East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.260316.061016a

Vol. 7, No. 2, pp. 227-247
May 2017

# Finite Volume Method for Pricing European and American Options under Jump-Diffusion Models

Xiao-Ting Gan[1,2], Jun-Feng Yin[1,*] and Yun-Xiang Guo[1]

[1] *School of Mathematical Science, Tongji University, Shanghai 200092, PR China.*
[2] *School of Mathematics and Statistics, Chuxiong Normal University, Chuxiong 675000, Yunnan Province, PR China.*

*Received 26 March 2016; Accepted (in revised version) 6 October 2016.*

**Abstract.** A class of finite volume methods is developed for pricing either European or American options under jump-diffusion models based on a linear finite element space. An easy to implement linear interpolation technique is derived to evaluate the integral term involved, and numerical analyses show that the full discrete system matrices are $M$-matrices. For European option pricing, the resulting dense linear systems are solved by the generalised minimal residual (GMRES) method; while for American options the resulting linear complementarity problems (LCP) are solved using the modulus-based successive overrelaxation (MSOR) method, where the $H_+$-matrix property of the system matrix guarantees convergence. Numerical results are presented to demonstrate the accuracy, efficiency and robustness of these methods.

**AMS subject classifications**: 65M08, 65M12, 65M60, 65N40

**Key words**: Finite volume method, option pricing, jump-diffusion models, linear complementarity problems, GMRES method, modulus-based successive overrelaxation method.

## 1. Introduction

The standard Black-Scholes equation has been successfully and widely used in finance, since the underlying model and related assumptions are simple and elegant [5]. However, empirical findings have shown that the standard Black-Scholes assumption of lognormal stock diffusion with constant volatility is not consistent with market prices. This is often referred to as the volatility skew or smile, and occurs in all major stock markets. In order to address this phenomenon, researchers have made various extensions to the Black-Scholes model. The jump-diffusion model first introduced by Merton allows for jump risks on option prices, and can account for large price changes due to rare events. Unlike in the original Black-Scholes model, the price is then intrinsically no longer a continuous function of time,

*Corresponding author. *Email addresses:* 9xtgan@tongji.edu.cn (X.-T. Gan), yinjf@tongji.edu.cn (J.-F. Yin), 103646@tongji.edu.cn (Y.-X. Guo)

 ©2017 Global-Science Press

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

but implied volatility curves yielded by jump-diffusion models are closer to the volatility smiles observed on markets.

It is challenging to provide numerical methods to solve partial integral differential equations (PIDE) for pricing options under jump-diffusion models, since a non-local integral term is involved [1, 2, 10, 11, 15]. For either European or American options under Kou's jump-diffusion models, Toivanen [13] recently developed an easy-to-implement recursion formula for the integral terms with optimal computational cost. A generalised iterative method for the linear complementarity problem (LCP) with a banded matrix instead of the full matrix has been provided by Salmi *et al.* [12]. Toivanen [14] has also proposed a high-order front-tracking finite difference method for pricing American options under jump-diffusion models, giving an easy way to implement fourth-order accurate discretisations as well. A fitted finite volume method was presented for the numerical solution of European and American jump-diffusion models by Zhang *et al.* [16, 17], essentially based on a finite volume formulation with a fitted local approximation to the solution. The finite volume approach usually possesses a special feature of local conservation of the numerical fluxes, and has been gaining popularity [9].

Numerical methods for pricing European and American options under Merton's jump-diffusion models are further studied here. We apply the finite volume scheme in space, and the backward Euler and Crank-Nicolson schemes in time. For European options, the resulting dense linear systems are solved using the generalised minimal residual (GMRES) method. For American options, a new way to solve the LCP is introduced — viz. the modulus-based successive over-relaxation (MSOR) method, and we show that the $H_+$-matrix property of the system matrix guarantees its convergence. European and American options under Merton's jump-diffusion models are briefly reviewed in Section 2. The finite volume discretisation of the PIDE is discussed in Section 3, where we prove the stability and convergence of our methods. The modulus-based matrix splitting method is introduced in Section 4. Numerical experiments reported in Section 5 verify their effectiveness and robustness, and our conclusions are drawn in Section 6.

## 2. Mathematical Model

We briefly describe the PIDE for European or American options under Merton's jump-diffusion model. In Merton's jump-diffusion model [12, 13] the value of an underlying asset $x$ is given by

$$\frac{dx(t)}{x(t-)} = \mu dt + \sigma dW(t) + d\left(\sum_{j=1}^{N(t)} V_j\right), \tag{2.1}$$

where $\mu$ denotes the drift rate, $\sigma$ the volatility, $W(t)$ a standard Brownian motion and $N(t)$ a Poisson process with rate $\lambda$. The set $\{V_j\}$ is a sequence of independent identically distributed random variables from a distribution with log-normal density

$$f_{ln}(y) := \frac{1}{y\delta\sqrt{2\pi}} \exp\left(-\frac{(\log y - \mu)^2}{2\delta^2}\right), \tag{2.2}$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

where $\delta$ determines the variance of the jumps.

Under the general jump-diffusion model (2.1) asumptions, the value $v$ of a European option can be obtained by solving an initial value problem involving the PIDE

$$v_\tau - \frac{1}{2}\sigma^2 x^2 v_{xx} - (r-\lambda k)x v_x + (r+\lambda)v - \lambda\left(\int_0^\infty v(xy)f(y)\,dy\right) = 0\,, \tag{2.3}$$

where $(\tau, x) \in (0,T]\times[0,\infty)$ with $\tau = T-t$, $t$ is the current time, $T$ the maturity date, $r$ the risk free interest rate and $k$ the expected relative jump size — i.e.

$$k = E(y-1) = \exp(\mu + \delta^2/2) - 1\,.$$

For a European put option, the boundary conditions are

$$v(\tau,0) = Ke^{-r\tau} \quad \text{and} \quad \lim_{x\to\infty} v(\tau,x) = 0\,, \quad \tau\in(0,T]\,, \tag{2.4}$$

and the initial condition is

$$v(0,x) = g(x) \text{ where } g(x) = \max\{K-x,0\} \tag{2.5}$$

involving the strike price $K$. For a European call option, the boundary conditions are

$$v(\tau,0) = 0 \quad \text{and} \quad \lim_{x\to\infty}\left[v(\tau,x) - (x - Ke^{-r\tau})\right] = 0\,, \quad \tau\in(0,T]\,, \tag{2.6}$$

and the initial condition is

$$v(0,x) = g(x) \text{ where } g(x) = \max\{x-K,0\}\,. \tag{2.7}$$

The value of an American put option $v$ satisfies an LCP

$$\begin{cases} \mathscr{L}v \geq 0\,, \\ v - g \geq 0\,, \\ \mathscr{L}v\cdot(v-g) = 0\,, \end{cases} \tag{2.8}$$

where

$$\mathscr{L}v := v_\tau - \frac{1}{2}\sigma^2 x^2 v_{xx} - (r-\lambda k)x v_x + (r+\lambda)v - \lambda\left(\int_0^\infty v(xy)f(y)\,dy\right),$$

subject to the initial value condition (2.5) and the boundary conditions

$$v(\tau,0) = K \quad \text{and} \quad \lim_{x\to\infty} v(\tau,x) = 0, \quad \tau\in(0,T]\,. \tag{2.9}$$

For the finite volume method discretisation, it is convenient to first transform the PIDE (2.3) into the variable coefficient parabolic equation

$$\frac{\partial v}{\partial \tau} - \frac{\partial}{\partial x}\left(p(x)\frac{\partial v}{\partial x}\right) + q(x)\frac{\partial v}{\partial x} + cv - \lambda\left(\int_0^\infty v(xy)f(y)\,dy\right) = 0 \tag{2.10}$$

where $p(x) = \frac{1}{2}\sigma^2 x^2$, $q(x) = (\sigma^2 - r + \lambda k)x$ and $c = r+\lambda$, which is subject to the corresponding initial and boundary conditions. For the European and American put options, appropriate finite volume methods are then adopted for the space and time discretisation, and call options can be handled in a similar way.

