magnitude compared with competing approaches. Our method is straightforward to implement, fast enough for real-time ticking risk systems and, if properly tuned, capable of producing high-precision price estimates at a rate of many tens of thousands of options per second per CPU.

While a significant improvement on all known (to us) speed and accuracy records of a classical benchmark problem should, in itself, be of interest to many, the primary objective of this paper was a concrete, practical one: the real-time risk management of American option portfolios. As discussed in Section 1, the Black–Scholes dynamics remain a standard reference model for virtually all practical trading activity in American puts and calls, especially on exchanges and for short- and medium-dated over-the-counter structures.

While the development of our method as well as our test cases focused on the Black–Scholes dynamics, extensions are certainly possible, as we briefly discussed in Section 7. An obvious avenue for future research would be a fuller development (and testing) of the various extensions in Section 7, perhaps especially the challenging problems of handling jumps, discrete dividends and stochastic volatility. Extensions to more payouts more complicated than puts and calls is also an interesting problem.

Beyond these topics, certain questions about the constant-parameter lognormal case fell outside the scope of this paper, and these could be considered in future work. For instance, the logic (and level of adaptiveness) around choosing values for l, m, n and p to optimize the precision-speed tradeoff for any specific option was only addressed cursorily in our paper through rough "rules of thumb". While this may be adequate for many applications, a more sophisticated analysis would surely improve our computational results even further. Finally, it remains an open question whether

[30] For each τ, Ziogas and Chiarella (2005) consider two carefully chosen levels of ϑ to form a linked system of integral equations for b_0 and b_1.

any of the many different possible integral equations for the exercise boundary (see Section 2.3) may have even better numerical properties than those considered in our paper.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

ACKNOWLEDGEMENTS

The authors are grateful for comments received from their colleagues and from participants at the January 2014 Aarhus University Quant Week, the January 2014 Bloomberg Quant Seminar, the March 2014 For Python Quants Conference and the July 2014 Risk Magazine Quant Congress USA.

REFERENCES

Ahn, C., Choe, H., and Lee, K. (2009). A long time asymptotic behavior of the free boundary for an American put. *Proceedings of the American Mathematical Society* **137**(10), 3425–3436 (http://doi.org/ff48h6).

AitSahlia, F., and Carr, P. (1997). American options: a comparison of numerical methods. In *Numerical Methods in Finance*, pp. 67–87. Cambridge University Press (http://doi.org/bhsq).

AitSahlia, F., and Lai, T. L. (2001). Exercise boundaries and efficient approximations to American option prices and hedge parameters. *The Journal of Computational Finance* **4**(4), 85–104 (http://doi.org/bhsr).

Andersen, L. (2007). Notes on fast Bermudan swaption pricing with fees. Technical Report, Bank of America.

Andersen, L., and Broadie, M. (2004). Primal–dual simulation algorithm for pricing multidimensional American options. *Management Science* **50**(9), 1222–1234 (http://doi.org/ckrp3p).

Andersen, L., Lake, M., and Offengenden, D. (2014). High performance American option pricing: discrete dividends. Technical Report, Bank of America. In preparation.

Andreasen, J. (2007). The fastest Bermudan pricer in the West. Technical Report, Bank of America.

Atkinson, K. E. (1992). A survey of numerical methods for solving nonlinear integral equations. *Journal of Integral Equations and Applications* **4**(1), 15–46 (http://doi.org/b5vcq7).

Bailey, D. H., Jeyabalan, K., and Li, X. S. (2005). A comparison of three high-precision quadrature schemes. *Experimental Mathematics* **14**(3), 317–329 (http://doi.org/ft77nc).

Barles, G., Burdeau, J., Romano, M., and Samsoen, N. (1995). Critical stock price near expiration. *Mathematical Finance* **5**(2), 77–95 (http://doi.org/dp582s).

Barone-Adesi, G., and Elliott, R. J. (1991). Approximations for the values of American options. *Stochastic Analysis and Applications* **9**(2), 115–131 (http://doi.org/d3q5b6).

Barone-Adesi, G., and Whaley, R. E. (1987). Efficient analytic approximation of American option values. *Journal of Finance* **42**(2), 301–320 (http://doi.org/bhss).

Bayraktar, E., and Xing, H. (2009). Analysis of the optimal exercise boundary of American options for jump diffusions. *SIAM Journal on Mathematical Analysis* **41**(2), 825–860 (http://doi.org/dnbrcz).

Berrut, J.-P., and Trefethen, L. N. (2004). Barycentric Lagrange interpolation. *SIAM Review* **46**(3), 501–517 (http://doi.org/csgngt).

Bjerksund, P., and Stensland, G. (1993a). Closed-form approximation of American options. *Scandinavian Journal of Management* **9**, S87–S99 (http://doi.org/db8rt5).

Bjerksund, P., and Stensland, G. (1993b). American exchange options and a put–call transformation: a note. *Journal of Business Finance and Accounting* **20**(5), 761–764 (http://doi.org/fth4ht).

Brennan, M. J., and Schwartz, E. S. (1978). Finite difference methods and jump processes arising in the pricing of contingent claims: a synthesis. *Journal of Financial and Quantitative Analysis* **13**(3), 461–474 (http://doi.org/bkhp84).

Broadie, M., and Detemple, J. (1996). American option valuation: new bounds, approximations, and a comparison of existing methods. *Review of Financial Studies* **9**(4), 1211–1250 (http://doi.org/dhrwft).

Brunner, H. (1984). The numerical solution of integral equations with weakly singular kernels. In *Numerical Analysis*, pp 50–71. Springer (http://doi.org/bz7xt7).

Brunner, H. (1985). The numerical solution of weakly singular Volterra integral equations by collocation on graded meshes. *Mathematics of Computation* **45**(172), 417–437 (http://doi.org/fpvgmh).

Brunner, H. (2004). *Collocation Methods for Volterra Integral and Related Functional Differential Equations*, Volume 15. Cambridge University Press (http://doi.org/fr9q7f).

Bunch, D. S., and Johnson, H. (2000). The American put option and its critical stock price. *Journal of Finance* **55**(5), 2333–2356 (http://doi.org/ft99kj).

Carr, P. (1998). Randomization and the American put. *Review of Financial Studies* **11**(3), 597–626 (http://doi.org/cjv3x4).

Carr, P., and Faguet, D. (1994). Fast accurate valuation of American options. Technical Report, Cornell University.

Chadam, J. (2010). Integral equation methods for free boundary problems. In *Encyclopedia of Quantitative Finance*. Wiley (http://doi.org/dssz74).

Chadam, J., and Chen, X. (2003). Analytical and numerical approximations for the early exercise boundary for American put options. *Dynamics of Continuous Discrete and Impulsive Systems* A **10**, 649–660.

Chen, T., and Joshi, M. (2010). Truncation and acceleration of the Tian tree for the pricing of American put options. SSRN Working Paper Series, Social Science Research Network (http://doi.org/bhst).

Chen, X., and Chadam, J. (2007). A mathematical analysis of the optimal exercise boundary for American put options. *SIAM Journal on Mathematical Analysis* **38**(5), 1613–1641 (http://doi.org/dwpbt6).

Chen, X., Cheng, H., and Chadam, J. (2011). Far-from-expiry behavior of the American put option on a dividend-paying asset. *Proceedings of the American Mathematical Society* **139**(1), 273–282 (http://doi.org/fbz4xw).

Chen, X., Cheng, H., and Chadam, J. (2013). Nonconvexity of the optimal exercise boundary for an American put option on a dividend-paying asset. *Mathematical Finance* **23**(1), 169–185 (http://doi.org/bhsv).

Cheng, J., and Zhang, J. E. (2012). Analytical pricing of American options. *Review of Derivatives Research* **15**(2), 157–192 (http://doi.org/fx4znh).

Cook, J. (2009). Asymptotic analysis of American-style options. PhD Thesis, University of Bath.

Cortazar, G., Medina, L., and Naranjo, L. (2013). A parallel algorithm for pricing American options. Working Paper. URL: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2325377 (http://doi.org/bhsw).

Cox, J. C. (1975). Notes on option pricing I: constant elasticity of variance diffusions. Unpublished note, Stanford University, Graduate School of Business.

Cox, J. C., Ross, S. A., and Rubinstein, M. (1979). Option pricing: a simplified approach. *Journal of Financial Economics* **7**(3), 229–263 (http://doi.org/fdz9rx).

Elnagar, G. N., and Kazemi, M. (1996). Chebyshev spectral solution of nonlinear Volterra–Hammerstein integral equations. *Journal of Computational and Applied Mathematics* **76**(1), 147–158 (http://doi.org/dpr288).

Evans, J. D., Kuske, R., and Keller, J. B. (2002). American options on assets with dividends near expiry. *Mathematical Finance* **12**(3), 219–237 (http://doi.org/c2bqq7).

Forsyth, P. A., and Vetzal, K. R. (2002). Quadratic convergence of a penalty method for valuing American options. *SIAM Journal on Scientific Computation* **23**(6), 2095–2122 (http://doi.org/bdsqm8).

Frontczak, R. (2013). Simple analytical approximations for the critical stock price of American options. Working Paper. URL: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2227626 (http://doi.org/bhsx).

Gukhal, C. R. (2001). Analytical valuation of American options on jump-diffusion processes. *Mathematical Finance* **11**(1), 97–115 (http://doi.org/c539m3).

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* **6**(2), 327–343 (http://doi.org/fg525s).

Hou, C., Little, T., and Pant, V. (2000). A new integral representation of the early exercise boundary for American put options. *The Journal of Computational Finance* **3**, 73–96 (http://doi.org/bhsz).

Huang, J.-Z., and Subrahmanyam, M. G. (1996). Pricing and hedging American options: a recursive integration method. *Review of Financial Studies* **9**(1), 277–300 (http://doi.org/b6d24x).

Jacka, S. D. (1991). Optimal stopping and the American put. *Mathematical Finance* **1**(2), 1–14 (http://doi.org/fq38p3).

Jamshidian, F. (1992). An analysis of American options. *Review of Futures Markets* **11**(1), 72–80.

Johnson, H. E. (1983). An analytic approximation for the American put price. *Journal of Financial and Quantitative Analysis* **18**(1), 141–148 (http://doi.org/csrhzv).

Joshi, M. S. (2009). The convergence of binomial trees for pricing the American put. *The Journal of Risk* **11**(4), 87–108 (http://doi.org/bhvj).

Ju, N. (1998). Pricing by American option by approximating its early exercise boundary as a multipiece exponential function. *Review of Financial Studies* **11**(3), 627–646 (http://doi.org/cm8c84).

Ju, N., and Zhong, R. (1999). An approximate formula for pricing American options. *Journal of Derivatives* **7**(2), 31–40 (http://doi.org/cs659g).

Kallast, S., and Kivinukk, A. (2003). Pricing and hedging American options using approximations by Kim integral equations. *European Finance Review* **7**(3), 361–383 (http://doi.org/c6qpf3).

Kim, I. J. (1990). The analytic valuation of American options. *Review of Financial Studies* **3**(4), 547–572 (http://doi.org/fjtssk).

Kim, I. J., Jang, B.-G., and Kim, K. T. (2013). A simple iterative method for the valuation of American options. *Quantitative Finance* **13**(6), 885–895 (http://doi.org/bhs2).

Kuske, R. A., and Keller, J. B. (1998). Optimal exercise boundary for an American put option. *Applied Mathematical Finance* **5**(2), 107–116 (http://doi.org/dkngj2).

Lee, J., and Paxson, D. A. (2003). Confined exponential approximations for the valuation of American options. *European Journal of Finance* **9**(5), 449–474 (http://doi.org/cn96g9).

Leisen, D. P. J., and Reimer, M. (1996). Binomial models for option valuation-examining and improving convergence. *Applied Mathematical Finance* **3**(4), 319–346 (http://doi.org/dqbtjj).

Li, M. (2008). A quasi-analytical interpolation method for pricing American options. Technical Report, Georgia Institute of Technology.

Li, M. (2009). Analytical approximations for the critical stock prices of American options: a performance comparison. Technical Report, University Library of Munich (http://doi.org/fx59b4).

Longstaff, F. A., and Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies* **14**(1), 113–147 (http://doi.org/b38b5q).

MacMillan, L. W. (1986). Analytic approximation for the American put option. *Advances in Futures and Options Research* **1**(1), 119–139.

McDonald, R. L., and Schroder, M. (1998). A parity result for American options. *The Journal of Computational Finance* **1**(3), 5–13 (http://doi.org/bhvh).

Merton, R. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science* **4**, 141–183 (http://doi.org/cvh6pc).

Ostrov, D. N., and Goodman, J. (2002). On the early exercise boundary of the American put option. *SIAM Journal on Applied Mathematics* **62**(5), 1823–1835 (http://doi.org/c7s47w).

Pham, H. (1997). Optimal stopping, free boundary, and American option in a jump-diffusion model. *Applied Mathematics and Optimization* **35**(2), 145–164 (http://doi.org/dgr7bf).

Press, W. H. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.

Subrahmanyam, M. G., and Yu, G. G. (1993). Pricing and hedging American options: a unified method and its efficient implementation. Working Paper, New York University Salomon Center, Leondard N. Stern School of Business.

Tang, T., Xu, X., and Cheng, J. (2008). On spectral methods for Volterra integral equations and the convergence analysis. *Journal of Computational Mathematics* **26**(6), 825–837.

Tavella, D., and Randall, C. (2000). *Pricing Financial Instruments: The Finite Difference Method*, Volume 13. Wiley.

Traub, J. F. (1982). *Iterative Methods for the Solution of Equations*. American Mathematical Society.

Trefethen, L. N. (2008). Is Gauss quadrature better than Clenshaw–Curtis? *SIAM Review* **50**(1), 67–87 [DOI:10.1137/060659831] (http://doi.org/fqzfrs).

Trefethen, L. N. (2012). Six myths of polynomial interpolation and quadrature. *Mathematics Today* **47**(4), 184–188.

Tsitsiklis, J. N., and Van Roy, B. (1999). Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control* **44**(10), 1840–1851 (http://doi.org/d86z39).

Wilmott, P., Howison, S., and Dewynne, J. (1995). *The Mathematics of Financial Derivatives: A Student Introduction*. Cambridge University Press (http://doi.org/bhs3).

Xie, D., Edwards, D. A., Schleiniger, G., and Zhu, Q. (2011). Characterization of the American put option using convexity. *Applied Mathematical Finance* **18**(4), 353–365 (http://doi.org/dk8g9c).

Yue-Kuen, K. (1998). *Mathematical Models of Financial Derivatives*. Springer.

Zhang, J. E., and Li, T. (2010). Pricing and hedging American options analytically: a perturbation method. *Mathematical Finance* **20**(1), 59–87 (http://doi.org/dg7m38).

Zhu, S.-P. (2006). A new analytical approximation formula for the optimal exercise boundary of American put options. *International Journal of Theoretical and Applied Finance* **9**(7), 1141–1177 (http://doi.org/ctr9w5).

Zhu, S.-P., and He, Z.-W. (2007). Calculating the early exercise boundary of American put options with an approximation formula. *International Journal of Theoretical and Applied Finance* **10**(7), 1203–1227 (http://doi.org/dq9r9c).

Ziogas, A., and Chiarella, C. (2005). Pricing American options under stochastic volatility. Technical Report, Society for Computational Economics.

Journal of Computational Finance **20**(1), 89–111
DOI: 10.21314/JCF.2016.309

Risk journals

Research Paper

Adjusting exponential Lévy models toward the simultaneous calibration of market prices for crash cliquets

Peter Carr,[1] Ajay Khanna[2] and Dilip B. Madan[3]

[1]Finance and Risk Engineering, New York University Tandon School of Engineering, 12 Metrotech Center, Brooklyn, NY 11201, USA; email: pcarr@nyc.rr.com

[2]Apartment 5/6W, 49–51 Warren Street, New York, NY 10007, USA; email: ajayk62@yahoo.com

[3]Robert H. Smith School of Business, University of Maryland, College Park, MD 20742, USA; email: dbm@rhsmith.umd.edu

(Received August 5, 2014; revised August 6, 2015; accepted October 16, 2015)

ABSTRACT

In this paper, option-calibrated exponential Lévy models are observed to typically overprice crash cliquets. Typical model Lévy tails are then not crash-market consistent. A general tail-thinning strategy is introduced that may be implemented on a class of parametric Lévy models closed under exponential tilting. Implementation on the Carr–Geman–Madan–Yor (CGMY) model leads to the CGAKMY model with a thinning function of $(1 + A|x|)^{-K}$. It is observed that this model adjustment can be crash-market consistent.

Keywords: completely monotone function; Gauss Laguerre quadrature; gap risk pricing; beta exposure pricing; CGMY model; negative binomial process.

Corresponding author: D. B. Madan

Print ISSN 1460-1559 | Online ISSN 1755-2850

1 INTRODUCTION

Consider a claim that pays the one-time excess of a daily drop in the Standard & Poor's 500 index (S&P 500) over 40%. Such a claim now trades, and its price on November 11, 2009 was 9 basis points (bps). The event in question has never occurred, and hopefully it never will, but it is a theoretical possibility. Working out the actual probability of such an event is an exercise in the extrapolation of physical jump arrival rates, with the pricing of such events being a comparable exercise for the risk-neutral jump arrival rates. For an empirical study in this direction, we refer to Bollerslev and Todorov (2011), who estimate a risk-neutral probability of 862bps for a 20% drop for a 2bps actual probability in such an event. Leaving aside physical probabilities, as one cannot observe the relative frequency of what is outside the realm of experience, we turn our attention to the observed market prices of crash cliquets that pay out the one-time excess moves beyond the specified crash strikes.

We find that a variety of exponential Lévy models, when calibrated to near-the-money option prices, typically overprice the crash cliquets when one employs the pricing formulas for these based on the calibrated Lévy measure (as presented, for example, in Tankov (2010)). These models were built for the pricing of near-the-money options, and they may possess Lévy measures with calibrated tails that are too fat for the information being revealed by observed market prices for crash cliquets. We are thus led to seek some general tail-thinning strategies that may be employed on existing models to get them better in line with the prices of crash cliquets.

It is also possible that markets are currently mispricing these claims, and when these markets correct themselves, we may be more interested in tail-thickening strategies. That is, however, an issue for another time, and given that prices, as they are, are determined in markets, our interest for the moment turns toward tail-thinning while maintaining computational tractability.

An additional advantage of the thinning strategy being introduced here is that it can be implemented on many existing models as well as for a fairly large variety of thinning candidates. The actual implementation is illustrated here on just the CGMY model of Carr, Geman, Madan and Yor (2002) for only a specific thinning function with a single additional parameter.

The resulting model is termed the CGKMY model, where the parameter K now controls the calibration to crash cliquet prices. A further two-parameter extension termed CGAKMY is also entertained.[1] Crash cliquets on single names are then modeled by extending the Carr and Madan (2012) index beta exposure approach. The index risk is priced using the CGAKMY model, coupled with a charge for residual

[1] We are indebted to King Wang for the suggestion to include the additional parameter extension.

idiosyncratic risk modeled as a CGMY process. The beta and the residual CGMY process parameters are estimated by calibrating weekly option prices on the single names.

The thinning strategy is introduced in Section 2. Details for the generalization of the CGMY to the CGKMY and CGAKMY models are presented in Section 3, as are results for the calibration to market prices for November 11, 2009. Section 4 provides results based on calibration to weekly options and crash cliquets for the S&P 500 on May 2, 2014 and a sample of single-name weekly options on this date. The beta exposure approach to single-name cliquets is developed in Section 6, and results on implementing this approach are presented in Section 7. Section 8 concludes.

2 THE THINNING STRATEGY

Consider an initial exponential pure jump Lévy process $X = (X(t), t > 0)$, with characteristic function given by

$$\begin{aligned}\phi_{X(t)}(u) &= E[\exp(\mathrm{i}uX(t))]\\ &= \exp(t[\mathrm{i}ub + \psi_X(u)]),\end{aligned}$$

where

$$\psi_X(u) = \int_{-\infty}^{\infty} (\mathrm{e}^{\mathrm{i}ux} - 1 - \mathrm{i}ux\mathbf{1}_{|x|\leqslant 1})k(x)\,\mathrm{d}x$$

and $k(x)$ is the Lévy density of the process X satisfying

$$\int_{-\infty}^{\infty} (x^2 \wedge 1)k(x)\,\mathrm{d}x < \infty.$$

In the special case of finite variation, we have

$$\int_{-\infty}^{\infty} |x|k(x)\,\mathrm{d}x < \infty,$$

and then we may write

$$\psi_X(u) = \int_{-\infty}^{\infty} (\mathrm{e}^{\mathrm{i}ux} - 1)k(x)\,\mathrm{d}x.$$

In the development, we shall focus our attention on the finite variation case, with the extension to the infinite variation case, following from classical arguments.

We suppose the Lévy density $k(x)$ belongs to a parametric class with a parameter vector θ that is closed under negative exponential tilting. In particular, for every constant $a > 0$ and initial parameter vector θ, we suppose there exists a parameter perturbation function $\chi(a)$ such that

$$\mathrm{e}^{-a|x|}k(x;\theta) = k(x;\theta + \chi(a)). \tag{2.1}$$

Denote the log characteristic function for $k(x;\theta)$ by $\psi_X(u;\theta)$. Many Lévy processes used in practical applications have such a closure property under negative exponential tilting.

As a candidate for a thinning function, consider a completely monotone function $m(x)$, $x > 0$, defined on the positive half line. By Bernstein's theorem, there exists a positive measure such that

$$m(x) = \int_0^\infty \rho(\mathrm{d}u)\mathrm{e}^{-ux}.$$

Assuming the measure ρ has a density with respect to the Lebesgue measure given by $\eta(u)$, we may write

$$m(x) = \int_0^\infty \eta(u)\mathrm{e}^{-ux}\,\mathrm{d}u.$$

With a view to minimizing the perturbation to $k(x)$ near zero, we consider defining a new perturbed Lévy measure $\tilde{k}(x)\,\mathrm{d}x$:

$$\tilde{k}(x) = k(x)m(1+x).$$

It follows that

$$\begin{aligned}\tilde{k}(x) &= k(x)\int_0^\infty \eta(u)\mathrm{e}^{-u(1+x)}\,\mathrm{d}u\\ &= k(x)\int_0^\infty \eta(u)\mathrm{e}^{-ux}\mathrm{e}^{-u}\,\mathrm{d}u.\end{aligned}$$

The integration with respect to the argument u over the half line is now with respect to an exponential weight function; one may approximate by a finite sum using Gauss–Laguerre quadrature and write

$$\tilde{k}(x) \approx k(x)\sum_{i=1}^{N} w_i\eta(u_i)\mathrm{e}^{-u_i x}. \tag{2.2}$$

Alternatively, we may just take the right-hand side of (2.2) as the perturbed model.

The perturbed process $\tilde{X} = (\tilde{X}(t), t > 0)$ has a perturbed Lévy density given by

$$\tilde{\tilde{k}}(x) = k(x)\sum_{i=1}^{N} w_i\eta(u_i)\mathrm{e}^{-u_i x}. \tag{2.3}$$

Its characteristic function is then given by

$$\begin{aligned}\phi_{\tilde{X}(t)}(u) &= E[\exp(\mathrm{i}u\tilde{X}(t))]\\ &= \exp\left(t\sum_{i=1}^{N} w_i\eta(u_i)\psi_X(u;\theta+\chi(u_i))\right).\end{aligned} \tag{2.4}$$

With a closed form for the characteristic function of the log price, one may employ the fast Fourier transform methods of Carr and Madan (1999) to price options under the perturbed process $\tilde{X}$.

2.1 Pricing of crash cliquets

A crash cliquet pays between initiation and maturity, one time only, the first excess percentage daily drop in the market over a prespecified strike times a notional. For a 40% drop and a notional of N, the payoff is zero if $S_j \geqslant 0.6S_{j-1}$, $j = 1, \ldots, t$; otherwise, let $\tau = \inf\{j \mid S_j < 0.6S_{j-1}\}$, and then there is a payoff at time τ in the amount

$$N\left(0.6 - \frac{S_\tau}{S_{\tau-1}}\right).$$

Other typical strikes are $0.65, 0.7, \ldots, 0.9$, reflecting market drops of $35, 30, \ldots, 10$ percentage points.

We model such large daily drops by the large jumps of a continuous-time pure jump process. Let a be the logarithm of $(1-k)$ for a crash of $k\%$. We are then interested in pricing a put option on the size of a down jump in log prices of x that pays for a unit notional of the value

$$(\mathrm{e}^a - \mathrm{e}^x)^+.$$

For a Lévy process with Lévy density $k(x)$, the arrival rate of such a jump is

$$\lambda(a) = \int_{-\infty}^{a} k(x)\,\mathrm{d}x.$$

The claim, if it pays on all such jumps, then pays to time t the value of

$$\int_0^t \int_{-\infty}^{a} (\mathrm{e}^a - \mathrm{e}^x)\mu(\mathrm{d}x, \mathrm{d}u),$$

where $\mu(\mathrm{d}x, \mathrm{d}u)$ is the integer-valued random measure of the jumps.

The price, ignoring discounting, is given by

$$\int_0^t \int_{-\infty}^{a} (\mathrm{e}^a - \mathrm{e}^x)k(x)\,\mathrm{d}x\,\mathrm{d}u.$$

One may incorporate inhomogeneous Lévy systems into the price to get

$$\int_0^t \int_{-\infty}^{a} (\mathrm{e}^a - \mathrm{e}^x)k(x,u)\,\mathrm{d}x\,\mathrm{d}u.$$

For an inhomogeneous Lévy system with discounting, we need to evaluate

$$\int_0^t \int_{-\infty}^{a} \mathrm{e}^{-ru}(\mathrm{e}^a - \mathrm{e}^x)k(x,u)\,\mathrm{d}x\,\mathrm{d}u.$$

We may rewrite this expression as

$$\int_0^t \lambda(u)\mathrm{e}^{-ru}\int_{-\infty}^{a}(\mathrm{e}^a-\mathrm{e}^x)\,p(x,u)\,\mathrm{d}x\,\mathrm{d}u,$$

where

$$\lambda(u)=\int_{-\infty}^{a}k(x,u)\,\mathrm{d}x,$$
$$p(x,u)=\frac{1}{\lambda(u)}k(x,u).$$

The arrival rate of down jumps larger than $(1-\mathrm{e}^a)$ at time u is $\lambda(u)$, and $p(x,u)$ is the conditional density of the jump size, given that such a jump has occurred at time u.

