Journal of Computational and Applied Mathematics 224 (2009) 668–678

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Exponential time integration for fast finite element solutions of some financial engineering problems

N. Rambeerich, D.Y. Tangman, A. Gopaul, M. Bhuruth *

Department of Mathematics, University of Mauritius, Reduit, Mauritius

ARTICLE INFO

Article history:
Received 4 March 2008
Received in revised form 8 May 2008

MSC:
35A35
65M60

Keywords:
Partial integro-differential equation
Finite element discretisations
Exponential time integration
Jump-diffusion model

ABSTRACT

We consider exponential time integration schemes for fast numerical pricing of European, American, barrier and butterfly options when the stock price follows a dynamics described by a jump-diffusion process. The resulting pricing equation which is in the form of a partial integro-differential equation is approximated in space using finite elements. Our methods require the computation of a single matrix exponential and we demonstrate using a wide range of numerical tests that the combination of exponential integrators and finite element discretisations with quadratic basis functions leads to highly accurate algorithms for cases when the jump magnitude is Gaussian. Comparison with other time-stepping methods are also carried out to illustrate the effectiveness of our methods.

1. Introduction

In the numerical pricing of financial options, the time-stepping algorithm which has been most often employed is the Crank–Nicolson scheme. However it is now well-known that for the Black–Scholes equation [3] for pricing European vanilla options, although the scheme yields second-order accurate option prices, poor accuracy is achieved for the two hedging parameters, delta and gamma, which are respectively the first and second derivatives of the option price with respect to the stock price. For a digital call option which has a Heaviside function as payoff, it has been observed that no convergence results even in the option price when the grid is refined for fixed mesh size ratio. This behaviour of the scheme has been analyzed by Giles and Carter [11] and it has been shown that by replacing the first two time-steps of the Crank–Nicolson scheme by four half-time-steps of the Backward Euler scheme, second-order convergence for delta and gamma can be obtained.

In a recent paper [21], we introduced an exponential time integration (ETI) scheme in combination with central space discretisations for numerical financial option pricing. We showed how to incorporate time-dependent boundaries in the time-marching scheme and our numerical results indicated unconditional second-order convergence rates for European, barrier and butterfly options under the Black–Scholes model.

Exponential time integration has gained importance following the work of Cox and Matthews [7] and with recent developments in efficient methods for computing the matrix exponential [9,20], this time evolution method is likely to be a popular choice for solving large semi-discrete systems arising in various numerical computations.

Our work here concerns the application of exponential time integration in a finite element method-of-lines for pricing various financial derivatives when the stock price process follows a dynamics described by Merton's jump-diffusion model [16]. The pricing equation for single-factor options is a partial integro-differential equation (PIDE) where the jump

* Corresponding author.
E-mail addresses: nisha.rambeerich@umail.uom.ac.mu (N. Rambeerich), desire.tangman1@umail.uom.ac.mu (D.Y. Tangman), a.gopaul@uom.ac.mu (A. Gopaul), mbhuruth@uom.ac.mu (M. Bhuruth).

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.05.047

term is to be integrated over $\mathbb{R}$. For this jump-diffusion PIDE, a second-order BDF scheme is described in [1] and an implicit-explicit (IMEX) scheme is used in [4]. A more recent work of Feng and Linetsky [10] proposes an extrapolation approach in combination with the first-order accurate IMEX-Euler scheme and the numerical results indicate that the extrapolation method improves significantly over the first-order IMEX-Euler scheme in solving the jump-diffusion PIDE.

The present work considers a finite element method with quadratic basis functions and our aim is to show that a higher-order finite element discretisation in combination with an exponential time integration scheme leads to a fast and highly accurate algorithm for pricing single-factor options. Our choice of a finite element discretisation for the pricing equation is mainly motivated by the finite element discretisation with linear elements employed in [10] as we aim to compare the performance of our technique with that of the extrapolation approach. We also note that for most financial options, the initial data which is the payoff function has discontinuities in its first and second derivatives and consequently this imposes less restriction on the finite element method since we deal with the variational form whereas discontinuities of the payoff function have been observed to affect the convergence of high-order finite difference approximations [22] and a grid-stretching technique [23] is necessary to recover the expected high-order convergence rate.

Although European options can be priced in a much faster way by the fast Fourier Transform (FFT) method of Carr and Madan [6], it is common practice in the numerical option pricing literature to use the European option as a basis for assessing the computational efficiency of a new method and then apply the technique for more complex options such as the path-dependent barrier option or the American option. We adopt such an approach here and we show that for European options, the proposed technique compares favourably in computational time with both the second-order BDF and Crank–Nicolson schemes.

We then show that more complex options can also be effectively priced and we give numerical evidence in the case of American, barrier and butterfly spread options. An advantage of using a higher-order finite element method is that a specified accuracy can be reached with fewer finite elements. For illustration, our results show that for pricing an American put option, we achieve an accuracy of the order of 4.3×10^{-5} with 160 quadratic elements in 7.748 s whereas with linear elements we obtain an accuracy of only $9.9 \times 10^{-4} \approx 10^{-3}$ in 10.1 s and 640 elements. We also provide numerical evidence that for specified accuracy levels, faster convergence is obtained with quadratic basis functions and exponential time integration than with the extrapolation approach described in [10].

The structure of this paper is as follows: In Section 2, we describe option pricing under jump-diffusion models and in Section 3 we consider spatial approximations for the pricing equation using finite elements. Exponential time integration for the semi-discrete systems for the different options are described in Section 4. Numerical results are presented in Section 5 where we also give comparisons with the extrapolation approach.

2. The option pricing model

We let the filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ define a financial market model consisting of two assets, a bond (risk-free asset) with price process $\{B_t\}_{t\geq 0}$ and a stock (risky asset) with price process $\{S_t\}_{t\geq 0}$, where B_t and S_t denote the respective prices for the bond and the stock at time t. A European call option (respectively put option) on the risky asset is a financial contract that gives its holder the right, but not the obligation to buy (respectively to sell) the asset for a specified price K, called the strike price at the time-to-maturity, T. In contrast, an American option is exercisable at any time prior to or at maturity.

The price of the bond is given by the solution of the ordinary differential equation

$$\mathrm{d}B_t = rB_t\,\mathrm{d}t,$$

with r being the risk-free rate of interest and the price S_t satisfies the stochastic differential equation

$$\frac{\mathrm{d}S_t}{S_t} = (r - d - \lambda\kappa)\mathrm{d}t + \sigma\mathrm{d}W_t + (\eta - 1)\mathrm{d}N_t,$$

where $\{W_t\}_{t\geq 0}$ is a standard Brownian motion under the equivalent martingale measure (EMM) $\mathbb{Q}$ such that the discounted stock price process $\left\{\mathrm{e}^{-rt}S_t\right\}_{t\geq 0}$ is a $\mathbb{Q}$-martingale, σ is the volatility, $\{N_t\}_{t\geq 0}$ is a Poisson process with intensity λ and is independent of $\{W_t\}_{t\geq 0}$, $\eta - 1$ represents the impulse function for a jump from S to ηS and its expectation, $E[\eta - 1]$, is denoted by κ. The notation d denotes the dividend yield.

