A Componentwise Splitting Method for Pricing American Options Under the Bates Model

Jari Toivanen

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

Summary. A linear complementarity problem (LCP) is formulated for the price of American options under the Bates model which combines the Heston stochastic volatility model and the Merton jump-diffusion model. A finite difference discretization is described for the partial derivatives and a simple quadrature is used for the integral term due to jumps. A componentwise splitting method is generalized for the Bates model. It is leads to solution of sequence of one-dimensional LCPs which can be solved very efficiently using the Brennan and Schwartz algorithm. The numerical experiments demonstrate the componentwise splitting method to be essentially as accurate as the PSOR method, but order of magnitude faster. Furthermore, pricing under the Bates model is less than twice more expensive computationally than under the Heston model in the experiments.

1 Introduction

During the last couple of decades, the trading of options has grown to tremendous scale. The most basic options give either the right to sell (put) or buy (call) the underlying asset with the strike price. European options can be exercised only at the expiry time while American options can be exercised any time before the expiry. Usually American options need to be priced numerically due to the early exercise possibility. One approach is to formulate a linear complementarity problem (LCP) or variational inequality with a partial (integro-)differential operator for the price and then solve it numerically after discretization. Since the books by Glowinski, Lions, and Trémolières [17] and by Glowinski [14], these problems have been extensively studied.

For pricing options, a model is needed for the behavior of the value of the underlying asset. Many such models of varying complexity have been developed. More complicated models reproduce more realistic paths for the value and match between the market price and model prices of options is better, but they also make pricing more challenging. In the Black–Scholes model [5], the value is a geometric Brownian motion. The Merton model [26] adds log-normally distributed jumps to the Black–Scholes model while in the Kou

W. Fitzgibbon et al. (eds.), *Applied and Numerical Partial Differential Equations*,
Computational Methods in Applied Sciences 15, DOI 10.1007/978-90-481-3239-3_16,

model [23], the jumps are log-doubly-exponentially distributed. The Heston model [19] makes the volatility also stochastic in the Black–Scholes model. The Bates model [4] which is also sometimes called as the Heston–Merton model adds to the Heston model log-normally distributed jumps. The correlated jump model [12] allows also the volatility in the Bates model to jump.

Many methods have been proposed for solving the resulting LCPs. The Brennan and Schwartz algorithm [6] is a direct method for pricing American options under the Black–Scholes model; see also [21]. Numerical methods pricing under the Heston model have been considered in [8, 20, 22, 27, 35]. The treatment of the jumps in the Merton and Kou models have been studied in [2, 3, 9, 10, 25, 32]. Pricing under the Bates model has been considered in [7] and under the correlated jump model in [13].

In this paper, we consider pricing American call options under the Bates model. We discretize the spatial partial derivatives in the resulting partial integro-differential operator using a seven-point finite difference stencil. The integral term is discretized using a simple quadrature. The Rannacher scheme [29] is employed in the time stepping. We treat the LCP by introducing a generalization for the componentwise splitting method in [20]. The numerical experiments demonstrate that the proposed method is orders of magnitude faster than the PSOR method.

The outline of the paper is the following. The Bates model and an LCP for an American call option is described in Section 2. The discretization of LCPs is constructed in Section 3. The componentwise splitting method is proposed in Section 4. Numerical experiments are presented in Section 5 and conclusions are given in Section 6.

2 Option Pricing Model

In the following, we give coupled stochastic differential equations describing the Bates model. Then, we give an LCP for the price of an American call option when the market prices of the volatility and jump risks are zero.

2.1 Bates Model

The Bates stochastic volatility model with jumps [4] combines the Merton jump model [26] and the Heston stochastic volatility model [19]. It describes the behavior of the asset value x and its variance y by the coupled stochastic differential equations

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

where 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. The jump size J is taken from a distribution

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

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

2.2 Linear Complementarity Problem for American Options

We define a partial integro-differential operator L acting on a price function u as

$$\begin{aligned} Lu = u_\tau - \frac{1}{2} y x^2 u_{xx} - \rho\sigma y x u_{xy} - \frac{1}{2}\sigma^2 y u_{yy} - (r - q - \lambda\xi) x u_x \\ - \kappa(\theta - y) u_y + (r + \lambda) u - \lambda \int_0^\infty u(Jx, y, \tau) f(J) dJ, \end{aligned} \tag{3}$$

where $\tau = T - t$ is the time to expiry and q is the dividend yield. For computations, the unbounded domain is truncated to be

$$(x, y, \tau) \in (0, X) \times (0, Y) \times (0, T] \tag{4}$$

with sufficiently large X and Y.