Let $N(t)$ be the number of arrivals before t. The distribution of $N(t)$ is Poisson, with mean Λ and

$$P(N(t)=n)=\frac{\Lambda^n\mathrm{e}^{-\Lambda}}{n!},\quad \Lambda=\int_0^t\lambda(u)\,\mathrm{d}u.$$

The density of the arrival time is

$$g(u)=\frac{\lambda(u)}{\Lambda}.$$

If we pay for just a single jump, then the value is

$$(1-\mathrm{e}^{-\Lambda})\int_0^t g(u)\mathrm{e}^{-ru}\int_{-\infty}^{a}(\mathrm{e}^a-\mathrm{e}^x)\,p(x,u)\,\mathrm{d}x\,\mathrm{d}u$$
$$=\frac{(1-\mathrm{e}^{-\Lambda})}{\Lambda}\int_0^t\mathrm{e}^{-ru}\int_{-\infty}^{a}(\mathrm{e}^a-\mathrm{e}^x)k(x,u)\,\mathrm{d}x\,\mathrm{d}u.$$

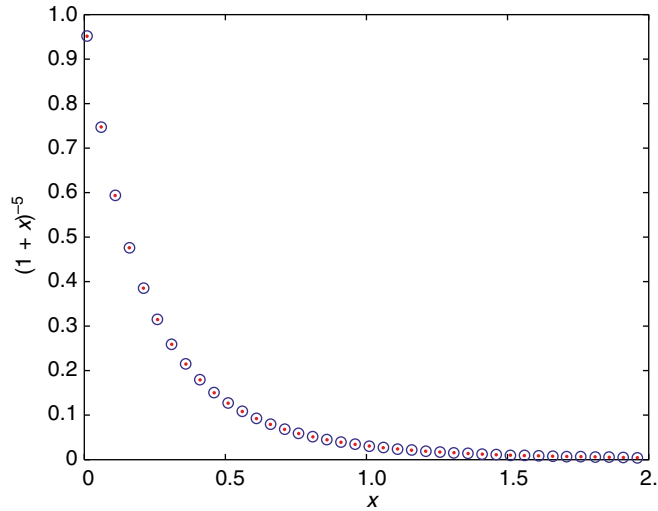
Since the perturbed Lévy density of (2.3) is a weighted average of exponential tilts applied to the original Lévy density, and since, by (2.1), this is equivalent to just a parameter shift, one may also evaluate crash prices for the perturbed density, given that they may be computed for the original measure.

3 FROM CGMY TO CGKMY AND CGAKMY

Consider the thinning function

$$\frac{1}{(1+x)^K}=\frac{1}{\Gamma(K)}\int_0^\infty u^{K-1}\mathrm{e}^{-(1+x)u}\,\mathrm{d}u$$
$$\approx\sum_{i=1}^{N}w_i u_i^{K-1}\mathrm{e}^{-u_i x}.$$

FIGURE 1 Five-point Gauss–Laguerre approximation for $(1 + x)^{-5}$.



For a five-point Gauss–Laguerre quadrature, we may in fact write

$$\frac{1}{(1+|x|)^K} \approx \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\Gamma(a)} \mathrm{e}^{-u_i|x|}.$$

Figure 1 presents a graph of the target and the approximation for $K = 5$. Table 1 presents the precise numeric approximating values at a sample of values for the x variable.

We may note that the five-point approximation is very good for values of K at or above unity. For $K < 1$, there is a departure of the approximation that remains a completely monotone function, but it is not equal to the target of $(1 + |x|)^{-K}$; hence, the CGMY case of $K = 0$ is not a special case.

For the special case of this specific thinning function, one may observe that the thinning function is also the expectation of the exponential of $-Ux$, for the random variable U being gamma distributed with shape parameter K, or the distribution of the gamma process at time K. Recognizing that the negative binomial process with parameter p converges on scaling by p to the gamma process (Kozubowski and Podgórski 2007), we may also write the approximation for small p as

$$\frac{1}{(1+|x|)^K} \approx \sum_{k=0}^{\infty} \binom{K+k-1}{k} p^K (1-p)^k \exp(-pkx), \tag{3.1}$$

TABLE 1 Five-point Gauss quadrature.

x	y	Quadrature
0.0010	0.9950	0.9950
0.1010	0.6181	0.6181
0.2010	0.4002	0.4002
0.3010	0.2683	0.2682
0.4010	0.1853	0.1851
0.5010	0.1312	0.1310
0.6010	0.0951	0.0948
0.7010	0.0702	0.0699
0.8010	0.0528	0.0524
0.9010	0.0403	0.0400
1.0010	0.0312	0.0310
1.1010	0.0244	0.0243
1.2010	0.0194	0.0193
1.3010	0.0155	0.0156
1.4010	0.0125	0.0127
1.5010	0.0102	0.0104
1.6010	0.0084	0.0086
1.7010	0.0070	0.0072
1.8010	0.0058	0.0061
1.9010	0.0049	0.0051

which yields an approximation based on the scaled negative binomial distribution. Alternatively, one may take the right-hand side of (3.1) as the proposed thinning function itself.

The CGMY Lévy measure may be written as

$$k_{\mathrm{CGMY}}(x) = C\frac{\exp(\frac{1}{2}(G-M)x - \frac{1}{2}(G+M)|x|)}{|x|^{1+Y}};$$

the perturbed Lévy measure is then

$$k_{\mathrm{CGKMY}}(x) = \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\Gamma(K)} \exp(-u_i|x|).$$

To ensure that we have the right value at $x = 0$, we normalize by the sum of the weights as opposed to $\Gamma(K)$. Hence, we write

$$\hat{\Gamma}(K) = \sum_{i=1}^{5} w_i u_i^{K-1}$$

and then define the CGKMY Lévy measure by

$$k_{\mathrm{CGKMY}}(x) = C\frac{\exp(\frac{1}{2}(G-M)x - \frac{1}{2}(G+M)|x|)}{|x|^{1+Y}}\sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\hat{\Gamma}(K)}\exp(-u_i|x|).$$

From the analysis of the preceding section, we obtain the log characteristic function for CGKMY:

$$\ln\phi_{\mathrm{CGKMY}}(u) = \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\hat{\Gamma}(K)} C\Gamma(-Y)[(M-\mathrm{i}u+u_i)^Y - (M+u_i)^Y + (G+\mathrm{i}u+u_i)^Y - (G+u_i)^Y].$$

By the property of closure under exponential tilting displayed in (2.1), and given that the only parameter restrictions for the CGMY model are positivity of C, G, M and $0 \leqslant Y < 2$, we note that the log characteristic function for the CGKMY model is well defined.

3.1 CGKMY crash cliquet pricing

For a Lévy process, we have to evaluate

$$\lambda = \int_{-\infty}^{a} k(x)\,\mathrm{d}x,$$

and our crash cliquet value is

$$V(a) = \frac{(1-\exp(-\lambda t))}{\lambda}\left(\mathrm{e}^{-|a|}\lambda - \int_{-\infty}^{a}\mathrm{e}^{x}k(x)\,\mathrm{d}x\right).$$

For CGKMY, we have

$$\begin{aligned}
&\lambda(G,a,K,Y)\\
&= \int_{|a|}^{\infty}\frac{\exp(-G|x|)}{|x|^{1+Y}}\sum_{i=1}^{5} w_i\frac{u_i^{K-1}}{\hat{\Gamma}(K)}\exp(-u_i x)\\
&= \sum_{i=1}^{5} w_i\frac{u_i^{K-1}}{\hat{\Gamma}(K)}\int_{|a|}^{\infty}\frac{\exp(-(G+u_i)x)}{x^{1+Y}}\,\mathrm{d}x\\
&= \sum_{i=1}^{5} w_i\frac{u_i^{K-1}}{\hat{\Gamma}(K)}(G+u_i)^Y\int_{(G+u_i)|a|}^{\infty} w^{-Y-1}\mathrm{e}^{-w}\,\mathrm{d}w\\
&= \sum_{i=1}^{5} w_i\frac{u_i^{K-1}}{\hat{\Gamma}(K)}(G+u_i)^Y\left[\frac{w^{-Y}}{-Y}\mathrm{e}^{-w}\Big|_{(G+u_i)|a|}^{\infty} - \frac{1}{Y}\int_{(G+u_i)|a|}^{\infty} w^{-Y}\mathrm{e}^{-w}\,\mathrm{d}w\right]
\end{aligned}$$

$$= \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\hat{\Gamma}(K)} (G+u_i)^Y \left[\frac{1}{Y}(G+u_i)|a|^{-Y} \mathrm{e}^{-(G+u_i)|a|} \right.$$
$$\left. - \frac{1}{Y}\frac{w^{1-Y}}{1-Y}\mathrm{e}^{-w}\Big|_{(G+u_i)|a|}^{\infty} - \frac{1}{Y(1-Y)}\int_{(G+u_i)|a|}^{\infty} w^{1-Y}\mathrm{e}^{-w}\,\mathrm{d}w \right]$$
$$= \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\hat{\Gamma}(K)} (G+u_i)^Y \left[\frac{1}{Y}((G+u_i)|a|)^{-Y} \mathrm{e}^{-(G+u_i)|a|} \right.$$
$$+ \frac{1}{Y(1-Y)}((G+u_i)|a|)^{1-Y}\mathrm{e}^{-(G+u_i)|a|}$$
$$\left. - \frac{\Gamma(2-Y)}{Y(1-Y)}\,\mathrm{gammainc}((G+u_i)|a|, 2-Y, \mathrm{upper}) \right].$$

The function gammainc$(x, a,$ 'upper') is the upper integral of the incomplete gamma function in MATLAB, defined as

$$\mathrm{gammainc}(x, a, \text{'upper'}) = \frac{1}{\Gamma(a)}\int_x^{\infty} u^{a-1}\mathrm{e}^{-u}\,\mathrm{d}u$$

for the computation of

$$\int_{-\infty}^{a} \mathrm{e}^x k(x)\,\mathrm{d}x = \int_{|a|}^{\infty} \frac{\exp(-Gx)}{x^{1+Y}} \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\hat{\Gamma}(K)} \exp(-u_i|x|)\exp(-x)\,\mathrm{d}x$$
$$= \sum_{i=1}^{5} w_i \frac{u_i^{K-1}}{\hat{\Gamma}(K)} \int_{|a|}^{\infty} \frac{\exp(-(G+1+u_i)x)}{x^{1+Y}}\,\mathrm{d}x,$$

and thus we have a computation similar to that for λ above, except that we now replace G by $G+1$. So, we can write the price of the CGKMY crash cliquet as

$$V(a) = \frac{1-\exp(\lambda(G,a,K,Y)t)}{\lambda(G,a,K,Y)}(\exp(-|a|)\lambda(G,a,K,Y) - \lambda(G+1,a,K,Y)).$$

A further extension to the two-parameter thinning function $(1+A|x|)^{-K}$ is easily incorporated upon noting that

$$\frac{1}{(1+Ax)^K} = \frac{1}{\Gamma(K)}\int_0^{\infty} u^{K-1}\mathrm{e}^{-(1+Ax)u}\,\mathrm{d}u$$
$$\approx \sum_{i=1}^{N} w_i u_i^{K-1}\mathrm{e}^{-Au_i x}.$$

TABLE 2 Crash cliquet prices on November 11, 2009.

Strike	Price (bps)
60	9
65	14
70	23
75	31
80	53
85	95
90	164

4 CGKMY AND CGAKMY RESULTS ON S&P 500 NOVEMBER 11, 2009

For the seven strikes of 60% to 90% in steps of 5%, the crash cliquet prices for this date are presented in Table 2.

We first calibrated the CGMY model to seventy option prices, with maturities from one to two years. Figure 2 presents the fit to the option prices. The average percentage error of 2.17% is a better-than-typical performance for such models on such prices (see, for example, Schoutens 2003).

Figure 3 presents the prices for the crash cliquets using the option-calibrated Lévy measure parameters.

As may be observed, the option-calibrated prices are much higher than the market prices. The extrapolation of the Lévy measure to the large tail probabilities is thus not relevant for the pricing of crash cliquets. This is not surprising, as near-the-money options have little chance of incorporating such information. Crash cliquet prices are new pieces of market information, and adjustments in the parameter K may help by thinning out the tails of the Lévy measure.

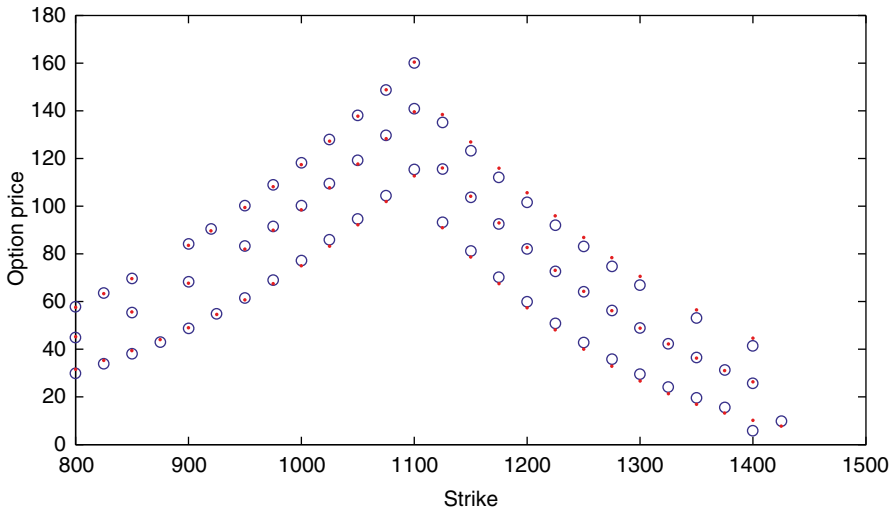
Figure 4 presents a sample of crash prices for different values of K in addition to the prices reported in Figure 3.

We observe that one may lower the crash cliquet prices by raising the value of K.

Finally, we report on the joint calibration of option prices and crash cliquet prices, using CGKMY with a 10% weight on the crash prices, as there were seven crash prices and seventy option prices. Figure 5 presents the fit to the option prices, while Figure 6 presents the crash prices.

We observe that, with just a small weight on the crash prices, the parameter K adjusts to the level 1.97 and fits the crash prices. There is a cost in terms of fitting out-of-the-money option prices, but at-the-money and near-the-money options receive an acceptable fit.

FIGURE 2 Fit of CGMY to seventy S&P 500 options at three maturities between one and two years.



Parameter values and fit statistics are shown on the graph. The market prices are presented as circles, while the model prices are shown as dots. $C = 0.0689$. $G = 0.5139$. $M = 36.36$. $Y = 1.1360$. RMSE $= 2.07$. AAE $= 1.6357$. APE $= 0.0217$.

We also report on the thinning function $(1 + A|x|)^{-K}$, with an extra parameter for the model CGAKMY. This had a comparable fit (presented in Figures 7 and 8).

5 RESULTS ON CALIBRATION TO WEEKLY OPTIONS FOR THE S&P 500, GOOG, GS AND JNJ ON MAY 2, 2014

We obtained data on weekly options for the S&P 500, Alphabet Inc (GOOG), Goldman Sachs (GS) and Johnson & Johnson (JNJ) on market close of May 2, 2014. Figure 9 presents a graph of the option prices for these underliers.

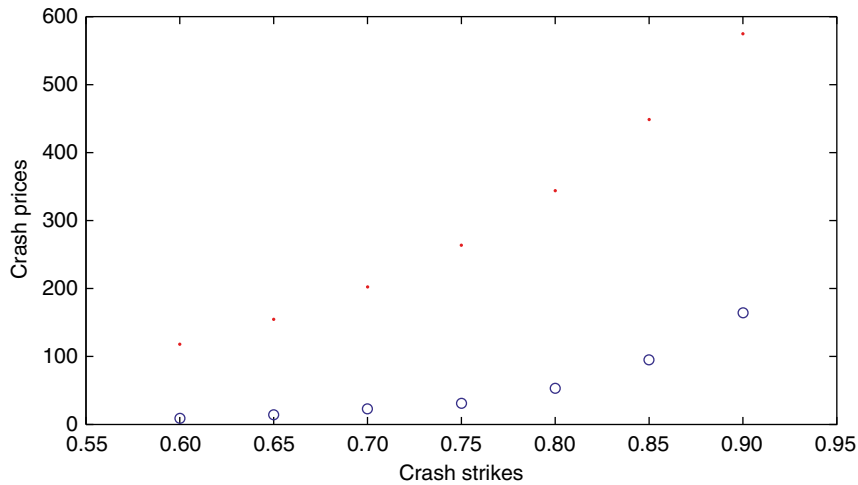
We observe that the S&P 500 prices are below the single-name prices reported, which are also ordered with JNJ the lowest, GOOG the highest and GS in the middle.

First, we estimated the CGMY model on all four underliers and found the fits to be adequate. We report in Tables 3 and 4 the estimated parameter values and the implied prices for crash puts of three-month maturity.

The crash cliquet prices were as given in Table 4.

For the S&P 500, we also obtained market prices for the three-month crash cliquet, and we present the results of calibrating the model CGAKMY. The estimated parameters are reported in Table 5.

FIGURE 3 Crash cliquet prices inferred from option-calibrated parameters.



Market prices are represented by circles, and model prices are represented by squares.

FIGURE 4 CGKMY crash prices using option-calibrated values for the CGMY parameters and a sample of values for K.

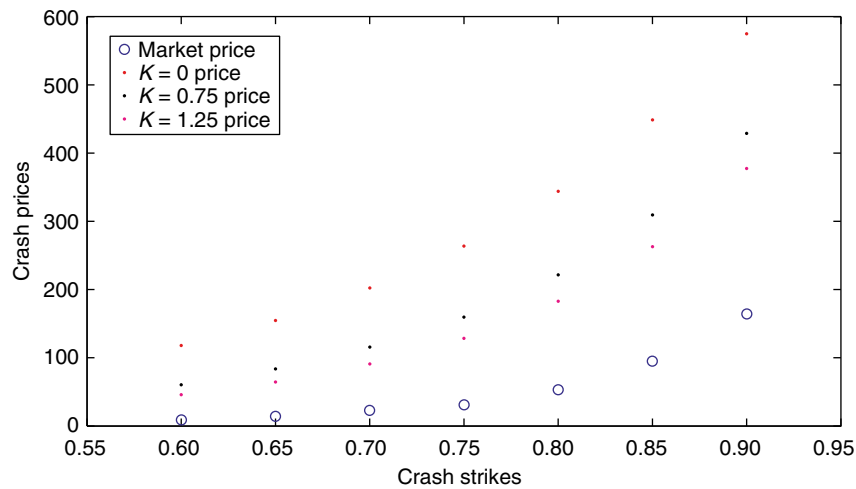
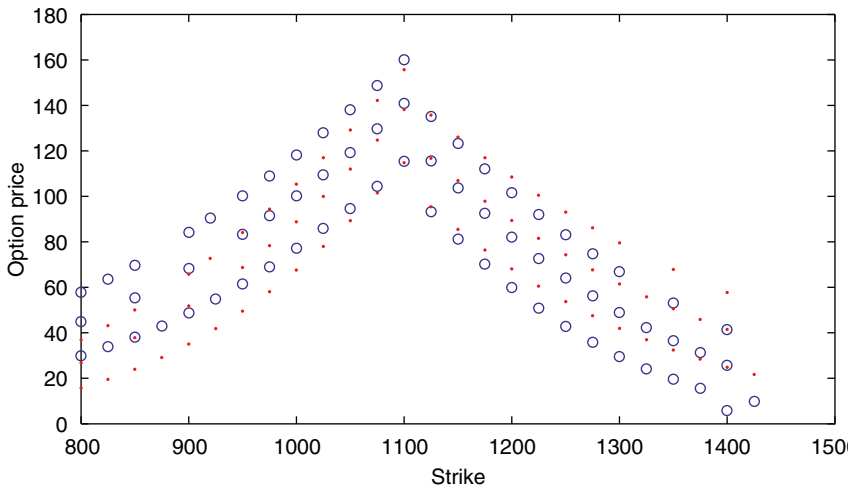


FIGURE 5 Fit of CGKMY to option prices.



Figures 10 and 11 present the fit to the S&P 500 options as well as the three-month crash cliquets for this CGAKMY joint calibration to both sets of prices.

6 THE S&P 500 BETA EXPOSURE APPROACH TO SINGLE-NAME CRASH CLIQUET PRICING

There are markets in crash cliquets on the index, and an interest in attempting to infer from these prices the prices for single-name crash cliquets. We take the view here that some part of the crashes in the prices of single-name stocks are related to a crash in the level of the index. We therefore adopt for this purpose a beta exposure approach.

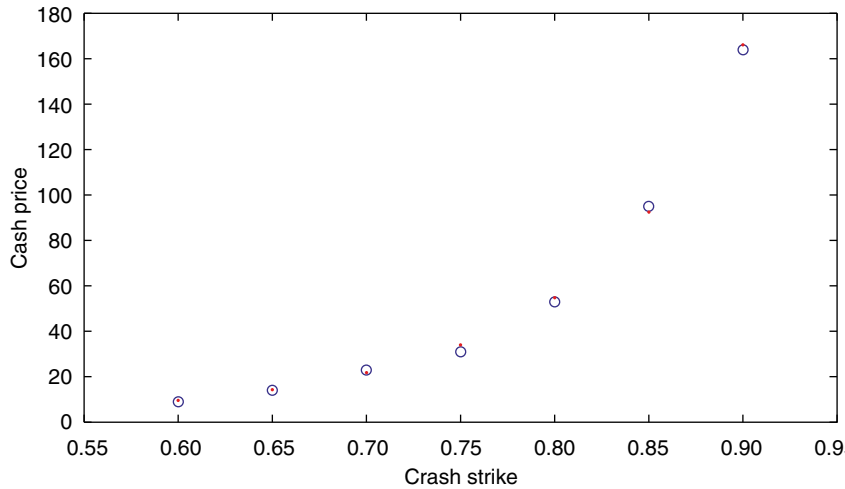
The beta exposure approach to option pricing was introduced in Carr and Madan (2012). Here, it is extended to the pricing of crash cliquets. Let the jump in the single-name log price be s, while x is the jump in the logarithm of the index. Under the beta exposure modeling approach, we write

$$s = \beta x + u,$$

where u is an independent or idiosyncratic jump. However, we further suppose that x, u are never both nonzero at exactly the same instant in time, so we have either

$$s = \beta x$$

FIGURE 6 Fit of CGKMY to crash cliquet prices.



$C = 0.01395$. $G = 5.5476\text{E}{-7}$. $K = 1.9764$. $M = 116.30$. $Y = 1.7619$.

or

$$s = u.$$

The single-name crash cliquet using Lévy models is then given in terms of the Lévy measure of s, say, $v(s)$, as follows. Let $h(u)$ be the Lévy density for idiosyncratic jumps.

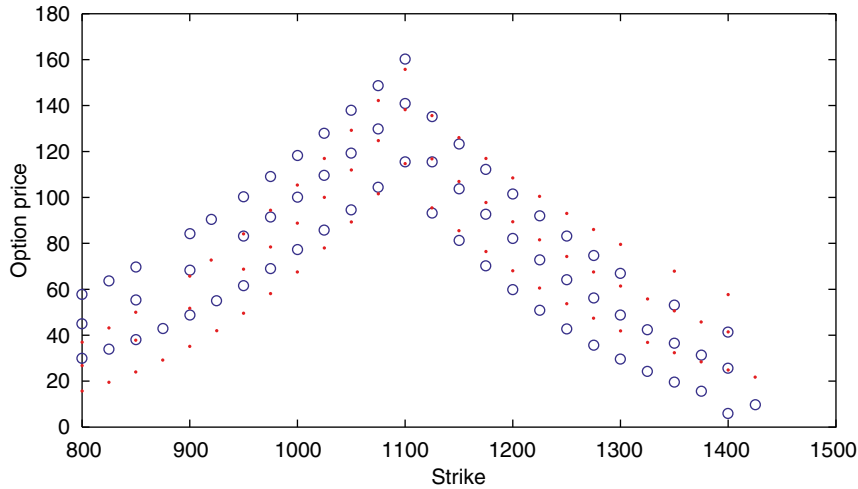
The log characteristic function for s is then

$$\begin{aligned}\int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta s}-1)v(s)\,\mathrm{d}s &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta(\beta x+u)}-1)(k(x)+h(u))\,\mathrm{d}x\,\mathrm{d}u\\ &= \int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta\beta x}-1)k(x)\,\mathrm{d}x+\int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta u}-1)h(u)\,\mathrm{d}u\\ &= \int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta s}-1)\left(\frac{1}{\beta}k\left(\frac{s}{\beta}\right)\right)+\int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta s}-1)h(s)\,\mathrm{d}s\\ &= \int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}\zeta s}-1)\left(\frac{1}{\beta}k\left(\frac{s}{\beta}\right)+h(s)\right)\mathrm{d}s.\end{aligned}$$

Hence,

$$v(s) = \frac{1}{\beta}k\left(\frac{s}{\beta}\right)+h(s),$$

FIGURE 7 CGAKMY fit to S&P 500 option on November 11, 2009.



so we define

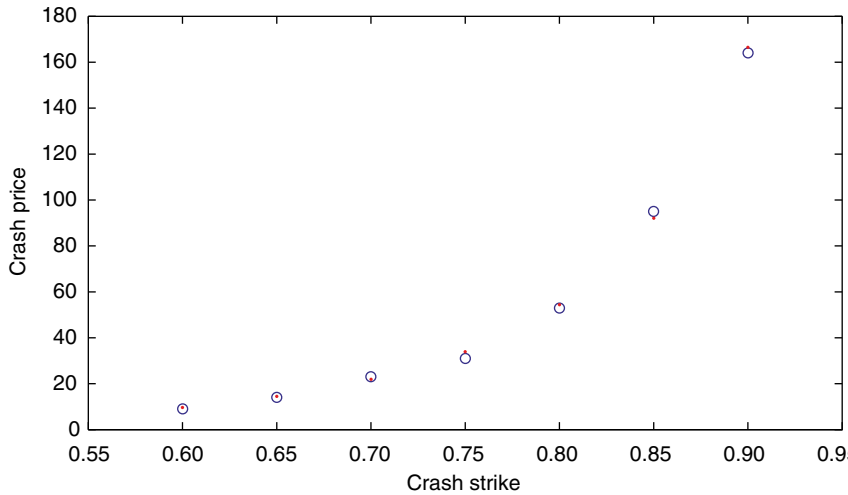
$$\begin{aligned}\lambda &= \int_{-\infty}^{a} \frac{1}{\beta} k\left(\frac{s}{\beta}\right) + h(s)\,\mathrm{d}s \\ &= \int_{-\infty}^{a/\beta} k(x)\,\mathrm{d}x + \int_{-\infty}^{a} h(s)\,\mathrm{d}s \\ &= \lambda_X\left(\frac{a}{\beta}\right) + \lambda_U(a).\end{aligned}$$

The crash cliquet price for the single names is then computed as follows:

$$\begin{aligned}c(a) &= \frac{1-\mathrm{e}^{-\lambda t}}{\lambda} \int_{-\infty}^{a} (\mathrm{e}^{a} - \mathrm{e}^{s})\left(\frac{1}{\beta} k\left(\frac{s}{\beta}\right) + h(s)\right)\mathrm{d}s \\ &= \frac{1-\mathrm{e}^{-\lambda t}}{\lambda}\left(\int_{-\infty}^{a/\beta} (\mathrm{e}^{a} - \mathrm{e}^{\beta x}) k(x)\,\mathrm{d}x + \int_{-\infty}^{a} (\mathrm{e}^{a} - \mathrm{e}^{s}) h(s)\,\mathrm{d}s\right) \\ &= \frac{1-\mathrm{e}^{-\lambda t}}{\lambda}\left(\mathrm{e}^{a}\lambda\left(\frac{a}{\beta}, C, G, A, K, M, Y\right)\right. \\ &\qquad \left. - \lambda\left(\frac{a}{\beta}, C, G+\beta, A, K, M, Y\right) + \frac{c_U(a)\lambda_U(a)}{1-\mathrm{e}^{-\lambda_U(a)t}}\right).\end{aligned}$$

We may estimate β and the CGMY parameters for U from the single-name option surface, and the characteristic function for s as $\beta x + u$.