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

## 3. Finite Volume Discretisation

To discretise the problem (2.10) in the finite volume method, we reformulate the problems in a truncated region $I=[0,X]$, where $X$ denotes a sufficiently large number to ensure solution accuracy. First, we define a uniform grid partition $T_h$ of the computational domain $[0,X]$:

$$0=x_0<x_1<x_2<\cdots<x_{m-1}<x_m=X\,,$$

with the sub-intervals $I_i=[x_{i-1},x_i]$ $(i=1,\cdots,m)$, $h=x_i-x_{i-1}=X/m$. The trial function space $V_h$ adopted is the linear element space with respect to $T_h$, which consists of all functions $v_h$ satisfying

(i) $v_h\in C(I)$, $v_h(X)=0$, and

(ii) $v_h$ is linear on each $I_i$, and uniquely determined by its endpoint values.

The basis function with respect to $x_i$ is

$$\phi_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{h}, & x\in I_i\,,\\ \dfrac{x_{i+1}-x}{h}, & x\in I_{i+1}\,,\\ 0\,, & \text{elsewhere}\,,\end{cases}\tag{3.1}$$

so the functions $\{\phi_i(x):i=0,1,\cdots,m-1\}$ form a basis for $V_h$ where

$$v_h=\sum_{i=0}^{m-1}v_i\phi_i\,,\quad \forall\, v_h\in V_h\,,\tag{3.2}$$

with $v_i=v_h(x_i)$. Following Ref. [8], we have

$$v_h=v_{i-1}(1-\xi)+v_i\xi\ \text{ where }\ \xi=\frac{x-x_{i-1}}{h},\quad x\in I_i\,,\tag{3.3}$$

$$v_h'=\frac{v_i-v_{i-1}}{h},\qquad x\in I_i\,.\tag{3.4}$$

Further, we define a dual grid $T_h^*$ with nodes

$$0=x_0<x_{\frac12}<x_{\frac32}<\cdots<x_{m-\frac12}<x_m=X\,,$$

where $x_{i-\frac12}=(x_{i-1}+x_i)/2$, $i=1,2,\cdots,m$. The dual elements are denoted by $I_0^*=[x_0,x_{\frac12}]$, $I_i^*=[x_{i-\frac12},x_{i+\frac12}]$, $i=1,2,\cdots,m-1$, $I_m^*=[x_{m-\frac12},x_m]$.

Accordingly, we choose the test function space $V_h^*$ as the piecewise constant function space with respect to $T_h^*$, consisting of all functions $\bar v_h$ that satisfy

(i) $\bar v_h(x)=0$ for $x\in I_m^*$, and

(ii) $\bar v_h$ is a constant on each $I_i^*(i=0,1,\cdots,m-1)$.

The basis functions of $V_h^*$ are $(i=0,1,\cdots,m-1)$:

$$\Phi_i(x)=\begin{cases}1\,, & x\in I_i^*\,,\\ 0\,, & x\overline{\in}I_i^*\,.\end{cases}\tag{3.5}$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

For any $\bar{v}_h \in V_h^*$ we have the form

$$\bar{v}_h = \sum_{i=0}^{m-1} \bar{v}_i \Phi_i \,, \tag{3.6}$$

where $\bar{v}_i = \bar{v}_h(x_i)$.

### 3.1. Approximation of the integral term

The integral term

$$Q = \int_0^\infty v(xy) f(y)\, dy \tag{3.7}$$

is approximated using the linear interpolation technique in Ref. [12], to be used in the spatial finite volume discretisation of the problems considered in the next section. First, on performing a change of variable $y = z/x$ in Eq. (3.7) we get

$$Q = \int_0^\infty v(z) f(z/x)/x \; dz \,, \tag{3.8}$$

and for each grid point $x_i$ $(i = 1, \cdots, m-1)$ we have

$$Q \approx Q_i = \int_0^{x_{m-1}} v(z) f(z/x_i)/x_i \; dz = \sum_{j=0}^{m-1} Q_{i,j} \,, \tag{3.9}$$

where

$$Q_{i,j} = \int_{x_j}^{x_{j+1}} v(z)\, f(z/x_i)/x_i \; dz \,.$$

Now using the linear interpolation for $v$ between the grid points, we get the approximation

$$Q_i \approx A_i = \sum_{j=0}^{m-1} A_{i,j}, \tag{3.10}$$

at each grid point $x_i$ $(i = 1, \cdots, m-1)$, where

$$A_{i,j} = \int_{x_j}^{x_{j+1}} \left( \frac{x_{j+1} - z}{h} v(x_j) + \frac{z - x_j}{h} v(x_{j+1}) \right) \frac{f(z/x_i)}{x_i} \; dz \,. \tag{3.11}$$

Then substituting the log-normal distribution $f(y) = f_{ln}(y)$ into Eq. (3.11), we obtain

$$A_{i,j} = \frac{1}{\delta\sqrt{2\pi}} \int_{x_j}^{x_{j+1}} \left( \frac{x_{j+1} - z}{h} v(x_j) + \frac{z - x_j}{h} v(x_{j+1}) \right) \exp\left( -\frac{(\log(z/x_i) - \mu)^2}{2\delta^2} \right) / z \; dz \,. \tag{3.12}$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

For each $j = 0, 1, \cdots, m-1$, on integrating Eq. (3.12) over $[x_j, x_{j+1}]$ we have

$$A_{i,j} = \frac{1}{2h}\left[\left(\operatorname{erf}\left(\frac{\mu - \log\left(\frac{x_{j+1}}{x_i}\right)}{\sqrt{2}\delta}\right) - \operatorname{erf}\left(\frac{\mu - \log\left(\frac{x_j}{x_i}\right)}{\sqrt{2}\delta}\right)\right)\alpha_j + \left(\operatorname{erf}\left(\frac{\mu + \delta^2 - \log\left(\frac{x_{j+1}}{x_i}\right)}{\sqrt{2}\delta}\right) - \operatorname{erf}\left(\frac{\mu + \delta^2 - \log\left(\frac{x_j}{x_i}\right)}{\sqrt{2}\delta}\right)\right) x_i \beta_j\right] \tag{3.13}$$

where erf(·) denotes the error function, and

$$\alpha_j = v(x_{j+1})x_j - v(x_j)x_{j+1} \quad \text{and} \quad \beta_j = \left(v(x_j) - v(x_{j+1})\right) e^{\mu + \delta^2/2}. \tag{3.14}$$

The following theorem states that the above approximation of the integral is second-order accurate under a suitable regularity assumption.

**Theorem 3.1.** *For a given $\tau \in (0, T]$, if $v(\tau, x)$ is twice continuously differentiable with respect to $x$ in $[0, X]$ then $|Q_i - A_i| = \mathcal{O}(h^2)$.*

*Proof.* Let us begin by considering an approximation in the interval $[x_j, x_{j+1}]$ for some $j \in \{0, 1, \cdots, m-1\}$, based on the linear interpolation of $v(\tau, z)$ that we denote by $\bar{v}(z)$. A well-known result states that if $v$ is regular then for any $z \in [x_j, x_{j+1}]$ the interpolation error is

$$v(\tau, z) - \bar{v}(z) = v_{xx}(\tau, \xi)(z - x_j)(z - x_{j+1})/2 \tag{3.15}$$

for some $\xi \in [x_j, x_{j+1}]$, so we have

$$\left|v(\tau, z) - \bar{v}(z)\right| \le Ch^2/8$$

where $C = \max_{\xi \in [0,X]} v_{xx}(\tau, \xi)$. Consequently,

$$\begin{aligned}\left|Q_{i,j} - A_{i,j}\right| &= \int_{x_j}^{x_{j+1}} \left|v(\tau, z) - \bar{v}(z)\right| f(z/x_i)/x_i \, dz \\ &= \frac{1}{\delta\sqrt{2\pi}} \int_{x_j}^{x_{j+1}} \left|v(\tau, z) - \bar{v}(z)\right| \exp\left(-\frac{(\log(z/x_i) - \mu)^2}{2\delta^2}\right)/z \, dz \le Ch^2,\end{aligned}$$

and summing over all of the intervals $j \in \{0, 1, \cdots, m-1\}$ we get

$$\begin{aligned}\left|Q_i - A_i\right| &\le \sum_{j=0}^{m-1} \left|Q_{i,j} - A_{i,j}\right| = \sum_{j=0}^{m-1} \int_{x_j}^{x_{j+1}} \left|v(\tau, z) - \bar{v}(z)\right| f(z/x_i)/x_i \, dz \\ &\le \sum_{j=0}^{m-1} \frac{1}{\delta\sqrt{2\pi}} \int_{x_j}^{x_{j+1}} \left|v(\tau, z) - \bar{v}(z)\right| \exp\left(-\frac{(\log(z/x_i) - \mu)^2}{2\delta^2}\right)/z \, dz \le Ch^2.\end{aligned}$$