The initial value for u is defined by the payoff function $g(x, y)$ 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

$$g(x, y) = \max\{x - K, 0\}, \quad x \in (0, X),\ y \in (0, Y). \tag{5}$$

The price u of an American option satisfies an LCP

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

We pose the boundary conditions

$$\begin{gathered} u(0, y, \tau) = g(0, y), \quad u(X, y, \tau) = g(X, y), \quad y \in (0, Y), \\ u_y(x, Y, \tau) = 0, \quad x \in (0, X). \end{gathered} \tag{7}$$

Beyond the boundary $x = X$, the price u is approximated to be the same as the payoff g, that is, $u(x, y, \tau) = g(x, y)$ for $x \geq X$. On the boundary $y = 0$, the LCP (6) holds and no additional boundary condition needs to be posed.

3 Discretization

We approximate the price u on a space–time grid defined by the grid points (x_i, y_j, τ_k), $0 \le i \le m$, $0 \le j \le n$, $0 \le k \le l$.

3.1 Discretization of Spatial Differential Operator

We use a uniform space grid with the grid steps in the x-direction and y-direction being $\Delta x = X/m$ and $\Delta y = Y/n$, respectively. Figure 1 shows a coarse space grid. A semidiscrete approximation for the price u is given by the time-dependent grid point values

$$u_{i,j}(\tau) \approx u(x_i, y_j, \tau) = u(i\Delta x, j\Delta y, \tau), \quad 0 \le i \le m,\ 0 \le j \le n. \tag{8}$$

We need to discretize the spatial partial derivatives in L given by

$$a_{11} u_{xx} + a_{12} u_{xy} + a_{22} u_{yy} + b_1 u_x + b_2 u_y + cu, \tag{9}$$

where

$$\begin{aligned} a_{11} &= -\frac{1}{2} y x^2, & a_{12} &= -\rho\sigma y x, & a_{22} &= -\frac{1}{2}\sigma^2 y, \\ b_1 &= -(r - q - \lambda\xi)x, & b_2 &= -\kappa(\theta - y), & c &= r + \lambda. \end{aligned} \tag{10}$$

The spatial partial derivatives are discretized using finite differences. For the non cross-derivatives, we use the standard central difference approximations

$$\begin{aligned} u_{xx}(x_i, y_j, \tau) &\approx \frac{1}{(\Delta x)^2}\left(2u(x_i, y_j, \tau) - u(x_i - \Delta x, y_j, \tau) - u(x_i + \Delta x, y_j, \tau)\right), \\ u_{yy}(x_i, y_j, \tau) &\approx \frac{1}{(\Delta y)^2}\left(2u(x_i, y_j, \tau) - u(x_i - \Delta x, y_j, \tau) - u(x_i + \Delta x, y_j, \tau)\right), \\ u_x(x_i, y_j, \tau) &\approx \frac{1}{2\Delta x}\left(u(x_i + \Delta x, y_j, \tau) - u(x_i - \Delta x, y_j, \tau)\right), \\ u_y(x_i, y_j, \tau) &\approx \frac{1}{2\Delta y}\left(u(x_i + \Delta x, y_j, \tau) - u(x_i - \Delta x, y_j, \tau)\right). \end{aligned} \tag{11}$$

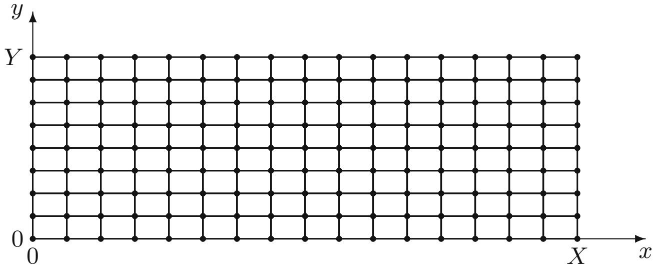


Fig. 1. A coarse 17×9 uniform grid for the computational domain $(0, X) \times (0, Y)$.

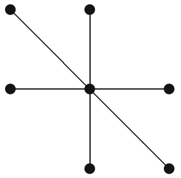


Fig. 2. A seven-point finite difference stencil used with a negative correlation $\rho < 0$ between the Wiener processes for the asset value x and its variance y.

In this paper, we assume that the correlation ρ is negative and we use a seven-point stencil shown in Figure 2. A similar stencil has been described in [7]. For a positive correlation ρ, a suitable seven-point stencil is given in [20, 22]. The cross-derivative u_{xy} is approximated by

$$\begin{aligned} u_{xy}(x_i, y_j, \tau) \approx \frac{1}{2\Delta x \Delta y} &\left(2u(x_i, y_j, \tau) - u(x_i - \Delta x, y_j + \Delta y)\right. \\ &\left. -u(x_i + \Delta x, y_j - \Delta y) - (\Delta x)^2 u_{xx}(x_i, y_j, \tau) - (\Delta y)^2 u_{yy}(x_i, y_j, \tau)\right). \quad (12) \end{aligned}$$