FIGURE 8 CGAKMY fit on crash cliquet prices for S&P 500 on November 11, 2009.



$C = 0.0171$. $G = 0.02561$. $A = 2.4336$. $K = 1.2750$. $M = 1.1700$. $Y = 1.7353$.

FIGURE 9 Put option prices relative to spot as a function of strike, relative to spot for four underliers (S&P 500, GOOG, GS and JNJ) at market close on May 2, 2014.

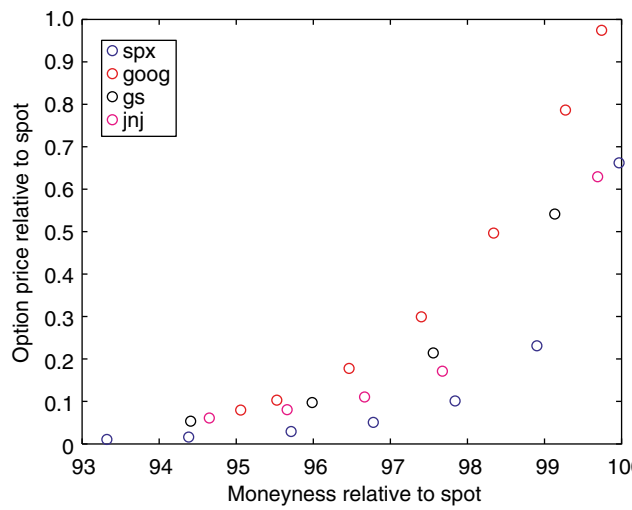


TABLE 3 Calibrated CGMY parameters.

	C	*G*	*M*	*Y*
S&P 500	0.1606	6.5245	9.6797	0.9206
GOOG	7.3128	24.5975	9.2368	3.11e−4
GS	0.0472	8.3442	9.6029	1.4028
JNJ	3.8148	18.7579	8.0830	0.0014

TABLE 4 Implied crash cliquet prices.

Strike	S&P 500	GOOG	GS	JNJ
0.6	0.3127	0.0001	0.0309	0.0017
0.65	0.7232	0.0009	0.0895	0.0098
0.7	1.62	0.007	0.25	0.0497
0.75	3.58	0.05	0.69	0.2304
0.8	7.89	0.32	1.91	0.999
0.85	17.65	1.95	5.47	4.13
0.9	41.15	11.19	17.06	16.62
0.95	103.97	61.35	63.11	65.61

TABLE 5 Estimated parameters for CGAKMY.

C	*G*	*A*	*K*	*M*	*Y*
0.0043	0.0233	0.0595	0.3266	1.2634	1.7136

6.1 Characteristic function for single-name driven by a CGAKMY factor and independent CGMY process

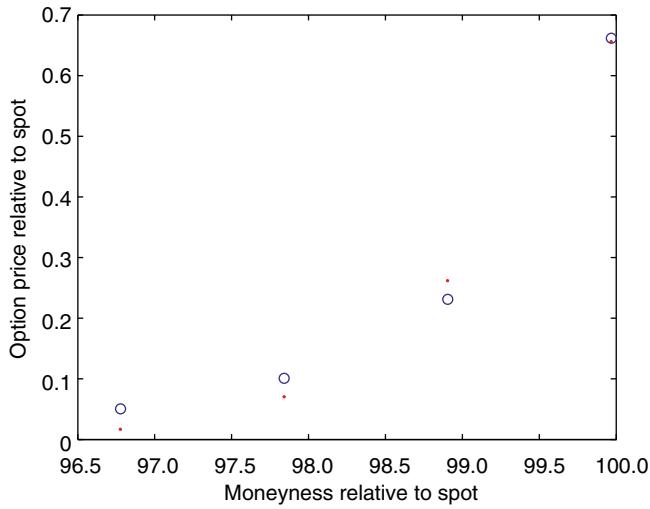
Let $S(t)$ be a pure jump stock price process, with jumps s in the log price $\ln(S(t))$. Further, let $X(t)$, $U(t)$ be pure jump Lévy processes with jumps x, u. The beta exposure model then translates to

$$\ln(S(t)) = \beta X(t) + U(t),$$

where we now take X to be in the CGAKMY, while U is a CGMY process. We shall estimate X first and assume it is known. We then estimate β and the CGMY parameters of U.

The parameters are then C_X, G_X, A_X, K_X, M_X and Y_X, which are known, followed by β, C_U, G_U, M_U and Y_U. We term this model CGMYRCGAKMYF, for CGMY

FIGURE 10 Fit of CGAKMY to S&P 500 weekly options on May 2, 2014.



Market prices are represented by circles, while model prices are shown as squares.

residual and CGAKMY factor. We have

$$\begin{aligned} E[\exp(\mathrm{i}u \ln S(t))] &= E[\exp(\mathrm{i}u\beta X(t) + \mathrm{i}uU(t))] \\ &= [\phi_{\mathrm{CGAKMY}}(u\beta)\phi_{\mathrm{CGMY}}(u)]^t. \end{aligned}$$

We may also observe then that

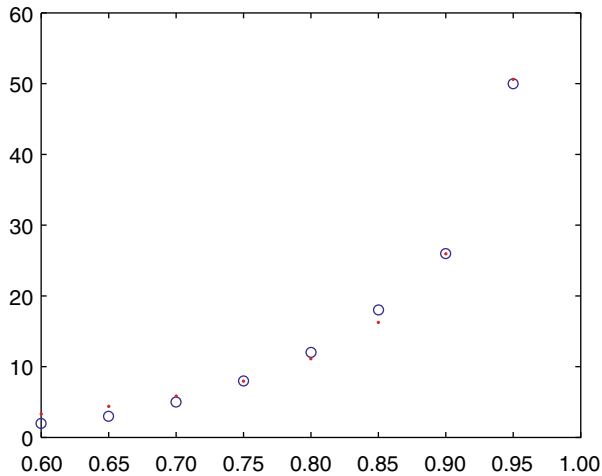
$$\begin{aligned} &\phi_{\mathrm{CGMYRCGAKMYF}}(u) \\ &\quad = \exp\left(\int_{-\infty}^{\infty}(\mathrm{e}^{\mathrm{i}us} - 1)\left(\frac{1}{\beta}k_{\mathrm{CGAKMY}}\left(\frac{s}{\beta}\right) + k_{\mathrm{CGMY}}(s)\right)\mathrm{d}s\right). \end{aligned}$$

With the usual convexity correction, we have

$$\begin{aligned} \phi_{\ln S}(u) = \exp(\mathrm{i}u[\log(S(0)) + rt - t\log(\phi_{\mathrm{CGAKMYRCGMYF}}(-1\mathrm{i}))]) \\ \times [\phi_{\mathrm{CGAKMY}}(u\beta)\phi_{\mathrm{CGMY}}(u)]^t. \end{aligned}$$

We estimate β and the CGMY parameters for the residual on data for options on the single name. In this exercise, the characteristic function for the index is fixed, as it was estimated from joint data on index options and index crash cliquets. We are then ready to price single-name crash cliquets.

FIGURE 11 Fit of CGAKMY to three-month crash put prices on S&P 500 for May 2, 2014.



7 RESULTS FOR SINGLE NAMES

We present results for GOOG, GS and JNJ, as calibrated at market close for May 2, 2014, in three short subsections. In each section, we first report the CGMY and β parameters, as calibrated to the single-name option prices. Next, we report the crash cliquet prices in basis points and the proportion contributed by the idiosyncratic component π_R bps. We observe that the contribution of idiosyncratic component rises as the strike rises, but it can fall when we get closer to at-the-money.

7.1 GOOG

The beta exposure parameters are

$$\begin{aligned} \beta &= 1.4439, \\ C_U &= 0.0156, \\ G_U &= 7.0493, \\ M_U &= 4.29, \\ Y_U &= 0.0907. \end{aligned}$$

The crash cliquet results are given in Table 6.

TABLE 6 The crash cliquet results for beta.

	Strike							
	0.60	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
Price	6	9	11	15	22	32	50	94
π_R	24	40	62	93	132	176	218	226

TABLE 7 Crash cliquet results for GS.

	Strike							
	0.60	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
Price	11	14	19	26	36	52	81	146
π_R	60	77	96	115	133	145	147	123

7.2 GS

The beta exposure parameters are

$$\begin{aligned} \beta &= 1.9324, \\ C_U &= 0.0068, \\ G_U &= 4.1733, \\ M_U &= 4.6729, \\ Y_U &= 0.0721. \end{aligned}$$

The crash cliquet results are given in Table 7.

7.3 JNJ

The beta exposure parameters are

$$\begin{aligned} \beta &= 1.878, \\ C_U &= 0.0135, \\ G_U &= 3.7942, \\ M_U &= 4.6672, \\ Y_U &= 0.1256. \end{aligned}$$

The crash cliquet results are given in Table 8.

TABLE 8 The crash cliquet results for JNJ.

	Strike							
	0.60	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
Price	10	14	18	25	35	50	79	142
π_R	174	218	263	307	346	371	368	305

8 CONCLUSION

It is observed that widely used option-calibrated exponential Lévy models do overprice crash cliquets. This was observed for numerous models, with results presented here for just the CGMY model. It was interpreted that extrapolations of typical model Lévy tails are, then, not crash cliquet market consistent. A general strategy of tail thinning was developed and proposed for a class of models with parametric Lévy densities closed under exponential tilting. Many models in the literature have this property, as they essentially capture skew by exponential tilting. The models CGKMY and CGAKMY were introduced by employing the thinning functions of $(1 + |x|)^{-K}$ and $(1 + A|x|)^{-K}$. Adjustments in the parameters A, K helped the development of adjusted Lévy processes capable of being consistent with the prices of crash cliquets and near-the-money option prices. Single-name crash cliquet prices were then developed via a beta exposure to the index, modeled as a CGAKMY process plus an independent CGMY component. Results were illustrated for weekly option data at market close on May 2, 2014 on a sample of underliers. Additionally, results on the STX were also presented for data on November 11, 2009.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

REFERENCES

Bollerslev, T., and Todorov, V. (2011). Tails, fears and risk premia. *Journal of Finance* **66**, 2165–2211 (http://doi.org/b5p3vr).

Carr, P., and Madan, D. B. (1999). Option valuation using the fast Fourier transform. *The Journal of Computational Finance* **2**(4), 61–73 (http://doi.org/bkrb).

Carr, P., and Madan, D. B. (2012). Factor models for option pricing. *Asia Pacific Financial Markets* **19**, 319–329 (http://doi.org/dx9s7g).

Carr, P., Geman, H., Madan, D., and Yor, M. (2002). The fine structure of asset returns: an empirical investigation. *Journal of Business* **75**(2), 305–332 (http://doi.org/fqvqsv).

Kozubowski, T. J., and Podgórski, K. (2007). Invariance properties of the negative binomial Lévy process and stochastic self-similarity. *International Mathematical Forum* **2**, 1457–1468.

Schoutens, W. (2003). *Lévy Processes in Finance: Pricing Financial Derivatives*. Wiley (http://doi.org/bkj7xq).

Tankov, P. (2010). Pricing and hedging gap risk. *The Journal of Computational Finance* **13**(3), 33–59 (http://doi.org/bkrc).

Journal of Computational Finance **20**(1), 113–137
DOI: 10.21314/JCF.2016.311

Risk journals

Research Paper

An exact and efficient method for computing cross-Gammas of Bermudan swaptions and cancelable swaps under the Libor market model

Mark S. Joshi and Dan Zhu

Centre for Actuarial Studies, Department of Economics, University of Melbourne, VIC 3010, Australia; emails: mark@markjoshi.com, d.zhu2@student.unimelb.edu.au

(Received December 22, 2014; revised June 10, 2015; accepted June 15, 2015)

ABSTRACT

We introduce a new simulation algorithm for computing the Hessians of Bermudan swaptions and cancelable swaps. The resulting pathwise estimates are unbiased and accurate. Given the exercise strategy, the pathwise angularities are removed by a sequence of measure changes. The change of measure at each exercise time is chosen to be optimal in terms of minimizing the variance of the likelihood ratio terms. Numerical results for the Hessian of cancelable swaps are presented to demonstrate the speed and efficacy of the method.

Keywords: Monte Carlo simulation; Bermudan products; exercise strategy; Hessian; measure changes.

1 INTRODUCTION

Bermudan swaptions and cancelable swaps are among the most liquidly traded exotic interest rate derivative contracts. Consequently, their pricing and risk management are

Corresponding author: M. S. Joshi

Print ISSN 1460-1559 | Online ISSN 1755-2850

of high practical importance. Such derivatives are typically Delta-hedged: the Hessian (or Gamma-matrix) of the price measures the rates of change in the Deltas with respect to changes in the underlying forward rates. Thus, they explain the profit-and-loss in a Delta-hedging strategy. The objective of this paper is to develop an efficient and accurate algorithm to compute the Hessians of Bermudan swaptions and cancelable swaps that can be used in practice for hedging purposes.

The pricing of Bermudan swaptions and cancelable swaps is generally implemented under the Libor market model (LMM) framework (Brace *et al* 1997). It is widely used by practitioners and the academic community for valuing exotic interest rate derivatives. The model has been studied extensively in the past: for more details, we refer the reader to Brace (2007), Fries (2007) and Joshi (2011). Monte Carlo simulation is the most common method of its implementation in practice. Until recently, pricing early exercisable interest rate derivatives using Monte Carlo simulation was regarded as very hard. Such difficulties arise because the exercise decisions embedded in Bermudan-type products require comparisons between the value received upon exercise and the continuation value. The continuation value is the expected value of the unexercised product, which is not easily obtainable in a simulation.

A number of breakthroughs have been made to address this problem. Most of these methods focus on producing a lower-bound algorithm, which produces a lower-biased estimate of the price. We shall focus on lower bounds in this paper. The regression approach (typically least squares), first introduced by Carriere (1996) and then Longstaff and Schwartz (2001), is widely used to approximate the continuation values in calculating the lower bound. The approach is usually adequate to make the correct exercise decisions when options are deeply in- or out-of-the-money, but it is less effective when the decision is not so clear. Based on this observation, Beveridge *et al* (2013) used two regressions at each exercise time. Joshi (2014) recently adopted this idea but took it further. The author suggested using multiple regressions, typically about five, discarding a fixed fraction of paths furthest from the boundary each time and regressing the remaining ones. The major advantage of using multiple regressions over a double regression is that, by only discarding a small fraction of paths, we are unlikely to be affected by substantiable misestimation of the continuation value close to the boundary. In these regression-based techniques, a second pass is typically used to compute a lower-biased estimate of the price using the exercise strategy developed in the first pass; a better exercise strategy via the multiple regression method provides a tighter lower bound of the price, ie, the downward bias is smaller in the second pass. We shall use the multiple regression approach in this paper to estimate the price.

Along with pricing, another important task in practice for quantitative analysts is to compute accurate and fast Greeks. The three main Monte Carlo methods for computing Greeks are the finite-differencing method, the pathwise (PW) method and the

likelihood ratio (LR) method. The pathwise method, when applicable, produces unbiased estimates and the smallest standard errors among the three (Glasserman 2004). However, it is only applicable to computing Hessians when the first-order derivatives of the payoff function are Lipschitz continuous everywhere and differentiable almost surely. (We shall call a function with such properties $\hat{C}^2$.)

Piterbarg (2004a) showed that, given the optimal exercise strategy that maximizes the value of the option over all exercise times, one can apply the pathwise method to produce unbiased estimates of Deltas while fixing the optimal exercise time. The author further suggested that the output of the first pass regressions can be used to replace the optimal one in the pathwise Delta estimates. However, Korn and Liang (2015) emphasized that, due to the lack of Lipschitz-continuous first-order derivatives, the basis for applying the pathwise method to calculating Gammas of Bermudan-type products is not justified. Previously, they were only estimated with some smoothing techniques. For example, Joshi and Yang (2011) used a localized smoothing of the Monte Carlo algorithm at each exercise time, and the estimated Gammas for cancelable swaps were consequently biased.

Computing Greeks of financial products with discontinuous and angular payoffs has been studied extensively. To compute the Hessian of products with angular payoffs, one can compute the Deltas by the pathwise method and then apply the likelihood ratio method to produce unbiased estimates of the Hessian. This approach suffers from the problem of producing estimates with large standard errors, and it is not applicable if the reduced-factor LMM is used. To compute Deltas for products with discontinuous payoffs, Chan and Joshi (2015) introduced a continuously differentiable change of measure function to ensure that the payoff function under the new algorithm is Lipschitz continuous everywhere and differentiable almost surely; the pathwise method is then applicable to the new algorithm for computing Deltas. Joshi and Zhu (2016) extended this idea: they used a twice-differentiable change of measure function to produce a $\hat{C}^2$ algorithm for computing the price. For products with angular payoffs, they recognized that a differentiable change of measure function is sufficient to remove the pathwise discontinuities of the first-order derivatives. Their method for computing the Hessian of the price for angular products is called the first-order optimal partial proxy method of calculating Hessians (HOPP(1)). The measure changes if these methods are optimal in terms of minimizing the variance of the likelihood ratio weight.

In this paper, we modify the HOPP(1) method to produce an unbiased estimate of the Hessian for Bermudan-type products. The difficulties of applying the HOPP(1) method here are discussed in Section 3.1. To address the problems, at each exercise time we replace one standard uniform for simulating the state variables with a new change of measure function, which is a weighted average of the original standard uniform and the HOPP(1) change of measure function. The weight depends on the

location of the unbumped path. In particular, the weight of the HOPP(1) change of measure is zero if the unbumped path is far from the boundary. The pathwise method is then applicable to the new $\hat{C}^2$ pricing algorithm to produce unbiased estimates of the Hessian: we shall call it the Hessian by optimal measure changes (HOMC). The virtue of our method for computing sensitivities is its wide applicability, since it only requires the smoothness of cashflow functions and monotonic approximated continuation values.

Besides the lack of Lipschitz-continuous first-order derivatives, another difficulty encountered in computing the Hessian of Libor exotics arises from the large number of Greeks and the complexity of the underlying model. To reduce the computational complexity of these cases, Giles and Glasserman (2006) applied the adjoint and automatic differentiation method to calculating the price sensitivities of financial products. Joshi and Yang (2011) introduced a methodology for computing the Hessian of smooth functions by decomposing the algorithm into elementary operations, and then calculating the pathwise estimator of the Hessian using a backward method. Since the HOMC algorithm for computing the price is $\hat{C}^2$, we shall use the algorithmic Hessian method to save computational effort.

The paper is organized as follows. In Section 2, we review the mathematical background of the LMM, give product descriptions of some Libor exotics and offer a brief discussion of the multiple regression method. In Section 3, we present the basic idea of the HOMC algorithm. Numerical results on cancelable swaps are presented in Section 4.

2 THE LIBOR MARKET MODEL AND THE MULTIPLE REGRESSION ALGORITHM

2.1 The Libor market model under the spot measure

In this section, we briefly summarize the Libor market model (Brace *et al* 1997). Take the set of tenor dates

$$0 < T_0 < T_1 < T_2 \cdots < T_{n-1} < T_n,$$

where the time difference between two reset dates T_{i+1} and T_i is τ_i. Let $B_i(t)$ denote the value of zero-coupon bonds maturing at T_i, observed at time t. We define the Libor spanning over the period $[T_i, T_{i+1})$ at time $t < T_i$ as

$$f_i(t) = \frac{B_i(t) - B_{i+1}(t)}{\tau_i B_{i+1}(t)},$$

with dynamic

$$\mathrm{d}f_i(t) = \mu_i(f,t) f_i(t)\,\mathrm{d}t + \sigma_i(t) f_i(t)\,\mathrm{d}W_t^i, \tag{2.1}$$

where W_t is the F-dimensional standard Brownian motion. For $t > T_i$, we have $f_i(t) = f_i(T_i)$. Typically, the instantaneous volatility curve $\sigma_i(t)$ is chosen to be time homogeneous, and the correlations between the rates are assumed to be constant.

Define a function $\eta\colon [0, T_n) \to 0, 1, 2, \ldots, n-1$ to be the index of the next Libor reset date at time t. The modeling dynamics of f_i under the spot measure with the log-Euler scheme is

$$f_i(T_{j+1}) = f_i(T_j) \exp\left(\mu_i(T_j) + \sum_{k=1}^{F}(a_{ik}Z_k - 0.5a_{ik}^2)\right) \tag{2.2}$$

for $T_{j+1} \leqslant T_i$. Here, a_{ik} is the ikth element of the pseudo-square root of the covariance matrix. The first principal component of stock correlation is typically flat when computed from historical data. Lord and Pelsser (2007) showed that, in certain cases, this assumption implies that all elements of the first component are positive. We compute the pseudo-square root of the covariance matrix via spectral decomposition to reduce the number of factors from n to F for $F \leqslant n$. The first column of the result is then guaranteed to be positive, under the assumptions of Lord and Pelsser (2007).

The drift is computed using the method introduced by Joshi (2003). Under the spot measure, the drift is

$$\mu_i(T_j) = \sum_{k=1}^{F} a_{ik}e_{ik}, \tag{2.3}$$

with

$$e_{ik} = \sum_{s=\eta(T_j)}^{i} \frac{\tau_s f_s}{1 + \tau_s f_s} a_{sk}. \tag{2.4}$$

The computational order of implementing the LMM is $\mathcal{O}(nF)$ per step by this method. The drift in the model is state-dependent.

This model has been extended by various authors. One example of these extensions is the displaced-diffusion model (Joshi and Rebonato 2003). The displaced-diffusion coefficients of forward rates allow for skewness in the caplet volatility surface, a persistent market feature. This model collapses to the original lognormal LMM by setting displaced-diffusion coefficients to zero. Mercurio (2010) introduced another extension of the LMM that is compatible with the current market practice of building different yield curves for different tenors and for discounting. It is based on modeling the joint evolution of forward rate agreement (FRA) rates and forward rates belonging to the discount curve. Our numerical examples for computing the Hessians of Bermudan swaptions and cancelable swaps are performed under the original LMM;

however, we believe there are no particular barriers to implementing them under the displaced-diffusion LMM or the multi-curve LMM.

2.2 Libor product descriptions

A payer swap is a swap where the holder pays the fixed rates, K, and receives the floating rates. The payoff of a payer swap is

$$\sum_{i=0}^{n-1} \frac{(f_i(T_i) - K)\tau_i}{1 + \tau_i f_i(T_i)} P_i(T_i).$$

One can compute the price as well as the Greeks of a swap analytically. Payer swaps increase in value as forward rates rise, and vice versa. The Gammas of a payer swap are negative due to the interaction of the swap rate and the discounting bonds.

A receiver Bermudan swaption is where the holder of the option has the right but not the obligation to enter into a receiver swap at a set of predetermined reset times. The swap always terminates at time T_n, so the length of the swap decreases with time. Thus, we have the payoff of a receiver Bermudan swaption,

$$(K - \mathrm{SR}_j)_+ \sum_{i=j}^{n-1} \tau_i P_{i+1},$$

where SR_j denotes the swap rate starting at T_j and ending at T_n.

A cancelable swap is where one of the counterparties has the right but not the obligation to terminate the swap on one or more predetermined dates during the life of the swap. A payer-cancelable swap can be viewed as a combination of a vanilla swap and a receiver Bermudan swaption. We concentrate on cancelable swaps throughout this paper. The exercise value of a cancelable swap at exercise time T_j for $j = 0, 1, \ldots, n-1$ is zero, and we have the accumulated cashflows from T_0 to T_j. Once we compute the Hessian of the price for a cancelable swap, it is trivial to obtain the Hessian for its Bermudan swaption counterpart. This is because we can compute the Hessian of the price of vanilla swaps analytically.

2.3 The multiple regression algorithm

Pricing products with early exercisable features via Monte Carlo simulation is one of the hard problems in financial engineering. This is because exercise decisions require knowledge of continuation values, which are not easily obtained in simulations. Regression-based techniques are popular methods of obtaining approximations of the continuation values from simulated paths (we refer the reader to Joshi (2011) for detailed explanations of these methods). Here, we present a brief sketch of the least

squares algorithm. Before we start our description, consider the following notation. For $i = T_0, T_1, T_2, \ldots, T_{n-1}$, let

- F_i denote the vector of the current basis functions and current state variables,
- PC_i denote the pathwise discounted continuation value,
- $C_i(F_i)$ denote the approximated continuation value as a function of the current basis,
- $E_i(F_i)$ denote the current exercise value, which depends on the current state variables.