This complete the proof. □

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

We denote by $\mathbf{R} = (R_{ij})$ the full matrix resulting from the integral term, with

$$R_{ij} = R_1 + R_2 \qquad \text{for } i, j = 1, 2, \cdots, m-1\,, \tag{3.16}$$

where

$$\begin{aligned} R_1 = -\frac{\lambda}{2h}\Bigg[ & x_{j-1}\left(\operatorname{erf}\left(\frac{\mu - \log\left(\frac{x_j}{x_i}\right)}{\sqrt{2}\delta}\right) - \operatorname{erf}\left(\frac{\mu - \log\left(\frac{x_{j-1}}{x_i}\right)}{\sqrt{2}\delta}\right)\right) \\ & - x_i e^{\mu+\delta^2/2}\left(\operatorname{erf}\left(\frac{\mu + \delta^2 - \log\left(\frac{x_j}{x_i}\right)}{\sqrt{2}\delta}\right) - \operatorname{erf}\left(\frac{\mu + \delta^2 - \log\left(\frac{x_{j-1}}{x_i}\right)}{\sqrt{2}\delta}\right)\right)\Bigg]\,, \end{aligned}$$

$$\begin{aligned} R_2 = -\frac{\lambda}{2h}\Bigg[ & x_i e^{\mu+\delta^2/2}\left(\operatorname{erf}\left(\frac{\mu + \delta^2 - \log\left(\frac{x_{j+1}}{x_i}\right)}{\sqrt{2}\delta}\right) - \operatorname{erf}\left(\frac{\mu + \delta^2 - \log\left(\frac{x_j}{x_i}\right)}{\sqrt{2}\delta}\right)\right) \\ & - x_{j+1}\left(\operatorname{erf}\left(\frac{\mu - \log\left(\frac{x_{j+1}}{x_i}\right)}{\sqrt{2}\delta}\right) - \operatorname{erf}\left(\frac{\mu - \log\left(\frac{x_j}{x_i}\right)}{\sqrt{2}\delta}\right)\right)\Bigg]\,. \end{aligned}$$

Evidently, all of the elements of $\mathbf{R}$ are negative, so we have the following good properties of matrix $\mathbf{R} + \lambda\mathbf{I}$ where $\mathbf{I}$ denotes a unit matrix.

**Theorem 3.2.** *The matrix $\boldsymbol{R} + \lambda \boldsymbol{I}$ is a diagonally dominant Z-matrix with a nonnegative diagonal — i.e.*

$$\sum_j (\boldsymbol{R} + \lambda \boldsymbol{I})_{i,j} \geq 0 \;\; \textit{and} \;\; \boldsymbol{R}_{i,j} \leq 0, \;\; \forall j \neq i.$$

The proof of this theorem is similar to Theorem 3.4 in Ref. [13], so omitted here.

## 3.2. Finite volume schemes

In this subsection, we propose one semi-discrete and two kinds of full discrete finite volume scheme for Eq. (2.10). The semi-discrete finite volume scheme is:

Find $v_h = v_h(\tau, \cdot) \in V_h$ such that

$$\begin{cases} \left(\dfrac{\partial v_h}{\partial \tau}, \bar{v}_h\right) + a_h(v_h, \bar{v}_h) = 0\,, & \forall \bar{v}_h \in V_h^*, \;\; 0 < \tau \leq T\,, \\ v(0, x) = g(x), & x \in [0, X]\,, \end{cases} \tag{3.17}$$

or equivalently

$$\begin{cases} \left(\dfrac{\partial v_h}{\partial \tau}, \Phi_i\right) + a_h(v_h, \Phi_i) = 0\,, & i = 0, \cdots, m-1\,, \\ v(0, \tau) = g(x), & x \in [0, X]\,, \end{cases} \tag{3.18}$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

where

$$a_h(v_h,\Phi_i) = -\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x}\left(p(x)\frac{\partial v}{\partial x}\right) dx + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x)\frac{\partial v_h}{\partial x}\, dx + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} c v_h\, dx - \lambda \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} Q\, dx, \quad i = 1\cdots m-1. \tag{3.19}$$

For each $i = 1,\cdots,m-1$, on integrating all of the terms on the right-hand side of Eq. (3.19) over $I_i^*$ and then applying Eq. (3.4) and the mid-point quadrature rule, we have

$$-\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x}\left(p(x)\frac{\partial v}{\partial x}\right) dx = p_{i-\frac{1}{2}} v_h'(x_{i-1/2}) - p_{i+\frac{1}{2}} v_h'(x_{i+1/2}) = p_{i-\frac{1}{2}}\frac{v_i - v_{i-1}}{h} - p_{i+\frac{1}{2}}\frac{v_{i+1}-v_i}{h}, \tag{3.20}$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x)\frac{\partial v_h}{\partial x} dx = q(x)\, v_h\Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q'(x)\, v_h\, dx = q_{i+\frac{1}{2}} v_h(x_{i+\frac{1}{2}}) - q_{i-\frac{1}{2}} v_h(x_{i-\frac{1}{2}}) - q(x)\, v_h\Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v_h(x_i) = q_{i+\frac{1}{2}}\frac{v_{i+1}+v_i}{2} - q_{i-\frac{1}{2}}\frac{v_i+v_{i-1}}{2} - (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})v_i, \tag{3.21}$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} c\, v_h dx = c\, v_i\, h, \tag{3.22}$$

$$-\lambda \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} Q\, dx = -\lambda Q_i h. \tag{3.23}$$

Substituting Eqs, (3.20)-(3.23) into Eq. (3.19) we obtain

$$a_h(v_h,\Phi_i) = p_{i-\frac{1}{2}}\frac{v_i - v_{i-1}}{h} - p_{i+\frac{1}{2}}\frac{v_{i+1}-v_i}{h} + q_{i+\frac{1}{2}}\frac{v_{i+1}+v_i}{2} - q_{i-\frac{1}{2}}\frac{v_i+v_{i-1}}{2} - (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}})v_i + c\, v_i h - \lambda Q_i h, \quad i = 1\cdots m-1, \tag{3.24}$$

where $v_i = v_h(\tau, x_i)$, $v_m = 0$, $p_{i-\frac{1}{2}} = p(x_{i-\frac{1}{2}})$, $q_{i-\frac{1}{2}} = q(x_{i-\frac{1}{2}})$. In particular, for any $x \in I_0^*$ we have

$$a_h(v_h,\Phi_0) = \left(\frac{p_{\frac{1}{2}}}{h} - \frac{q_{\frac{1}{2}}}{2} + \frac{h}{2}c\right) v_0 + \left(\frac{q_{\frac{1}{2}}}{2} - \frac{p_{\frac{1}{2}}}{h}\right) v_1. \tag{3.25}$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

On the other hand,

$$\begin{cases} \left(\dfrac{\partial v_h}{\partial \tau}, \Phi_0\right) = \dfrac{h}{2}\dfrac{\partial v_0}{\partial \tau}, \\ \left(\dfrac{\partial v_h}{\partial \tau}, \Phi_i\right) = h\dfrac{\partial v_i}{\partial \tau}, \quad i = 1, 2, \cdots, m-1. \end{cases} \tag{3.26}$$

Consequently, after the above space finite volume discretisation the semi-discrete finite volume scheme (3.17) leads to the following matrix form:

$$\frac{\partial \mathbf{v}}{\partial \tau} + \mathbf{S}\mathbf{v} = \mathbf{f}, \tag{3.27}$$

where $\mathbf{v} \in R^{(m-1)\times 1}$ is an unknown vector and $\mathbf{S} = \mathbf{P} + \mathbf{R}$, where $\mathbf{P}$ is a tridiagonal matrix and $\mathbf{R}$ is a full matrix resulting from the integral term. Specifically,

$$\mathbf{P} = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{m-2} \\ & & & a_{m-1} & b_{m-1} \end{pmatrix}_{(m-1)\times(m-1)},$$