Due to additional derivative terms in (12), we define modified coefficients for u_{xx} and u_{yy} as

$$\tilde{a}_{11} = a_{11} + \frac{1}{2}\frac{\Delta x}{\Delta y}a_{12}, \quad \text{and} \quad \tilde{a}_{22} = a_{22} + \frac{1}{2}\frac{\Delta y}{\Delta x}a_{12}. \quad (13)$$

It is well-known that the central finite differences can lead to positive weights in difference stencil when the convection dominates the diffusion. To avoid positive weights, we add some artificial diffusion according to

$$\hat{a}_{11} = \min\left\{\tilde{a}_{11}, -\frac{1}{2}b_1\Delta x, \frac{1}{2}b_1\Delta x\right\} \quad (14)$$

and

$$\hat{a}_{22} = \min\left\{\tilde{a}_{22}, -\frac{1}{2}b_2\Delta y, \frac{1}{2}b_2\Delta y\right\}. \quad (15)$$

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

3.2 Discretization of Integral Term

The integral term due to the jumps in (3) needs to computed at each grid point $x = x_i$. We denoted it by

$$I_i = \int_0^\infty u(Jx_i, y, \tau) f(J) dJ. \quad (16)$$

In order to perform the integration, we make a change of variable $J = e^s$ which leads to

$$I_i = \int_{-\infty}^{\infty} u(e^s x_i, y, \tau) p(s) ds, \tag{17}$$

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

$$p(s) = \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{[s - (\gamma - \delta^2/2)]^2}{2\delta^2} \right). \tag{18}$$

We decompose I_i into integrals over grid intervals as

$$I_i = \sum_{j=0}^{n-1} I_{i,j} + \int_{\ln x_n - \ln x_i}^{\infty} g(e^s x_i, y) p(s) ds, \tag{19}$$

where

$$I_{i,j} = \int_{\ln x_{j+1} - \ln x_i}^{\ln x_j - \ln x_i} u(e^s x_i, y, \tau) p(s) ds. \tag{20}$$

The price function $u(x, y, \tau)$ needs to be approximated between each grid point pair (x_i, x_{i+1}). For this, we use a piecewise linear interpolation

$$u(x, y, \tau) \approx \frac{x_{i+1} - x}{x_{i+1} - x_i} u(x_i, y, \tau) + \frac{x - x_i}{x_{i+1} - x_i} u(x_{i+1}, y, \tau) \tag{21}$$

for $x \in [x_j, x_{j+1}]$.

By performing the integration, we obtain

$$\begin{aligned} I_{i,j} \approx \frac{e^\gamma}{2} &\left[\operatorname{erf}\left(\frac{s_{i,j+1} - \gamma - \delta^2/2}{\delta\sqrt{2}} \right) - \operatorname{erf}\left(\frac{s_{i,j} - \gamma - \delta^2/2}{\delta\sqrt{2}} \right) \right] \alpha_j x_i \\ &+ \frac{1}{2} \left[\operatorname{erf}\left(\frac{s_{i,j+1} - \gamma + \delta^2/2}{\delta\sqrt{2}} \right) - \operatorname{erf}\left(\frac{s_{i,j} - \gamma + \delta^2/2}{\delta\sqrt{2}} \right) \right] \beta_j x_i, \end{aligned} \tag{22}$$

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

$$\alpha_j = \frac{u(x_{j+1}, y, \tau) - u(x_j, y, \tau)}{x_{j+1} - x_j}, \quad \text{and} \quad \beta_j = \frac{u(x_j, y, \tau) x_{j+1} - u(x_{j+1}, y, \tau) x_j}{x_{j+1} - x_j}. \tag{23}$$

3.3 Semidiscrete LCP

The space discretization leads to an LCP

$$\begin{cases} \mathbf{u}_\tau + \mathbf{A}\mathbf{u} + \mathbf{a} \geq \mathbf{0}, \quad \mathbf{u} \geq \mathbf{g}, \\ (\mathbf{u}_\tau + \mathbf{A}\mathbf{u} + \mathbf{a})^T (\mathbf{u} - \mathbf{g}) = 0, \end{cases} \tag{24}$$

where $\mathbf{A}$ is $(m+1)(n+1) \times (m+1)(n+1)$ matrix, $\mathbf{a}$ is a vector resulting from the second term in (19), $\mathbf{u}$ and $\mathbf{g}$ are vectors containing the grid point values of the price u and the payoff g, respectively. In the above LCP, the inequalities hold componentwise. The entries in the rows of $\mathbf{A}$ corresponding to the grid points on the boundaries $x = 0$ and $x = X$ are set to zero. The submatrix of $\mathbf{A}$ corresponding to the grid points not on the boundaries $x = 0$ and $x = X$ is an M-matrix. When the numbering of the grid points first goes through the grid points in the x-direction and then in the y-direction, the $(n+1) \times (n+1)$ diagonal blocks of $\mathbf{A}$ are essentially full matrices due to the jump term.