The first pass of the algorithm is based on a backward induction procedure starting at T_{n-1}; this is repeated and moved backward until T_0. At each exercise time (T_i), the continuation value (C_i) is a regression of PC_i on F_i. Typically, it is approximated by a linear combination of current basis functions. Least squares regression is then used to estimate the coefficients. The approximated continuation value from the least squares regression is compared to E_i; the value of the product at each exercise time is set to E_i if it is greater, otherwise it is set to the pathwise discounted continuation value, PC_i. We then move backward to step T_{i-1}. We discount the value obtained from T_i and add any additional cashflows between the steps to obtain PC_{i-1}. Then, we regress again and repeat the same procedure until the first step.

A second independent pass is then generated. At each exercise time (T_i), we compare the current exercise value and the estimated continuation value with the coefficients determined in the first pass. If $E_i > C_i$, we exercise the product and compute the corresponding discounted cashflows; otherwise, we move forward to the next exercise time. The result gives an unbiased estimate of the lower bound, given the exercise strategy determined in the first pass.

The least squares regression method is widely used for calculating lower bounds. However, the accuracy of the output is heavily dependent on the choice of basis in the regression (see Beveridge *et al* 2013). What to use for basis functions is generally not obvious, and a significant amount of time is required to investigate the appropriate set of basis functions for a particular product. For Bermudan swaptions and cancelable swaps, second-order polynomials in forward rates and swap rates are popular choices, following Piterbarg (2004b).

One enhancement of the least squares method is the multiple regression method introduced by Joshi (2014), which is an efficient algorithm with less dependence on the choice of basis functions. It is based on the observation that the least squares method is usually adequate to make the correct exercise decisions when options are deeply in- or out-of-the-money, ie, when $|C_i - E_i|$ is large. Joshi (2014) used multiple regressions to obtain a better fit to continuation values when the option is close to the

boundary. The approximated continuation value is dependent on the distance of the path from the boundary.

Let $C_{i,j}(F_i)$ denote the approximated continuation value from the jth regression at T_i as a function of the current basis. For a fixed fraction $\alpha \in (0,1)$ and regression depth d, the regression step in the first pass at each time T_i in the least squares algorithm is instead performed as follows.

- Calculate the pathwise discounted continuation value PC_i.
- Perform an initial regression using the least squares method with these PC_i, and produce the first estimate of the expected discounted continuation value, $C_{i,1}(F_i)$.
- Use these estimates $C_{i,1}(F_i)$ to determine the critical value $L_{i,1}$, such that a fraction $1-\alpha$ of the paths have $|C_{i,1}(F_i) - E_i(F_i)|$ greater than it.
- Discard the paths that have $|C_{i,1}(F_i) - E_i(F_i)|$ greater than $L_{i,1}$, and perform a second regression with the remaining PC_i to obtain another set of coefficients and a critical value $L_{i,2}$.
- Perform similar exercises for other regressions until the dth regression.

The approximated continuation value from the sequence of regressions at each time T_i is

$$C_i(F_i) = \sum_{j=1}^{d-1} \left(C_{i,j}(F_i) \mathbb{I}_{|C_{i,j}(F_i) - E_i(F_i)| > L_{i,j}} \prod_{k=1}^{j-1} \mathbb{I}_{|C_{i,k}(F_i) - E_i(F_i)| < L_{i,k}} \right) + C_{i,d}(F_i) \prod_{k=1}^{d-1} \mathbb{I}_{|C_{i,k}(F_i) - E_i(F_i)| < L_{i,k}}. \tag{2.5}$$

Given the parameter input $\theta \in \mathbb{R}^m$ and a sequence of standard uniform random variates $V \in \mathbb{R}^{n \times F}$, the pathwise estimate of the price for cancelable products can now be expressed as a function $P : \mathbb{R}^m \times \mathbb{R}^{n \times F} \to \mathbb{R}$, such that

$$P(\theta, V) = N(0) \sum_{i=0}^{n-1} \frac{\mathrm{CF}(T_i, F_i(\theta, V))}{N(T_i, F_i(\theta, V))} \prod_{j=0}^{i} \mathbb{I}_{C_i(F_i(\theta,V)) > E_i(F_i(\theta,V))}, \tag{2.6}$$

where N is the numeraire and CF are the cashflows of the product. The exercise value E_i of a cancelable swap is zero.

3 THE HESSIAN BY OPTIMAL MEASURE CHANGES (HOMC) ALGORITHM

In this section, we present the HOMC algorithm. The first pass of the algorithm is exactly the same as explained in the previous section, which determines the exercise strategy. In the second pass, we perform measure changes to ensure the pathwise estimate of the price has Lipschitz-continuous first-order derivatives. We only need to ensure that small bumps in the parameters of interest do not lead to the bumped path finishing on a different side of the angularity, since the pathwise estimate of the price is smooth away from the pathwise angularities. The change of measure performed on each step is a modification of the HOPP(1) method (Joshi and Zhu 2016); measure changes are only performed if the unbumped path is within the innermost region, and a smooth function is introduced to provide the transition of the change of measure function between the innermost and the outer regions.

The new algorithm for computing the weighted pathwise estimate of the price $\hat{P}$ is a product of the pathwise estimate of the price and a sequence of likelihood ratio weights with the following properties:

- $\hat{P}(\theta_0) = P(\theta_0, V)$;
- $\mathbb{E}[\hat{P}(\theta)] = \mathbb{E}[P(\theta, V)]$;
- $\hat{P}(\theta)$ is $\hat{C}^2$;

so, we can apply the pathwise method to compute the unbiased estimate of the Hessian. In this section, we briefly recap the basic idea of the HOPP(1) algorithm and discuss the difficulties of applying the method to the problems we consider in this paper. We then present our solution, the HOMC algorithm, which is a modification of the HOPP(1) algorithm, to address the problem of computing the Hessian for Bermudan-type products.

3.1 The HOPP(1) algorithm

In this section, we briefly summarize the HOPP(1) algorithm. The application of the algorithm requires knowledge of the point of angularity, which can be described the set where some proxy constraint function equals a certain level. In our case of Bermudan-type products, the payoff angularity is where the continuation value equals the exercise value at each exercise time. Thus, the natural choice for the proxy constraint function is the continuation value minus the exercise value. The HOPP(1) method is applicable for computing second-order derivatives as long as the proxy constraint function is monotonic in the first standard uniform for simulating the forward rates across the step.

In particular, to simulate the forward rates across the step $[T_{i-1}, T_i]$ according to (2.2) and using the inverse transform method, one needs to generate a vector of random uniforms v_i. Assume that we observe the standard uniforms of each step $v_{i,j}$ for $j > 1$ before the first standard uniform $v_{i,1}$. The evolution of state variables across each step is divided into two phases. In the first phase, we perform the ordinary evolution of the state variables, except for the first random uniform. That is, for each active forward rate, we compute

$$f_j^*(T_i) = f_j(T_{i-1}) \exp\left(\mu_i(T_{i-1}) - 0.5a_{j1}^2 + \sum_{k=2}^{F}(a_{jk}Z_k - 0.5a_{jk}^2)\right).$$

Let F_i^* denote the vector of current state variables after the first phase; for instance, it can be the vector of $f_j^*(T_i)$. In the second phase, we perform the change of variable on $v_{i,1}$, ie, the standard random uniform for generating Z_1, to remove pathwise discontinuities of the first-order derivatives of the pathwise estimate of price with respect to θ. The only two requirements for this to apply are

- that the underlying pathwise estimate of the price, $P(\theta, V)$, is smooth away from the angularities,
- that, given F_i^*, we can either compute or approximate (via numerical techniques such as the Newton–Raphson method) the differentiable critical value function of the first standard uniform, $a_i(F_i^*)$, for the function $P(\theta, V)$ to cross the angularity.

The pathwise price function in (2.6) clearly satisfies the first requirement. However, depending on the choice of the approximated continuation functions, it may not satisfy the second one.

To clarify the exact method of computing the critical value function, we take the payoff of a caplet as an example. It has an angularity at $f_0 = K$. The critical value function is

$$a_0(F_0^*) = \Phi\left(\frac{\log(K) - \log(f_0^*)}{a_{01}}\right),$$

where Φ is the cumulative density function of the standard normal distribution, and the current state variable F_0^* is only the forward rate after the first phase. In this case,

$$v_{0,1} \geqslant a_0(F_0^*) \Leftrightarrow f_0(f_0^*, v_{0,1}) \geqslant K,$$

and the critical value function is determined explicitly. For the payoff of a cancelable swap, this has an angularity at the point where the approximated continuation value (C_t) at time t equals zero. It is not possible to find the explicit solution of $a_t(F_t^*)$ such that

$$v_{t,1} \geqslant a_t(F_t^*) \Leftrightarrow C_t(F_t^*, v_{t,1}) \geqslant 0$$

if the approximated continuation value is a second-order polynomial in forward rates and swap rates. For these situations, the Newton–Raphson method is required to numerically approximate these critical value functions. Joshi and Zhu (2016) used one iteration of the Newton–Raphson method for financial products with angular payoffs, and two iterations for products with discontinuous payoffs. Since the critical value functions are approximations, there is still a small possibility that the bumped and unbumped paths will finish on different sides of angularities or discontinuities. Joshi and Zhu (2016) suggested removing this remaining discontinuity by using the unbumped path event for the bumped path. This creates a bias in the bumped path's expectation. They showed that the bias created vanishes at the limit, ie, as $\|\theta-\theta_0\| \to 0$, for computing first- and second-order derivatives.

Given the critical value functions, the basic idea of HOPP(1) is to perform measure changes in order to eliminate the limits' dependence on the parameter of interest, so pathwise discontinuities of the first-order derivatives of the price are removed. Although the evolution of state variables may be multidimensional, the change of variable is performed only on one standard uniform per step. Since the first column of the pseudo-square root of the covariance matrix is positive, the approximated continuation value is guaranteed to be monotonic in the first standard uniform at every step. The change of variable is performed by replacing the first standard uniform with a function $U_i^{\mathrm{HOPP}(1)}(\theta, v_{i,1})$ with the following properties:

- $U_i^{\mathrm{HOPP}(1)}(\theta_0, v_{i,1}) = v_{i,1}$;
- it is bijective on $[0, 1]$ for fixed $F_i^*(\theta)$;
- $U_i^{\mathrm{HOPP}(1)}(\theta, a_i(F_i^*(\theta_0))) = a_i(F_i^*(\theta))$;
- it is differentiable as a function of θ and piecewise differentiable as a function of $v_{i,1}$.

There are many functions that satisfy the conditions listed above. Joshi and Zhu (2016) chose the appropriate change of measure functions based on the observation that pathwise methods, when applicable, generally lead to lower variances (Glasserman 2004). Using a likelihood ratio method discards all benefits from the regularity properties of the cashflows and uses only the density's smoothness. Joshi and Zhu (2016) made use of the integrand's regularity properties away from the angularities by putting the maximal weight that is possible without differentiating the angularities on the pathwise method.

For computing the Hessians of the price for Bermudan swaptions and cancelable swaps given F_i^* and $a_i(F_i^*(\theta))$, the critical value of the first standard uniform such

that $C_i(F_i(\theta, a_i)) = E_i(F_i(\theta, a_i))$, the HOPP(1) change of measure function is

$$U_i^{\mathrm{HOPP}(1)}(\theta, v_{i,1}) = \begin{cases} \dfrac{1 - a_i(F_i^*(\theta))}{1 - a_i(F_i^*(\theta_0))}(v_{i,1} - a_i(F_i^*(\theta_0))) + a_i(F_i^*(\theta)) & \text{if } v_{i,1} > a_i(F_i^*(\theta_0)), \\ \dfrac{a_i(F_i^*(\theta))}{a_i(F_i^*(\theta_0))} v_i & \text{otherwise.} \end{cases} \tag{3.1}$$

Chan and Joshi (2015) showed that this change of measure is optimal in terms of minimizing the following expression:

$$\int_0^1 \left(\frac{\partial U_i^{\mathrm{HOPP}(1)}(\theta, v)}{\partial v} \right)^2 \mathrm{d}v.$$

The resulting likelihood ratio weight is

$$\frac{\partial U_i^{\mathrm{HOPP}(1)}(\theta, v_{i,1})}{\partial v_{i,1}} = \begin{cases} \dfrac{1 - a_i(F_i^*(\theta))}{1 - a_i(F_i^*(\theta_0))} & \text{if } v_{i,1} > a_i(F_i^*(\theta_0)), \\ \dfrac{a_i(F_i^*(\theta))}{a_i(F_i^*(\theta_0))} & \text{otherwise.} \end{cases} \tag{3.2}$$

As a result of the above measure change,

- the resulting discounted payoff function is twice-differentiable almost everywhere and has Lipschitz-continuous first-order derivatives,
- this change of variable is also optimal in terms of minimizing the variance of the likelihood ratio part of the second-order derivative estimates.

Joshi and Zhu (2016) showed the efficacy of HOPP(1) for computing sensitivities of various exotic derivative products. However, it is difficult to apply the HOPP(1) algorithm to the cases we consider in this paper. In particular, when the least squares method is used to estimate the price, we face three main issues. First, the bias in the estimated lower bound is large when the basis for approximating continuation values is inadequate. Second, many authors use second-order polynomials in forward rates and swap rates as basis functions, following Piterbarg (2004b). The numerical search for the critical value function in these cases requires additional computational effort. Third, the second-order polynomials may not be monotonic in the first random uniform; thus, the HOPP(1) may not even be applicable. To address these problems, we modify the HOPP(1) method to compute the sensitivities of the prices for Bermudan-type products.

3.2 The HOMC algorithm

The first step of our solution is to use the multiple regression method (Joshi 2014) for approximating the continuation values, as it provides better exercise strategies than the ordinary least squares method. Further, the results of the multiple regression method are less dependent on the choices of the basis, which gives us the freedom to choose the basis so that the approximated continuation value is linear in the first standard uniform. This choice allows us to obtain analytical solutions for the critical values and ensures that the proxy constraint function is monotonic in the first standard uniform.

Using the discontinuous approximated continuation value in (2.5) obtained via the multiple regression method, we need to make some modifications to the HOPP(1) change of measure. First, if the unbumped path is not within the innermost region, it is impossible for the bumped path with a small bump size to cross the angularity. Thus, we shall only use measure changes if the approximated continuation value is determined by the innermost regression, ie, $C_i(F_i) = C_{i,d}(F_i)$. Second, if the unbumped path is within the innermost region, the possibility of crossing the angularity reduces as the unbumped path moves away from the boundary. Thus, we shall only use the full HOPP(1) algorithm close to the boundary.

The following change of measure function is used to replace the first generated standard uniform $v_{i,1}$ at step i:

$$U_i(\theta, v_{i,1}) = g(\theta_0, v_{i,1}) U_i^{\mathrm{HOPP}(1)}(\theta, v_{i,1}) + (1 - g(\theta_0, v_{i,1}))v_{i,1}. \tag{3.3}$$

This function is a weighted average of the original standard uniform $v_{i,1}$ and the HOPP(1) change of measure function $U_i^{\mathrm{HOPP}(1)}$. The weight g is a function depending on the location of the unbumped path. In particular, we want to use more of the HOPP(1) change of measure near the boundary, since the possibility of the bumped path with a small bump size crossing the angularity is significant, and vice versa. We require the function g to satisfy the following conditions:

- $g(\theta_0, v_{i,1}) = 0$ if $|C_{i,d-1} - E_i| > L_{i,d-1}$;
- $g(\theta_0, v_{i,1}) = 1$ if $C_{i,d} = E_i$;
- $\partial g/\partial C_{i,d} > 0$ if $C_{i,d} < E_i$ and $\partial g/\partial C_{i,d} < 0$ if $C_{i,d} > E_i$;
- it is continuously differentiable as a function of $v_{i,1}$.

Thus, we need to choose a function, g, which depends on both $C_{i,d-1}$ and $C_{i,d}$.

There are many functions that satisfy the above conditions; for computing the Hessians of Bermudan swaptions and cancelable swaps, we employ the following:

$$g(\theta_0, v_{i,1}) = \begin{cases} 0 & \text{if } |C_{i,d-1} - E_i| > L_{i,d-1}, \\ \exp\left(-\dfrac{(C_{i,d}(F_i(\theta_0, v_{i,1})) - E_i(F_i(\theta_0, v_{i,1})))^2}{L_{i,d-1}^2 - (C_{i,d-1}(F_i(\theta_0, v_{i,1})) - E_i(F_i(\theta_0, v_{i,1})))^2}\right) & \text{otherwise.} \end{cases} \tag{3.4}$$

It is easy to verify that the above equation satisfies the conditions on the function g for it to be used in (3.3). In particular, we have constructed the function such that it is smooth in $v_{i,1}$.

Since we are in the innermost region, the function $U_i^{\mathrm{HOPP}(1)}$ is determined by the innermost approximated continuation value, ie, a_i is the critical value of $v_{i,1}$ such that $C_{i,d} = E_i$. This change of measure function is continuously differentiable in the parameter of interest θ, since $U_i^{\mathrm{HOPP}(1)}$ is a differentiable function, and it is optimal in terms of minimizing the variance of the likelihood ratio terms.

The resulting likelihood ratio weight is

$$W_i(\theta, v_{i,1}) = g(\theta_0, v_{i,1}) \frac{\partial U^{\mathrm{HOPP}(1)}(\theta, v_{i,1})}{\partial v_{i,1}} + (1 - g(\theta_0, v_{i,1})) + (U^{\mathrm{HOPP}(1)}(\theta, v_{i,1}) - v_{i,1}) \frac{\partial g(\theta_0, v_{i,1})}{\partial v_{i,1}}. \tag{3.5}$$

The HOMC pathwise estimate of the price is

$$\hat{P}(\theta) = P(\theta, U) \prod_{i=0}^{n-1} W_i(\theta, v_{i,1}), \tag{3.6}$$

where U is the set of modified standard uniforms, ie, the first standard uniform at each step, $v_{i,1}$, is replaced by (3.3). The new algorithm satisfies the following conditions:

- it is an unbiased estimate of the price, ie, $\mathbb{E}[P(\theta, V)] = \mathbb{E}[\hat{P}(\theta)]$;
- it is $\hat{C}^2$, since the sequence of measure changes has removed the pathwise discontinuities of the first-order derivatives;
- it is twice-differentiable almost surely, since the composite of differentiable functions is differentiable;

- the finite-differencing estimate (Glasserman 2004) of the Hessian is uniformly integrable.

Thus, it satisfies the conditions for applying the pathwise method to compute second-order derivatives.

4 APPLICATION TO CANCELABLE SWAPS AND NUMERICAL RESULTS

In this section, we shall perform numerical experiments to demonstrate the efficacy and speed of the HOMC algorithm. Since a cancelable swap is the sum of a vanilla swap and a Bermudan swaption, we shall only present the algorithm for a cancelable swap.

Our numerical examples are presented for cancelable swaps; however, we believe that there are no particular barriers to implementing them for more complicated Bermudan-type products. Here, we provide a brief discussion of the application of HOMC for snowballs. The exercise value of a callable snowball is not analytically computable. Piterbarg (2004b) suggested a regression approach to estimate the exercise value. Beveridge and Joshi (2008) showed numerous simplifications if one instead works with the cancelable product (see also Joshi 2011; Amin 2003). In particular, one only needs to use regressions to approximate the continuation values of cancelable snowballs, which is the same as for our cancelable swap example. One subtlety associated with snowballs is, however, that their payoff functions are also angular. To apply HOMC, we need to modify the change of measure function (U_i) at T_i, ie, we need to incorporate an additional critical point in order to ensure that the bumped path is on the same side as the unbumped paths for the point where the exercise value equals the approximated continuation value, as well as the point where the payoff at T_i hits zero.

4.1 Product description and LMM setup

We consider an n-period cancelable swap that can be canceled at each reset date. The estimated pathwise price of the product is as shown in (2.6), with the cashflow function

$$\mathrm{CF}(T_i, f_i) = \frac{\tau_i (f_i(T_i) - K)}{1 + \tau_i f_i(T_i)},$$

where K is the strike rate.

We consider the six-month Libors, ie, $T_i - T_{i-1} = 0.5$. A flat volatility structure with volatility $\sigma_i = 0.1$ is used. The instantaneous correlations are given by

$$\rho_{i,j} = \exp(-0.1|T_i - T_j|).$$

We use the spectral decomposition algorithm to reduce this into a lower-factor pseudo-square root for computational benefit. The rates are driven by a five-dimensional Brownian motion. Korn and Liang (2013) studied similar examples with a one-dimensional Brownian motion setup.

For the multiple regression method in the first pass, we use three basis variables, $\log(f_i)$, $\log(f_{i+1})$ and $\sum_{j=i+2}^{n-1}\log(f_j)$, at each T_i; the estimated $C_{i,j}$ are affine functions of these basis variables. Quadratic functions are typical examples used in the literature for the basis. However, the multiple regression algorithm makes the choice of basis functions for the least squares regression much less important. Linear basis functions of the log forward rates are also linear in the first standard normal random variable at each step. Thus, we can easily compute the critical value function, a_i, of the first standard uniform in (3.1). This property helps to reduce the computational cost of implementing the HOMC algorithm.

4.2 Comparison with the PWLR method

We first consider an at-the-money cancelable swap, with the first reset date $T_0 = 0.5$ and $K = 0.05$. The first pass is conducted using a 50 000 path sample and a regression depth of $d = 5$ for the multiple regression algorithm to fix the exercise strategy. The regression is conducted so that the innermost regression has 10 000 paths. Since we are working with multiple regressions, a significant number of paths is required for the innermost region to produce an accurate approximation of the continuation value close to the boundary. This approach was developed in GPU programs, where a large number of paths can be rapidly computed. Joshi (2014) used 327 680 paths for the first pass to develop the exercise strategy. In the second pass, we use 5000 paths for the HOMC algorithm. We shall use the PWLR method to benchmark our results. The PWLR method is not applicable to the reduced-factor LMM. Therefore, we only benchmark our method against the PWLR method for a five-rate cancelable swap, due to the computational cost of PWLR. In order to achieve a similar level of precision to HOMC, we need to run 500 000 paths in the second pass for PWLR. The results demonstrate that the HOMC method produces unbiased estimates of the Hessian, and it is more efficient than the PWLR method (see Tables 1 and 2).

As pointed out in Joshi and Zhu (2016), the PWLR approach can produce very large standard errors when the initial reset time is small. In our next numerical experiment, using the same setup as in the first experiment, except setting $T_0 = 0.1$, we need to use 2 000 000 paths in the second pass for the PWLR method in order to produce estimates with a similar level of precision to our HOMC with 5000 paths. The results are summarized in Tables 3 and 4.

TABLE 1 The mean of the Hessian computed by the HOMC algorithm using 5000 paths in the second pass, and the PWLR method using 500 000 paths in the second pass for an at-the-money five-rate cancelable swap when $T_0 = 0.5$.

	f_0	f_1	f_2	f_3	f_4
f_0	19.79 v. 19.22	8.37 v. 8.44	6.51 v. 6.39	5.08 v. 5.21	3.17 v. 3.25
f_1	8.37 v. 8.44	15.61 v. 16.22	8.52 v. 8.62	6.65 v. 6.50	3.69 v. 3.58
f_2	6.51 v. 6.39	8.52 v. 8.62	13.61 v. 13.66	7.65 v. 7.52	4.3 v. 4.97
f_3	5.08 v. 5.21	6.65 v. 6.50	7.65 v. 7.52	12.31 v. 11.91	4.42 v. 4.3
f_4	3.17 v. 3.25	3.69 v. 3.58	4.3 v. 4.97	4.42 v. 4.3	14.55 v. 14.06

50 000 paths are used in the first pass for both methods.

TABLE 2 The standard error of the Hessian computed by the HOMC algorithm using 5000 paths in the second pass, and the PWLR method using 500 000 paths in the second pass for an at-the-money five-rate cancelable swap when $T_0 = 0.5$.

	f_0	f_1	f_2	f_3	f_4
f_0	1.34 v. 0.77	1.02 v. 0.72	0.97 v. 0.7	1.06 v. 0.84	0.95 v. 1.1
f_1	1.02 v. 0.72	1.19 v. 0.99	1.02 v. 0.95	1.05 v. 1.15	0.96 v. 1.51
f_2	0.97 v. 0.7	1.02 v. 0.95	1.24 v. 0.95	1.29 v. 1.15	1.08 v. 1.54
f_3	1.06 v. 0.84	1.05 v. 1.15	1.29 v. 1.15	1.73 v. 1.15	1.43 v. 1.53
f_4	0.95 v. 1.1	0.96 v. 1.51	1.08 v. 1.54	1.43 v. 1.53	3.82 v. 1.12

50 000 paths are used in the first pass for both methods.

TABLE 3 The mean of the Hessian computed by the HOMC algorithm using 5000 paths in the second pass, and the PWLR method using 2 000 000 paths in the second pass for an at-the-money five-rate cancelable swap when $T_0 = 0.1$.

	f_0	f_1	f_2	f_3	f_4
f_0	10.62 v. 11.11	4.26 v. 4.07	2.98 v. 2.27	1.39 v. 1.43	1.32 v. 1.96
f_1	4.26 v. 4.07	21.89 v. 23.73	9.85 v. 10.54	5.88 v. 5.97	5.38 v. 5.36
f_2	2.98 v. 2.27	9.85 v. 10.54	21.43 v. 20.76	8.15 v. 8.93	5.37 v. 5.21
f_3	1.39 v. 1.43	5.88 v. 5.97	8.15 v. 8.93	17.42 v. 17.41	5.23 v. 5.85
f_4	1.32 v. 1.96	5.38 v. 5.36	5.37 v. 5.21	5.23 v. 5.85	11.68 v. 11.77

50 000 paths are used in the first pass for both methods.

We plot the sum of standard errors produced by the two methods using the same setup and 50 000 paths for both passes, but varying the first reset date from 0.05 to 0.5 (see Figure 1).