The value for a European option with payoff function $\Psi(S_T)$ is its discounted expected payoff under the EMM $\mathbb{Q}$. Using the log transformation $x = \ln(S/K)$ and letting $\tau = T - t$, computation of the option value requires solving the PIDE

$$u_\tau = \mathcal{L}u = \frac{1}{2}\sigma^2 u_{xx} + \left(r - d - \frac{1}{2}\sigma^2\right)u_x - ru + \int_{\mathbb{R}} \left[u(x+y,\tau) - u(x,\tau) - \left(\mathrm{e}^y - 1\right)u_x\right] g(y)\,\mathrm{d}y, \tag{1}$$

for $(x, \tau) \in \mathbb{R} \times (0, T]$ with initial condition $u(x, 0) = \psi(x) = \Psi(K\mathrm{e}^x)$. Under Merton's model [16], the jump sizes are normally distributed with density function $g(y)$ given by

$$g(y) = \frac{\lambda}{\sqrt{2\pi}\sigma_J} \exp\left[-\frac{1}{2}\left(\frac{y - \mu_J}{\sigma_J}\right)^2\right].$$

The American option pricing problem is posed as a linear complementarity problem (LCP) of the form

$$\begin{aligned}
&u_\tau - \mathcal{L}u(x,\tau) \geq 0,\\
&u(x,\tau) - u(x,0) \geq 0, \quad u(x,0) = \psi(x),\\
&(u_\tau - \mathcal{L}u(x,\tau))\,(u(x,\tau) - u(x,0)) = 0, \quad (x,\tau) \in \mathbb{R}\times(0,T].
\end{aligned} \tag{2}$$

The boundary conditions for a European put option are set as

$$u(x,\tau) = K\mathrm{e}^{-r\tau} - K\mathrm{e}^{x-d\tau} \quad \text{as } x \to -\infty; \qquad u(x,\tau) = 0 \quad \text{as } x \to \infty.$$

In the case of an American put option, we use

$$u(x,\tau) = K - K\mathrm{e}^{x} \quad \text{as } x \to -\infty; \qquad u(x,\tau) = 0 \quad \text{as } x \to \infty,$$

and

$$u(x,\tau) = 0 \quad \text{as } x \to -\infty; \qquad u(x,\tau) = K\mathrm{e}^{x} - K \quad \text{as } x \to \infty,$$

for an American call option.

For European options, we consider the case when $d = 0$. Following [1,15], it is common to put $r = 0$ in (1) since the solution when $r > 0$ can be obtained using $\mathrm{e}^{-r\tau}u(x + r\tau, \tau)$, where $u(x,\tau)$ is the solution of (1) for $r = 0$. Hence, we homogenize the PIDE (1) with $r = 0$ by using the transformation $\bar{u} = u - \psi(x)$. This leads to solving for an excess to payoff value in the equation below

$$\bar{u}_\tau - \frac{1}{2}\sigma^2\bar{u}_{xx} + \left(\frac{1}{2}\sigma^2 + \lambda\kappa\right)\bar{u}_x + \lambda\bar{u} - \int_{\mathbb{R}} \bar{u}(x+y,\tau)g(y)\,\mathrm{d}y = f(x), \tag{3}$$

with initial condition $\bar{u}(x,0) = 0$ and boundary conditions: $\lim_{|x|\to\infty}\bar{u}(x,\tau) = 0$. The expression for $f(x)$ in (3) under Merton's model with payoff $\psi(x) = \max(K - K\mathrm{e}^x, 0) = K(1-\mathrm{e}^x)^+$ is given by

$$f(x) = \frac{1}{2}\sigma^2 K\delta_0 + [\lambda(1+\kappa)K\mathrm{e}^x - \lambda K]\mathbf{1}_{x\leq 0} + \lambda K\Phi\left(\frac{-x-\mu_J}{\sigma_J}\right) - \lambda K\mathrm{e}^{x+\mu_J+\sigma_J^2/2}\Phi\left(\frac{-x-\mu_J-\sigma_J^2}{\sigma_J}\right), \tag{4}$$

and for $\psi(x) = K(\mathrm{e}^x - 1)^+$, we have

$$f(x) = \frac{1}{2}\sigma^2 K\delta_0 - [\lambda(1+\kappa)K\mathrm{e}^x - \lambda K]\mathbf{1}_{x\geq 0} + \lambda K\mathrm{e}^{x+\mu_J+\sigma_J^2/2}\Phi\left(\frac{x+\mu_J+\sigma_J^2}{\sigma_J}\right) - \lambda K\Phi\left(\frac{x+\mu_J}{\sigma_J}\right),$$

where δ_a denotes the Dirac delta concentrated at point a and $\Phi(z)$ in (4) is the cumulative normal distribution defined by

$$\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} \mathrm{e}^{-\varrho^2/2}\mathrm{d}\varrho.$$

3. Galerkin finite element spatial discretisation

In this section, we describe the application of the finite element method to the option pricing problem. The infinite domain $\mathbb{R}$ is first truncated to a finite interval $\Omega = (x_{\min}, x_{\max})$. We divide Ω into M sub-domains and construct a finite element mesh with nodes denoted by x_i and constant space-step $h = (x_{\max} - x_{\min})/M$.

The variational form of (3) is established by multiplying (3) by a weight function $\vartheta(x) \in L^2(\Omega)$ (the space of square integrable functions on Ω) and then integrating over the domain Ω. Within the finite element mesh, we denote a typical element by Ω_e and consider the weak form given below that is written in terms of the sum of integrals over Ω_e,

$$\begin{aligned}
&\sum_{e=1}^{M}\left[\int_{\Omega_e}\vartheta\bar{u}_\tau\,\mathrm{d}x + \int_{\Omega_e}\left(\frac{1}{2}\sigma^2\vartheta_x\bar{u}_x + \left(\frac{1}{2}\sigma^2 + \lambda\kappa\right)\vartheta\bar{u}_x + \lambda\vartheta\bar{u}\right)\mathrm{d}x - \int_{\Omega_e}\vartheta\left(\int_{\mathbb{R}}\bar{u}(x+y,\tau)g(y)\,\mathrm{d}y\right)\mathrm{d}x\right]\\
&\quad = \sum_{e=1}^{M}\int_{\Omega_e}\vartheta f(x)\,\mathrm{d}x,
\end{aligned} \tag{5}$$

with initial condition given by

$$\sum_{e=1}^{M}\int_{\Omega_e}\vartheta\bar{u}(x,0)\,\mathrm{d}x = 0.$$

Within each element Ω_e, we replace $\bar{u}$ in (5) using the summation

$$\bar{u}(x,\tau) = \sum_{j=1}^{n} \bar{u}_j^{(e)}(\tau)\varphi_j^{(e)}(x),$$

where n is the number of nodes x_j in Ω_e, $\bar{u}_j^{(e)}(\tau)$ is the value for $\bar{u}$ at time τ corresponding to a point x_j of the element and $\varphi_j^{(e)}(x)$ represent the element shape functions that are piecewise continuous [14,19] and are derived via the Lagrange interpolation formula. In this paper, we consider both linear and quadratic shape functions and the weight functions are chosen similarly as the shape functions, which is typical of the Galerkin method.