3.4 Time Discretization

We use the Rannacher scheme [29] with nonuniform time steps. It takes a few first time steps with the implicit Euler method and then it uses the Crank–Nicolson method. This leads to better stability properties than using just the Crank–Nicolson method. The solution vector $\mathbf{u}$ is approximated at times

$$\tau_k = \begin{cases} \left(\dfrac{k}{2l}\right)^2 T, & k = 0, 1, 2, 3, \\ \left(\dfrac{k-2}{l-2}\right)^2 T, & k = 4, 5, \ldots, l. \end{cases} \tag{25}$$

In order to simplify the following notations, we define time step sizes $\Delta\tau_k = \tau_{k+1} - \tau_k$, $k = 0, 1, \ldots, l-1$.

In order to simplify the notations in the following, we denote by $\mathrm{LCP}(\mathbf{B}, \mathbf{u}, \mathbf{b}, \mathbf{g})$ the linear complementarity problem

$$\begin{cases} (\mathbf{B}\mathbf{u} - \mathbf{b}) \geq \mathbf{0}, \quad \mathbf{u} \geq \mathbf{g}, \\ (\mathbf{B}\mathbf{u} - \mathbf{b})^T (\mathbf{u} - \mathbf{g}) = 0. \end{cases} \tag{26}$$

The Rannacher time stepping leads to the solution of the following sequence of LCPs:

$$\mathrm{LCP}(\mathbf{B}^{(k+1)}, \mathbf{u}^{(k+1)}, \mathbf{b}^{(k+1)}, \mathbf{g}), \tag{27}$$

where $\mathbf{u}^{(k)}$ denotes the vector $\mathbf{u}$ at the time τ_k. For the first four time steps $k = 0, 1, 2, 3$, we use the implicit Euler method defined by

$$\mathbf{B}^{(k+1)} = \mathbf{I} + \Delta\tau_k \mathbf{A} \qquad \text{and} \qquad \mathbf{b}^{(k+1)} = \Delta\tau_k \mathbf{u}^{(k)} - \Delta\tau_k \mathbf{a}. \tag{28}$$

The rest of the time steps $k = 4, 5, \ldots, l-1$ are performed using the Crank–Nicolson method defined by

$$\mathbf{B}^{(k+1)} = \mathbf{I} + \frac{1}{2}\Delta\tau_k \mathbf{A} \qquad \text{and} \qquad \mathbf{b}^{(k+1)} = \left(\mathbf{I} - \frac{1}{2}\Delta\tau_k \mathbf{A}\right)\mathbf{u}^{(k)} - \Delta\tau_k \mathbf{a}. \tag{29}$$

4 Componentwise Splitting Method

Componentwise splitting methods are inspired by ADI (Alternating Direction Implicit) schemes which were introduced in [11, 28]. Instead of treating a part of operator explicitly, we use fully implicit splittings considered in [15, 16, 24, 33], for example. For the Heston model, the componentwise splitting method were introduced in [20] with a positive correlation ρ. In [7], the splitting method was considered in the case of a negative correlation.

The matrix $\mathbf{A}$ is split into three matrices which correspond to the couplings in the x-direction, y-direction, and diagonal direction. Figure 3 shows the matrix splitting and also the corresponding splitting of the finite difference stencil. The simplest fractional step method based on the implicit Euler method is given in Figure 4. The formal accuracy of this method is $\mathcal{O}(\Delta\tau_{l-1}) = \mathcal{O}\left(\frac{1}{l}\right)$.

We increase the accuracy of the splitting method by performing a Strang symmetrization [30] and use the Crank–Nicolson method; see also [15]. This leads one time step to have the following fractional steps:

$$\begin{aligned}
&\text{Step 1. LCP}\left(\mathbf{I}+\frac{\Delta\tau_k}{4}\mathbf{A}_y, \mathbf{u}^{(k+1/5)}, \left(\mathbf{I}-\frac{\Delta\tau_k}{4}\mathbf{A}_y\right)\mathbf{u}^{(k)}, \mathbf{g}\right)\\
&\text{Step 2. LCP}\left(\mathbf{I}+\frac{\Delta\tau_k}{4}\mathbf{A}_d, \mathbf{u}^{(k+2/5)}, \left(\mathbf{I}-\frac{\Delta\tau_k}{4}\mathbf{A}_y\right)\mathbf{u}^{(k+1/5)}, \mathbf{g}\right)\\
&\text{Step 3. LCP}\left(\mathbf{I}+\frac{\Delta\tau_k}{2}\mathbf{A}_x, \mathbf{u}^{(k+3/5)}, \left(\mathbf{I}-\frac{\Delta\tau_k}{2}\mathbf{A}_x\right)\mathbf{u}^{(k+2/5)} - \Delta\tau_k\mathbf{a}, \mathbf{g}\right)\\
&\text{Step 4. LCP}\left(\mathbf{I}+\frac{\Delta\tau_k}{4}\mathbf{A}_d, \mathbf{u}^{(k+4/5)}, \left(\mathbf{I}-\frac{\Delta\tau_k}{4}\mathbf{A}_y\right)\mathbf{u}^{(k+3/5)}, \mathbf{g}\right)\\
&\text{Step 5. LCP}\left(\mathbf{I}+\frac{\Delta\tau_k}{4}\mathbf{A}_y, \mathbf{u}^{(k+1)}, \left(\mathbf{I}-\frac{\Delta\tau_k}{4}\mathbf{A}_y\right)\mathbf{u}^{(k+4/5)}, \mathbf{g}\right)
\end{aligned}$$