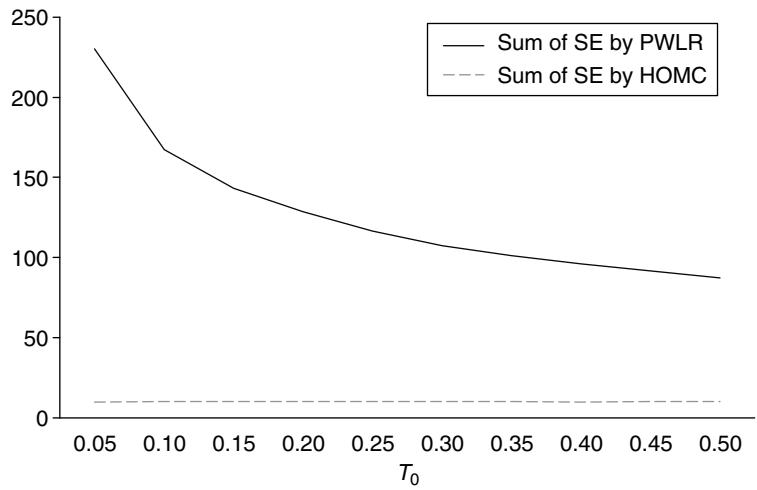
The maximum sum of standard errors computed by the PWLR method is 230.0428 when $T_0 = 0.05$, and the corresponding sum of standard errors computed by HOMC

TABLE 4 The standard errors of the Hessian computed by the HOMC algorithm using 5000 paths in the second pass, and the PWLR method using 2 000 000 paths in the second pass for an at-the-money five-rate cancelable swap when $T_0 = 0.1$.

	f_0	f_1	f_2	f_3	f_4
f_0	1.65 v. 0.75	1.36 v. 0.81	1.16 v. 0.75	1.09 v. 0.83	0.92 v. 1.08
f_1	1.36 v. 0.81	1.36 v. 1.12	1.1 v. 1.02	1.08 v. 1.14	0.97 v. 1.49
f_2	1.16 v. 0.75	1.1 v. 1.02	1.12 v. 1.02	1.14 v. 1.14	1.03 v. 1.49
f_3	1.09 v. 0.83	1.08 v. 1.14	1.14 v. 1.14	1.59 v. 1.14	1.28 v. 1.48
f_4	0.92 v. 1.08	0.97 v. 1.49	1.03 v. 1.49	1.28 v. 1.48	1.72 v. 1.07

50 000 paths are used in the first pass for both methods.

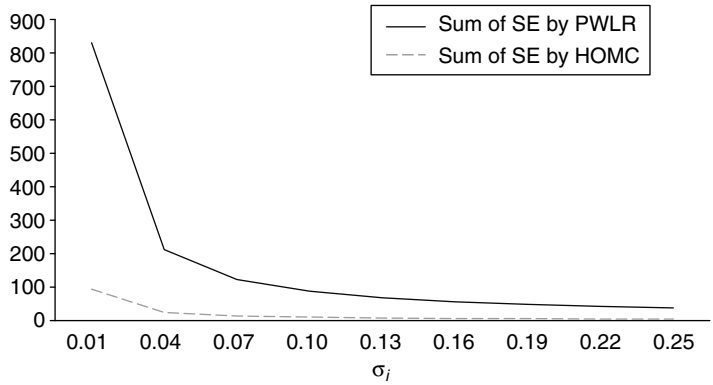
FIGURE 1 Comparison of the sum of standard errors with varying T_0 computed by the HOMC algorithm and the PWLR method, using 50 000 paths for both passes.



is 9.982. The minimum sum of standard errors computed by the PWLR method is 87.3943 when $T_0 = 0.5$, and the corresponding sum of standard errors computed by HOMC is 10.137. Overall, the PWLR method produces large standard errors as the initial reset time approaches zero, and our HOMC produces estimates of the Hessian with stable standard errors.

We further plot the sum of standard errors produced by the two methods, using the same setup and 50 000 paths for both passes, but varying the flat volatility, σ_i, from 0.02 to 0.2 (see Figure 2).

FIGURE 2 Comparison of the sum of standard errors with varying σ_i computed by the HOMC algorithm and the PWLR method, using 50 000 paths for both passes.



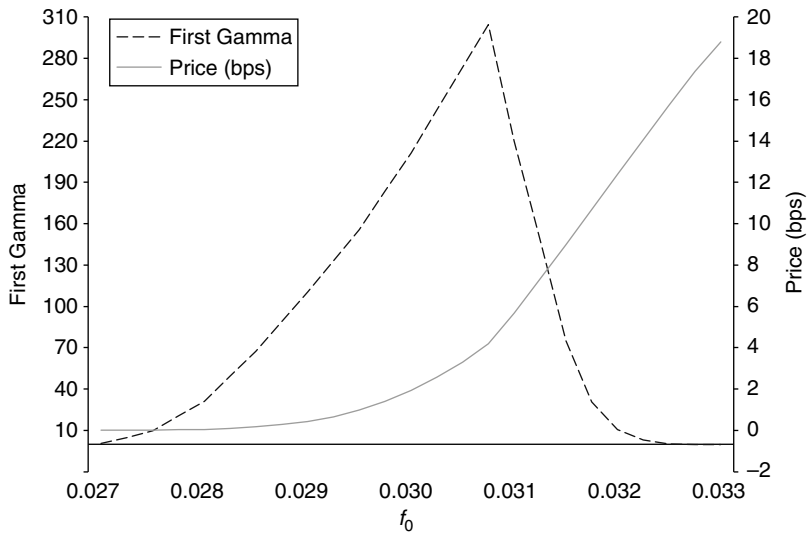
The maximum sum of standard errors computed by the PWLR method is 417.573 when $\sigma_i = 0.02$, and the corresponding sum of standard errors computed by HOMC is 47.473. The minimum sum of standard errors computed by the PWLR method is 45.605 when $\sigma_i = 0.2$, and the corresponding sum of standard errors computed by HOMC is 5.071. Overall, both the PWLR method and the HOMC method produce large standard errors as the flat volatility, σ_i, approaches zero, but our HOMC method always significantly outperforms the PWLR method.

In the next experiment, we consider a ten-rate at-the-money cancelable swap. Joshi and Yang (2011) computed the Deltas and Gammas of this product using the same setup as in our numerical experiment. They used a smoothing technique to deal with the angularity. Thus, the estimated Gammas are biased. For this particular product, the price we computed is 105.03 basis points (bps) (105bps in Joshi and Yang (2011)), with a standard error of 7.428E-05 and using 50 000 paths for both passes. We compute the Hessian and the Deltas of this product, and the results are summarized in Tables 6 and 7 (see the online appendix). The results show that the Delta and Hessian computed by HOMC are within the standard error of the results in Joshi and Yang (2011).

4.3 The shape of the Gamma and the price

The Gamma is an estimate of how much the Delta of an option changes when the underlying parameter shifts. As a tool, the Gamma describes how stable the current Delta is. A big Gamma means that the Delta can change dramatically for even a small move in the underlying forward rate. For a Delta-hedged position, constantly monitoring Gammas is especially important in order to ensure the adequacy of the

FIGURE 3 The price and the first Gamma of a ten-rate cancelable swap, varying the initial value of f_0 from 0.027 to 0.033, with 65 535 paths for the first pass and 50 000 paths for the second, when $T_0 = 0.01$.

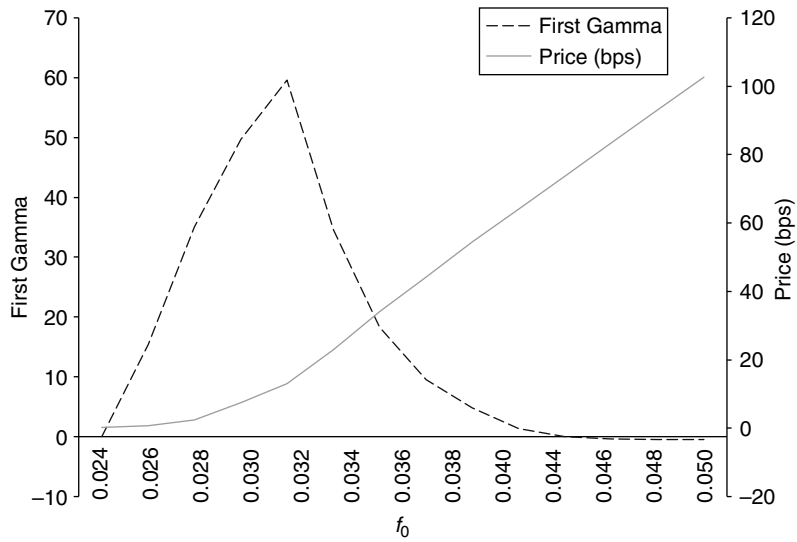


current Delta position. We shall examine the second-order derivative of the price with respect to the first forward rate, ie, the first Gamma, by the HOMC algorithm, as the first forward rate varies.

In Figures 3, 4 and 5, we plot the price and the corresponding first Gammas for a ten-rate cancelable swap, using the same setup as before, with varying f_0. In order to produce a stable plot, we use 65 535 paths for the first pass and 50 000 paths for the second pass in these experiments. Since the PWLR method also produces unbiased estimates of the Hessian, one can use it to produce these plots. However, due to the significant standard errors of the PWLR estimates (shown in the previous numerical examples), it is computationally expensive to use PWLR for producing plots with the same shape as our results. We consider the different first reset dates with different ranges for the first forwards, and we present the plots for $T_0 = 0.01$ in Figure 3, $T_0 = 0.05$ in Figure 4 and $T_0 = 0.5$ in Figure 5.

The three graphs demonstrate that Gammas are big near the exercise boundary and reduce away from it. The big Gammas are consistent with the plot of the price, demonstrating a greater curvature near the boundary. This is because a small change in the first forward is likely to shift the path across the angularity close to the boundary. As the time to the first reset date draws nearer, Gammas of at-the-boundary

FIGURE 4 The price and the first Gamma of a ten-rate cancelable swap, varying the initial value of f_0 from 0.024 to 0.05, with 65 535 paths for the first pass and 50 000 paths for the second, when $T_0 = 0.05$.



options increase. We also observe negative Gammas for deeply in-the-money cancelable swaps, which is consistent with the structure of the underlying product. The cancelable swap behaves very much like a vanilla swap when the product is deeply in-the-money and the vanilla swap has negative Gammas.

4.4 Computational cost

Finally, we investigate the computational cost of the HOMC algorithm. When the drift is calculated using the method introduced by Joshi (2003), the computational order of calculating the price is $\mathcal{O}(nF)$ per step, where n is the number of underlying forward rates and F is the number of Brownian motions driving the evolution of the forward rates. Given the number of steps (N) and the number of state variables (M), ie, the variables that depend on θ and are necessary to describe the state of the computation on a given step, the algorithmic Hessian method of Joshi and Yang (2011) is able to compute all the Gammas with order $\mathcal{O}(nFNM)$. However, their estimated Hessian is biased due to the localized smoothing applied at each exercise time. Here, we compute the unbiased estimate of the Hessian for cancelable swaps with the same computational order.

FIGURE 5 The price and the first Gamma of a ten-rate cancelable swap, varying the initial value of f_0 from 0.01 to 0.08, with 65 535 paths for the first pass and 50 000 paths for the second, when $T_0 = 0.5$.

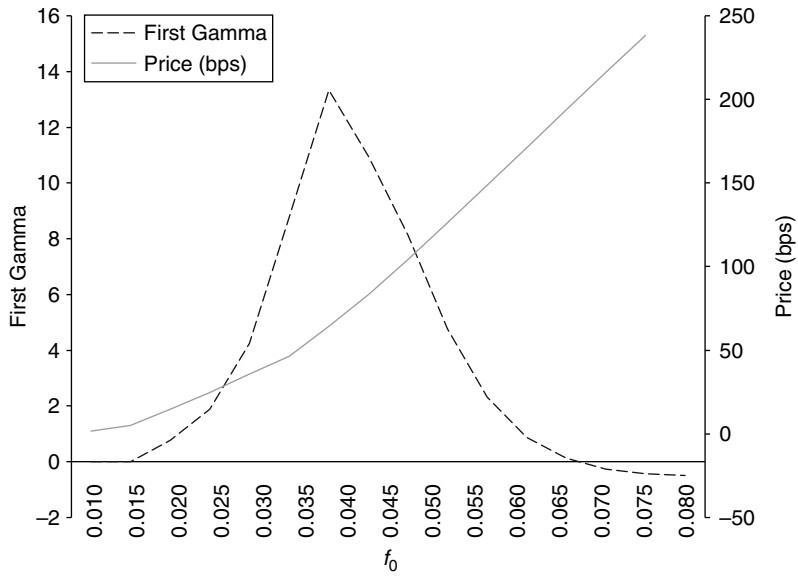


TABLE 5 Summary of Figures 3–5.

T_0	Max Gamma	Max price	Min Gamma	Min price
0.01	304.60 (3.9853 SE) when $f_0 = 0.0308$	18.885bps (6.571E−05 SE) when $f_0 = 0.033$	−0.4635 (7.403E−04 SE) when $f_0 = 0.033$	0.00011bps (1.549E−07 SE) when $f_0 = 0.027$
0.05	59.61 (1.77468 SE) when $f_0 = 0.032$	102.683bps (7.563E−05 SE) when $f_0 = 0.05$	−0.4708 (0.001601 SE) when $f_0 = 0.05$	0.19176bps (5.409E−06 SE) when $f_0 = 0.024$
0.5	13.34 (0.4235 SE) when $f_0 = 0.035$	238.303bps (7.712E−05 SE) when $f_0 = 0.08$	−0.4876 (0.007823 SE) when $f_0 = 0.08$	1.481bps (1.229E−05 SE) when $f_0 = 0.01$

For this experiment, we compute the price and the Greeks (the Hessian and the Deltas are computed simultaneously) with a sample size of 50 000 paths for both the first and second passes for an n-rate at-the-money cancelable swap, using the same setup as before. Since we are using n steps with n rates, the time taken to compute

TABLE 6 Times (in seconds) for pricing, and additional times for Greeks, of an n-rate cancelable swap, followed by the same times divided by n^2 and n^3, respectively, using 50 000 paths for both passes.

	n							
	5	**10**	**15**	**20**	**25**	**30**	**35**	**40**
Price	0.0653	0.173	0.327	0.521	0.769	1.044	1.386	1.776
Price $\times n^{-2}$	0.00261	0.00174	0.00145	0.00130	0.00123	0.00116	0.00113	0.00111
Greeks	0.230	1.007	2.462	4.758	8.070	12.695	18.600	27.462
Greeks $\times n^{-3}$	0.00184	0.00101	0.00073	0.00059	0.00053	0.00047	0.00043	0.00043

the price should have an order of n^2, and the Hessian should be n^3. The results are shown in Table 5. We also divide the times by n^2 and n^3 to compare the results. These are bounded as expected. The results show that the measure change performed at each step does not greatly increase the computational burden. Therefore, the HOMC algorithm can be implemented in practice to compute fast and accurate estimates of the Greeks for Bermudan swaptions and cancelable swaps.

5 CONCLUSION

We have successfully derived a methodology for computing the Hessians of Bermudan swaptions and cancelable swaps. The key to our approach is to perform a measure change at each exercise point if the path is near the boundary in order to ensure that the first-order derivatives of the pathwise estimate of the price are Lipschitz continuous. The measure change is selected so that the variance of the likelihood ratio part is minimized. The pathwise estimate of the price under the new scheme is $\hat{C}^2$, and the algorithmic Hessian method is then applied to calculate the pathwise estimate of the Hessian. Our numerical results suggest that the HOMC algorithm outperforms the PWLR method. The exact and efficient Hessian of the price computed by HOMC could be used in practice for hedging Bermudan swaptions and cancelable swaps.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

REFERENCES

Amin, A. (2003). Multi-factor cross currency Libor market models: implementation, calibration and examples. SSRN Working Paper (http://doi.org/fzkzdh).

Beveridge, C., and Joshi, M. (2008). Juggling snowballs. *Risk* **21**, 100–104.

Beveridge, C., Joshi, M., and Tang, R. (2013). Practical policy iteration: generic methods for obtaining rapid and tight bounds for Bermudan exotic derivatives using Monte Carlo simulation. *Journal of Economic Dynamics and Control* **37**(7), 1342–1361.

Brace, A. (2007). *Engineering BGM*. CRC Press, Boca Raton, FL.

Brace, A., Gątarek, D., and Musiela, M. (1997). The market model of interest rate dynamics. *Mathematical Finance* **7**(2), 127–155.

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

Chan, J. H., and Joshi, M. S. (2015). Optimal limit methods for computing sensitivities of discontinuous integrals including triggerable derivative securities. *IIE Transactions* **47**(9), 978–997 (http://doi.org/bhwt).

Fries, C. (2007). *Mathematical Finance: Theory, Modeling, Implementation*. Wiley.

Giles, M., and Glasserman, P. (2006). Smoking adjoints: fast Monte Carlo Greeks. *Risk* **19**(1), 88–92.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*, Volume 53. Springer.

Joshi, M. S. (2003). Rapid computation of drifts in a reduced factor Libor market model. *Wilmott Magazine* **5**, 84–85.

Joshi, M. S. (2011). *More Mathematical Finance*. Pilot Whale Press.

Joshi, M. S. (2014). Kooderive: multi-core graphics cards, the Libor market model, least squares Monte Carlo and the pricing of cancellable swaps. SSRN Working Paper (http://doi.org/bhww).

Joshi, M. S., and Rebonato, R. (2003). A stochastic-volatility, displaced-diffusion extension of the Libor market model. *Quantitative Finance* **3**(6), 458–469.

Joshi, M. S., and Yang, C. (2011). Algorithmic Hessians and the fast computation of cross-Gamma risk. *IIE Transactions* **43**(12), 878–892.

Joshi, M. S., and Zhu, D. (2016). Optimal partial proxy method for computing Gammas of financial products with discontinuous and angular payoffs. *Applied Mathematical Finance* **23**(1), 22–56 (http://doi.org/bkg4).

Korn, R., and Liang, Q. (2013). Robust and accurate Monte Carlo simulation of (cross-) Gammas for Bermudan swaptions in the Libor market model. *The Journal of Computational Finance* **17**(3), 87–110.

Korn, R., and Liang, Q. (2015). Correction to "Robust and accurate Monte Carlo simulation of (cross-)Gammas for Bermudan swaptions in the Libor market model". *The Journal of Computational Finance* **18**(3), 129–133.

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

Lord, R., and Pelsser, A. (2007). Level–slope–curvature: fact or artefact? *Applied Mathematical Finance* **14**(2), 105–130.

Mercurio, F. (2010). Modern Libor market models: using different curves for projecting rates and for discounting. *International Journal of Theoretical and Applied Finance* **13**(1), 113–137.

Piterbarg, V. (2004a). Computing Deltas of callable Libor exotics in forward Libor models. *The Journal of Computational Finance* **7**(3), 107–144.

Piterbarg, V. (2004b). A practioner's guide to pricing and hedging callable Libor exotics in forward Libor models. *The Journal of Computational Finance* **8**(2), 65–119.

Journal of Computational Finance **20**(1), 139–172
DOI: 10.21314/JCF.2017.337

Risk journals

Research Paper

Efficient computation of exposure profiles on real-world and risk-neutral scenarios for Bermudan swaptions

Qian Feng,[1] Shashi Jain,[3] Patrik Karlsson,[3] Drona Kandhai[3,4] and Cornelis W. Oosterlee[1,2]

[1]Centrum Wiskunde and Informatica (CWI), Science Park 123, 1098 XG Amsterdam, The Netherlands; emails: qian@cwi.nl, c.w.oosterlee@cwi.nl

[2]Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

[3]ING Bank, Foppingadreef 7, PO Box 1800, 1000 BV Amsterdam, The Netherlands; emails: shashi.jain@ingbank.com, patrik.karlsson@ingbank.com

[4]University of Amsterdam, PO Box 94216, 1090 GE, Amsterdam, The Netherlands; email: drona.kandhai@ingbank.com

(Received June 6, 2016; accepted June 7, 2016)

ABSTRACT

This paper presents a computationally efficient technique for the computation of exposure distributions at any future time under the risk-neutral and some observed real-world probability measures; these are needed for the computation of credit valuation adjustment (CVA) and potential future exposure (PFE). In particular, we present a valuation framework for Bermudan swaptions. The essential idea is to approximate the required value function via a set of risk-neutral scenarios and use this approximated value function on the set of observed real-world scenarios. This technique significantly improves the computational efficiency by avoiding nested Monte Carlo simulation and using only basic methods such as regression. We demonstrate the benefits

Corresponding author: Q. Feng

Print ISSN 1460-1559 | Online ISSN 1755-2850

of this technique by computing exposure distributions for Bermudan swaptions under the Hull–White and G2++ models.

Keywords: credit valuation adjustment (CVA); credit exposure; potential future exposure (PFE); Bermudan swaption; risk-neutral measure; real-world measure.

1 INTRODUCTION

The aim of the regulatory capital base in the Basel framework is to improve a bank's resilience against future losses due to defaults of counterparties (Basel Committee on Banking Supervision 2010). Credit exposure to counterparties occurs due to financial transactions or investments via over-the-counter (OTC) derivatives products. It is defined as the market value of the replacement costs of transactions if a counterparty defaults, assuming no recovery. Banks are required to hold regulatory capital to back exposure in the future to all their counterparties.

The Basel Committee gives specific definitions for the credit exposure metrics and adjustments regarding the future credit risk to banks/firms (Basel Committee on Banking Supervision 2005). For example, the expected exposure (EE) is the mean of the exposure distribution at any particular future date. The potential future exposure (PFE) is a high quantile (typically 97% or 99%) of the exposure distribution at any particular future date. The (unilateral) credit valuation adjustment (CVA) is the market value of the credit risk of the counterparty to the bank, which is typically calculated via an integral over time of the product of the discounted EE, the default probability and the percentage of loss given default (LGD) (Zhu and Pykhtin 2007).

EE and PFE are important indicators for the safety of a bank's portfolio to market movements. They are therefore used as metrics for capital requirements by regulators in Basel II and III (Gregory 2010). PFE is used for trading limits for portfolios with counterparties, as it may indicate at any future date the maximum amount of exposure with a predefined confidence. For example, the 99% PFE is the level of potential exposure that can be exceeded with a probability of 1%. CVA is a charge that has a direct impact on the balance sheet and the income statement of a firm, as it is an adjustment to the value of financial derivatives.

There are three basic steps in calculating future distributions of exposure (Gregory 2010):

- the generation of scenarios using the models that represent the evolution of the underlying market factors;
- the valuation of the portfolio for each scenario at each monitoring date;
- the determination of exposure values at each date for each scenario.

There is no doubt that CVA must be computed under the risk-neutral measure, as it is the market price of counterparty default risk. It is the cost of setting up a hedge portfolio to mitigate the credit risk that arises from exposure against a counterparty. In the setting of a CVA computation, scenarios are generated under the risk-neutral measure to compute "risk-neutral exposure distributions".

In contrast, for risk analysis, it is argued that expectations (EEs) and quantiles (PFEs) of future exposure values must be obtained via scenarios that can reflect the real world in a realistic way. We know that the risk-neutral probability measure used in the pricing process does not reflect the real-world probability of future outcomes, as it has been adjusted based on the assumption that market participants are risk neutral.

The Girsanov theorem states that the risk-neutral volatility should be equal to the real-world volatility when an equivalent measure exists (Andersen and Piterbarg 2010). However, it is well known that in practice the risk-neutral market-implied volatility differs from the observed real-world volatility (Hull *et al* 2014; Stein 2013). The observed historical dynamics and the calibrated risk-neutral dynamics may exhibit a different behavior, which is a challenge for risk management, as the computational cost becomes high.

In practice, calculation of exposure values on each real-world scenario at each monitoring date needs to be performed under a risk-neutral measure. For certain products, such as Bermudan swaptions, the valuation is based on Monte Carlo simulations, which can be computationally intensive, especially since pricing then requires another nested set of Monte Carlo paths. The computational cost increases drastically due to the number of real-world scenarios, risk-neutral paths and monitoring dates.

Employing a simplification, ie, assuming that the observed real-world scenarios are close to the risk-neutral scenarios and calculation takes place under just one measure, may lead to serious problems, as there are significant differences between the resulting distributions. Stein (2014) showed that exposures computed under the risk-neutral measure depend on the choice of numéraire and can be manipulated by choosing a different numéraire. As a conclusion, it is crucial that calculations of EE and PFE are done under the real-world instead of the risk-neutral measure.

The computational problem poses a great challenge to practitioners to enhance computational efficiency. Available solutions include reduction of the number of monitoring dates and Monte Carlo paths, application of variance reduction techniques and using interpolation and enhanced computational platforms such as graphics processing units (GPUs). Even with all these efforts, calculations cost a lot of time (Stein 2014).

For Bermudan swaptions, Joshi and Kwon (2016) provided an efficient approach for approximating CVA, which relies only on an indicator of future exercise time along scenarios, the decision of which is based on the regressed functions. The expected

exposure at a monitoring date is then obtained from the corresponding deflated pathwise cashflows. However, this approximation method cannot, in a straightforward fashion, be used for PFE on the real-world scenarios. For PFE computations, Stein (2013) proposed to avoid nested Monte Carlo simulations by combining the real-world and the risk-neutral probability measures. The computed results lie between the computed PFE values under the real-world and risk-neutral probability measures.

In this paper, we will focus on accurate computation of these risk measures for a heavily traded OTC derivative, the Bermudan swaption. There are well-developed methods that can be used to compute the time-zero value of Bermudan swaptions, such as regression and simulation-based Monte Carlo methods, eg, the least squares method (LSM) (Andersen 1999; Longstaff and Schwartz 2001) or the stochastic grid bundling method (SGBM) (Jain and Oosterlee 2012, 2015; Karlsson *et al* 2014), the finite difference (FD) PDE method or the Fourier expansion-based COS method (Fang and Oosterlee 2009).

This paper presents an efficient method to significantly enhance the computational efficiency of exposure values computation without the nested simulation. The key is to approximate the value function by a linear combination of basis functions obtained by risk-neutral scenarios, and to compute the expected payoff using the approximated value function to determine the optimal early exercise strategy on the paths representing the observed real-world scenarios. Only two sets of scenarios, one under the risk-neutral and one under the observed historical dynamics, are needed to compute the exposure distributions at any future time under the two measures. We apply this numerical scheme within the context of the LSM and SGBM approaches.