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{m-2} \\ v_{m-1} \end{pmatrix}_{(m-1)\times 1}, \qquad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{m-2} \\ f_{m-1} \end{pmatrix}_{(m-1)\times 1},$$

where the $\mathbf{P}$ entries are

$$a_i = -\frac{1}{2}\sigma^2\left(i-\frac{1}{2}\right)^2 - \frac{1}{2}(\sigma^2 - r + \lambda k)\left(i-\frac{1}{2}\right), \qquad i = 2, \cdots, m-1,$$

$$\begin{aligned} b_i =& \frac{1}{2}\sigma^2\left(i-\frac{1}{2}\right)^2 + \frac{1}{2}\sigma^2\left(i+\frac{1}{2}\right)^2 - \frac{1}{2}(\sigma^2 - r + \lambda k)\left(i+\frac{1}{2}\right) \\ &+ \frac{1}{2}(\sigma^2 - r + \lambda k)\left(i-\frac{1}{2}\right) + c, \qquad i = 1, \cdots, m-1, \end{aligned}$$

$$c_i = -\frac{1}{2}\sigma^2\left(i+\frac{1}{2}\right)^2 + \frac{1}{2}(\sigma^2 - r + \lambda k)\left(i+\frac{1}{2}\right), \qquad i = 1, \cdots, m-2,$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

and the **f** entries where $v_0 = v(\tau, 0)$, $\log(0) = -\text{Inf}$, and $\text{erf}(\text{Inf}) = 1$ are

$$\begin{aligned} f_1 =& \left[\frac{3}{8}\sigma^2 - \frac{1}{4}(r-\lambda k)\right] v_0 + \lambda \left( h \left[ \text{erf}\left( \frac{\mu + \delta^2 - \log(1)}{\sqrt{2}\delta} \right) - \text{erf}\left( \frac{\mu + \delta^2 - \log(0)}{\sqrt{2}\delta} \right) \right] \right. \\ & \left. \cdot \exp\left(\mu + \frac{\delta^2}{2}\right) - h \left[ \text{erf}\left( \frac{\mu - \log(1)}{\sqrt{2}\delta} \right) - \text{erf}\left( \frac{\mu - \log(0)}{\sqrt{2}\delta} \right) \right] \right) v_0 , \\ f_i =& \lambda \left( ih \left[ \text{erf}\left( \frac{\mu + \delta^2 - \log(h/(ih))}{\sqrt{2}\delta} \right) - \text{erf}\left( \frac{\mu + \delta^2 - \log(0)}{\sqrt{2}\delta} \right) \right] \exp\left(\mu + \frac{\delta^2}{2}\right) \right. \\ & \left. - h \left[ \text{erf}\left( \frac{\mu - \log(h/(ih))}{\sqrt{2}\delta} \right) - \text{erf}\left( \frac{\mu - \log(0)}{\sqrt{2}\delta} \right) \right] \right) v_0 , \qquad i = 1, \cdots, m-1 . \end{aligned}$$

The following theorem states that the matrix **P** has the desirable properties.

**Theorem 3.3.** *Let **P** be the semi-discrete matrix of PIDE* (2.10). *Then assuming that* $r > 0$, $\lambda > 0$, $\frac{1}{2}\sigma^2 + r > \lambda k$ *and* $-\frac{5}{2}\sigma^2 + r < \lambda k$, *the matrix* $\boldsymbol{P} - \lambda \boldsymbol{I}$ *is a strictly diagonally dominant with positive diagonal elements M-matrix — i.e. for* $i, j = 1, 2, \cdots, m-1$ *we have*

$$\sum_j (\boldsymbol{P} - \lambda \boldsymbol{I})_{i,j} > 0 , \quad \boldsymbol{P}_{i,j} \le 0, \quad \forall j \ne i \ \text{ and } \ (\boldsymbol{P} - \lambda \boldsymbol{I})_{i,i} > 0 .$$

*Proof.* First we show that the off-diagonal $c_i$ in **P** are nonpositive. From the definition of the $c_i$, when $i = 1$ we have

$$c_1 = \frac{3}{4}\left( -\frac{1}{2}\sigma^2 - r + \lambda k \right) < 0$$

because $r > 0$, $\lambda > 0$ and $\frac{1}{2}\sigma^2 + r > \lambda k$. Consequently, when $i \ge 2$ we have

$$c_i = \frac{3 + 2(i-1)}{4} \left[ -\frac{1 + 2(i-1)}{2}\sigma^2 - r + \lambda k \right] < 0$$

because $[1 + 2(i-1)]/2 > 1/2$ for $i \ge 2$. Similarly, the off-diagonal $a_i$ can be shown to be nonpositive under the assumptions $r > 0$, $\lambda > 0$ and $-\frac{5}{2}\sigma^2 + r < \lambda k$. Furthermore, from the entries $b_i$ in **P** it follows that

$$b_i + a_i + c_i = c = r + \lambda > 0 ,$$

which implies

$$\sum_j (\mathbf{P} - \lambda \mathbf{I})_{i,j} = r > 0 \ \text{ and } \ (\mathbf{P} - \lambda \mathbf{I})_{i,i} > 0 .$$

This complete the proof. □

Combining Theorem 3.2 and Theorem 3.3, we obtain the following result.

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

**Corollary 3.1.** *The matrix $\boldsymbol{S}=\boldsymbol{P}+\boldsymbol{R}$ is a strictly diagonally dominant with positive diagonal elements M-matrix.*

The full discrete finite volume schemes are now proposed. The partition of the time interval $[0,T]$ is $0=\tau_0<\tau_1<\cdots<\tau_n=T$, and we consider the time step $\Delta\tau=T/n$. The full discrete finite volume scheme for (3.17) is:

Find $v_h^l\in V_h$ for $1\le l\le n$ such that

$$\begin{cases} (\partial_t v_h^l,\bar{v}_h)+a_h(v_h^{l,\theta},\bar{v}_h)=0\,, & \forall\,\bar{v}_h\in V_h^*\,,\\ v_h^0=g(x)\,, & x\in I\,, \end{cases} \tag{3.28}$$

where

$$\partial_t v_h^l=\frac{v_h^l-v_h^{l-1}}{\Delta\tau}\,,\qquad v_h^{l,\theta}=\theta\,v_h^l+(1-\theta)v_h^{l-1}\,,\qquad v_h^l=v_h(\tau_l),\quad \tau_l=l\Delta\tau\,.$$

This is the backward Euler and Crank-Nicolson finite volume scheme when $\theta$ is equal to 1 and $1/2$, respectively. We can rewrite (3.28) in the following matrix form:

$$\left(\frac{1}{\Delta\tau}\mathbf{I}+\theta\mathbf{S}\right)\mathbf{v}^l=\left(\frac{1}{\Delta\tau}\mathbf{I}-(1-\theta)\mathbf{S}\right)\mathbf{v}^{l-1}+\mathbf{f}\,,\qquad l=1,2,\cdots,n\,, \tag{3.29}$$

where $\mathbf{v}^0=\mathbf{g}$ and the vector $\mathbf{g}$ contains the values of the payoff function $g(x)$ at the grid points — viz. $g_i=g(x_i)$ for $i=1,\cdots,m-1$.

The following Corollary can be obtained from Corollary 3.1 directly.

