Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2013) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2013

Iterative Methods for Pricing American Options under the Bates Model

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

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

Abstract

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

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

1. Introduction

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

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

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

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

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

2. Option Pricing Model

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

$$\begin{aligned} ds &= (\mu - \lambda\xi)s\,dt + \sqrt{y}s\,dw_1 + (J-1)s\,dn, \\ dy &= \kappa(\theta - y)dt + \sigma\sqrt{y}\,dw_2. \end{aligned}$$

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

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

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

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

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

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

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

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

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

For the computations, the unbounded domain is truncated to

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

with sufficiently large S and Y.

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

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

We impose the boundary conditions

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

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

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

3. Discretization

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

where

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

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

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

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

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

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

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

The spatial discretization leads to a semi-discrete LCP

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

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

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

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

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

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

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

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

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

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

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

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

4. The solution of LCPs

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

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

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

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

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

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

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

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

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

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

Algorithm One iteration of PSOR($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)

```
for $i = 1, \dots, \mathrm{dim}\boldsymbol{B}$
    $\boldsymbol{r}_i = \boldsymbol{b}_i - \sum_{j=1}^{\mathrm{dim}\boldsymbol{B}} \boldsymbol{B}_{ij}\boldsymbol{u}_j$
    $\boldsymbol{u}_i = \boldsymbol{u}_i + \omega \boldsymbol{r}_i / \boldsymbol{B}_{ii}$
    $\boldsymbol{u}_i = \max(\boldsymbol{u}_i, \boldsymbol{g}_i)$
end for
```

Algorithm One iteration of PAMG($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)

```
if coarsest level then
    solve LCP($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
else
    PS($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
    $\boldsymbol{u}^c = \boldsymbol{0}$
    $\boldsymbol{r}^c = \tilde{\boldsymbol{R}}(\boldsymbol{b} - \boldsymbol{B}\boldsymbol{u})$
    $\boldsymbol{g}^c = \hat{\boldsymbol{R}}(\boldsymbol{g} - \boldsymbol{u})$
    PAMG($\boldsymbol{B}^c, \boldsymbol{u}^c, \boldsymbol{r}^c, \boldsymbol{g}^c$)
    $\boldsymbol{u} = \boldsymbol{u} + \tilde{\boldsymbol{P}}\boldsymbol{u}^c$
    PS($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
end if
```

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

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

Algorithm

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

5. Numerical Experiment

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

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

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

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

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

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

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

For the iterations, we use the stopping criterion

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

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

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

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

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

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

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

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

6. Conclusions

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

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

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Political Economy 81 (1973) 637–654.

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

[3] S. G. Kou, A jump-diffusion model for option pricing, Management Sci. 48 (8) (2002) 1086–1101.

[4] S. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, Rev. Financial Stud. 6 (1993) 327–343.

[5] D. S. Bates, Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options, Review Financial Stud. 9 (1) (1996) 69–107.

[6] D. Duffie, J. Pan, K. Singleton, Transform analysis and asset pricing for affine jump-diffusions, Econometrica 68 (6) (2000) 1343–1376.

[7] M. J. Brennan, E. S. Schwartz, The valuation of American put options, J. Finance 32 (1977) 449–462.

[8] S. Ikonen, J. Toivanen, Pricing American options using LU decomposition, Appl. Math. Sci. 1 (49-52) (2007) 2529–2551.

[9] N. Clarke, K. Parrott, Multigrid for American option pricing with stochastic volatility, Appl. Math. Finance 6 (1999) 177–195.

[10] S. Ikonen, J. Toivanen, Componentwise splitting methods for pricing American options under stochastic volatility, Int. J. Theor. Appl. Finance 10 (2) (2007) 331–361.

[11] S. Ikonen, J. Toivanen, Efficient numerical methods for pricing American options under stochastic volatility, Numer. Methods Partial Differential Equations 24 (1) (2008) 104–126.

[12] K. Ito, J. Toivanen, Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility, SIAM J. Sci. Comput. 31 (4) (2009) 2646–2664.

[13] C. W. Oosterlee, On multigrid for linear complementarity problems with application to American-style options, Electron. Trans. Numer. Anal. 15 (2003) 165–185.