In order to maintain the good stability of the Rannacher scheme, we use the implicit Euler method instead the Crank–Nicolson method for the first four time steps $k = 0, 1, 2, 3$ in the above symmetrized splitting method.

4.1 Solution of One-Dimensional LCPs

For an American call option, typical early exercise boundaries at different times are shown in Figure 5. The boundary can be described by a relation

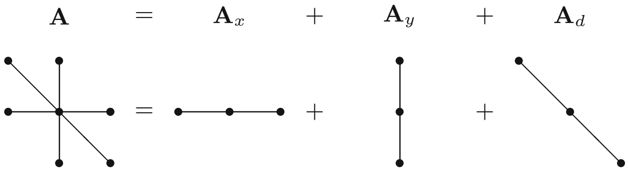
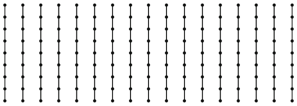
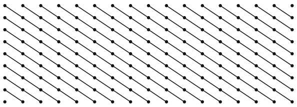


Fig. 3. The matrix splitting of $\mathbf{A}$ and the corresponding splitting of the finite difference stencil.

1. $\mathrm{LCP}(\mathbf{I} + \Delta\tau_k \mathbf{A}_y, \mathbf{u}^{(k+1/3)}, \Delta\tau_k \mathbf{u}^{(k)}, \mathbf{g})$
   Solve the sequence of one-dimensional LCPs:



2. $\mathrm{LCP}(\mathbf{I} + \Delta\tau_k \mathbf{A}_d, \mathbf{u}^{(k+2/3)}, \Delta\tau_k \mathbf{u}^{(k+1/3)}, \mathbf{g})$
   Solve the sequence of one-dimensional LCPs:



3. $\mathrm{LCP}(\mathbf{I} + \Delta\tau_k \mathbf{A}_x, \mathbf{u}^{(k+1)}, \Delta\tau_k \mathbf{u}^{(k+2/3)} - \Delta\tau_k \mathbf{a}, \mathbf{g})$
   Solve the sequence of one-dimensional LCPs:

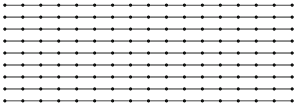


Fig. 4. Three fractional splitting steps for performing the time step from τ_k to τ_{k+1}.

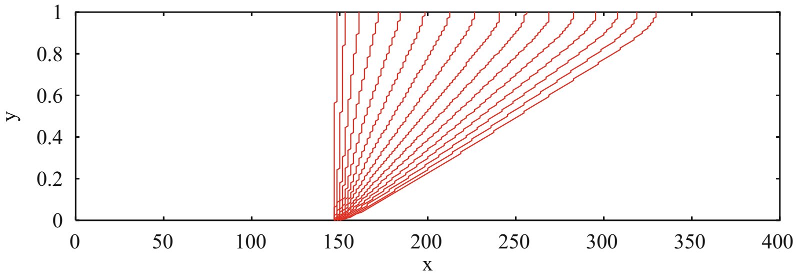


Fig. 5. The time evolution of the early exercise boundaries for an American call option.

$y = h(x, \tau)$, where h an increasing function with respect to x. Thus, a given point (x, y, τ) belongs to

- The hold region if $y > h(x, \tau)$ or
- The early exercise region if $y \leq h(x, \tau)$

Similarly, the early exercise boundary divides each x-directional line, y-directional line, and $(1, -1)$-directional line into two parts. Due to this solution structure and the tridiagonal matrices defining the LCPs in the y-direction and $(1,-1)$-direction, the Brennan and Schwartz algorithm can be used to solve these problems. The LCPs in the x-direction have full matrices due to the integral term. An iterative solution procedure for these problems is described in the end of this section.