The paper is organized as follows. Section 2 presents the background mathematical formulation of EE, PFE and CVA as well as the dynamic programming framework for pricing Bermudan swaptions. Section 3 explains the essential insight for computation under two measures based on the risk-neutral scenarios, and describes the algorithms for computing the exposure profiles for the SGBM and LSM. We provide reference values for exposure, based on Fourier-cosine expansions, in Section 4. Section 5 presents numerical results with the algorithms developed for the one-factor Hull–White and two-factor G2++ models.

2 CREDIT VALUATION ADJUSTMENT, EXPECTED EXPOSURE AND POTENTIAL FUTURE EXPOSURE AS RISK MEASURES

In this section, we present the general framework for computing the exposure measurements. It is important to choose suitable probability measures to compute CVA, EE and PFE. We will discuss the practical background and the choice of probability measures.

2.1 Calibration and backtesting

It is well known that there are differences between calibrated historical dynamics and the dynamics implied by market prices. The reason is that models calibrated to historical data tend to reflect future values based on historical observations, and models calibrated to market prices reflect market participants' expectations about the future. Some research on building a joint framework in the real and risk-neutral worlds is done by Hull *et al* (2014). They propose a joint measure model for the short rate, in which both historical data and market prices can be used for calibration, and the calibrated risk-neutral and real-world measures are equivalent.

The practical setting with respect to calibrating model parameters is involved, however. Backtesting of counterparty risk models is required by the Basel Committee for those banks with an internal model method approval, for which PFE is an important indicator for setting limits. Backtesting refers to comparison of the outcomes of a bank's model against realized values in the past. The bank's model must be consistent with regulatory constraints; in other words, it must be able to pass the backtesting of PFE. A bank has to strike a balance between managing its risk and meeting the expectations of the shareholders. An overconservative estimate of market factors for exposure computation would lead to high regulatory capital reservings.

In short, a model used by a bank for generating scenarios should be able to reflect the real world: it should be able to meet the requirements of backtesting limits by regulators and the return rate by investors. Based on this, Kenyon *et al* (2015) proposed a risk-appetite measure that would fit in with these requirements. When a calibrated model under this risk-appetite measure cannot pass the backtesting, the bank needs to reconsider its preferences. From backtesting, one may find a so-called PFE-limit implied volatility of a model, by which, combined with a given budget, a bank's risk preference can be computed.

Ruiz (2012) called the model that describes the evolution of the underlying market factors the risk factor evolution (RFE) model, on which the backtesting is done periodically. The related probability measure is called the RFE measure. In that work, the model used to describe the real world is introduced first, and the relevant probability measure is defined based on the model. In some sense, there are different probability measures induced by the backtesting setting that describe the outcome, assuming the underlying factors evolve according to the calibrated model.

2.2 Mathematical formulation

Consider an economy within a finite time horizon $[0, T]$. The probability space $(\Omega, \mathcal{F}, \mathbb{P})$ describes the uncertainty and information, with Ω being the sample space consisting of outcome elements w, with $\mathcal{F}$ being a σ-algebra on Ω, and with $\mathbb{P}\colon \mathcal{F} \to [0, 1]$ being the probability measure that specifies the probability of events

happening on the measure space $(\Omega, \mathcal{F})$. Information up to time t is included in the filtration $\{\mathcal{F}_t, t \in [0, T]\}$.

Further assume a complete market without arbitrage opportunities. There exists an equivalent martingale measure such that a price associated to any attainable claim is computed as an expectation under this probability measure with respect to the associated numéraire. We choose to use a risk-neutral probability measure, denoted by $\mathbb{Q}\colon \mathcal{F} \to [0, 1]$, with numéraire $B_t(w) = \exp(\int_0^t r_s(w)\,\mathrm{d}s)$, where $\{r_s, s \in [0, t]\}$ is the risk-neutral short rate. The numéraire B_t represents a savings account with $B_0(w) = 1$.

Inspired by Kenyon *et al* (2015) and Ruiz (2012), we define a probability measure of observed history that can pass the backtesting. We use the notation $\mathbb{A}\colon \Omega' \to [0, 1]$ to present the observed historical probability measure on some measure space $(\Omega', \mathcal{F}')$ that we choose to reflect the probability of events in the real world. The probability measure $\mathbb{A}(\Omega') = 1$. The observed historical measure $\mathbb{A}$ may not be equivalent to the chosen risk-neutral measure $\mathbb{Q}$. As a probability space that includes realized outcomes in the past, the observed measure space should satisfy $\Omega' \subset \Omega$ and the associated filtration $\mathcal{F}'_t \subset \mathcal{F}_t$.

Let the stochastic process $\{\boldsymbol{X}_t \in \mathbb{R}^d,\ t \in [0, T]\}$ on $(\Omega, \mathcal{F})$ represent all influential market factors. We further define the market factor $\{\boldsymbol{X}_t\}_0^T$ on the space $(\Omega', \mathcal{F}')$ as the same mapping as the one on $(\Omega, \mathcal{F})$, ie, for an outcome w that may happen in both Ω and Ω' with different probability, one will have the same realized values for the market factors. Fixing an outcome $w \in \Omega' \subset \Omega$, the stochastic process is a function of time t, ie, $\boldsymbol{X}_t(w)\colon [0, T] \to \mathbb{R}^d$, which is a path of $\boldsymbol{X}_t$.

2.3 Definition of exposure, credit valuation adjustment and potential future exposure

Let the value of a portfolio v at time t be denoted by random variable $v_t\colon \Omega \to \mathbb{R}$; $v_t(w)$ is the value of the portfolio at time t on a path, which is the mark-to-market value of the portfolio computed under the risk-neutral measure $\mathbb{Q}$.

We define exposure as the replacement costs of the portfolio, given by

$$E_t(w) = \max(0, v_t(w)), \tag{2.1}$$

where $w \in \Omega$. Once the contract expires or, in the case of early exercise options, when the contract is exercised before expiry, the exposure of the portfolio is equal to zero.

Assume the percentage of LGD to be a constant over time, and let $\mathrm{PS}(t)$ represent the default probability up to time t, which is retrieved from credit default swap (CDS) market data under the risk-neutral probability measure. Assume the independence of

exposure and the probability of default. The CVA formula is then given by

$$\mathrm{CVA}_0 = \mathrm{LGD} \int_0^T \mathrm{EE}^*(t)\,\mathrm{d}\,\mathrm{PS}(t), \tag{2.2}$$

where the notation $\mathrm{d}\,\mathrm{PS}(t)$ represents the probability that the default event occurs during the interval $[t, t + \mathrm{d}t]$, and the discounted expected exposure EE^* is the conditional expectation of discounted exposure computed with the probability measure $\mathbb{Q}$, given by

$$\mathrm{EE}^*(t) = \mathbb{E}^{\mathbb{Q}}\left[\frac{E_t}{B_t}\right] = \int_{\Omega} \frac{E_t(w)}{B_t(w)}\,\mathrm{d}\mathbb{Q}(w), \tag{2.3}$$

where $\mathbb{E}^{\mathbb{Q}}$ is the risk-neutral expectation.

The curve $\mathrm{PFE}(t)$ is a function of future time t until the expiry of the transactions T. Its peak value indicates the maximum potential exposure of a portfolio over the horizon $[0, T]$. We define the PFE curve at time $t \in [0, T]$ as the 99% quantile of the exposure distribution, measured by the observed probability measure $\mathbb{A}$, given by

$$\mathrm{PFE}(t) = \inf\{y \mid \mathbb{A}(\{w\colon E_t(w) < y\}) \geqslant 99\%\}, \tag{2.4}$$

where $w \in \Omega'$ and $\boldsymbol{X}_0(w) = x$.

The maximum PFE (MPFE) is used to measure the peak value at the PFE curve over the time horizon $[0, T]$, given by

$$\mathrm{MPFE} = \max_{t\in[0,T]} \mathrm{PFE}(t). \tag{2.5}$$

In a similar way, another measure of credit risk of a portfolio is the EE, which is the average exposure at any future date, denoted by $\mathrm{EE}(t)$. The value of the EE curve at a monitoring date t under the observed measure $\mathbb{A}$ is given by

$$\mathrm{EE}(t) = \mathbb{E}^{\mathbb{A}}[E_t] = \int_{\Omega'} E_t(w)\,\mathrm{d}\mathbb{A}(w), \tag{2.6}$$

where $w \in \Omega'$ and $\boldsymbol{X}_0(w) = x$. The real-world expected positive exposure (EPE) over a time period $[0, T]$ is given by

$$\mathrm{EPE}(0, T) = \frac{1}{T} \int_0^T \mathrm{EE}(t)\,\mathrm{d}s. \tag{2.7}$$

In particular, we are interested in Bermudan swaptions, the pricing dynamics of which are presented in the following section.

2.4 Pricing of Bermudan swaptions

A Bermudan swaption is an option where the owner has the right to enter into an underlying swap either on the swaption's expiry or at a number of other predefined exercise dates before the expiry date. As soon as the swaption is exercised, the underlying swap starts. We assume here that the expiry date of the swap is predefined, so the duration of the swap is calculated from the swaption exercise date until a fixed end date. The underlying dynamics for the short rate governing the Bermudan swaption are either the one-factor Hull–White model or the two-factor G2++ model. Details of these well-known governing dynamics, either under the risk-neutral or the observed real-world dynamics, are presented in Appendixes 1 and 2 (available online).

We assume that the exercise dates coincide with the payment dates of the underlying swaps. Then, we consider an increasing maturity structure, $0 < T_1 < \cdots < T_N < T_{N+1}$, with T_{N+1} the fixed end date of the underlying swap and T_1, T_N the first and last opportunities to enter, respectively. We define $T_0 = 0$. We assume that when an investor enters a swap at time T_n, $n = 1, 2, \ldots, N$, the payments of the underlying swap will occur at $T_{n+1}, T_{n+2}, \ldots, T_{N+1}$, with time fraction $\tau_n = T_{n+1} - T_n$. We let N_0 represent the notional amount and K be the fixed strike. We use indicator $\delta = 1$ for a payer Bermudan swaption and $\delta = -1$ for a receiver Bermudan swaption.

The payoff for entering the underlying swap at time T_n associated with payment times $\mathcal{T}_n = \{T_{n+1}, \ldots, T_{N+1}\}$, conditional on $\boldsymbol{X}_{T_n} = x$, is given by (Brigo and Mercurio 2007)

$$U_n(x) = N_0\Bigg(\sum_{k=n}^{N} P(T_n, T_{k+1}, x)\tau_k\Bigg)\max(\delta(S(T_n, \mathcal{T}_n, x) - K), 0), \tag{2.8}$$

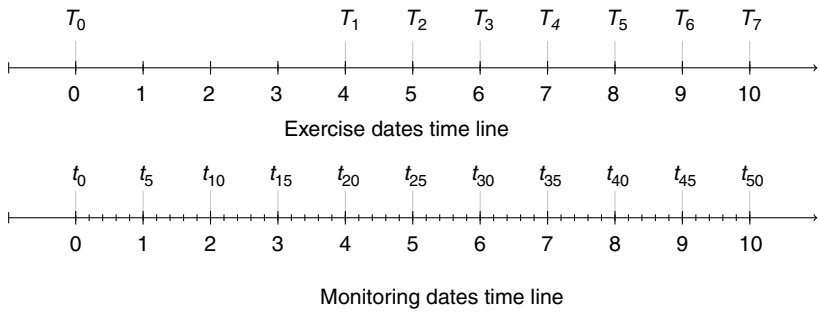
where the forward swap rate $S(t, \mathcal{T}_n, x)$ at time $t \leqslant T_n$ associated with time $T_n, \ldots, T_{N+1}$ is defined by

$$S(T_n, \mathcal{T}_n, x) = \frac{1 - P(T_n, T_{N+1}, x)}{\sum_{k=n}^{N} P(T_n, T_{k+1}, x)\tau_k}, \tag{2.9}$$

and $P(T_n, T_k, x)$ is the price of a zero-coupon bond (ZCB), conditional on $\boldsymbol{X}_{T_n} = x$, associated with times T_n and T_k. The analytic formula of the ZCB is related to the risk-neutral model for the underlying variable (see, for example, Appendixes 1 and 2, available online).

We refer to a function U_n, a bounded Borel function, as the exercise function, which represents the value of the future payments on any given scenario, when the option will be exercised at time T_n. For completeness, we define $U_0 \equiv 0$. We choose for the stochastic process $\{\boldsymbol{X}_t, t \in [0, T_N]\}$ an Ito diffusion. In that case, $U_n(\boldsymbol{X}_{T_n})$

FIGURE 1 Time lines.



is a continuous variable, as $\boldsymbol{X}_{T_n}$ is a continuous random variable. The value of not exercising the option at $t \in [0, T_N)$ is the value of continuing the option at time t.

Let time $t \in [T_n, T_{n+1})$, where the exercise opportunities are restricted to dates $\{T_{n+1}, \ldots, T_n\}$. The value of the Bermudan claim is the risk-neutral expectation of the (discounted) future payoff when exercising optimally (Øksendal 2003). With the strong Markov property of the Ito diffusions (Øksendal 2003), the value of this Bermudan claim at time t, conditional on $\boldsymbol{X}_t = x$, is the value that is obtained by maximizing the following object function (Glasserman 2003):

$$C(t,x) = \max_{I \in \{n+1,\ldots,N\}} B_t \mathbb{E}^{\mathbb{Q}}\left[\frac{U_I(\boldsymbol{X}_{T_I})}{B_{T_I}} \,\middle|\, \boldsymbol{X}_t = x\right], \tag{2.10}$$

where $n = 0, \ldots, N-1$. We refer to the value function $C(t,\cdot)$ as the continuation function at time t.

We wish to determine the exposure at a set of discrete monitoring dates, $\{0 = t_0 < t_1 < \cdots < t_M = T_N\}$, with time step $\Delta t_k = t_{k+1} - t_k$, $k = 0, \ldots, M-1$. These monitoring dates include the exercise dates $\{T_1, T_2, \ldots, T_N\}$, and t_M is equal to T_N. There are some dates between each two exercise dates, as we are also interested in the exposure at those intermediate dates.

Figure 1 presents the time lines of the exercise dates of a Bermudan swaption and the monitoring dates used for exposure computation as an example. This Bermudan swaption can be exercised seven times between year four and year ten, ie, year four is the first exercise date and year ten is the expiry (the last exercise date). The exposure monitoring dates are each one-fifth of a year from time zero until year ten. The monitoring date $t_{20} = 4$ coincides with the first exercise date,and the monitoring date $t_{50} = 10$ is equal to the last exercise opportunity.

We compute the exposure of a Bermudan claim at monitoring dates $\{t_m\}_{m=0}^{M}$. Value function V then satisfies (Glasserman 2003)

$$V(t_m, x) = \begin{cases} U_N(x), & t_M = T_N, \\ \max(C(t_m, x), U_n(x)), & t_m = T_n,\ n < N, \\ C(t_m, x), & T_n < t_m < T_{n+1},\ n < N, \end{cases} \tag{2.11}$$

where the continuation function C is computed as the conditional expectation of the future option value, given by

$$C(t_m, x) = B_{t_m} \mathbb{E}^{\mathbb{Q}} \left[\frac{V(t_{m+1}, \boldsymbol{X}_{t_{m+1}})}{B_{t_{m+1}}} \,\middle|\, \boldsymbol{X}_{t_m} = x \right], \tag{2.12}$$

which can be proven to be equivalent to (2.10) by induction.

The optimal exercise strategy is now as follows. At state $\boldsymbol{X}_{T_n} = x$, exercise takes place when $U_n(x) > C(T_n, x)$, and the option is kept at all non-exercise monitoring dates t_m. The value function V and continuation function C are defined over the time period $[0, T_N]$ and space $\mathcal{D} \in \mathbb{R}^d$.

The pricing dynamics in (2.11) are most conveniently handled by means of a backward recursive iteration. From known value U_N at time $t_M = T_N$, we compute $V(t_{M-1}, \cdot)$, and subsequently function $V(t_{M-2}, \cdot)$, and so on, until time zero. The essential problem, hence, becomes to determine the value function V and continuation function C at all monitoring dates $\{t_m\}_{m=1}^{\mathrm{M}}$.

REMARK 2.1 Given a fixed path $w' \in \Omega'$ or $w \in \Omega$, we compute the option values for the scenario as $V(t_m, \boldsymbol{X}_{t_m}(w))$ at any monitoring date t_m by (2.11). Once the option for scenario w is exercised at a specific date, the option terminates, and the exposure values regarding this option along the scenario from the exercise date to T become zero.

When a sufficient number of scenarios for the risk-neutral model are generated, the option value can be determined at all monitoring dates for any scenario, and we obtain a matrix of exposure values called the exposure profile.

The exposure profile, computed from observed real-world scenarios that are calibrated based on historical data, is an empirical real-world exposure density from which we can estimate real-world EEs and PFEs at each monitoring date. However, with risk-neutral short rate processes, the exposure profiles on risk-neutral scenarios are needed to compute the discounted EE.

We see that the key to computing exposure profiles on generated scenarios is to know the value function V and the continuation function C at all monitoring dates $\{t_m\}_{m=1}^{\mathrm{M}}$.

Nested Monte Carlo simulation is often used when a simulation-based algorithm is employed for the valuation; this is expensive, as simulations of risk-neutral paths are

needed for each (real-world) scenario at each monitoring date. Suppose that accurate pricing requires K_I risk-neutral paths at M monitoring dates; then, the computational time would be $O(M^2 K_I K_a)$ for K_a real-world scenarios for computing EE and PFE profiles.

3 REGRESSION-BASED MONTE CARLO ALGORITHMS

The computation of the conditional risk-neutral expectation is the most expensive part in the algorithm for dynamics (2.11). We propose algorithms that can approximate the continuation function in (2.12) by basic functions (for example, polynomial functions), based on the risk-neutral scenarios. Using these functions, we can perform simulations with risk-neutral expectations on the real-world scenarios without nested simulations. To compute CVA and PFE, we only need one set of K_q risk-neutral scenarios and one set of K_a real-world scenarios.

The proposed algorithms are based on the approximation of the continuation function within the SGBM and LSM simulation techniques. In this section, details of the algorithms are presented as well as the differences between the LSM and SGBM.

3.1 Stochastic grid bundling method

The SGBM approach, based on regression, bundling and simulation, was developed by Jain and Oosterlee (2015) for pricing Bermudan options. The SGBM can be very naturally generalized toward the efficient computation of exposure profiles because of its high accuracy in approximating expected payoffs on each Monte Carlo path. The SGBM has been used to compute risk-neutral exposure profiles (for computing CVA) of Bermudan-style claims in Karlsson *et al* (2014) and Feng and Oosterlee (2014).

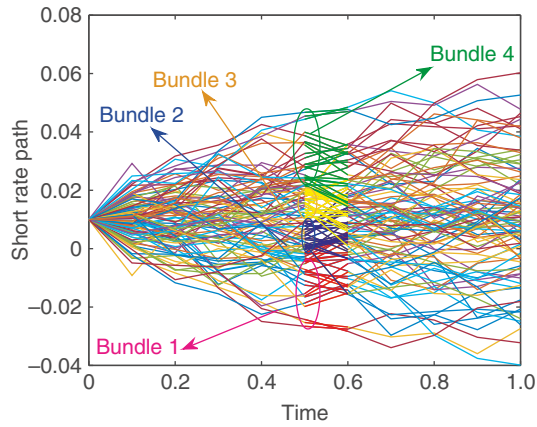
Pricing in the context of the SGBM approach is based on risk-neutral scenarios. Computation of discounted expected option values is performed locally in so-called bundles by means of local regression. We will store the bundle-wise approximated continuation functions and use them to compute exposure profiles for the observed real-world scenarios for a Bermudan swaption.

3.1.1 Risk-neutral scenarios

Let $\{X^q_{1,h}, \ldots, X^q_{M,h}\}_{h=1}^{K_q}$ be K_q scenarios, where the underlying factor evolves with the risk-neutral model. Pricing is done by a backward-in-time iteration, as in (2.11), from time t_M to time $t_0 = 0$.

To initialize the computation, the option value at expiry $t_M = T_N$ is computed as the immediate payoff U_N, ie, the option value realized on the hth scenario at time

FIGURE 2 Bundles and the disjoint sub-domains at time 0.5.



t_M, $\hat{v}^q_{M,h} = U_N(X^q_{M,h})$. As the option either expires or is exercised at time t_M, the exposure equals zero for all paths at time t_M, $\{\hat{E}^q_{M,h} = 0\}_{h=1}^{K_q}$.

At monitoring dates t_m, $m = M-1, \ldots, 1$, J partitions $\{\mathcal{B}_{m,j}\}_{j=1}^{J}$, which are called bundles, are defined; these consist of Monte Carlo path values at t_m and have very similar realized values based on the cross-sectional risk-neutral samples $\{X^q_{m,h}\}_{h=1}^{K_q}$. The realized values of the risk-neutral paths form a bounded domain, and these bundles divide the domain into disjoint sub-domains $\{D_{m,j}\}_{j=1}^{J}$. For a one-dimensional variable, these disjoint sub-domains connected to bundles can be presented by

$$\mathcal{D}_{m,j} = \Big(\max_{h\in\mathcal{B}_{m,j-1}}(X^q_{m,h}), \max_{h\in\mathcal{B}_{m,j}}(X^q_{m,h})\Big], \tag{3.1}$$

where $j = 2, 3, \ldots, J-1$. In particular, we define the first sub-domain at the left-side boundary as

$$\mathcal{D}_{m,1} = \Big(\min_{h\in\mathcal{B}_{m,1}}(X^q_{m,h}), \max_{h\in\mathcal{B}_{m,1}}(X^q_{m,h})\Big]. \tag{3.2}$$

Figure 2 shows the Monte Carlo paths in four bundles and the associated disjoint sub-domains at a monitoring date. The bundles are based on the values of the realized short rate at time 0.5.

The continuation function $C(t_m, \cdot)$ is approximated "in a bundle-wise fashion" for each domain by approximating the value function at time t_{m+1} for the paths in a bundle.

For $j = 1, \ldots, J$, on the Monte Carlo paths in bundle $\mathcal{B}_{m,j}$, the value function $V(t_{m+1}, \cdot)$ is approximated by a linear combination of basis functions $\{\phi_k\}_{k=1}^{B}$, ie,

$$V(t_{m+1}, y) \approx \sum_{k=1}^{B} \beta_k(m, j)\phi_k(y), \tag{3.3}$$

where the coefficients $\beta_k(m, j)$ of the kth basis function minimize the sum of squared residuals over the paths in bundle $\mathcal{B}_{m,j}$, ie,

$$\sum_{h \in \mathcal{B}_{mj}} \left(\hat{v}^q_{m+1,h} - \sum_{k=1}^{B} \beta_k(m, j)\phi_k(X^q_{m+1,h}) \right)^2, \tag{3.4}$$

with $\{\hat{v}^q_{m+1,h}\}_{h=1}^{K_q}$ the option values at time t_{m+1} on the cross-sectional sample $\{X^q_{m+1,h}\}_{h=1}^{K_q}$.

Using the approximated value function in (3.3) instead of the "true value" in (2.12), the continuation function on $\mathcal{D}_{m,j}$ can be approximated by

$$C(t_m, x) \approx \sum_{k=1}^{B} \beta_k(m, j)\psi_k(x, t_m, t_{m+1}), \tag{3.5}$$

where $x \in \mathcal{D}_{m,j}$ and function ψ_k is the conditional risk-neutral discounted expectation of basis function ϕ_k, defined by

$$\psi_k(x, t_m, t_{m+1}) := B_{t_m} \mathbb{E}^{\mathbb{Q}} \left[\frac{\phi_k(\boldsymbol{X}_{t_{m+1}})}{B_{t_{m+1}}} \,\middle|\, \boldsymbol{X}_{t_m} = x \right]. \tag{3.6}$$

The formulas for $\{\psi_k\}_{k=1}^{B}$ can be obtained easily, and often analytically, when polynomial terms are chosen as the basis functions (see Section 3.1.4).

The expected values on the paths of the bundle $\mathcal{B}_{m,j}$ can then be approximated by

$$\hat{c}^q_{m,h} \approx \sum_{k=1}^{B} \beta_k(m, j)\psi_k(X^q_{m,h}, t_m, t_{m+1}), \tag{3.7}$$

where $h \in \mathcal{B}_{m,j}$.

After computation of the continuation values for all paths $\{\hat{c}^q_{m,h}\}_{h=1}^{K_q}$ at time t_m, we determine the option value at time t_m by

$$\hat{v}^q_{m,h} = \begin{cases} \max(U_n(X^q_{m,h}), \hat{c}^q_{m,h}), & t_m = T_n, \\ \hat{c}^q_{m,h}, & t_m \in (T_n, T_{n+1}), \end{cases} \tag{3.8}$$

where U_n is the exercise function.

The exposure value on the hth path from time t_m to expiry t_M is updated by the following scheme.

(1) When exercised at exercise time $t_m = T_n$, a value of zero is assigned to the exposures along the path from time t_m to expiry, ie,

$$\hat{E}^q_{k,h} = 0, \quad k = m, \ldots, M.$$

(2) When the option is "alive" at an exercise date, or when t_m is a monitoring date between two exercise dates, the exposure at the path is equal to the approximated continuation value, $\hat{E}^q_{m,h} = \hat{c}^q_{m,h}$, and the exposure values at later times remain unchanged.

The algorithm proceeds by moving one time step backward to t_{m-1}, where the paths are again divided into new bundles, based on the realized values $\{X^q_{m-1,h}\}^{K_q}_{h=1}$, and the continuation function is approximated in a bundle-wise fashion. Option values are evaluated, and the exposure profile is updated. The algorithm proceeds, recursively, back to $t_0 = 0$. At time t_0, we do not need bundles, and regression takes place for all paths to get the coefficients $\{\beta_k(0)\}^B_{k=1}$, ie, the option value at time zero is approximated by

$$\hat{v}^q_0 \approx \sum_{k=1}^{B} \beta_k(0)\psi_k(x_0, t_0, t_1). \tag{3.9}$$

During the backward recursive iteration, information about the boundaries of the disjoint sub-domains, $\mathcal{D}_{m,j}$, is stored, along with the associated coefficients $\{\beta_k(m, j)\}^B_{k=1}$ for each index, $j = 1, \ldots, J$, at each monitoring date, t_m, $m = 0, \ldots, M-1$. Based on this information, we can retrieve the piecewise approximated continuation function for each time t_m.