For the jump integral term in (5), we have

$$\begin{aligned}\sum_{e=1}^{M}\int_{\Omega_e}\vartheta\left(\int_{\mathbb{R}}\bar{u}(x+y,\tau)g(y)\,\mathrm{d}y\right)\mathrm{d}x &= \sum_{e=1}^{M}\int_{\Omega_e}\vartheta\left(\int_{\mathbb{R}}\bar{u}(z,\tau)g(z-x)\,\mathrm{d}z\right)\mathrm{d}x,\\ &\approx \sum_{e=1}^{M}\int_{\Omega_e}\vartheta\left(\sum_{e=1}^{M}\int_{\Omega_e}\bar{u}(z,\tau)g(z-x)\,\mathrm{d}z\right)\mathrm{d}x. \end{aligned} \tag{6}$$

The integrals in (6) are approximated by applying Newton–Cotes quadrature successively, first in the z-direction and then in the x-direction. For 2-node linear elements, the trapezoidal rule is applied to

$$J_{\mathrm{Trap}_x} = \sum_{e=1}^{M}\int_{\Omega_e}\left[\varphi_1^{(e)}(x),\varphi_2^{(e)}(x)\right]^{\mathrm{T}} J_{\mathrm{Trap}_z}\,\mathrm{d}x,$$

where

$$\begin{aligned} J_{\mathrm{Trap}_z} &\approx \sum_{e=1}^{M}\int_{\Omega_e}\sum_{j=1}^{2}\bar{u}_j^{(e)}(\tau)\varphi_j^{(e)}(z)g(z-x)\,\mathrm{d}z,\\ &= \frac{h}{2}\left[\bar{u}_0(\tau)g(x_0-x) + 2\sum_{j=1}^{M-1}\bar{u}_j(\tau)g(x_j-x) + \bar{u}_M(\tau)g(x_M-x)\right]. \end{aligned}$$

Similarly, for 3-node quadratic elements, Simpson's rule gives

$$\begin{aligned} J_{\mathrm{Simp}_z} &\approx \sum_{e=1}^{M}\int_{\Omega_e}\sum_{j=1}^{3}\bar{u}_j^{(e)}(\tau)\varphi_j^{(e)}(z)g(z-x)\,\mathrm{d}z,\\ &= \frac{h}{6}\left[\bar{u}_0(\tau)g(x_0-x) + 4\sum_{j=1}^{M}\bar{u}_{2j-1}(\tau)g(x_{2j-1}-x) + 2\sum_{j=1}^{M-1}\bar{u}_{2j}(\tau)g(x_{2j}-x) + \bar{u}_{2M}(\tau)g(x_{2M}-x)\right], \end{aligned}$$

in the z-direction and in the x-direction, it is applied to the integrals below

$$J_{\mathrm{Simp}_x} = \sum_{e=1}^{M}\int_{\Omega_e}\left[\varphi_1^{(e)}(x),\varphi_2^{(e)}(x),\varphi_3^{(e)}(x)\right]^{\mathrm{T}} J_{\mathrm{Simp}_z}\,\mathrm{d}x.$$

The finite element spatial approximation for (5) leads to a system of ODEs of the form given by

$$\begin{aligned} &\mathbf{M}_1\bar{\mathbf{u}}'(\tau) + (\mathbf{M}-\mathbf{J})\bar{\mathbf{u}}(\tau) = \mathbf{f},\\ &\mathbf{M}_1\bar{\mathbf{u}}(0) = \mathbf{0}, \end{aligned} \tag{7}$$

where $\mathbf{M}_1$ is the mass matrix, $\mathbf{J}$ corresponds to the dense jump matrix, $\mathbf{f}$ is the load vector and $\mathbf{M}$ is the system matrix obtained from the representation given below,

$$\sum_{e=1}^{M}\int_{\Omega_e}\sum_{j=1}^{n}\bar{u}_j^{(e)}(\tau)\left[\frac{1}{2}\sigma^2\frac{d\varphi_i^{(e)}}{\mathrm{d}x}\frac{d\varphi_j^{(e)}}{\mathrm{d}x} + \left(\frac{1}{2}\sigma^2+\lambda\kappa\right)\varphi_i^{(e)}\frac{d\varphi_j^{(e)}}{\mathrm{d}x} + \lambda\varphi_i^{(e)}\varphi_j^{(e)}\right]\mathrm{d}x,$$

for $i = 1, 2, \ldots, n$.

3.1. Barrier and butterfly options

We consider cases for knock-out barrier options that are known to be worthless if the underlying price reaches a fixed barrier level. An up-and-out call option has the same payoff and lower boundary conditions as that of a European call option

except for an upper condition at the barrier x_u in which case $u(x_u, \tau) = 0$. Conversely, a down-and-out put option has the same payoff and upper condition as the European put option while the lower condition at barrier x_l is $u(x_l, \tau) = 0$.

A butterfly call option has three singularities at strike prices K_1, K_2 and $K_3 = (K_1 + K_2)/2$ and its payoff function is given by $\psi(x) = (K_3 e^x - K_1)^+ + (K_3 e^x - K_2)^+ - 2(K_3 e^x - K_3)^+$. For both barrier and butterfly options, we let $\bar{u} = u$ and proceed with the finite element approximations in a similar way as described above for the case of the European option. The option valuation problem reduces to solving $\bar{\mathbf{u}}$ in

$$\mathbf{M}_1 \bar{\mathbf{u}}'(\tau) + (\mathbf{M} - \mathbf{J})\bar{\mathbf{u}}(\tau) = \mathbf{0},$$

with initial condition $\mathbf{M}_1 \bar{\mathbf{u}}(0) = \mathbf{c}$, where vector $\mathbf{c}$ is the discrete analogue for $\mathcal{C}$ given by

$$\mathcal{C} = \sum_{e=1}^{M} \int_{\Omega_e} \vartheta \psi(x)\, \mathrm{d}x.$$