Brennan and Schwartz Algorithm

The Brennan and Schwartz algorithm for American put options under the Black–Scholes model was described in [6]. The algorithm can be modified to use a standard **LU**-decomposition [1, 21]. We formulate it for a tridiagonal linear complementarity problem:

$$\mathbf{T}\mathbf{x} = \begin{pmatrix} \mathbf{T}_{1,1} & \mathbf{T}_{1,2} & & \\ \mathbf{T}_{2,1} & \ddots & \ddots & \\ & \ddots & \mathbf{T}_{m-1,m-1} & \mathbf{T}_{m-1,m} \\ & & \mathbf{T}_{m,m-1} & \mathbf{T}_{m,m} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{pmatrix} \geq \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{pmatrix} = \mathbf{b}, \qquad (30)$$

$$\mathbf{x} \geq \mathbf{g}, \qquad (\mathbf{T}\mathbf{x} - \mathbf{b})^T(\mathbf{x} - \mathbf{g}) = 0. \qquad (31)$$

The Brennan and Schwartz algorithm assumes the solution $\mathbf{x}$ to be such that for some integer k it holds that

$$\begin{aligned} \mathbf{x}_i &> g_i, \quad i = 1, \ldots, k, \qquad \text{and} \\ \mathbf{x}_i &= g_i, \quad i = k+1, \ldots, m. \end{aligned} \qquad (32)$$

The algorithm with **LU**-decomposition is described as follows:

```
*Brennan and Schwartz algorithm*
Computation of **LU**-decomposition and forward substitution:
$\mathbf{U}_{1,1} = \mathbf{T}_{1,1}$
$\mathbf{y}_1 = \mathbf{b}_1$
Do $i = 2, \ldots, m$
    $\mathbf{L}_{i,i-1} = \mathbf{T}_{i,i-1}/\mathbf{U}_{i-1,i-1}$
    $\mathbf{U}_{i-1,i} = \mathbf{T}_{i-1,i}$
    $\mathbf{U}_{i,i} = \mathbf{T}_{i,i} - \mathbf{L}_{i,i-1}\mathbf{U}_{i-1,i}$
    $\mathbf{y}_i = \mathbf{b}_i - \mathbf{L}_{i,i-1}\mathbf{y}_{i-1}$
End Do
Backward substitution with a projection:
$\mathbf{x}_m = \mathbf{y}_m/\mathbf{U}_{m,m}$
$\mathbf{x}_m = \max\{\mathbf{x}_m, \mathbf{g}_m\}$
Do $i = m-1, \ldots, 1$
    $\mathbf{x}_i = (\mathbf{y}_i - \mathbf{U}_{i,i+1}\mathbf{x}_{i+1})/\mathbf{U}_{i,i}$
    $\mathbf{x}_i = \max\{\mathbf{x}_i, \mathbf{g}_i\}$
End Do
```

In this algorithm the only modification to a standard solution with **LU**-decomposition is the additional projection in the backward substitution.

After a suitable numbering of unknowns the assumption (32) holds for the one-dimensional LCPs in all three directions. The Brennan and Schwartz algorithm can be use directly to solve the one-dimensional LCPs in the y-direction and in the $(1,-1)$-direction.

LCPs with Full Matrices Associated to the $\mathbf{x}$*-Direction*

A matrix associated to one-dimensional LCP in the x-direction is denoted by $\mathbf{B}$. It has a regular splitting [34]

$$\mathbf{B} = \mathbf{T} - \mathbf{J}, \tag{33}$$

where $-\mathbf{J}$ is a full matrix resulting from the integral term and $\mathbf{T}$ is the rest which a tridiagonal matrix. We generalize a fixed point iteration described in [31] and used in [2,10,32]. The fixed point iteration for LCP$(\mathbf{B}, \mathbf{x}, \mathbf{b}, \mathbf{g})$ reads

$$\mathrm{LCP}(\mathbf{T}, \mathbf{x}^{j+1}, \mathbf{b} + \mathbf{J}\mathbf{x}^{j}, \mathbf{g}), \quad j = 0, 1, \dots \tag{34}$$

Each iteration requires the solution of an LCP with the tridiagonal $\mathbf{T}$ and the multiplication of a vector by $\mathbf{J}$. The Brennan and Schwartz algorithm can be used to solve the LCPs (34). The iteration converges very rapidly and only a couple of iterations are needed to reach sufficient accuracy for practical purposes.

5 Numerical Experiments

In the numerical experiments for call options, we use the model parameter values:

- The risk free interest rate $r = 0.03$
- The dividend yield $q = 0.05$
- The strike price $K = 100$
- The correlation between the price and variance processes $\rho = -0.5$
- The mean level of the variance $\theta = 0.04$
- The rate of reversion to the mean level $\kappa = 2.0$
- The volatility of the variance $\sigma = 0.25$
- The jump rate $\lambda = 0.2$
- The mean jump $\gamma = -0.5$
- The variance of jump $\delta = 0.4$

The computational domain is $(x, y, \tau) \in [0, 400] \times [0, 1] \times [0, 0.5]$.

Our first experiment compares the PSOR method and the Strang symmetrized componentwise splitting method for call options under the Heston model, that is, $\lambda = 0$. In this case, the LCPs in the x-direction are tridiagonal and they can be solved using the Brennan and Schwartz algorithm without the iteration (34).