**Corollary 3.2.** *The full discrete matrices $\frac{1}{\Delta\tau}\boldsymbol{I}+\theta\boldsymbol{S}$ resulting from the backward Euler ($\theta=1$) and Crank-Nicolson ($\theta=1/2$) finite volume schemes are strictly diagonally dominant with positive diagonal elements M-matrices.*

Noting the form (3.29), we let

$$\mathbf{B}=\frac{1}{\Delta\tau}\mathbf{I}+\theta\mathbf{S}\ \text{ and }\ \mathbf{C}=\frac{1}{\Delta\tau}\mathbf{I}-(1-\theta)\mathbf{S}\,,\quad \theta=\frac{1}{2}\text{ or }1\,. \tag{3.30}$$

The space and time discretisation of American option model (2.8) then leads to a sequence of linear complementarity problems (LCP):

$$\begin{cases} \mathbf{B}\mathbf{v}^l\geqslant\mathbf{C}\mathbf{v}^{(l-1)}+\mathbf{f}\,,\\ \mathbf{v}^{(l)}\geqslant\mathbf{g}\,,\\ (\mathbf{B}\mathbf{v}^{(l)}-\mathbf{C}\mathbf{v}^{(l-1)}-\mathbf{f})^{\mathrm{T}}(\mathbf{v}^{(l)}-\mathbf{g})=0\,, \end{cases} \tag{3.31}$$

for each $l=1,2,\cdots,n$. Further, letting

$$A:=\mathbf{B}\,,\ \ z:=\mathbf{v}^{(l)}-\mathbf{g}\ \text{ and }\ q:=\mathbf{B}\mathbf{g}-\mathbf{C}\mathbf{v}^{(l-1)}-\mathbf{f}\,,$$

we can transform (3.31) into the standard linear complementarity problem:

$$w:=Az+q\ge 0\,,\ \ z\ge 0\ \text{ and }\ z^Tw=0\,, \tag{3.32}$$

where the early exercise constraint equations are set to zero instead of the payoff function for convenience, abbreviated as LCP$(q,A)$.

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

## 4. Solutions of the Linear Complementarity Problem

So far, the most popular algorithm to solve linear complementarity problem has been the projected successive overrelation (PSOR) method [6] that is generally fast and robust for many kinds of American option pricing problems [3]. However, this method exhibits exponential solution-time behaviour as the number of space discretisation points increases, and is not well equipped to handle sparse matrix systems — especially in the case of multi-asset options [7, 19, 20]. Below we discuss another efficient numerical method in detail — viz. the modulus-based matrix splitting method. The following theorem implies that the LCP($q$,$A$) is equivalent to a fixed-point problem.

**Theorem 4.1** (cf. Ref [4]). *Let $A = M - N$ be a splitting of the matrix $A \in R^{n\times n}$, $\Omega_1$ and $\Omega_2$ be $n \times n$ nonnegative diagonal matrices, and $\Omega$ and $\Gamma$ be $n \times n$ positive diagonal matrices such that $\Omega = \Omega_1 + \Omega_2$. For the LCP($q$,$A$), the following statements hold:*

*(1) if $(\omega, z)$ is a solution of LCP($q$,$A$), then $x = (\Gamma^{-1}z - \Omega^{-1}\omega)/2$ satisfies the implicit fixed-point equation*

$$(M\Gamma + \Omega_1)x = (N\Gamma - \Omega_2)x + (\Omega - A\Gamma)|x| - q\ ;\ \ and \tag{4.1}$$

*(2) if $x$ satisfies the implicit fixed-point equation* (4.1)*, then*

$$z = \Gamma(|x| + x)\ \text{ and }\ \omega = \Omega(|x| + x) \tag{4.2}$$

*is a solution of the LCP($q$,$A$).*

By specifying

$$\Omega = \Omega_1\,,\ \ \Omega_2 = 0\ \text{ and }\ \Gamma = \frac{1}{\gamma} I$$

in Eq. (4.1), we obtain the simplified implicit fixed-point equation

$$(M + \Omega)x = Nx + (\Omega - A)|x| - \gamma q\ . \tag{4.3}$$

The modulus-based matrix splitting iteration method can therefore be defined by the linear system

$$(M + \Omega)x^{(k+1)} = Nx^{(k)} + (\Omega - A)|x^{(k)}| - \gamma q \tag{4.4}$$

such that

$$z^{(k+1)} = \frac{1}{\gamma}\left(|x^{(k+1)}| + x^{(k+1)}\right)\ ,$$

where $\Omega$ is a positive diagonal matrix and $\gamma$ is a positive constant. Specifically, if we let $A = D - L - U$ where $D$, $L$ and $U$ are the respective diagonal, strictly lower-triangular and strictly upper triangular matrices of $A$, and set

$$M = (1/\alpha)D - L\ \text{ and }\ N = (1/\alpha - 1)D + U\ ,$$

then Eq. (4.4) produces the modulus-based successive overrelaxation (MSOR) method — viz.

$$(D + \Omega - \alpha L)x^{(k+1)} = \left[(1-\alpha)D + \alpha U\right]x^{(k)} + (\Omega - \alpha A)|x^{(k)}| - \alpha q\,,\ \gamma = 1,\ k = 0, 1, \cdots \tag{4.5}$$

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

with

$$z^{(k+1)} = |x^{(k+1)}| + x^{(k+1)} \text{ and } \omega^{(k+1)} = \Omega\left(|x^{(k+1)}| - x^{(k+1)}\right),$$

which converges to the solution of LCP($q$,$A$) where $\alpha$ is the relaxation factor. In particular, if $\alpha = 1$, the MSOR method leads to the modulus-based Gauss-Seidel (MGS) method. The MSOR algorithm is as follows.

**Algorithm 4.1. Modulus-based Successive Overrelaxation (MSOR) method**

1. **Choose** $x, \Omega, \alpha, \text{tol}, \text{maxit}$;
2. **For** $\text{it} = 1, 2, \ldots, \text{maxit}$
3. $z = |x| + x$;
4. $b = [(1-\alpha)D + \alpha U]x + (\Omega - \alpha A)|x| - \alpha q$;
5. $\text{Res} = \|\min(Az + q, z)\|_2$;
6. **If** $\text{Res} < \text{tol}$
7. **break**;
8. **End**
9. Solve $(D + \Omega - \alpha L)x = b$;
10. **End For**

Convergence conditions for the MSOR method are provided in the following theorem.

**Theorem 4.2** (cf. Ref. [4]). *Let $A \in R^{n\times n}$ be an $H_+$-matrix, and $A = M - N$ be an $H$-compatible splitting of the matrix $A$ — i.e. $\langle A\rangle = \langle M\rangle - |N|$. Assuming $\Omega$ is a positive diagonal matrix and and $\gamma$ is a positive constant, if the parameter matrix $\Omega$ satisfies $\Omega \geq \frac{1}{2}\text{diag}(M)$ then the iteration sequence $\{z^k\}_{k=1}^{+\infty} \subset R_+^n$ generated by the MSOR method converges to the unique solution $z^* \in R_+^n$ of the LCP(q,A) for any initial vector $x_0 \in R^n$.*

Finally, we establish the following convergence theorem for the MSOR method that prices American options under Merton's jump-diffusion model.

**Theorem 4.3.** *Let $\boldsymbol{S}$ and $\boldsymbol{B} = \frac{1}{\Delta\tau}\boldsymbol{I} + \theta\boldsymbol{S}$ be the semi-discrete and full discrete matrix of the PIDE (2.10), respectively. If $r > 0$, $\lambda > 0$, $\frac{1}{2}\sigma^2 + r > \lambda k$ and $-\frac{5}{2}\sigma^2 + r < \lambda k$, and the parameter matrix $\Omega$ satisfies $\Omega \geq \frac{1}{2}\text{diag}(M)$, then the MSOR method for pricing American option under Merton's jump-diffusion model converges to the unique solution for any initial vector.*

*Proof.* According to Corollary 3.2, the full discrete matrix $\mathbf{B}$ is a strictly diagonally dominant with positive diagonal elements $M$-matrix, hence $\mathbf{B}$ is an $H_+$-matrix [18]. Also, $\mathbf{B} = D - L - U$ is an $H$-compatible splitting of the matrix $\mathbf{B}$. Then from Theorem 4.2, if the parameter matrix $\Omega \geq \frac{1}{2}\text{diag}(M)$, the MSOR method converges to the unique solution for any initial vector. □

Unlike the PSOR method, the MSOR method does not require the projection of the iterated solution onto the space $R_+^n := \{x \in R^n | x \geq 0\}$, which may be costly and complicated in actual implementations. Furthermore, the linear system of the equation in line 9 of the algorithm 4.1 may be solved exactly, and the coefficient matrix $D + \Omega - \alpha L$ can be more diagonal dominant than $D - \alpha L$ in the PSOR algorithm for the positive diagonal matrix $\Omega$. Consequently, the MSOR method may be more efficient numerically than the PSOR method.