3.2. American options

For the case of an American option, we use an operator splitting technique [12]. The inequalities in (2) are written as equalities as given below

$$u_\tau - \mathcal{L}u(x, \tau) = \Lambda(\tau), \tag{8}$$

with $\Lambda(\tau)$ being an auxiliary term which acts as a penalty term satisfying $\Lambda(\tau) \geq 0$. Additional constraints are then enforced to ensure that the value of the American option is at least the payoff,

$$(u(x, \tau) - \psi(x)) \cdot \Lambda(\tau) = 0,$$
$$u(x, \tau) \geq \psi(x), \quad (x, \tau) \in \mathbb{R} \times (0, T].$$

Eq. (8) is transformed using $\bar{u} = u - \psi(x)$, the excess to payoff value and the resulting equation is discretised using the Galerkin finite element method. In this case, the semi-discrete system to be time integrated has the form

$$\mathbf{M}_1 \bar{\mathbf{u}}'(\tau) + (\mathbf{M} - \mathbf{J})\bar{\mathbf{u}}(\tau) = \mathbf{f} + \mathbf{h} \cdot \Lambda(\tau),$$

where matrix $\mathbf{M}$ and vector $\mathbf{f}$ now also involve the rate of interest r, the dividend yield d and vector $\mathbf{h}$ is the discrete analogue for

$$\mathcal{H} = \sum_{e=1}^{M} \int_{\Omega_e} \vartheta\, \mathrm{d}x.$$

Also, $\mathbf{h} \cdot \Lambda(\tau)$ is the vector obtained by componentwise multiplications of the elements in $\mathbf{h}$ and $\Lambda(\tau)$.

To enforce the boundary conditions $\bar{u}(x, \tau) \to 0$ as $|x| \to \infty$, we strip the first and last rows and columns of the matrices $\mathbf{M}_1$, $\mathbf{M}$ and $\mathbf{J}$. Similarly, for the vectors $\bar{\mathbf{u}}'(\tau)$, $\bar{\mathbf{u}}(\tau)$, $\mathbf{f}$, $\mathbf{h}$ and $\Lambda(\tau)$, the first and last rows are eliminated.

4. Exponential Time Integration (ETI)

In this section, we describe exponential time integration for the semi-discrete systems for the different options. For the European option, the system of ODEs in (7) is rewritten in the form

$$\bar{\mathbf{u}}'(\tau) = \mathbf{A}\bar{\mathbf{u}}(\tau) + \mathbf{b}, \quad 0 \leq \tau \leq T, \tag{9}$$

where matrix $\mathbf{A} = -\mathbf{M}_1^{-1}(\mathbf{M} - \mathbf{J})$ and vector $\mathbf{b} = \mathbf{M}_1^{-1}\mathbf{f}$ is independent of τ and the initial condition is $\bar{\mathbf{u}}(0) = \mathbf{0}$. Integrating (9) on the interval $[0, T]$ leads to the scheme

$$\bar{\mathbf{u}}(T) = \mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{I}\right)\mathbf{b}, \tag{10}$$

where $\mathbf{I}$ is the identity matrix. Note that computation of the price of European option using (10) requires forming the matrix $\mathbf{A}$ and evaluation of $\mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{I}\right)$. Since $\mathbf{M}_1$ is a banded matrix, the computation of $\mathbf{A}$ and the vector $\mathbf{b}$ can be efficiently carried out. There are several methods for computing $\mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{I}\right)$. A recent comparison of different techniques with respect to accuracy, stability, efficiency, memory requirements and ease of implementation is described in [2] and it is shown that a matrix decomposition algorithm [17] is the cheapest in terms of computational time and has accuracy comparable to the explicit evaluation of the formula $\mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{I}\right)$. In our implementation, we have computed this term explicitly using Matlab's expm and inv functions. The option price at time T is then obtained by adding the payoff $\psi(x)$ to the numerical solution $\bar{\mathbf{u}}(T)$ in (10).

For the butterfly and barrier options, the semi-discrete systems are of the form

$$\bar{\mathbf{u}}'(\tau) = \mathbf{A}\bar{\mathbf{u}}(\tau), \quad 0 \leq \tau \leq T,$$

Table 1
Comparison of time-stepping schemes with linear element space discretisation

Elements	ETI			Crank–Nicolson			BDF-2		
	Price	Error	CPU	Price	Error	CPU	Price	Error	CPU
20	14.6696	3.6541 (−1)	0.008	14.6699	3.6508 (−1)	0.026	14.6687	3.6626 (−1)	0.031
40	14.9474	8.7613 (−2)	0.010	14.9474	8.7540 (−2)	0.080	14.9471	8.7859 (−2)	0.093
80	15.0133	2.1716 (−2)	0.027	15.0133	2.1699 (−2)	0.317	15.0132	2.1778 (−2)	0.328
160	15.0296	5.4188 (−3)	0.136	15.0296	5.4169 (−3)	1.134	15.0296	5.4346 (−3)	1.852
320	15.0336	1.3551 (−3)	0.964	15.0336	1.3546 (−3)	7.874	15.0336	1.3595 (−3)	13.113
640	15.0346	3.3980 (−4)	7.343	15.0346	3.3968 (−4)	58.401	15.0346	3.4182 (−4)	98.438
1280	15.0349	8.5980 (−5)	60.900	15.0349	8.5992 (−5)	468.062	15.0349	8.8384 (−5)	764.375
Exact price	15.034989								

The data set is: $K = 100$, $T = 0.5$, $\sigma = 0.3$, $\sigma_J = 0.5$, $\lambda = 1$.

with initial condition $\bar{\mathbf{u}}(0) = \mathbf{M}_1^{-1}\mathbf{c}$. Thus, we get

$$\bar{\mathbf{u}}(T) = \mathrm{e}^{\mathbf{A}T}\bar{\mathbf{u}}(0).$$

For the American option, we need to solve

$$\bar{\mathbf{u}}'(\tau) = \mathbf{A}\bar{\mathbf{u}}(\tau) + \mathbf{b}(\tau),$$

where the vector $\mathbf{b}(\tau) = \mathbf{M}_1^{-1}(\mathbf{f} + \mathbf{h} \cdot \Lambda(\tau))$ with initial condition $\bar{\mathbf{u}}(0) = \mathbf{0}$. We consider a uniform time-step k with $Nk = T$ and denote $\tau^n = nk$ for $n = 0, 1, \ldots, N$. Then for a time-step from τ^n to τ^{n+1}, we first solve the equation

$$\widehat{\mathbf{u}}\left(\tau^{n+1}\right) = \mathrm{e}^{\mathbf{A}k}\bar{\mathbf{u}}\left(\tau^n\right) + \mathbf{A}^{-1}\left(\mathrm{e}^{\mathbf{A}k} - \mathbf{I}\right)\mathbf{b}\left(\tau^n\right).$$

To compute $\bar{\mathbf{u}}\left(\tau^{n+1}\right)$, we have to enforce the constraints

$$\bar{\mathbf{u}}\left(\tau^{n+1}\right) = \max\left(\mathbf{0}, \widehat{\mathbf{u}}\left(\tau^{n+1}\right) - k\Lambda(\tau^n)\right),$$

and

$$\Lambda(\tau^{n+1}) = \Lambda(\tau^n) + \frac{1}{k}\left(\bar{\mathbf{u}}\left(\tau^{n+1}\right) - \widehat{\mathbf{u}}\left(\tau^{n+1}\right)\right),$$

respectively, where $\Lambda(\tau^0) = \mathbf{0}$. The American option value is then given by the formula $\bar{\mathbf{u}}\left(\tau^{N+1}\right) + \psi(x)$.