With the risk-neutral exposure profiles, $\{\hat{E}^q_{1,h}, \ldots, \hat{E}^q_{M,h}, \}^{K_q}_{h=1}$, the discounted EE of a Bermudan swaption can be approximated by

$$\mathrm{EE}^*(t_m) \approx \frac{1}{K_q} \sum_{h=1}^{K_q} \exp\left(- \sum_{k=0}^{m+1} \tfrac{1}{2}(\hat{r}^q_{k,h} + \hat{r}^q_{k+1,h})\Delta t_k \right) \hat{E}^q_{m,h}, \tag{3.10}$$

where $\{\hat{r}^q_{1,j}, \ldots, \hat{r}^q_{M,j}\}^{K_q}_{j=1}$ represents simulated risk-neutral short rate values.

3.1.2 Real-world scenarios

During the computations on the risk-neutral scenarios, we have stored the bundle-wise coefficients $\{\beta_k(m, j)\}^B_{k=1}$ and the associated sub-domains $\{\mathcal{D}_{m,j}\}^J_{j=1}$, by which we can perform valuation and exposure computation for any scenario without nested simulation.

We present the steps to compute exposure profiles on a set of K_a observed real-world scenarios $\{X^a_{1,h}, \ldots, X^a_{M,h}\}$. These profiles are also determined by a backward iteration from time t_M until time t_0.

At expiry date t_M, the exposure equals zero,

$$\{\hat{E}^a_{M,h} = 0\}_{h=1}^{K_a}.$$

At monitoring dates $t_m < t_M$, for each index $j = 1, \ldots, J$, we determine those paths for which $X^a_{m,h} \in \mathcal{D}_{m,h}$; we compute the continuation values for these paths by

$$\hat{c}^a_{m,h} \approx \sum_{k=1}^{B} \beta_k(m, j)\psi_k(X^a_{m,h}, t_m, t_{m+1}), \tag{3.11}$$

where $X^a_{m,h} \in \mathcal{D}_{m,j}$.

Based on these continuation values, we update the exposure profile on this set of real-world scenarios.

At an exercise time $t_m = T_n$, we compare the approximated continuation value $\hat{c}^a_{m,h}$ with the immediate exercise values $U_n(X^a_{m,h})$ for each path; when the immediate exercise value is largest, the option is exercised at this path at time t_m, and exposure values at this path from time t_m to expiry are set to zero, ie, $\hat{E}^a_{k,h} = 0, \quad k = m, \ldots, M$.

Otherwise, $\hat{E}^a_{m,h} = \hat{c}^a_{m,h}$ and the later exposure values remain unchanged.

When t_m is an intermediate monitoring date, the exposure values are equal to the continuation values in (3.11).

Note that the time-zero option value is the same for the risk-neutral and real-world scenarios, ie, $\hat{v}^q_0 = \hat{v}^a_0$. Values of the observed real-world PFE and EE curves at monitoring dates t_m can be approximated by

$$\begin{aligned} \text{PFE}(t_m) &= \text{quantile}(\hat{E}^a_{m,h}, 99\%), \\ \text{EE}(t_m) &= \frac{1}{K_a}\sum_{h=1}^{K_a} \hat{E}^a_{m,h}. \end{aligned} \tag{3.12}$$

3.1.3 Stochastic grid bundling method bundling technique

An essential technique within the SGBM is the bundling of asset path values at each monitoring date, based on the cross-sectional risk-neutral samples. Numerical experiments have shown that the algorithm converges with respect to the number of bundles (Feng and Oosterlee 2014; Jain and Oosterlee 2015).

Various bundling techniques have been presented in the literature, such as the recursive-bifurcation method, k-means clustering (Jain and Oosterlee 2015) and the equal-number bundling method (Feng and Oosterlee 2014). Here, we use the

equal-number bundling technique. In this method, at each time step t_m, we rank the paths by their realized values, $\{X^q_{m,h}\}_{h=1}^{K_q}$, and place the paths with indexes between $(j-1)K_q/J+1$ and jK_q/J into the jth bundle, $\mathcal{B}_{m,j}$, $j=1,\ldots,J-1$. The remaining paths are placed in the Jth bundle, $\mathcal{B}_{m,J}$. Asset paths do not overlap among bundles at time t_m, and each path is placed in a bundle.

The advantage of the equal-number bundling technique is that the number of paths within each bundle is proportional to the total number of asset paths. An appropriate number of paths in each bundle is important for accuracy during the local regression. As mentioned, the bundling technique is also used to determine the disjoint subdomains on which the value function is approximated in a piece-wise fashion.

For high-dimensional problems, one can either use the equal-number bundling technique along each dimension, as employed in Feng and Oosterlee (2014), or one can project the high-dimensional vector onto a one-dimensional vector and then apply the equal-number bundling technique (see Jain and Oosterlee 2015; Leitao and Oosterlee 2015).

3.1.4 *Formulas for the discounted moments in the stochastic grid bundling method*

When we choose monomials as the basis functions within the bundles in the SGBM, the conditional expectation of the discounted basis functions is equal to the discounted moments. There is a direct link between the discounted moments and the discounted characteristic function (dChF), which we can also use to derive analytic formulas for the discounted moments.

As a one-dimensional example, in which the underlying variable represents the short rate, ie, $\boldsymbol{X}_t = r_t$, let the basis functions be $\phi_k(r_t) = (r_t)^{k-1}$, $k=1,\ldots,B$. The discounted moments ψ_k, conditional on $r_{t_m} = x$ over the period (t_m, t_{m+1}), are given by

$$\psi_k(x,t_m,t_{m+1}) := \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_{t_m}^{t_{m+1}} r_s\,\mathrm{d}s\right)(r_{t_{m+1}})^{k-1}\;\middle|\; r_{t_m}=x\right], \tag{3.13}$$

and the associated dChF is given by

$$\Phi(u;x,t_m,t_{m+1}) := \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_{t_m}^{t_{m+1}} r_s\,\mathrm{d}s + \mathrm{i}u r_{t_{m+1}}\right)\;\middle|\; r_{t_m}=x\right]. \tag{3.14}$$

When an explicit formula for the dChF is available, ψ_k can be derived by

$$\psi_k(x,t_m,t_{m+1}) = \frac{1}{(\mathrm{i})^{k-1}}\frac{\partial^{k-1}\Phi}{\partial u^{k-1}}(u;x,t_m,t_{m+1})\bigg|_{u=0}. \tag{3.15}$$

Using the relation in (3.15), we find analytic formulas for the discounted moments when the dChF is known. The dChFs of the Hull–White and G2++ models are presented in Appendixes 1 and 2, respectively (available online).

3.2 Least squares method

The LSM is also a regression-based Monte Carlo method that is very popular among practitioners. The objective of the LSM algorithm is to find for each path the optimal stopping policy at each exercise time T_n; the option value is computed as the average value of the generated discounted cashflows. The optimal early exercise policy for the in-the-money paths is determined by comparing the immediate exercise value and the approximated continuation value, which is approximated by a linear combination of (global) basis functions $\{\phi_k\}_{k=1}^{B}$.

One can always combine the (expensive) nested Monte Carlo simulation with the LSM for the computation of EE and PFE on observed real-world scenarios. We will adapt the original LSM algorithm to obtain a more efficient method for computing risk-neutral and real-world exposures. The technique is similar to that described for the SGBM: valuation on the risk-neutral scenarios, approximation of the continuation function and computation of risk-neutral and real-world exposure quantities.

The involved part in the LSM is that discounted cashflows, realized on a path, are not representative of the "true" continuation values. In the LSM algorithm, the approximated continuation values are only used to determine the exercise policy; therefore, one cannot use them to determine the maximum of the immediate exercise value and discounted cashflows in order to approximate the option value (Feng and Oosterlee 2014), as is done in the SGBM.

The challenge is to approximate exposure values by means of the realized discounted cashflows over all paths.

Joshi and Kwon (2016) present a way of employing realized discounted cashflows and the sign of the regressed values for an efficient computation of CVA on risk-neutral scenarios. However, since the average of discounted cashflows is not the value of a contract under the observed real-world measure, it cannot be used to compute real-world EE or PFE quantities.

Here, we propose two LSM-based algorithms for the approximation of continuation values with realized cashflows. They can be seen as alternative algorithms to the SGBM for the computation of exposure values when we do not have expressions for the discounted moments (or when the LSM is the method of choice for many other tasks). We will test the accuracy of the algorithms compared with the SGBM and reference values generated by the COS method in Section 5.

3.2.1 Risk-neutral scenarios

First of all, we briefly explain the original LSM algorithm with the risk-neutral scenarios. At the final exercise date, $t_M = T_N$, the option holder can either exercise an option or not, and the generated cashflows are given by $q_{M,h} = U_N(X^q_{M,h})$, $h = 1, \ldots, K^q$.

At monitoring dates $t_m \in (T_{n-1}, T_n)$, at which the option cannot be exercised, the realized discounted cashflows are updated by

$$q_{m,h} = q_{m+1,h} D_{m,h}, \tag{3.16}$$

with the discount factor $D_{m,h} = \exp\left(-\frac{1}{2}(\hat{r}^q_{m,h} + \hat{r}^q_{m+1,h})\Delta t_m\right)$.

At an exercise date $t_m = T_n$, prior to the last exercise opportunity, the exercise decision is based on the comparison of the immediate payoff by exercising and the continuation value when holding the option on the in-the-money paths; the continuation values at those in-the-money paths are approximated by projecting the (discounted) cashflows of these paths onto some global basis functions $\{\phi_1, \ldots, \phi_B\}$.

The option is exercised at an in-the-money path, where the payoff is larger than the continuation value. After determining the exercise strategy at each path, the discounted cashflows read

$$q_{m,h} = \begin{cases} U_n(X^q_{m,h}), & \text{exercised,} \\ q_{m+1,h} D_{m,h}, & \text{to be continued.} \end{cases} \tag{3.17}$$

Again, computation of the discounted cashflows at any monitoring date takes place recursively, backward in time. At time $t_0 = 0$, the option value is approximated by

$$\hat{v}^q_{0,h} \approx \frac{1}{K_q} \sum_{h=1}^{K_q} q_{0,h}.$$

During the backward recursion, the discounted cashflows realized on all paths at each monitoring date t_m are computed.

For the computation of the real-world EE and PFE quantities, valuation needs to be done on the whole domain of realized asset values, as we need the continuation values at each monitoring date for all paths. We therefore propose to use the realized discounted cashflows determined by (3.17) or (3.16) on the risk-neutral scenarios.

One possible algorithm in the LSM context involves employing two disjoint sub-domains, similar to in the SGBM. At each monitoring date $t_m \in (T_{n-1}, T_n]$, Monte Carlo paths are divided into two bundles based on the realized values of the underlying variable, so the approximation can take place in two disjoint sub-domains, given by

$$\mathcal{U}_{n,1} = \{x \mid U_n(x) \leqslant 0\}, \qquad \mathcal{U}_{n,2} = \{x \mid U_n(x) > 0\}. \tag{3.18}$$

The continuation function is approximated on these two sub-domains as

$$C(t_m, x) \approx \sum_{k=1}^{B} \zeta_k(t_m, j)\phi_k(x), \tag{3.19}$$

where $x \in \mathcal{U}_{n,j}$; the coefficients $\zeta_k(t_m, j)$ are obtained by minimizing the sum of squared residuals over the two bundles, respectively, which is given by

$$\sum_{X^q_{m,h} \in \mathcal{U}_{n,j}} \left(q_{m+1,h} D_{m,h} - \sum_{k=1}^{B} \zeta_k(t_m, j) \phi_k(X^q_{m,h}) \right)^2. \tag{3.20}$$

We refer to this technique as the LSM-bundle technique.

The other possible algorithm is to perform the regression over all Monte Carlo paths and compute the approximated continuation function on each path. The regression is as in (3.20), using basis functions and discounted cashflows but for all paths. We call this the LSM-all algorithm. Note that the exercise decision is still based on the in-the-money paths with approximated payoff, using (3.19) at exercise dates $T_n < T_N$, $n = 1, \ldots, N-1$.

We compute the risk-neutral exposure profiles with the approximated value functions in (3.19) by means of the same backward recursion procedure in Section 3.1.1.

3.2.2 Real-world scenarios

The LSM-bundle algorithm can be used for computing exposure on the observed real-world scenarios directly. It is based on the same backward iteration in Section 3.1.2; however, the continuation values are computed by the function in (3.19), ie,

$$\hat{c}^a_{m,h} \approx \sum_{k=1}^{B} \zeta_k(t_m, j) \phi_k(X^a_{m,h}). \tag{3.21}$$

At an early exercise date $t_m = T_n < T_N$, the early exercise policy is determined for in-the-money paths by comparing $\hat{c}^a_{m,h}$ with the immediate exercise value $U_n(X^a_{m,h})$. Exposure values along the path from time t_m to expiry are set to zero if the option at a path is exercised.

By the LSM-all algorithm, we use the continuation function approximated in (3.19) for determining the optimal early exercise time on each real-world path; the regressed function is based on all paths to compute exposure values. We will compare the LSM-bundle and LSM-all algorithms in Section 5.

3.3 Differences between the stochastic grid bundling method and least squares method algorithms

The SGBM differs from the LSM with respect to the bundling and the local regression based on the discounted moments. By these components, the SGBM approximates the continuation function in a more accurate way than the LSM, but at a (small) additional computational cost. Here, we give some insights into these differences.

The use of SGBM bundles may improve the local approximation on the disjoint sub-domains, and we can reduce the number of basis functions.

Another important feature of the SGBM is that option values are obtained from regression in order to obtain the coefficients for the continuation function.

Loosely speaking, the continuation function is approximated locally on the bounded sub-domains $\{\mathcal{D}_{m,j}\}_{j=1}^{J}$ by projection on the functions $\{\psi_k\}_{k=1}^{B}$.

Compared with the SGBM, the LSM is based on the discounted cashflows for regression to approximate the expected payoff; however, discounted cashflows do not represent the realized expected payoff on all Monte Carlo paths. In the LSM, the expected payoff is only used to determine the optimal early exercise time and not the option value. One cannot compute the option value by using the maximum of the expected payoff and the exercise value, as it will lead to an upward bias for the time-zero option value (Longstaff and Schwartz 2001).

The SGBM does not suffer from this, and the maximum of the exercise value and the regressed continuation values gives us the direct estimator. We recommend also computing the path estimator for convergence of the SGBM algorithm. Based on a new set of scenarios with the obtained coefficients to determine the optimal exercise policy on each path, we then take the average of the discounted cashflows as the time-zero option value. Upon convergence, the direct and path estimators should be very close (Jain and Oosterlee 2015).

The LSM approach is a very efficient and adaptive algorithm for computing option values at time zero. The LSM-based algorithms for computing exposure can be regarded as alternative ways of computing the future exposure distributions based on simulation. We will analyze the accuracy of all variants in Section 5.

4 THE COS METHOD

In this section, we explain the computation of the continuation function of Bermudan swaptions under the one-factor Hull–White model by the COS method. The COS method is an efficient and accurate method based on Fourier-cosine expansions. It can be used to determine reference values for the exposure. For Lévy processes and early exercise options, the computational speed of the COS method can be enhanced by incorporating the fast Fourier transform (FFT) into the computations. We cannot employ the FFT, because the resulting matrixes with the Hull–White model do not have the special form needed (Toeplitz and Hankel matrixes (see Fang and Oosterlee 2009)) to employ the FFT. For the G2++ model, the two-dimensional COS method developed in Ruijter and Oosterlee (2012) may be used for pricing Bermudan swaptions, but this is not pursued here.

When the short rate is a stochastic process, the discount factor is a random variable, which should be under the expectation operator when computing the continuation

values. In order to compute the discounted expectation of the future option values, we will work with the discounted density function. Let $p(y, z; t, T, x)$ be the joint density function of the underlying variables $\boldsymbol{X}_T = y$ and $z = -\log(B_t/B_T)$, conditional on $\boldsymbol{X}_t = x$. The discounted density function is defined as the marginal probability function p_X of $\boldsymbol{X}_T$, derived by integrating the joint density p over $z \in \mathbb{R}$:

$$p_X(y; t, T, x) := \int_{\mathbb{R}} \mathrm{e}^{-z} p(y, z; t, T, x)\,\mathrm{d}z. \tag{4.1}$$

The dChF is the Fourier transform of the discounted density function, ie,

$$\begin{aligned}\Phi(u; t, T, x) &= \mathbb{E}\left[\frac{B_t}{B_T}\exp(\mathrm{i}u\boldsymbol{X}_T) \;\middle|\; \boldsymbol{X}_t = x\right] \\ &= \int_{\mathbb{R}^n} \mathrm{e}^{\mathrm{i}uy} \int_{\mathbb{R}} \mathrm{e}^{-z} f(y, z; t, T, x)\,\mathrm{d}z\,\mathrm{d}y \\ &= \int_{\mathbb{R}^n} \mathrm{e}^{\mathrm{i}uy} p_X(y; t, T, x)\,\mathrm{d}y,\end{aligned} \tag{4.2}$$

where $u \in \mathbb{R}^d$ and $\boldsymbol{X}_t \in \mathbb{R}^d$.

In the one-dimensional setting, $\boldsymbol{X}_t = X_t$, the discounted density function p_X can be approximated by Fourier-cosine expansions (Fang and Oosterlee 2009). On an integration range $[a, b]$, we define $\{u_k\}_{k=0}^{Q-1}$ by

$$u_k = \frac{k\pi}{b-a}, \quad k = 0, \ldots, Q-1,$$

where Q represents the number of cosine terms used in the Fourier-cosine expansion of the discounted density, which is given by

$$p_X(y; t, T, x) \approx \frac{2}{b-a}\sum_{k=0}^{Q-1}{}' \mathcal{P}_k(x, t, T)\cos(u_k(y-a)). \tag{4.3}$$

The symbol $\sum'$ in (4.3) implies that the first term of the summation is multiplied by $\frac{1}{2}$ and the Fourier coefficients $\mathcal{P}_k$ are given by

$$\mathcal{P}_k(x, t, T) := \mathrm{Re}\{\Phi(u_k; t, T, x)\exp(-\mathrm{i}au_k)\}. \tag{4.4}$$

The integration range should be chosen such that the integral of the discounted density function over the region $[a, b]$ resembles very well the value of a ZCB between time t and T, given $\boldsymbol{X}_t = x$. The way of constructing the range is presented in Appendix 1 (available online).

With (2.12), the continuation function, conditional on $X_{t_m} = x$, at any monitoring date $t_m \in [0, T_N)$ can be computed as an integral over $[a, b]$:

$$\begin{aligned} C(t_m, x) &\approx \int_a^b V(t_{m+1}, y)\, p_X(y; t_m, t_{m+1}, x)\, \mathrm{d}y \\ &\approx \frac{2}{b-a} \sum_{k=0}^{Q-1}{}' \mathcal{P}_k(x, t_m, t_{m+1}) \mathcal{V}_k(t_{m+1}), \end{aligned} \tag{4.5}$$

where the coefficients $\mathcal{V}_k(t_{m+1})$ are defined by

$$\mathcal{V}_k(t_{m+1}) := \int_a^b V(t_{m+1}, y) \cos(u_k(y-a))\, \mathrm{d}y. \tag{4.6}$$

The coefficients $\{\mathcal{V}_k\}_{k=0}^{Q-1}$ can be computed at monitoring dates $\{t_m\}_{m=1}^{\mathrm{M}}$ by the backward recursion, as in (2.11). Analytic formulas in the case of the Hull–White model for the coefficients $\{\mathcal{V}_k\}_{k=0}^{Q-1}$ can be computed by backward recursion.

At the expiry date, $t_M = T_N$, the option value equals the payoff of the underlying swap, ie, $V(t_M, \cdot) = U_N(\cdot)$. We are only interested in the in-the-money region regarding the function U_n, for which we need to solve $U_N(x^*(T_N)) = 0$.

Function U_N is positive on the range $(a, x^*(T_N))$ for a receiver Bermudan swaption, and on the range $(x^*(T_N), b)$ for a payer Bermudan swaption. We compute the integral on the range in which $U_N > 0$ for the coefficients $\mathcal{V}_k(t_M)$. The formulas for the integral are given by

$$\begin{aligned} \mathcal{V}_k(t_M) &= \int_a^b U_N(y) \cos(u_k(y-a))\, \mathrm{d}y \\ &= \begin{cases} \mathcal{G}_k(a, x^*(T_N), T_N) & \text{for a receiver swaption,} \\ \mathcal{G}_k(x^*(T_N), b, T_N) & \text{for a payer swaption.} \end{cases} \end{aligned} \tag{4.7}$$

The coefficients $\mathcal{G}_k$ at time T_n over $[x_1, x_2]$ are computed by

$$\begin{aligned} \mathcal{G}_k(x_1, x_2, T_n) &= N_0 \int_{x_1}^{x_2} \cos(u_k(y-a)) U_N(y)\, \mathrm{d}y \\ &= N_0 \delta(\mathcal{A}_k^1(x_1, x_2) - \mathcal{A}_k^2(x_1, x_2, T_n)), \end{aligned} \tag{4.8}$$

where the coefficients are given by

$$\begin{aligned} &\mathcal{A}_k^1(x_1, x_2) \\ &\quad = \begin{cases} x_2 - x_1, & k = 0, \\ \dfrac{1}{u_k}[\sin(u_k(x_2 - a)) - \sin(u_k(x_1 - a))], & k \neq 0, \end{cases} \end{aligned} \tag{4.9}$$

$$
\begin{aligned}
&\mathcal{A}_k^2(x_1, x_2, T_n) \\
&\quad = \sum_{j=n}^{N} \frac{c_j \bar{A}(T_n, T_{j+1})}{(u_k)^2 + (\bar{B}(T_n, T_{j+1}))^2} \\
&\qquad \times [\exp\{-\bar{B}(T_n, T_{j+1})x_2\} \\
&\qquad\quad \times (u_k \sin(u_k(x_2 - a)) - \bar{B}(T_n, T_{j+1}) \cos(u_k(x_2 - a))) \\
&\qquad\quad - \exp\{-\bar{B}(T_n, T_{j+1})x_1\} \\
&\qquad\quad \times (u_k \sin(u_k(x_1 - a)) - \bar{B}(T_n, T_{j+1}) \cos(u_k(x_1 - a)))]. \qquad (4.10)
\end{aligned}
$$

Here, $c_j = \tau_j K$, $j = n, \ldots, N-1$, $c_N = 1 + \tau_N K$, and $\bar{A}$ and $\bar{B}$ are coefficients associated to the ZCB price, given in Appendix 1 (available online).

In the COS method, computation also takes place in backward fashion. We distinguish an early exercise date from an intermediate date between two exercise times. At an intermediate date, $t_m \in (T_{n-1}, T_n)$, $V(t_m, \cdot) = C(t_m, \cdot)$; thus, the coefficients $\mathcal{V}_k$ at time t_m are given by

$$
\mathcal{V}_k(t_m) = \int_a^b C(t_m, y) \cos(u_k(y - a)) \, \mathrm{d}y = \mathcal{C}_k(a, b, t_m), \qquad (4.11)
$$

where the coefficients $\mathcal{C}_k$ at time t_m over $[x_1, x_2]$ are computed via an integral

$$
\begin{aligned}
\mathcal{C}_k(x_1, x_2, t_m) &:= \int_{x_1}^{x_2} C(t_m, y) \cos(u_k(y - a)) \, \mathrm{d}y \\
&\approx \frac{2}{b-a} \sum_{j=0}^{Q-1}{}' \left(\int_{x_1}^{x_2} \mathcal{P}_j(y, t_m, t_{m+1}) \cos(u_k(y - a)) \, \mathrm{d}y \right) \mathcal{V}_j(t_{m+1}) \\
&= \frac{2}{b-a} \sum_{j=0}^{Q-1}{}' \mathrm{Re}\{\mathcal{W}_j(t_m, t_{m+1}) \mathcal{X}_{kj}(x_1, x_2, \Delta t_m)\} \mathcal{V}_j(t_{m+1}),
\end{aligned} \qquad (4.12)
$$

in which coefficients $\{V_j(t_{m+1})\}_{j=1}^{Q-1}$ have been computed at time t_{m+1}, and coefficients $\mathcal{W}$ and $\mathcal{X}$ are given by

$$
\begin{aligned}
&\mathcal{W}_j(t_m, t_{m+1}) \\
&\quad = \exp\left\{ - \int_{t_m}^{t_{m+1}} \theta(s) \, \mathrm{d}s + \mathrm{i}u_j \theta(t_{m+1}) \right. \\
&\qquad \left. - \mathrm{i}u_j a - \tilde{B}_g(u_j, \Delta t_m)\theta(t_m) + \tilde{A}_g(u_j, \Delta t_m) \right\}, \qquad (4.13)
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{X}_{kj}(x_1, x_2, \Delta t_m) \\
&\quad = \int_{x_1}^{x_2} \cos(u_k(y-a)) \exp(\tilde{B}_g(u_j, \Delta t_m) y)\, \mathrm{d}y \\
&\quad = \frac{1}{(u_k)^2 + (\tilde{B}_g(u_j, \Delta t_m))^2} \\
&\qquad \times [\exp\{\tilde{B}_g(u_j, \Delta t_m) x_2\} \\
&\qquad\quad \times (u_k \sin(u_k(x_2 - a)) + \cos(u_k(x_2 - a))\tilde{B}_g(u_j, \Delta t_m)) \\
&\qquad\quad - \exp\{\tilde{B}_g(u_j, \Delta t_m) x_1\} \\
&\qquad\quad \times (u_k \sin(u_k(x_1 - a)) + \cos(u_k(x_1 - a))\tilde{B}_g(u_j, \Delta t_m))]. \qquad (4.14)
\end{aligned}
$$

The analytic formulas for the coefficients $\tilde{A}_g$ and $\tilde{B}_g$, and the integral with the function θ in (4.13), are given in Appendix 1 (available online).

At an early exercise date $t_m = T_n$, $n = N - 1, \ldots, 1$, the option value is the maximum of the continuation value and the immediate exercise value; hence, we solve the following equation: $C(T_n, x^*(T_n)) - U_n(x^*(T_n)) = 0$. Solution $x^*(T_n)$ represents the optimal early exercise boundary at time T_n. The equation can be solved by some root-finding algorithm, such as the Newton–Raphson method.