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

## 5. Numerical Experiments

Some numerical examples are now presented to demonstrate the effectiveness of the finite volume method.

### 5.1. European options

We first present results for the pricing of European vanilla put and European butterfly options. The assumed parameters in the European vanilla put option and European butterfly option are

$$\begin{aligned} &r = 0.05, \quad \sigma = 0.15, \quad \mu = -0.9, \quad \delta = 0.45, \\ &\lambda = 0.10, \quad K = 100, \quad T = 0.25, \quad X = 300 \end{aligned} \tag{5.1}$$

and

$$\begin{aligned} &r = 0.05, \quad \sigma = 0.15, \quad \mu = -0.9, \quad \delta = 0.45, \quad \lambda = 0.10, \\ &K_1 = 90, \quad K_2 = 110, \quad T = 0.25, \quad X = 300, \end{aligned} \tag{5.2}$$

respectively. The butterfly option has the payoff

$$g(x) = \max(x - K_1, 0) - 2\max\big(x - (K_1 + K_2)/2, 0\big) + \max(x - K_2, 0)\,, \tag{5.3}$$

constructed by holding a long position in two calls at strike prices $K_1$ and $K_2$, and a short position in two calls at the strike price $(K_1 + K_2)/2$. The above parameters have also been adopted in Refs. [10, 11, 17]. The relative error is defined as

$$\text{Error} = \frac{\|v^* - v^\nu\|_2}{\|v^*\|_2}\,, \tag{5.4}$$

where $v^\nu$ is the $\nu$-th numerical solution and $v^*$ the reference solution, with $\|\cdot\|_2$ denoting the $l_2$-norm of a vector.

In our numerical experiments, the linear systems resulting from European options were solved by the GMRES method. All of the computations were run on a computer with CPU 3.4 GHz and memory 8.0 GB, using the programming language MATLAB R2014a.

The validity of the finite volume method for the European vanilla and European butterfly options can be seen in the numerical results presented in Table 1, where the prices produced are in good agreement with those obtained in Refs. [11, 17].

The convergence performance of two full discrete schemes were studied. In order to report the relative errors, a reference numerical solution for the European vanilla put option was computed on a fine grid with $(m, n) = (2400, 1600)$ via the Crank-Nicolson scheme and the GMRES method, where $m$ and $n$ are the respective number of grid steps in the $x$-direction and $\tau$-direction. In Table 2, the relative errors of the numerical solutions and elapsed CPU time are listed for the backward Euler and Crank-Nicolson schemes, where it is seen that the Crank-Nicolson scheme is more precise and requires less CPU time than the backward Euler scheme on the various grids.

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

Table 1: Values for European vanilla put and butterfly options using the Crank-Nicolson scheme.

| $(m,n)$ | $v(T,100)$, vanilla | $v(T,105)$, butterfly |
|---|---|---|
| $(300,200)$ | 3.14246 | 3.09323 |
| $(600,400)$ | 3.14749 | 3.09052 |
| $(1200,800)$ | 3.14866 | 3.08967 |
| $(2400,1600)$ | 3.14892 | 3.08985 |
| Ref. [11,17] | 3.14902 | 3.08980 |

Table 2: Comparison of two full discrete schemes based on different discretisation grids for the European vanilla put option.

| $(m,n)$ | backward Euler scheme | | Crank-Nicolson scheme | |
|---|---|---|---|---|
| | CPU(s) | Error | CPU(s) | Error |
| $(50,25)$ | 0.07 | 7.49e-04 | 0.14 | 6.54e-04 |
| $(100,50)$ | 0.07 | 1.88e-04 | 0.15 | 1.38e-04 |
| $(200,100)$ | 0.27 | 6.33e-05 | 0.22 | 3.71e-05 |
| $(400,200)$ | 1.30 | 2.23e-05 | 1.14 | 9.14e-06 |
| $(800,400)$ | 6.13 | 8.80e-06 | 5.81 | 2.72e-06 |

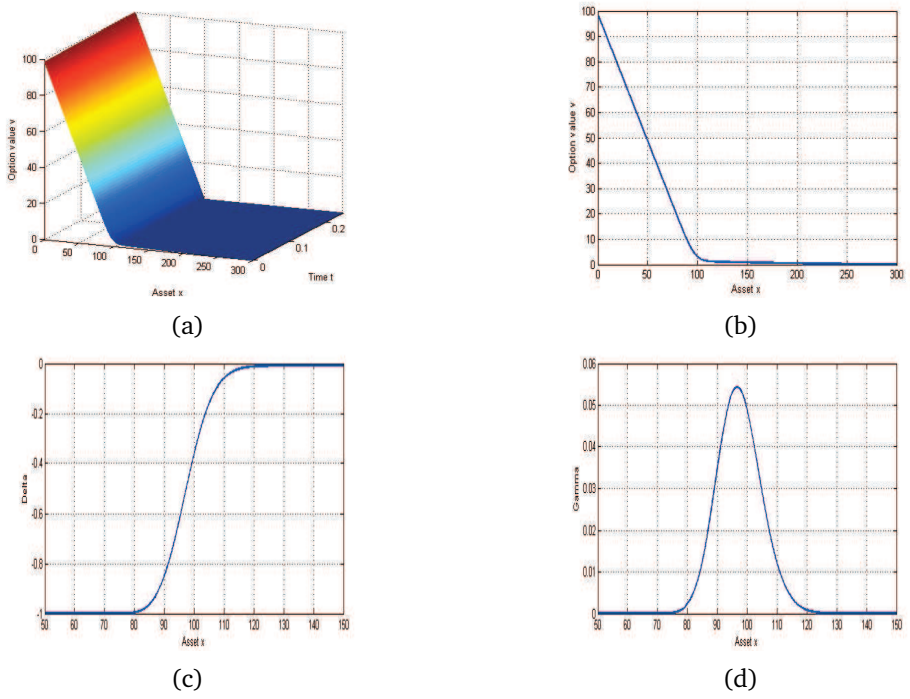



Figure 1: European vanilla put option: (a) the option-value surface; (b) the option value, (c) Delta and (d) Gamma at $\tau = T(t=0)$, using the Crank-Nicolson scheme (Grids: $(m,n) = (2400,1600)$).

Finally, in Figs. 1 and 2 we plot the option-value surface, the option value, together with the first derivative delta:=$[\partial v/\partial x \approx [v(x+\Delta x)-v(x)]/\Delta x$ and second derivative

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

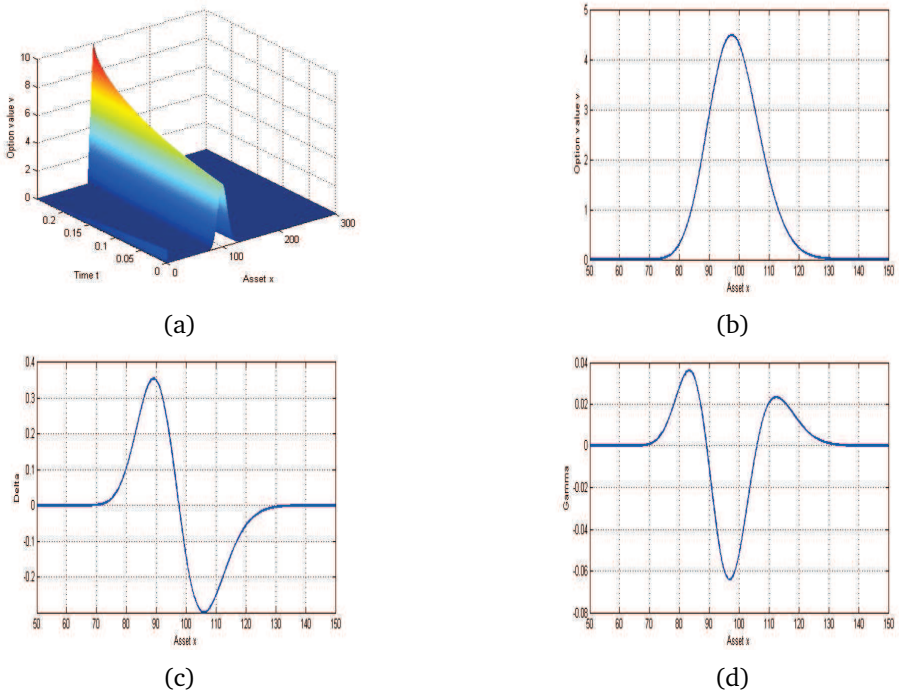



Figure 2: European butterfly option: (a) the option-value surface; (b) the option value, (c) Delta and (d) Gamma at $\tau = T(t=0)$, using the Crank-Nicolson scheme (Grids: $(m,n) = (2400, 1600)$).

gamma:$=\partial^2 v/\partial x^2 \approx [v(x+\Delta x)-2v(x)+v(x-\Delta x)]/(\Delta x)^2$, at the last time step of the European vanilla put option and butterfly option. Both figures show that the evaluated option value, delta and gamma are qualitatively quite good, with no spurious oscillation.