5. Numerical experiments

In this section, we illustrate the performance of the ETI scheme for pricing European, butterfly, barrier and American options using different test data. The 'Exact Price' for the European options under Merton's model is computed using the analytical formula in [16]. All codes are run using MATLAB®6.1 with 1 GB RAM and 3.00 GHz processor.

5.1. Numerical results for European options—Merton's model

We consider the numerical solutions for a European put option under Merton's model obtained using the ETI, Crank–Nicolson and BDF-2 schemes combined with linear finite element discretisation for the set of parameters: $T = 0.5$, $K = 100$, $\sigma = 0.3$, $\sigma_J = 0.5$, $\lambda = 1$ and truncated domain $x_{\min} = -2$ and $x_{\max} = 2$. The results are given in Table 1 at the point where the spot price $S = K$. For the Crank–Nicolson and BDF-2 schemes, on each refinement, the number of time-steps (N) is taken as twice the number of linear elements used. 'Error' is the difference between the 'Exact Price' and the numerical solutions.

A second-order convergence rate is achieved for all the three schemes. However, we observe that the ETI scheme is about 7 times faster than the Crank–Nicolson scheme and about 12 times faster than the BDF scheme. The poor performances for both the Crank–Nicolson and BDF-2 schemes are due to an iterative technique used to prevent the inversion of a dense matrix resulting from discretisation of the convolution integral term. For the Crank–Nicolson scheme [8], the vector $\bar{\mathbf{u}}\left(\tau^{n+1}\right)$ is obtained by solving

$$\left(\mathbf{M}_1 + \frac{k}{2}\mathbf{M}\right)\bar{\mathbf{u}}\left(\tau^{n+1}\right)^{l+1} = \left(\mathbf{M}_1 - \frac{k}{2}\mathbf{M}\right)\bar{\mathbf{u}}\left(\tau^n\right) + \frac{k}{2}\left(\mathbf{J}\bar{\mathbf{u}}\left(\tau^n\right) + \mathbf{J}\bar{\mathbf{u}}\left(\tau^{n+1}\right)^l\right) + k\mathbf{f}, \tag{11}$$

and for the BDF-2 scheme [1], the vector $\bar{\mathbf{u}}\left(\tau^{n+2}\right)$ is the solution of

$$\left(\frac{3}{2}\mathbf{M}_1 + k\mathbf{M}\right)\bar{\mathbf{u}}\left(\tau^{n+2}\right)^{l+1} = 2\mathbf{M}_1\bar{\mathbf{u}}\left(\tau^{n+1}\right) - \frac{1}{2}\mathbf{M}_1\bar{\mathbf{u}}\left(\tau^n\right) + k\mathbf{J}\bar{\mathbf{u}}\left(\tau^{n+2}\right)^l + k\mathbf{f}, \tag{12}$$

Table 2
Numerical results at spot prices 80, 90, 100, 110, 120 obtained by combining the ETI scheme with linear elements and with quadratic elements for parameters: $K = 100, T = 1, \sigma = 0.25, \sigma_J = 0.3, \lambda = 1$

Elements	$S = 80$		$S = 90$		$S = 100$		$S = 110$		$S = 120$	
	Price	Error	Price	Error	Price	Error	Price	Error	Price	Error
	Linear elements									
20	25.6345504	0.5226	19.7216277	0.2695	14.5606865	0.4590	11.1239878	4.5545 (−2)	7.9865277	0.2920
40	26.0374639	0.1197	19.8665661	0.1245	14.9050497	0.1146	11.0793653	9.0167 (−2)	8.2319154	4.6597 (−2)
80	26.1297519	2.7399 (−2)	19.9627730	2.8323 (−2)	14.9910499	2.8646 (−2)	11.1510605	1.8472 (−2)	8.2751491	3.3636 (−3)
160	26.1492470	7.9038 (−3)	19.9851301	5.9663 (−3)	15.0125354	7.1603 (−3)	11.1666040	2.9286 (−3)	8.2771143	1.3984 (−3)
320	26.1552610	1.8897 (−3)	19.9899007	1.1957 (−3)	15.0179058	1.7900 (−3)	11.1693138	2.1881 (−4)	8.2783890	1.2376 (−4)
640	26.1567089	4.4189 (−4)	19.9906847	4.1173 (−4)	15.0192483	4.4749 (−4)	11.1694052	1.2743 (−4)	8.2783240	1.8869 (−4)
	Quadratic elements									
20	26.1698384	1.2688 (−2)	19.9808484	1.0248 (−2)	15.0273480	7.6522 (−3)	11.1637668	5.7659 (−3)	8.2813312	2.8184 (−3)
40	26.1587448	1.5941 (−3)	19.9922735	1.1770 (−3)	15.0201644	4.6867 (−4)	11.1701538	6.2117 (−4)	8.2783257	1.8706 (−4)
80	26.1571656	1.4885 (−5)	19.9912875	1.9109 (−4)	15.0197250	2.9236 (−5)	11.1695973	6.4651 (−5)	8.2784700	4.2740 (−5)
160	26.1571338	1.6957 (−5)	19.9911250	2.8561 (−5)	15.0196976	1.8250 (−6)	11.1695404	7.7812 (−6)	8.2785172	4.4777 (−6)
320	26.1571474	3.3645 (−6)	19.9910976	1.1954 (−6)	15.0196959	1.1691 (−7)	11.1695331	4.9186 (−7)	8.2785124	3.5180 (−7)
Exact price	26.157150761		19.99109641		15.01969577		11.16953264		8.27851274	

with $\bar{\mathbf{u}}^1$ obtained using the implicit Euler method. The Toeplitz matrix–vector product $\mathbf{J}\bar{\mathbf{u}}$ in both Eqs. (11) and (12) is evaluated using FFT algorithm where each Toeplitz matrix is first embedded in a circulant matrix and then multiplied with a vector. In general, an FFT algorithm involves three operations: two discrete Fourier transforms (DFTs) and an inverse discrete Fourier transform (IDFT). From (11) and (12), we find that for both the Crank–Nicolson and BDF-2 schemes, the number of FFT operations depends on the number of iterations per time-step. By pre-computing the DFT corresponding to the first column vector of the circulant matrix, for both the Crank–Nicolson and the BDF-2 schemes, each iteration involves one DFT and one IDFT. Opposed to these schemes, for valuing the European put option using the ETI scheme, we do not require a Toeplitz matrix–vector product, neither do we need to invert the dense matrix associated with the convolution integral term over several time-steps.