The coefficients $\{\mathcal{V}_k(t_m)\}_{k=0}^{Q-1}$ at time $t_m = T_n$ with the optimal exercise value $x^*(T_n)$ are given by

$$
\begin{aligned}
\mathcal{V}_k(T_n) &= \int_a^b \max(C(T_n, y), U_n(y)) \cos(u_k(y-a))\, \mathrm{d}y \\
&= \begin{cases} \mathcal{G}_k(r^*(T_n), b, T_n) + \mathcal{C}_k(a, r^*(T_n), T_n) & \text{payer}, \\ \mathcal{G}_k(a, r^*(T_n), T_n) + \mathcal{C}_k(r^*(T_n), b, T_n) & \text{receiver}. \end{cases} \qquad (4.15)
\end{aligned}
$$

The computation of the coefficients $\{\mathcal{V}_k\}_{k=0}^{Q-1}$ depends on the early exercise boundary value at each exercise date T_n. The continuation function from the Fourier-cosine expansions in (4.5) converges with respect to the number of Fourier terms Q when the integration interval is chosen properly.

At each t_m, the continuation values for all scenarios can be computed by (4.5). Risk-neutral and real-world exposure profiles are obtained by backward iteration, as in Sections 3.1.1 and 3.1.2. One can employ interpolation to enhance the computational speed for the computation of the continuation values.

5 NUMERICAL EXPERIMENTS

We test the developed algorithms for different test cases under the one-factor Hull–White and two-factor G2++ models.

The notional amount of the underlying swap is set equal to 100. We define a $T_1 \times T_N$ Bermudan swaption as a Bermudan option written on the underlying swap that can be exercised between T_1 and T_N, the first and last early exercise opportunities. The option can be exercised annually after T_1, and the payment of the underlying swap is made by the end of each year until the fixed end date $T_{N+1} = T_N + 1$, ie, the time fraction $\tau_n = 1$.

We take a fixed strike K to be $40\%S$, $100\%S$ and $160\%S$, where S is the swap rate associated with date T_1 and payment dates $\mathcal{T}_1 = \{T_2, \ldots, T_{N+1}\}$ given by (2.9). It is the at-the-money strike of the European swaption that expires at date T_1 associated with payment dates $\mathcal{T}_1$.

5.1 Experiments with the Hull–White model

We generate risk-neutral and real-world scenarios using the Hull–White model presented in Appendix 1 (available online), with risk-neutral parameters λ and η obtained by market prices and real-world parameters κ and η obtained by historical data.

Table 1 reports the time-zero option values, CVA and real-world EPEs and MPFEs of 1Y×5Y and 4Y×10Y receiver Bermudan swaptions by the COS method, SGBM and LSM-bundle and LSM-all algorithms.

For the computation of future exposure distributions, one needs to combine the COS method computations with Monte Carlo scenario generation, so there are standard errors as well for the corresponding CVA, EPE and MPFE values. We present $100 \times$ CVA values instead of CVA to enlarge the differences and standard errors in Table 1.

The reference results by the COS method are obtained with $Q = 100$ cosine terms. In the SGBM algorithm, we use as basis functions $\{1, r, r^2\}$ for the approximation of the continuation values and ten bundles containing an equal number of paths. In the LSM, we choose a cubic function based on $\{1, r, r^2, r^3\}$ for the approximation. It is observed that SGBM and LSM converge with respect to the number of basis functions, and, from our experiments, we also find that for longer maturities a larger number of basis functions are required to maintain the accuracy.

As shown in Table 1, the differences in the computed time-zero option values between all algorithms are very small. The LSM-bundle and LSM-all algorithms return the same time-zero option value, as they are based on the same technique to determine the early exercise policy. Compared with the LSM, the SGBM has improved accuracy with smaller variances. The absolute difference in V_0-values between the SGBM and COS method is as small as 10^{-3}, and the standard errors are less than 1%. The largest difference in V_0 between the LSM and COS method is 6×10^{-3}, with a standard error between 1 and 2% in Table 1.

The SGBM is particularly accurate for computing the MPFE values. The results in Table 1 show that the absolute differences for MPFE computed by the SGBM and COS methods are less than 0.01. The LSM-all algorithm does not result in satisfactory results for the exposure values. MPFE is overestimated, while EPE is underestimated. The LSM-bundle algorithm, however, shows significant improvements with smaller errors.

For the computation of EPE and MPFE, the results obtained via these algorithms have a similar standard error. This shows that the dominating factor in the EPE and MPFE variances is connected to the number of generated scenarios.

Figure 3 compares the statistics of the risk-neutral and real-world exposure distributions: the mean in Figure 3(a) and the 99% quantile in Figure 3(b) for a 4Y/10Y receiver Bermudan swaption along time horizon $[0, 10]$. The significant difference between the curves shows that one cannot use quantiles computed by risk-neutral exposure distributions to represent the real-world PFE. There are downward jumps in the EE and PFE curves at each early exercise date $\{4Y, 5Y, \dots\}$ as the swaption on some of the paths is exercised.

The mean and 99% percentile of the real-world exposure are the required EE and PFE values. Figure 4 compares the EE and PFE curves obtained by the different algorithms for the periods 2Y–4Y and 6Y–8Y. The LSM tends to overestimate the PFE prior to the first early exercise opportunity and underestimate it afterwards. The SGBM results are as accurate as the reference values.

The main reason for the SGBM's excellent fit in the tails of the distributions is that, at each date, the algorithm provides an accurate local approximation of the continuation function for the whole realized domain of the underlying factor.

Figure 5(a) compares the reference continuation functions (by COS) with the approximated continuation functions (by the SGBM and LSM) on the bounded realized risk-neutral region at time 6.5Y, for the 4Y/10Y receiver swaption. The approximation by LSM-all is not accurate at the upper and lower regions, which explains its performance in Table 1. We observe an accuracy improvement in the results from the LSM-bundle algorithm. From the plot, we observe that the SGBM's approximated function well resembles the reference value on the whole domain. Figure 5(b) presents the empirical density of the risk-neutral short rate and the observed real-world short rate, where we see that the realized domain under the risk-neutral measure is more widely spread.

Table 2 gives the computational times for these algorithms. The SGBM is significantly faster than the reference COS algorithm, while the LSM is less accurate but faster than the SGBM. The experiments are performed on a computer with a CPU Intel Core i7-2600 3.40GHz $\times$ 8 processor and 15.6 Gigabytes of RAM. The computational cost increases proportionally with respect to parameter M.

TABLE 1 Bermudan receiver swaption under the Hull–White model.

(a) 1Y×5Y

K/S	Value	COS	SGBM	LSM-bundle	LSM-all
40%	V_0	4.126	4.127(0.00)	4.126(0.01)	4.126(0.01)
	MPFE	9.125(0.06)	9.118(0.06)	9.039(0.05)	8.8(0.05)
	EPE	1.704(0.00)	1.705(0.00)	1.708(0.01)	1.806(0.01)
	100CVA	15.87(0.01)	15.74(0.01)	15.75(0.08)	15.87(0.08)
100%	V_0	5.463	5.464(0.00)	5.461(0.01)	5.461(0.01)
	MPFE	11.07(0.05)	11.07(0.05)	11.06(0.05)	10.9(0.04)
	EPE	2.094(0.00)	2.096(0.00)	2.098(0.00)	2.215(0.00)
	100CVA	18.56(0.02)	18.33(0.02)	18.35(0.04)	18.44(0.04)
160%	V_0	7.11	7.11(0.00)	7.113(0.01)	7.113(0.01)
	MPFE	14.43(0.04)	14.42(0.04)	14.26(0.04)	13.92(0.04)
	EPE	2.368(0.00)	2.369(0.00)	2.372(0.01)	2.483(0.01)
	100CVA	21.28(0.02)	20.95(0.02)	20.98(0.05)	21.01(0.05)

(b) 4Y×10Y

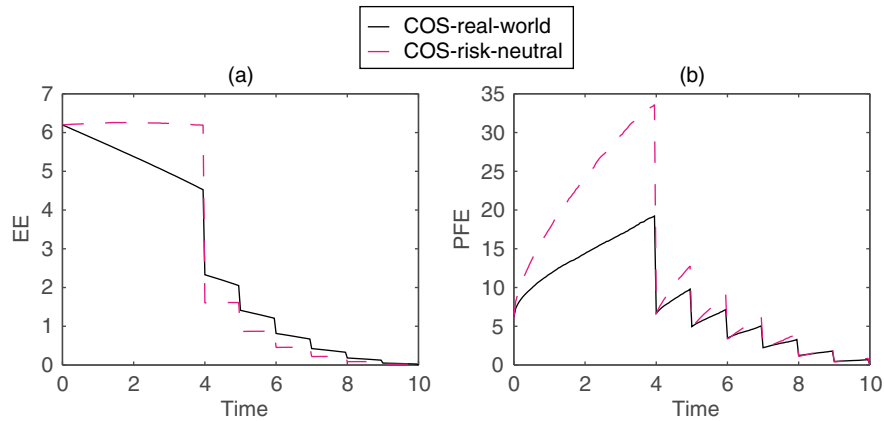
K/S	Value	COS	SGBM	LSM-bundle	LSM-all
40%	V_0	4.235	4.236(0.00)	4.237(0.01)	4.237(0.01)
	MPFE	14.12(0.12)	14.13(0.12)	13.86(0.11)	13.16(0.09)
	EPE	1.827(0.00)	1.829(0.00)	1.834(0.01)	1.91(0.01)
	100CVA	38.22(0.02)	37.98(0.02)	38.04(0.13)	38.34(0.13)
100%	V_0	6.199	6.199(0.00)	6.201(0.02)	6.201(0.02)
	MPFE	19.29(0.11)	19.29(0.11)	19.08(0.12)	18.08(0.10)
	EPE	2.606(0.00)	2.607(0.00)	2.616(0.01)	2.719(0.01)
	100CVA	53.35(0.05)	52.92(0.05)	53.03(0.14)	53.26(0.14)
160%	V_0	8.691	8.691(0.00)	8.687(0.02)	8.687(0.02)
	MPFE	24.33(0.09)	24.34(0.09)	24.28(0.09)	23.42(0.09)
	EPE	3.526(0.00)	3.527(0.00)	3.539(0.01)	3.628(0.01)
	100CVA	71.94(0.06)	71.35(0.06)	71.47(0.09)	71.5(0.10)

(a) $S \approx 0.0109$; risk-neutral: $\sigma = 0.010$, $\lambda = 0.020$; real-world: $\eta = 0.010$, $\kappa = 0.015$. (b) $S \approx 0.0113$; risk-neutral: $\sigma = 0.020$, $\lambda = 0.012$; real-world: $\eta = 0.006$, $\kappa = 0.008$. Risk-neutral and real-world scenarios are generated; the forward rate is flat, $f^{\mathrm{M}}(0,t) = 0.01$; the default probability function $\mathrm{PS}(t) = 1 - \exp(-0.02t)$ and LGD $= 1$; option values and CVA are based on $K_q = 100 \times 10^3$ risk-neutral scenarios; MPFE and EPE are based on $K_a = 100 \times 10^3$ real-world scenarios; the number of monitoring dates $M = T_N/\Delta t$ with $\Delta t = 0.05$; standard errors are in parentheses, based on ten independent runs.

5.2 Experiments with the G2++ model

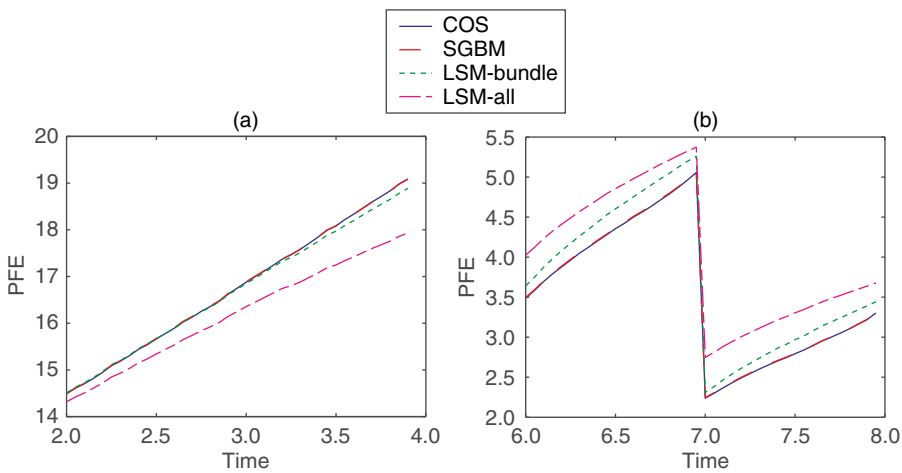
The dynamics of the risk-neutral and real-world G2++ models are given in Appendix 2 (available online), where the associated parameters (ie, the reversion speed κ_1, κ_2; the

FIGURE 3 Comparison of the mean and 99% quantile of the exposure distributions, computed by the COS method and based on risk-neutral and real-world scenarios, for the 4Y/10Y Bermudan receiver swaption, as specified in Table 1, when $K/S = 1$.



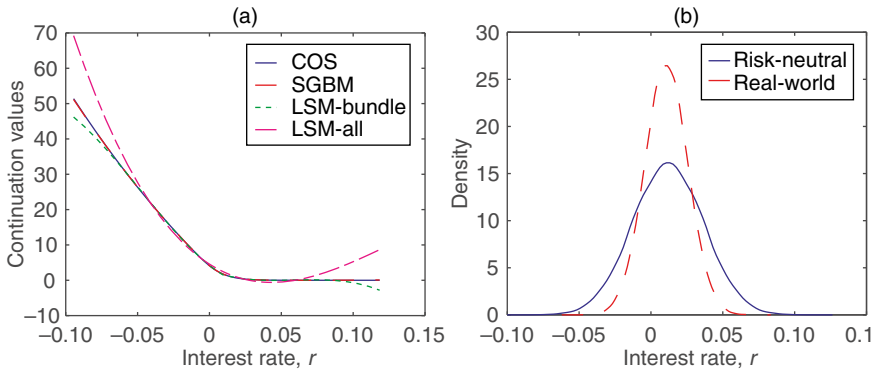
(a) Exposure average. (b) Exposure quantile 99%.

FIGURE 4 Comparison of PFE curves obtained by the COS method, SGBM and LSM for 2Y–4Y and 6Y–8Y, for the 4Y/10Y Bermudan receiver swaption specified in Table 1, when $K/S = 1$.



(a) PFE, 2Y–4Y. (b) PFE, 6Y–8Y.

FIGURE 5 Comparison of continuation functions via all algorithms at 6.5Y, for the 4Y/10Y Bermudan receiver swaption, specified in Table 1, when $K/S = 1$.



(a) Continuation function. (b) Short rate density.

TABLE 2 Computational costs (seconds) for computation of risk-neutral and real-world exposure distributions.

T_N	M	COS (s)	SGBM (s)	LSM-bundle (s)	LSM-all (s)
5Y	100	60.9	4.48	3.32	1.67
10Y	200	122.7	9.02	7.09	3.26

volatility η_1, η_2; and the correlation ξ) are based on historical data, and risk-neutral parameters (λ_1, λ_2, σ_1, σ_2 and ρ) are based on market prices.

In this two-dimensional model, we use the following monomials as the basis functions in the LSM algorithm

$$\{1, x_t, y_t, x_t^2, x_t y_t, y_t^2, x_t^3, x_t^2 y_t, x_t y_t^2, y_t^3\};$$

the basis functions in the SGBM algorithm are given by

$$\{1, x_t, y_t, x_t^2, x_t y_t, y_t^2\},$$

from which we observe that the number of basis functions increases rapidly with regard to the dimension of the underlying variable.

The associated discounted moments, required in the SGBM, can easily be derived from the analytic formula of the dChF of the G2++ model. As for the Hull–White model, we use $J = 10$ bundles in SGBM. In SGBM, we can either use the

TABLE 3 Receiver Bermudan swaption under the G2++ model.

(a) 1Y×5Y

K/S	Value	SGBM	LSM-bundle	Difference
40%	V_0	1.742(0.00)	1.747(0.01)	0.005
	MPFE	5.066(0.02)	5.037(0.17)	−0.029
	EPE	0.771(0.00)	0.773(0.00)	0.002
	100CVA	7.491(0.01)	7.51(0.03)	0.019
100%	V_0	2.897(0.00)	2.900(0.01)	0.003
	MPFE	6.535(0.04)	6.448(0.11)	−0.087
	EPE	1.113(0.00)	1.113(0.01)	0.000
	100CVA	10.05(0.01)	10.07(0.03)	0.02
160%	V_0	4.560(0.00)	4.563(0.01)	0.003
	MPFE	9.652(0.01)	9.601(0.09)	−0.51
	EPE	1.33(0.00)	1.337(0.01)	0.007
	100CVA	12.67(0.01)	12.7(0.04)	0.03

(b) 3Y×10Y

K/S	Value	SGBM	LSM-bundle	Difference
40%	V_0	0.861(0.00)	0.865(0.00)	0.005
	MPFE	2.784(0.03)	2.704(0.04)	−0.08
	EPE	0.446(0.00)	0.447(0.00)	0.001
	100CVA	8.678(0.01)	8.705(0.03)	0.027
100%	V_0	2.466(0.00)	2.475(0.01)	0.008
	MPFE	7.176(0.03)	7.115(0.03)	−0.061
	EPE	1.059(0.00)	1.063(0.00)	0.004
	100CVA	19.53(0.01)	19.61(0.04)	0.08
160%	V_0	5.42(0.00)	5.428(0.00)	0.008
	MPFE	12.1(0.03)	12.22(0.04)	0.012
	EPE	1.839(0.00)	1.846(0.00)	0.007
	100CVA	35.49(0.01)	35.62(0.04)	0.13

(a) $S \approx 0.0104$; risk-neutral: $\sigma_1 = 0.015$, $\sigma_2 = 0.008$, $\lambda_1 = 0.07$, $\lambda_2 = 0.08$, $\rho = -0.6$; real-world: $\eta_1 = 0.005$, $\eta_2 = 0.01$, $\kappa_1 = 0.54$, $\kappa_2 = 0.07$, $\xi = -0.8$. (b) $S \approx 0.0102$; risk-neutral: $\sigma_1 = 0.005$, $\sigma_2 = 0.008$, $\lambda_1 = 0.09$, $\lambda_2 = 0.15$, $\rho = -0.6$; real-world: $\eta_1 = 0.002$, $\eta_2 = 0.006$, $\kappa_1 = 0.04$, $\kappa_2 = 0.07$, $\xi = -0.8$. Risk-neutral and real-world scenarios are generated; forward rate $f^{\mathrm{M}}(0,t) = 0.01$; the default probability function $\mathrm{PS}(t) = 1-\exp(-0.02t)$ and LGD = 1; option values and CVA are based on $K_q = 100 \times 10^3$ risk-neutral scenarios; MPFE and EPE are based on $K_a = 100 \times 10^3$ real-world scenarios; the number of monitoring dates $M = T_N/\Delta t$ with $\Delta t = 0.05$.

two-dimensional equal-number bundling method, introduced in Feng and Oosterlee (2014), or the one-dimensional version based on projecting the high-dimensional variable onto a one-dimensional variable. Here, we create the bundles based on the realized values of $(x_t + y_t)$ on each path at time t_m.

FIGURE 6 PFE and the 99% quantile of the exposure distributions of a receiver Bermudan swaption, as specified in Table 3, when $K/S = 1$.

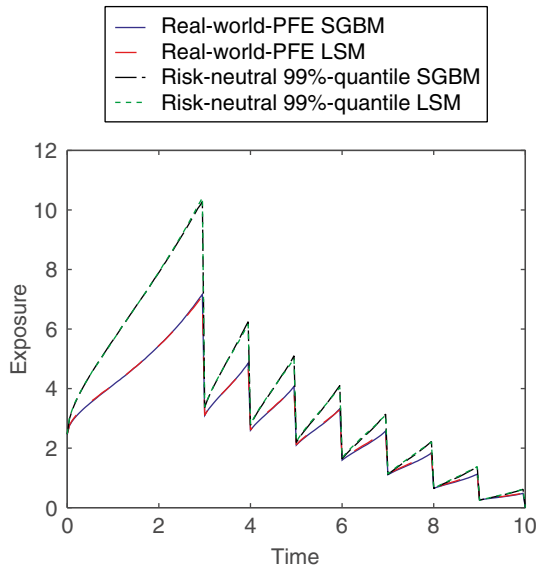


TABLE 4 Computational costs (seconds) for the computation of risk-neutral and real-world exposure distributions under the G2++ model.

T_N	M	LSM-bundle (s)	SGBM (s)
5Y	100	6.28	8.10
10Y	200	12.96	15.07

Table 3 reports the time-zero option value results for SGBM and LSM as well as the exposure measures for receiver Bermudan swaptions, where we can analyze the difference between the results by these algorithms.

Figure 6 presents the PFE curves computed on the real-world scenarios as well as the mean and 99% quantiles of the risk-neutral exposure distributions at each monitoring date. As expected, there is a clear difference between the statistics of the risk-neutral and real-world exposure distributions.

Table 4 presents the computational cost of the algorithms for this two-dimensional model. The cost increases with respect to the dimension of the variable.

6 CONCLUSION

This paper presents computationally efficient techniques for the simultaneous computation of exposure distributions under the risk-neutral and observed real-world probability measures. They are based on only two sets of scenarios, one generated under the risk-neutral dynamics and another under the observed real-world dynamics, as well as on basic techniques such as regression. Compared with nested Monte Carlo simulation, the techniques presented significantly reduce the computational cost and maintain high accuracy, which we demonstrated by using numerical results for Bermudan swaptions and comparing these with reference results generated by the Fourier-based COS method. We illustrated the ease of implementation for both the one-factor Hull–White and two-factor G2++ models.

We recommend the SGBM because of its accuracy and efficiency in the computation of continuation values. A highly satisfactory alternative is to use the LSM-bundle approach. The reference COS method is highly efficient for computing time-zero values of the Bermudan swaption, but for the computation of exposure, there is room for improvement in terms of computational speed.

The results for the parameter values chosen show that there are clear differences in exposure distributions for the risk-neutral and real-world scenarios. The proposed algorithms are based on the requirement that the sample space induced by the observed historical model is a subspace of the sample space under the risk-neutral measure.

The valuation framework presented is flexible and may be used efficiently for any type of Bermudan-style claim, such as Bermudan options and swaptions. For a Bermudan option, one can compute the sensitivities of CVA at the same time as using the SGBM, which is an additional benefit. The algorithms developed can be extended easily to the situation in which model parameters are piecewise constant over the time horizon.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the contents of the paper. The views expressed in this paper are those of the authors and do not necessarily reflect the position of their employers. Financial support from the Dutch Technology Foundation STW (project 12214) is gratefully acknowledged.

REFERENCES

Andersen, L. B. (1999). A simple approach to the pricing of Bermudan swaptions in the multifactor Libor market model. *The Journal of Computational Finance* **3**(2), 5–32 (http://doi.org/bkn9).

Andersen, L. B., and Piterbarg, V. V. (2010). *Interest Rate Modeling*. Atlantic Financial Press.

Basel Committee on Banking Supervision (2005). Annex 4 to "International convergence of capital measurement and capital standards: a revised framework". Report, Bank for International Settlements.

Basel Committee on Banking Supervision (2010). Basel III: a global regulatory framework for more resilient banks and banking systems. Report, Bank for International Settlements.

Brigo, D., and Mercurio, F. (2007). *Interest Rate Models – Theory and Practice: With Smile, Inflation and Credit*. Springer Science & Business Media.

Fang, F., and Oosterlee, C. W. (2009). Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik* **114**(1), 27–62 (http://doi.org/b2vhnx).

Feng, Q., and Oosterlee, C. W. (2014). Monte Carlo calculation of exposure profiles and Greeks for Bermudan and barrier options under the Heston Hull–White Model. SSRN Working Paper, arXiv:1412.3623.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*, Volume 53. Springer Science & Business Media (http://doi.org/bf7w).

Gregory, J. (2010). *Counterparty Credit Risk: The New Challenge for Global Financial Markets*, Volume 470. Wiley.

Hull, J. C., Sokol, A., and White, A. (2014). Modeling the short rate: the real and risk-neutral worlds. Working Paper 2403067, Rotman School of Management (http://doi.org/bkpb).

Jain, S., and Oosterlee, C. W. (2012). Pricing high-dimensional Bermudan options using the stochastic grid method. *International Journal of Computer Mathematics* **89**(9), 1186–1211 (http://doi.org/bkpc).

Jain, S., and Oosterlee, C. W. (2015). The stochastic grid bundling method: efficient pricing of Bermudan options and their Greeks. *Applied Mathematics and Computation* **269**, 412–431 (http://doi.org/bkpd).

Joshi, M. S., and Kwon, O. K. (2016). Least squares Monte Carlo credit value adjustment with small and unidirectional bias. SSRN Working Paper 2717250 (http://doi.org/bkpf).

Karlsson, P., Jain, S., and Oosterlee, C. W. (2014). Counterparty credit exposures for interest rate derivatives using the stochastic grid bundling method. SSRN Working Paper 2538173 (http://doi.org/bkpg).

Kenyon, C., Green, A. D., and Berrahoui, M. (2015). Which measure for PFE? The risk appetite measure A. SSRN Working Paper, December 15, arXiv:1512.06247.

Leitao, Á., and Oosterlee, C. W. (2015). GPU acceleration of the stochastic grid bundling method for early-exercise options. *International Journal of Computer Mathematics* **92**(12), 2433–2454 (http://doi.org/bkph).

Longstaff, F. A., and Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies* **14**(1), 113–147 (http://doi.org/b38b5q).

Øksendal, B. (2003). *Stochastic Differential Equations*. Springer (http://doi.org/dqpdqb).

Ruijter, M. J., and Oosterlee, C. W. (2012). Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing* **34**(5), B642–B671 (http://doi.org/bkpj).

Ruiz, I. (2012). Backtesting counterparty risk: how good is your model? Technical Report, iRuiz Consulting.

Stein, H. J. (2013). Joining risks and rewards. SSRN Working Paper 2368905.

Stein, H. J. (2014). Fixing underexposed snapshots: proper computation of credit exposures under the real world and risk neutral measures. SSRN Working Paper 2365540 (http://doi.org/bkpk).

Zhu, S. H., and Pykhtin, M. (2007). A guide to modeling counterparty credit risk. *GARP Risk Review* July/August(37), 16–22.