Table 1. The numerical results for the Heston model

Method	Grid (m, n, l)	Iteration	Error	Ratio	CPU
PSOR	$(64, 32, 8)$	34.6	0.14470		0.05
	$(128, 64, 16)$	42.3	0.05607	2.58	0.48
	$(256, 128, 32)$	95.3	0.01006	5.58	8.18
	$(512, 256, 64)$	196.6	0.00350	2.87	128.51
	$(1024, 512, 128)$	372.2	0.00066	5.31	1890.76
Componentwise splitting	$(64, 32, 8)$		0.14412		0.01
	$(128, 64, 16)$		0.05621	2.56	0.06
	$(256, 128, 32)$		0.01019	5.51	0.51
	$(512, 256, 64)$		0.00355	2.87	6.36
	$(1024, 512, 128)$		0.00067	5.28	58.27

Table 1 reports the numerical results. It (and also Table 2) has the following columns: Grid (m, n, l) defines the number of grid steps in x, y, and τ to be m, n, and l, respectively. Iteration gives the average number of PSOR iterations on each time step with the relaxation parameter $\omega = 1.5$. With the componentwise splitting method iteration specifies the number of iterations (34) to solve the LCPs in the x-direction at each time step. Error column gives the root mean square relative error given by

$$\text{error} = \left[\frac{1}{5} \sum_{i=1}^{5} \left(\frac{u(\mathbf{x}_i, \theta, T) - U(\mathbf{x}_i, \theta, T)}{U(\mathbf{x}_i, \theta, T)} \right)^2 \right]^{1/2}, \tag{35}$$

where $\mathbf{x} = (80, 90, 100, 110, 120)^T$ and U is the reference price. Ratio is the ratio of the consecutive root mean square relative errors. CPU gives the CPU time in seconds on a 3.8 GHz Intel Xeon PC. The reference prices under the Heston model at $(\mathbf{x}_i, \theta, T)$, $i = 1, 2, \ldots, 5$, are 0.131563, 1.255396, 4.999888, 11.680219, 20.325463 which were computed using the componentwise splitting method on the grid $(4096, 2048, 512)$.

We can observe from Table 1 that the discretizations with both methods appears to be roughly second-order accurate as the ratio is four on average. Furthermore, the splitting increases the error only about 2%. On the coarsest grid, the splitting method is five times faster than the PSOR method, and on the finest grid it is 32 times faster.

In our second experiment, we performed the same comparison under the Bates model. The reference prices computed using the componentwise splitting method on the grid $(4096, 2048, 512)$ are 0.328526, 2.109397, 6.711622, 13.749337, 22.143307. In the componentwise splitting method, the LCPs in the x-direction lead to full matrices and the iteration (34) is employed to solve them. Based on a few experiments, we observed that already after two iterations the accuracy is sufficient. Thus, we use two iterations in our comparison. The multiplication by the matrix $\mathbf{J}$ is the most expensive operation in the iteration. In order to perform it efficiently, we collected all n multiplications

Table 2. The numerical results for the Bates model

Method	Grid (m, n, l)	Iteration	Error	Ratio	CPU
PSOR	$(64, 32, 8)$	39.6	0.10887		0.16
	$(128, 64, 16)$	48.1	0.03803	2.86	2.14
	$(256, 128, 32)$	108.2	0.00670	5.68	138.92
	$(512, 256, 64)$	222.6	0.00209	3.20	8605.09
	$(1024, 512, 128)$	420.5	0.00034	6.13	275191.73
Componentwise splitting	$(64, 32, 8)$	2.0	0.10833		0.01
	$(128, 64, 16)$	2.0	0.03790	2.86	0.09
	$(256, 128, 32)$	2.0	0.00668	5.67	0.81
	$(512, 256, 64)$	2.0	0.00210	3.19	10.18
	$(1024, 512, 128)$	2.0	0.00035	6.07	109.45

corresponding to all x-grid lines together and then performed the resulting matrix–matrix multiplication using the optimized GotoBLAS library [18].

The numerical results under the Bates model are given in Table 2. Absolute errors are comparable to the ones under the Heston model, but as the option prices are higher under the Bates model the relative errors reported in the table are smaller. Again roughly second-order accuracy is observed with both methods. The CPU times with the componentwise splitting method were less than twice the times under the Heston model. The componentwise splitting method is 16 times faster on the coarsest grid, and it is about 2,500 times faster on the finest grid. On finer grids, the PSOR method leads to infeasible CPU times while the times with componentwise splitting method are still reasonable.

6 Conclusions

We described a linear complementarity problem (LCP) for pricing American options under the Bates model and we considered a finite difference discretization. We proposed a componentwise splitting method to solve approximately the LCPs. It leads to a sequence of LCPs with tridiagonal matrices. The Brennan and Schwartz algorithm can solve these LCPs efficiently.

Our numerical experiments showed that the additional splitting error do not essentially increase the discretization error. The componentwise splitting method is orders of magnitude faster than the PSOR method under the Bates model. Pricing under the Bates model was at most two times more expensive than under the Heston model with the componentwise splitting method.