We next consider the numerical option prices at spot prices 80, 90, 100, 110, 120 for a European put option with maturity $T = 1$ and strike price $K = 100$ and parameters: $\sigma = 0.25, \sigma_J = 0.3, \lambda = 1$ and $x_{\min} = -2$ and $x_{\max} = 2$. In Table 2, we present results obtained by the ETI scheme in combination with both linear and quadratic finite elements. To compare the linear solutions to quadratic solutions, we choose equal number of degrees of freedom (unknowns) for both the linear and quadratic finite element meshes. Hence, we examine the results obtained by M linear elements to those obtained using $M/2$ quadratic elements. The aim is to show that fewer quadratic elements produce highly accurate option values.

The numerical option prices displayed in Table 2 are computed as follows: We let $u^T(x)$ denote the time T value for the option at log-spot price $x = \ln(S/K)$. For linear elements with $x_i \le x \le x_{i+1}$, we denote the numerical option value at x_i by u_i^T and at x_{i+1} by u_{i+1}^T. Then

$$u^T(x) = \frac{x_{i+1} - x}{h} u_i^T + \frac{x - x_i}{h} u_{i+1}^T.$$

For quadratic elements with a log-spot price x such that $x_i \le x \le x_{i+2}$ and numerical option values given by u_i^T, u_{i+1}^T and u_{i+2}^T, we have

$$u^T(x) = \frac{2(x - x_{i+1})(x - x_{i+2})}{h^2} u_i^T - \frac{4(x - x_i)(x - x_{i+2})}{h^2} u_{i+1}^T + \frac{2(x - x_i)(x - x_{i+1})}{h^2} u_{i+2}^T.$$

From Table 2, we observe that at $S = K = 100$, an error of order 10^{-4} is reached with only 40 quadratic elements whereas with 80 linear finite elements, the accuracy attained is of order 10^{-2}. We also obtain accuracy of 10^{-7} for the option price with 320 quadratic elements compared to an accuracy of 10^{-4} obtained with 640 linear elements. Similar results are observed for the other spot prices.

We illustrate in Fig. 1 the pricing errors at $S = K = 100$ plotted as a function of number of elements. This clearly demonstrates that quadratic finite elements lead to very precise computed option prices compared to linear ones.

5.2. Numerical results for butterfly and barrier options—Merton's model

We consider the numerical prices for a European butterfly call option, a down-and-out put option and an up-and-out call option and show that the ETI scheme coupled with quadratic elements improves on the results obtained by linear elements. In this test example, we fix the upper boundary x_u for the up-and-out call option at $\ln(195/K)$ and the lower barrier level x_l for the down-and-out put option at $\ln(70/K)$, where $K = 100$. For the butterfly call option, the strike prices are $K_1 = 90$

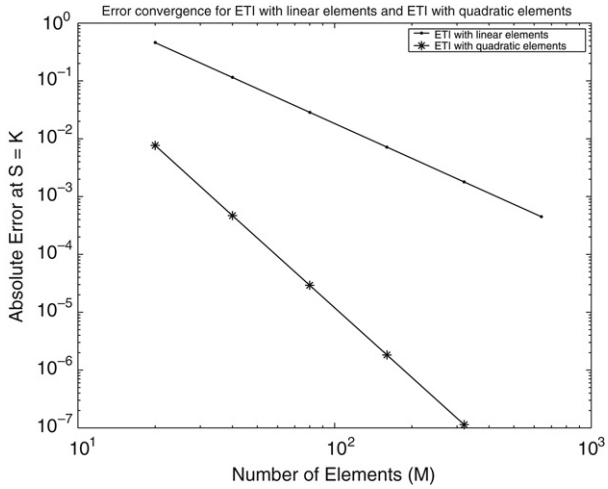

Fig. 1. Pricing errors at the money ($S = K$) European put option for linear elements and quadratic elements.

Table 3
Linear and quadratic solutions for option prices for butterfly call option, down-and-out put option and up-and-out call option by ETI scheme at the point where spot price equals strike for parameters: $T = 1, \sigma = 0.25, \sigma_J = 0.3, \lambda = 1, K_1 = 90, K_2 = 110, K = K_3 = 100$

		Linear elements					
		20	40	80	160	320	640
Butterfly call	Price	1.1627905	1.1297139	1.1250597	1.1239728	1.1237060	1.1236397
	Error	3.9173 (−2)	6.0963 (−3)	1.4419 (−3)	3.5510 (−4)	8.8363 (−5)	2.2046 (−5)
Down-and-out put	Price	3.3325786	3.3670511	3.3759920	3.3788110	3.3798368	3.3802556
	Error		3.4472 (−2)	8.9409 (−3)	2.8189 (−3)	1.0259 (−3)	4.1881 (−4)
Up-and-out call	Price	8.9034291	8.8525332	8.8417891	8.8390365	8.8382644	8.8380264
	Error		5.0896 (−2)	1.0744 (−2)	2.7526 (−3)	7.7217 (−4)	2.3801 (−4)
		Quadratic elements					
		20	40	80	160	320	Exact price
Butterfly call	Price	1.1214080	1.1235328	1.1236124	1.1236173	1.1236176	1.12361767
	Error	2.2097 (−3)	8.4908 (−5)	5.2239 (−6)	3.2373 (−7)	2.3043 (−8)	
Down-and-out put	Price	3.3797159	3.3801706	3.3802992	3.3803276	3.3803326	
	Error		4.5470 (−4)	1.2863 (−4)	2.8379 (−5)	5.0453 (−6)	
Up-and-out call	Price	8.8336616	8.8375976	8.8378758	8.8379012	8.8379048	
	Error		3.9359 (−3)	2.7823 (−4)	2.5365 (−5)	3.6078 (−6)	

and $K_2 = 110$. The set of parameters used for valuing all the three options are: $T = 1, \sigma = 0.25, \sigma_J = 0.3, \lambda = 1$ and $x_{\min} = -2$ and $x_{\max} = 2$. The results are reported in Table 3, at $S = K = 100$.