## 5.2. American options

We also investigated the performance of the finite volume method and MSOR method in pricing American options, adopting the same parameters listed in (5.1)-(5.3). With $r > 0$, $\lambda > 0$, $(1/2)\sigma^2 + r > \lambda k$ and $-(5/2)\sigma^2 + r < \lambda k$, the MSOR method is guaranteed to converge to the unique solution for any initial vector, according to Theorem 4.3. The iteration parameters were $\Omega = D$ and $\alpha = 1.2$ in the MSOR method, and in the PSOR method the relaxation factor was $\omega = 1.2$. For a fair comparison of the different methods, in all experiments the stopping criterion was

$$\text{Res}(z^\nu) := \|\min(Az^\nu + q, z^\nu)\|_2 < \text{tol} \tag{5.5}$$

with tol=$10^{-6}$, or when $\nu$ reached a prescribed maximum number of iteration steps (e.g. 1000), where $z^\nu$ represents the $\nu$-th numerical solution of the LCP$(q,A)$.

The numerical results for the American vanilla put and butterfly options obtained from the finite volume method and MSOR method are reported in Table 3, where we observe

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

Table 3: Values for American vanilla put and butterfly options using the Crank-Nicolson scheme.

| | vanilla option | | | butterfly option | |
|---|---|---|---|---|---|
| $(m,n)$ | $v(T,90)$ | $v(T,100)$ | $v(T,110)$ | $(m,n)$ | $v(T,105)$ |
| $(300,200)$ | 10.00000 | 3.23346 | 1.41828 | $(300,200)$ | 5.24693 |
| $(600,400)$ | 10.00297 | 3.23928 | 1.41941 | $(600,400)$ | 5.24980 |
| $(1200,800)$ | 10.00354 | 3.24075 | 1.41970 | $(1200,800)$ | 5.25126 |
| $(2400,1600)$ | 10.00379 | 3.24112 | 1.41978 | $(2400,1600)$ | 5.25128 |
| Ref. [11] | 10.00382 | 3.24122 | 1.41980 | Ref. [10] | 5.25129 |

Table 4: Comparison of two methods on different discretisation grids for American vanilla put option using the backward Euler scheme.

| | PSOR | | | MSOR | | |
|---|---|---|---|---|---|---|
| $(m,n)$ | IT | CPU | Error | IT | CPU | Error |
| $(50,25)$ | 9.8 | 0.03 | 4.93e-04 | 15.9 | 0.01 | 4.93e-04 |
| $(100,50)$ | 10.9 | 0.18 | 1.84e-04 | 23.9 | 0.08 | 1.84e-04 |
| $(200,100)$ | 12.4 | 0.78 | 7.02e-05 | 35.0 | 0.39 | 7.02e-05 |
| $(400,200)$ | 21.0 | 6.24 | 2.72e-05 | 49.0 | 4.91 | 2.73e-05 |
| $(800,400)$ | 39.7 | 117.32 | 1.45e-05 | 85.8 | 94.17 | 1.48e-05 |
| $(1200,800)$ | 45.0 | 598.80 | 1.24e-05 | 98.0 | 513.63 | 1.29e-05 |

Table 5: Comparison of two methods on different discretisation grids for American vanilla put option using the Crank-Nicolson scheme.

| | PSOR | | | MSOR | | |
|---|---|---|---|---|---|---|
| $(m,n)$ | IT | CPU | Error | IT | CPU | Error |
| $(50,25)$ | 9.7 | 0.03 | 4.31e-04 | 11.7 | 0.01 | 4.31e-04 |
| $(100,50)$ | 9.9 | 0.17 | 1.43e-04 | 17.0 | 0.08 | 1.43e-04 |
| $(200,100)$ | 10.9 | 0.67 | 4.69e-05 | 25.0 | 0.30 | 4.69e-05 |
| $(400,200)$ | 12.8 | 4.00 | 1.70e-05 | 37.0 | 3.81 | 1.70e-05 |
| $(800,400)$ | 23.0 | 69.24 | 1.18e-05 | 51.0 | 55.68 | 1.19e-05 |
| $(1200,800)$ | 25.8 | 340.84 | 1.17e-05 | 55.0 | 287.91 | 1.18e-05 |

that the prices obtained are in good agreement with those in Refs. [10, 11].

The convergence performance of the MSOR and PSOR methods were also compared. Similar to the computations for the European option, the reference numerical solution was obtained on a fine grid with $(m,n) = (2400,1600)$, via the Crank-Nicolson scheme and MSOR method. The number of average iteration steps (denoted by 'IT'), elapsed CPU time in seconds (denoted by 'CPU') and the relative error (denoted by 'Error') are listed in Tables 4 and 5. We observe that the accuracy of the numerical solution of the LCP improves as the discretisation grid is refined, where the number of iteration steps involved is higher for the MSOR method than for the PSOR method, but the MSOR method requires less CPU time. In addition, we note that the Crank-Nicolson scheme is more precise and requires less CPU time than the backward Euler scheme. In Fig. 3, we plot the curves for the CPU time of the MSOR and PSOR methods versus the space discretisation grid number $m$ when

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

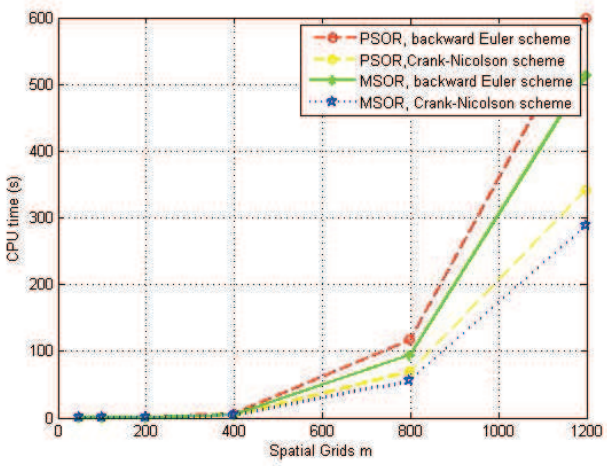



Figure 3: The CPU time of the PSOR and MSOR methods for American vanilla put option ($m = 2n$).