[14] R. Zvan, P. A. Forsyth, K. R. Vetzal, Penalty methods for American options with stochastic volatility, J. Comput. Appl. Math. 91 (2) (1998) 199–218.

[15] A. Almendral, C. W. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl. Numer. Math. 53 (1) (2005) 1–18.

[16] L. Andersen, J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Rev. Deriv. Res. 4 (3) (2000) 231–262.

[17] R. Cont, E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM Numer. Anal. 43 (4) (2005) 1596–1626.

[18] Y. d'Halluin, P. A. Forsyth, K. R. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, IMA J. Numer. Anal. 25 (1) (2005) 87–112.

[19] A.-M. Matache, C. Schwab, T. P. Wihler, Fast numerical solution of parabolic integrodifferential equations with applications in finance, SIAM J. Sci. Comput. 27 (2) (2005) 369–393.

[20] J. Toivanen, Numerical valuation of European and American options under Kou's jump-diffusion model, SIAM J. Sci. Comput. 30 (4) (2008) 1949–1970.

[21] C. Chiarella, B. Kang, G. H. Meyer, A. Ziogas, The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines, Int. J. Theor. Appl. Finance 12 (3) (2009) 393–425.

[22] J. Toivanen, A componentwise splitting method for pricing American options under the Bates model, in: Applied and numerical partial differential equations, Vol. 15 of Comput. Methods Appl. Sci., Springer, New York, 2010, pp. 213–227.

[23] L. Feng, V. Linetsky, Pricing options in jump-diffusion models: an extrapolation approach, Oper. Res. 56 (2) (2008) 304–325.

[24] R. Rannacher, Finite element solution of diffusion problems with irregular data, Numer. Math. 43 (2) (1984) 309–327.

[25] S. Salmi, J. Toivanen, An iterative method for pricing American options under jump-diffusion models, Appl. Numer. Math. 61 (7) (2011) 821–831.

[26] J. Toivanen, C. W. Oosterlee, A projected algebraic multigrid method for linear complementarity problems, Numer. Math. Theory Methods Appl. 5 (1) (2012) 85–98.

[27] M. B. Giles, R. Carter, Convergence analysis of Crank-Nicolson and Rannacher time-marching, J. Comput. Finance 9 (2006) 89–112.

[28] C. W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, SIAM J. Control 9 (1971) 385–392.

[29] R. U. Seydel, Tools for computational finance, 4th Edition, Universitext, Springer-Verlag, Berlin, 2009.

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

[31] D. M. Young, Iterative solution of large linear systems, Academic Press, New York, 1971.

[32] U. Trottenberg, C. W. Oosterlee, A. Schüller, Multigrid, Academic Press Inc., San Diego, CA, 2001, with contributions by A. Brandt, P. Oswald and K. Stüben.

[33] A. Brandt, C. W. Cryer, Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems, SIAM J. Sci. Statist. Comput. 4 (4) (1983) 655–684.

[34] R. H. W. Hoppe, Multigrid algorithms for variational inequalities, SIAM J. Numer. Anal. 24 (5) (1987) 1046–1065.

[35] M. Holtz, A. Kunoth, B-spline-based monotone multigrid methods, SIAM J. Numer. Anal. 45 (3) (2007) 1175–1199.

[36] S. Ikonen, J. Toivanen, Operator splitting methods for pricing American options under stochastic volatility, Numer. Math. 113 (2) (2009) 299–324.

[37] A. Ramage, L. von Sydow, A multigrid preconditioner for an adaptive Black-Scholes solver, BIT 51 (1) (2011) 217–233.

[38] C. Reisinger, G. Wittum, On multigrid for anisotropic equations and variational inequalities: Pricing multi-dimensional European and American options, Comput. Vis. Sci. 7 (3-4) (2004) 189–197.

[39] J. W. Ruge, K. Stüben, Algebraic multigrid, in: Multigrid methods, Vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.

[40] K. Stüben, Algebraic multigrid: An introduction with applications, in: Multigrid, Academic Press Inc., San Diego, CA, 2001.

[41] K. Goto, R. A. van de Geijn, Anatomy of high-performance matrix multiplication, ACM Trans. Math. Software 34 (3) (2008) Art. 12, 25.