Acknowledgement. The author thanks Prof. Roland Glowinski for introduction and discussions on operator splitting methods. The author is grateful to Dr. Samuli Ikonen for many fruitful discussions on numerical methods for option pricing.

References

1. Y. Achdou and O. Pironneau. *Computational methods for option pricing*, volume 30 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, PA, 2005.
2. A. Almendral and C. W. Oosterlee. Numerical valuation of options with jumps in the underlying. *Appl. Numer. Math.*, 53(1):1–18, 2005.
3. L. Andersen and J. Andreasen. Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing. *Rev. Deriv. Res.*, 4(3):231–262, 2000.
4. D. S. Bates. Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options. *Review Financial Stud.*, 9(1):69–107, 1996.
5. F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81:637–654, 1973.
6. M. J. Brennan and E. S. Schwartz. The valuation of American put options. *J. Finance*, 32:449–462, 1977.
7. C. Chiarella, B. Kang, G. H. Mayer, and A. Ziogas. The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines. Research Paper 219, Quantitative Finance Research Centre, University of Technology, Sydney, 2008.
8. N. Clarke and K. Parrott. Multigrid for American option pricing with stochastic volatility. *Appl. Math. Finance*, 6:177–195, 1999.
9. R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. *SIAM Numer. Anal.*, 43(4):1596–1626, 2005.
10. 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(1):87–112, 2005.
11. J. Douglas, Jr. and H. H. Rachford, Jr. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.
12. D. Duffie, J. Pan, and K. Singleton. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68(6):1343–1376, 2000.
13. L. Feng and V. Linetsky. Pricing options in jump-diffusion models: an extrapolation approach. *Oper. Res.*, 56:304–325, 2008.
14. R. Glowinski. *Numerical methods for nonlinear variational problems.* Springer, New York, 1984.
15. R. Glowinski. Finite element methods for incompressible viscous flow. In P. G. Ciarlet and J.-L. Lions, editors, *Handbook of Numerical Analysis, Vol. IX*, pages 3–1176. North-Holland, Amsterdam, 2003.
16. R. Glowinski and P. Le Tallec. *Augmented Lagrangian and operator splitting methods in nonlinear mechanics.* SIAM, Philadelphia, PA, 1989.
17. R. Glowinski, J.-L. Lions, and R. Trémolières. *Analyse numérique des inéquations variationnelles. Tome 1 & 2.* Dunod, Paris, 1976.
18. K. Goto and R. A. van de Geijn. Anatomy of high-performance matrix multiplication. *ACM Trans. Math. Software*, 34(3):Art. 12, 25 pp., 2008.
19. S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financial Stud.*, 6:327–343, 1993.
20. S. Ikonen and J. Toivanen. Componentwise splitting methods for pricing American options under stochastic volatility. *Int. J. Theor. Appl. Finance*, 10:331–361, 2007.

21. S. Ikonen and J. Toivanen. Pricing American options using LU decomposition. *Appl. Math. Sci.*, 1:2529–2551, 2007.
22. S. Ikonen and J. Toivanen. Efficient numerical methods for pricing American options under stochastic volatility. *Numer. Methods Partial Differential Equations*, 24:104–126, 2008.
23. S. G. Kou. A jump-diffusion model for option pricing. *Management Sci.*, 48(8):1086–1101, 2002.
24. G. I. Marchuk. Splitting and alternating direction methods. In P. G. Ciarlet and J.-L. Lions, editors, *Handbook of Numerical Analysis, Vol. I*, pages 197–462. North-Holland, Amsterdam, 1990.
25. A.-M. Matache, C. Schwab, and T. P. Wihler. Fast numerical solution of parabolic integrodifferential equations with applications in finance. *SIAM J. Sci. Comput.*, 27(2):369–393, 2005.
26. R. Merton. Option pricing when underlying stock returns are discontinuous. *J. Financial Econ.*, 3:125–144, 1976.
27. C. W. Oosterlee. On multigrid for linear complementarity problems with application to American-style options. *Electron. Trans. Numer. Anal.*, 15:165–185, 2003.
28. D. W. Peaceman and H. H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.
29. R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numer. Math.*, 43:309–327, 1982.
30. G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.
31. D. Tavella and C. Randall. *Pricing financial instruments: The finite difference method*. John Wiley & Sons, Chichester, 2000.
32. J. Toivanen. Numerical valuation of European and American options under Kou's jump-diffusion model. *SIAM J. Sci. Comput.*, 30:1949–1970, 2008.
33. N. N. Yanenko. *The method of fractional steps. The solution of problems of mathematical physics in several variables*. Springer, New York, 1971.
34. D. M. Young. *Iterative solution of large linear systems*. Academic Press, New York, 1971.
35. R. Zvan, P. A. Forsyth, and K. R. Vetzal. Penalty methods for American options with stochastic volatility. *J. Comput. Appl. Math.*, 91(2):199–218, 1998.