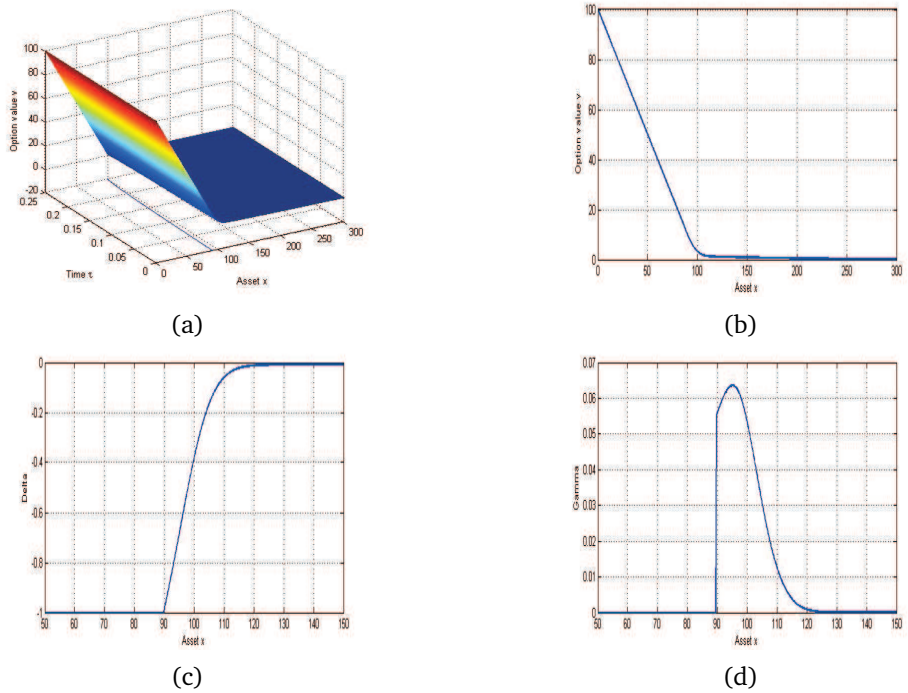



Figure 4: American vanilla put option: (a) the option-value surface and the optimal exercise boundary curve; (b) the option value, (c) Delta and (d) Gamma at $\tau = T(t = 0)$, using the Crank-Nicolson scheme and MSOR method (Grids: $(m, n) = (2400, 1600)$).

$m = 2n$, where we see that the PSOR method generally requires more CPU time than the MSOR method.

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

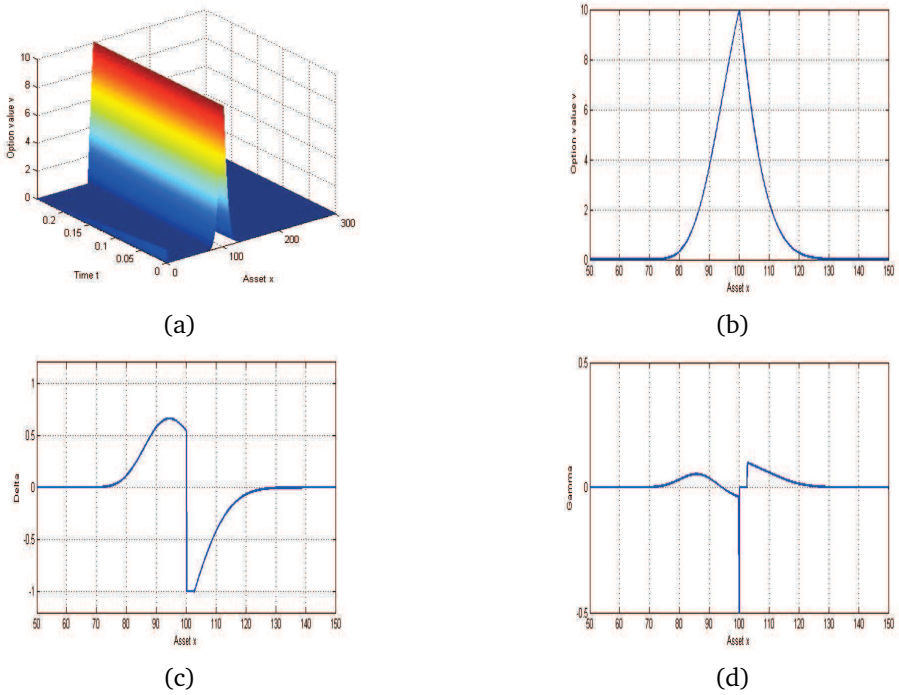



Figure 5: American butterfly option: (a) the option-value surface; (b) the option value, (c) Delta and (d) Gamma at $\tau = T(t = 0)$, using the Crank-Nicolson scheme and MSOR method (Grids: $(m, n) = (2400, 1600)$).

Finally, we plot the option-value surface and the optimal exercise boundary curve, the option value and their delta value and gamma value at the last time step for the American vanilla put option and butterfly option in Figs. 4 and 5. The stability and robustness of the Crank-Nicolson finite volume scheme and MSOR method are also evident in these two figures. Due to non-smoothness of the payoff function, the numerical results previously obtained by the Crank-Nicolson finite difference scheme [13] and the fitted finite volume scheme [16, 17] are quite erratic, so the Rannacher time-stepping technique had to be adopted. This erratic behaviour did not appear in our numerical experiments, indicating the robustness of the finite volume method.

## 6. Conclusion

A finite volume method for pricing European and American options under Merton's jump-diffusion model was investigated. An easy to implement linear interpolation for the integral approximation is second-order accurate, and the full discrete systems are strictly diagonally dominant $M$-matrices. For European option prices, the GMRES method was used to solve the associated algebraic equations. A new numerical method for American option pricing, the modulus-based successive overrelaxation (MSOR) method, was used

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

to solve the resulting linear complementary problem — and an associated convergence theorem was established. Numerical experiments confirm the stability and robustness of the proposed methods.

## Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions, which considerably improved the quality of this article. This work was supported by the National Natural Science Foundation of China (grant No. 11271289), the Fundamental Research Funds for the Central Universities, and also the Scientific Research Foundation of the Education Department of Yunnan Province (grant No. 2015Y443).

## References

[1] K. Amin, *Jump diffusion option valuation in discrete time*, J. Financial **48**, 1863-1883 (1993).
[2] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Rev. Deriv. Res. **4**, 231-262 (2000).
[3] A. Borici and H.J. Luthi, *Fast solution of complementarity formulations in American put option pricing*, J. Comp. Fin. **9**, 63-81 (2005).
[4] Z.Z. Bai, *Modulus-based matrix splitting iteration methods for linear complementarity problems*, Numer. Linear. Algebra. Appl. **17**, 917-933 (2010).
[5] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Polit. Econ. **81**, 637-654 (1973).
[6] C. W. Cryer, *The solution of a quadratic programming using systematic overrelaxation*, SIAM J. Control **9**, 385-392 (1971).
[7] P.A. Forsyth and K.R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput. **23**, 2095-2122 (2002).
[8] R.H. Li, Z.Y. Chen and W. Wu, *Generalized Difference Methods for Differential Equations (Numerical Analysis of Finite Volume Methods)*, Marcel Dekker, NewYork (2000).
[9] Y. Li, J. Lin and M. Yang, *Finite volume element methods: An overview on recent developments*, Int. J. Numer. Anal. Model, Series B **4**, 14-34 (2013).
[10] Y. d'Halluin, P.A. Forsyth and G.A Labahn, *A penalty method for American options with jump diffusion processes*, Numer. Math. **97**, 321-352 (2004).
[11] Y. d'Halluin, P.A. Forsyth and K.R. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA J. Numer. Anal. **25**, 87-112 (2005).
[12] S. Salmi and J. Toivanen, *An iterative method for pricing American options under jump-diffusion model*, Appl. Numer. Math. **61**, 821-831 (2011).
[13] J. Toivanen, *Numerical valuation of European and American options under Kou's jump-diffusion model*, SIAM J. Sci. Comput. **30**, 1949-1970 (2008).
[14] J. Toivanen, *A high-order front-tracking finite difference method for pricing American options under jump-diffusion models*, J. Comp. Fin. **13**, 61-79 (2010).
[15] X.L. Zhang, *Numerical analysis of American option pricing in a jump diffusion model*, Math. Oper. Res. **22**, 668-690 (1997).
[16] K. Zhang and S. Wang, *A computational scheme for options under jump diffusion processes*, Int. J. Numer. Anal. Mod. **6**, 110-123 (2009).
[17] K. Zhang and S. Wang, *Pricing options under jump diffusion processes with fitted finite volume method*, Appl. Math. Comput. **201**, 398-413 (2008).
[18] N. Zheng and J.F. Yin, *Modulus-based successive overrelaxation method for pricing American options*, J. Appl. Math. Informatics **31**, 769-784 (2013).

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press

[19] K. Zhang, X.Q. Yang and K.L. Teo, *Augmented Lagrangian method applied to American option pricing*, Automatica **42**, 1407-1416 (2006).
[20] K. Zhang, S. Wang, X.Q. Yang and K.L. Teo, *A power penalty approach to numerical solutions of two-asset American options*, Numer. Math: Theory, Methods and Applications **2**, 202-223 (2009).

https://doi.org/10.4208/eajam.260316.061016a Published online by Cambridge University Press