For the butterfly call option, an accuracy of 10^{-8} is achieved with only 320 quadratic elements. Conversely, 640 linear elements reach an accuracy of only 10^{-5}. We remark that the sharp kink in the butterfly spread payoff function at the point where the spot price equals the strike price $(K_1 + K_2)/2$ does not reduce the accuracy of the computed results. For the knock-out options, we again note that the quadratic elements give better solutions than the linear ones. Since, closed form solutions for barrier options under Merton's jump-diffusion model do not exist, the pricing error for these options is the difference in the option values for two successive levels of refinements [8]:

$$\text{Error} = |u_{2M}(x_K, T) - u_M(x_K, T)|, \tag{13}$$

where M is the number of elements and log-price $x_K = 0$, is the point where spot price equals strike price.

5.3. Numerical results for American options—Merton's model

In Table 4, we list the numerical values for both American call and put options at $S = K = 100$ that have been computed by the ETI scheme with both linear and quadratic elements as well as the computational time. In the case of the American call option, the data set is: $T = 1, \sigma = 0.15, \sigma_J = 0.25, r = 0.04, d = 0.02, \lambda = 1, x_{\min} = -2.2$ and $x_{\max} = 2.2$ and for the American put option, we use: $T = 0.5, \sigma = 0.15, \sigma_J = 0.3, r = 0.03, \lambda = 1, x_{\min} = -1.4$ and $x_{\max} = 1.4$. 'Error' is computed as given by (13). We find that fewer quadratic elements lead to more accurate American option prices. We also observe that the pricing errors corresponding to the quadratic solutions are decreasing at a much faster rate than those corresponding to the linear solutions. In the case of an American put option, we obtain an accuracy of the order of 4.3×10^{-5}

Table 4
Numerical solutions for American call and put options at spot price $S = 100$ obtained using the ETI scheme with linear elements and with quadratic elements

Time-steps	Linear				Quadratic			
	Elements	Price	Error	CPU	Elements	Price	Error	CPU
	American call							
20	20	10.7367427		0.013	10	11.9350075		0.013
40	40	11.3620031	6.2526 (−1)	0.017	20	11.6074310	3.2758 (−1)	0.028
80	80	11.5123909	1.5039 (−1)	0.044	40	11.5648950	4.2536 (−2)	0.116
160	160	11.5496884	3.7298 (−2)	0.202	80	11.5622688	2.6263 (−3)	0.830
320	320	11.5589963	9.3079 (−3)	1.450	160	11.5621078	1.6095 (−4)	7.555
640	640	11.5613221	2.3258 (−3)	9.833	320	11.5620979	9.9083 (−6)	85.516
	American Put							
20	20	7.0175734		0.013	10	7.4723527		0.014
40	40	7.3028070	2.8523 (−1)	0.019	20	7.4046191	6.7734 (−2)	0.028
80	80	7.3674628	6.4656 (−2)	0.050	40	7.3891845	1.5435 (−2)	0.095
160	160	7.3832077	1.5745 (−2)	0.220	80	7.3884078	7.7674 (−4)	0.830
320	320	7.3870462	3.8385 (−3)	1.502	160	7.3883649	4.2883 (−5)	7.748
640	640	7.3880336	9.8735 (−4)	10.089	320	7.3883626	2.3124 (−6)	85.165

Table 5
Numerical solutions for European put option by extrapolated IMEX-Euler scheme and ETI scheme

Elements	Extrap-IMEX-Euler				ETI		
	Time-steps (N)	Price	Error	CPU	Price	Error	CPU
20	15	6.2541278	2.0622 (−1)	0.0063	6.4648043	4.4535 (−3)	0.0125
40	15	6.4098864	5.0464 (−2)	0.0110	6.4605933	2.4247 (−4)	0.0399
80	15	6.4477521	1.2599 (−2)	0.0179	6.4603657	1.4845 (−5)	0.2305
160	15	6.4570612	3.2897 (−3)	0.0469	6.4603518	9.1515 (−7)	1.9602
320	15	6.4593892	9.6162 (−4)	0.1594	6.4603509	6.4836 (−8)	15.3563
640	15	6.4599719	3.7894 (−4)	0.5938			
Exact price					6.46035087		

Parameters used are: $T = 0.5$, $\sigma = 0.15$, $\sigma_J = 0.2$, $\lambda = 1$.

Table 6
Numerical solutions for butterfly call by extrapolated IMEX-Euler scheme and ETI scheme

Elements	Extrap-IMEX-Euler				ETI		
	Time-steps (N)	Price	Error	CPU	Price	Error	CPU
20	45	2.7702015	1.5286 (−2)	0.0117	2.7472332	7.6828 (−3)	0.0141
40	45	2.7539476	9.6834 (−4)	0.0196	2.7545255	3.9050 (−4)	0.0383
80	45	2.7544047	5.1125 (−4)	0.0391	2.7548920	2.3926 (−5)	0.2102
160	45	2.7547682	1.4778 (−4)	0.1078	2.7549145	1.5115 (−6)	1.9289
320	45	2.7548746	4.1357 (−5)	0.3930	2.7549159	9.6857 (−8)	14.9696
640	45	2.7549023	1.3700 (−5)	1.4344			
Exact price					2.75491597		

Parameters used are: $T = 0.5$, $\sigma = 0.15$, $\sigma_J = 0.2$, $\lambda = 1$.

with 160 quadratic elements in 7.748 s whereas with linear elements we obtain an accuracy of only $9.9 \times 10^{-4} \approx 10^{-3}$ in 10.1 s and 640 elements.

5.4. Comparison with extrapolated IMEX-Euler scheme

We give numerical evidence that the ETI scheme combined with quadratic finite elements leads to a faster algorithm with more accurate prices for options under Merton's jump-diffusion model. The PIDE problems for all the numerical examples given in this subsection are solved in the truncated domain $x_{\min} = -1$ and $x_{\max} = 1$.

We first consider a European put option with strike $K = 100$ and a butterfly call option with strike prices $K_1 = 90$ and $K_2 = 110$. The model parameters for pricing both options are: $T = 0.5$, $\sigma = 0.15$, $\sigma_J = 0.2$, $\lambda = 1$. From Table 5, for the European put option, we find that IMEX-Euler with extrapolation attains an error estimate of 10^{-3} in 0.0469 s in 15 time-steps and 160 linear elements. On the other hand, the ETI scheme attains the same error estimate in only 0.0125 s and 20 quadratic elements in just one time-step. We also note significant differences in the CPU timings between the ETI scheme where an accuracy of 10^{-5} is attained in just 0.2305 s while for the extrapolated IMEX-Euler scheme, we have an accuracy of 10^{-4} in 0.5938 s. From Table 6, where we list the numerical values for the butterfly spread call option, similar conclusions as in the case for the European put option are made.

Table 7
Numerical solutions for knock-out barrier options by extrapolated IMEX-Euler scheme and ETI scheme

Elements	Extrap-IMEX-Euler				ETI		
	Time-steps (N)	Price	Error	CPU	Price	Error	CPU
	Down-and-out put						
20	55	4.4378795		0.013	4.2922585		0.010
40	55	4.3365517	1.0133 (−1)	0.020	4.2953266	3.0682 (−3)	0.035
80	66	4.2967339	3.9818 (−2)	0.050	4.2953422	1.5545 (−5)	0.231
160	66	4.2959961	7.3779 (−4)	0.156	4.2953572	1.4997 (−5)	1.980
320	66	4.2956492	3.4696 (−4)	0.535	4.2953601	2.9422 (−6)	15.349
640	66	4.2954810	1.6813 (−4)	1.991			
	Up-and-out call						
20	55	4.2258521		0.012	4.1891728		0.012
40	66	4.2041307	2.1721 (−2)	0.025	4.1910504	1.8776 (−3)	0.035
80	66	4.1971553	6.9754 (−3)	0.052	4.1911972	1.4676 (−4)	0.239
160	66	4.1937072	3.4481 (−3)	0.181	4.1912185	2.1292 (−5)	1.976
320	66	4.1919738	1.7334 (−3)	0.770	4.1912215	3.0427 (−6)	15.344
640	66	4.1913023	6.7150 (−4)	2.998			

Parameters used are: $T = 0.5, \sigma = 0.15, \sigma_J = 0.2, \lambda = 0.1$.

The computational results for both the up-and-out call option with upper barrier $x_u = \ln(140/K)$ and the down-and-out put option with lower barrier $x_l = \ln(70/K)$ are listed in Table 7. The model parameters used to compute the option values are: $T = 0.5, \sigma = 0.15, \sigma_J = 0.2, \lambda = 0.1$. 'Error' is computed using Eq. (13). We again conclude that the ETI scheme with quadratic basis functions outperforms the extrapolated IMEX-Euler method.

6. Conclusion

We have considered a higher-order finite element discretisation in space in combination with an exponential time-marching method for pricing a variety of options under Merton's jump-diffusion model. We showed that by taking the number of quadratic to linear elements in the ratio one to two, higher accuracy and faster convergence are achieved using quadratic elements. The technique proposed is also efficient in pricing options that permit early exercise. In this work, the computation of the matrix function $\mathbf{A}^{-1}\left(e^{\mathbf{A}T} - \mathbf{I}\right)$ was carried out using an explicit computation using Matlab's expm function. We note that further efficiency in our technique can be brought about by approximating the matrix exponential via the Carathéodory–Fejér approximation described in [20].

However, we have observed that the finite element method based on quadratic basis functions does not achieve the expected accuracy level under Kou's model [13]. By treating Kou's model as a special case of the CGMY model [5], in the finite difference context with central space discretisations, the quadrature rule proposed in [24] yields the expected second-order convergence rate [18]. In the finite element context, it is not straightforward to apply such a technique and we are currently investigating this problem for infinite activity Lévy models and the results will be reported in a future work.

Acknowledgements

The authors thank the anonymous referee whose comments improved their work and the presentation of this paper. N. Rambeerich and D.Y. Tangman acknowledge the support of the University of Mauritius for carrying out this research.

References

[1] A. Almendral, C.W. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl. Numer. Math. 53 (2005) 1–18.
[2] H.A. Ashi, L.J. Cummings, P.C. Mathews, Comparison of methods for evaluating functions of a matrix exponential, Appl. Numer. Math. (2008) doi:10.1016/j.apnum.2008.03.039.
[3] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Pol. Econ. 81 (3) (1973) 637–654.
[4] M. Briani, R. Natalini, G. Russo, Implicit-explicit numerical schemes for jump-diffusion processes, Calcolo 44 (2007) 33–57.
[5] P. Carr, H. Geman, D.B. Madan, M. Yor, The fine structure of asset returns: An empirical investigation, J. Bus. 75 (2002) 305–332.
[6] P. Carr, D.B. Madan, Option valuation using the fast Fourier transform, J. Comput. Fin. 2 (4) (1998) 61–73.
[7] S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, J. Comput. Phys. 176 (2002) 430–455.
[8] Y. d'Halluin, P.A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numer. Math. 97 (2) (2004) 321–352.
[9] M. Eiermann, O.G. Ernst, A restarted Krylov subspace method for the evaluation of matrix functions, SIAM J. Numer. Anal. 44 (2006) 2481–2504.
[10] L. Feng, V. Linetsky, Pricing options in jump-diffusion models: An extrapolation approach, Oper. Res. (2007) doi:10.1287/opre.1070.0419.
[11] M.B. Giles, R. Carter, Convergence analysis of Crank–Nicolson and Rannacher time-marching, J. Comput. Fin. 9 (4) (2006) 89–112.
[12] S. Ikonen, J. Toivanen, Operator splitting methods for American option pricing, Appl. Math. Lett. 17 (2004) 809–814.
[13] S.G. Kou, H. Wang, Option pricing under a double exponential jump diffusion model, Manage. Sci. 50 (9) (2004) 1178–1192.
[14] Y.W. Kwon, H. Bang, The Finite Element Method using Matlab, CRC Press, 2000.
[15] A.M. Matache, T.V. Petersdorff, C. Schwab, Fast deterministic pricing of options on Lévy driven assets, M2AN Math. Model. Numer. Anal. 38 (1) (2004) 37–71.
[16] R.C. Merton, Option pricing when underlying stock returns are discontinuous, J. Fin. Econ. 3 (1976) 125–144.
[17] C. Moler, C.V. Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (1) (2003) 3–49.

[18] N. Rambeerich, Numerical pricing of options under jump-diffusion models. Technical report, University of Mauritius, April 2008.
[19] J.N. Reddy, An Introduction to the Finite Element Method, McGraw-Hill, 1985.
[20] T. Schmelzer, L.N. Trefethen, Evaluating matrix functions for exponential integrators via Carathéodory–Fejér approximation and contour integrals, ETNA 29 (2007) 1–18.
[21] D.Y. Tangman, A. Gopaul, M. Bhuruth, Exponential time integration and Chebychev discretisation schemes for fast pricing of options, Appl. Numer. Math. (2007) doi:10.1016/j.apnum.2007.07.005.
[22] D.Y. Tangman, A. Gopaul, M. Bhuruth, Numerical pricing of options using high-order compact finite difference schemes, J. Comput. Appl. Math. (2007) doi:10.1016/j.cam.2007.01.035.
[23] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, Wiley, New York, 2000.
[24] I.R. Wang, J.W.I. Wan, P.A. Forsyth, Robust numerical valuation of European and American options under the CGMY process, J. Comput. Fin. 10 (4) (2007) 31–69.