Numer. Math. (2009) 113:299–324
DOI 10.1007/s00211-009-0227-5

Numerische Mathematik

Operator splitting methods for pricing American options under stochastic volatility

Samuli Ikonen · Jari Toivanen

Received: 1 March 2007 / Revised: 4 August 2008 / Published online: 16 May 2009

Abstract We consider the numerical pricing of American options under Heston's stochastic volatility model. The price is given by a linear complementarity problem with a two-dimensional parabolic partial differential operator. We propose operator splitting methods for performing time stepping after a finite difference space discretization. The idea is to decouple the treatment of the early exercise constraint and the solution of the system of linear equations into separate fractional time steps. With this approach an efficient numerical method can be chosen for solving the system of linear equations in the first fractional step before making a simple update to satisfy the early exercise constraint. Our analysis suggests that the Crank–Nicolson method and the operator splitting method based on it have the same asymptotic order of accuracy. The numerical experiments show that the operator splitting methods have comparable discretization errors. They also demonstrate the efficiency of the operator splitting methods when a multigrid method is used for solving the systems of linear equations.

Mathematics Subject Classification (2000) 35K85 · 65M06 · 65M55 · 65Y20 · 91B28

1 Introduction

The option prices obtained using the Black–Scholes model [2] are usually not consistent with the market prices for options. One possible remedy is to make the volatility

S. Ikonen
Nordea Markets, 00020 Nordea, Finland
e-mail: Samuli.Ikonen@nordea.com

J. Toivanen (✉)
Department of Mathematical Information Technology, Agora,
40014 University of Jyväskylä, Jyväskylä, Finland
e-mail: Jari.Toivanen@mit.jyu.fi

to be a function of time and the value of the underlying asset. By calibrating this function consistent option prices can be obtained. Another approach is to assume that the volatility of the price process is also stochastic. Such models have been considered, for example, in [1,13,17,19]. In this paper, we will use the stochastic volatility model introduced by Heston for pricing American options [17]. The stochastic volatility models offer much better match with market prices than the Black–Scholes model, but they still cannot account rapid jump-like movements in price and volatility; see [6] and references therein.

The price of European options can be computed using analytical expressions for Heston's model [17]. The early exercise possibility of American options leads to nonlinear free boundary problems described by partial differential inequalities which are also called in different disciplines as variational inequalities, linear complementarity problems, and obstacle problems. Usually numerical methods are used to approximate the prices of American options. Already Brennan and Schwartz proposed a finite difference discretization for the one-dimensional Black–Scholes equation and then a direct solution method for pricing American options [4]. Huang and Pang consider various iterative methods for resulting linear complementarity problems in [18]. Heston's model has an additional space variable related to the volatility. In numerical approximations, it is usually chosen to be the variance of the price process, which is the square of the volatility. This additional variable makes the discretization and solution procedures more complicated as well as computationally more expensive. In the following, we review three papers and one report which have considered such option pricing problems.

Clarke and Parrott considered the discretization of a partial differential inequality for the price of an American option using finite differences for space derivatives and a slightly stabilized Crank–Nicolson method for the time derivative [8,9]. They performed a coordinate transformation to increase the accuracy of the space discretization with uniform grid step sizes. They proposed a special version of a projected full approximation scheme (PFAS) multigrid [3] for the solving arising linear complementarity problems. Their method uses a special projected line Gauss-Seidel smoother which in part made the considered method rather complicated and problem specific. The advantage of such a multilevel method is that the number of iterations required to solve a linear complementarity problem is essentially independent of the grid step size and, thus, the computational cost of this iterative solution is of optimal order.

Zvan et al. [30] discretized the partial differential inequality using finite element/volume method together with a nonlinear flux limiter for convection terms. They formulated linear complementarity problems using nonlinear penalty terms. The solutions of the arising nonlinear problems were obtained with an inexact Newton method and the linear problems with approximate Jacobians were solved with an incomplete LU preconditioned CGSTAB method. The method proposed by Zvan, Forsyth and Vetzal is simpler than the one by Clarke and Parrott, but the number of iterations required by the CGSTAB method increases when the grid step size decreases.

Oosterlee performed the space discretization using central differences for the diffusion and second-order upwind differences for the convection [25]. The time discretization was based on the second-order accurate backward finite difference scheme. Oosterlee studied the PFAS multigrid for linear complementarity problems. The qual-

ity of various smoothers for the multigrid was analyzed. This lead to the conclusion that only an alternating line smoother is robust. Also a recombination technique for iterates was proposed and tested. It improved the convergence of multigrid methods, but it could not regain a grid step size independent convergence if the multigrid method did not have this property without the recombination. Based on [25], it seems that a robust and efficient PFAS multigrid has to use a rather involved smoother for linear complementarity problems arising from the pricing American options using Heston's model.

Due to the previous observations we will pursue an alternative way to treat the early exercise constraint. Our approach is based on operator splitting methods which are commonly used to handle the incompressibility constraint in computational fluid dynamics [15]. Previously operator splitting methods have been applied for obstacle problems by Lions and Mercier [23], for example. In our earlier paper, [21], we proposed operator splitting methods for pricing American options under the Black–Scholes model. The idea is divide each time step into two fractional time steps. The first step integrates a partial differential equation with an auxiliary variable over the time step. The second step makes a simple update so that the solution and the auxiliary variable satisfies the linear complementarity conditions due to the early exercise constraint. Such a splitting introduces additional error and, thus, it is necessary to show that this error is sufficiently small so that this approach can be used to price options accurately. Our numerical experiments in [21] showed that for the Black–Scholes model the accuracy with a splitting is essentially the same as without it. The computational cost of the operator splitting method is about the same as the cost of the explicit payoff method which has been considered for one-dimensional American option pricing problems in [10,11,20], for example. The difference between these methods is that the operator splitting method uses the auxiliary variable to improve the accuracy.

The purpose of this paper is to introduce operator splitting methods for pricing American options using Heston's model and also study their accuracy and efficiency. We analyse the difference between the solutions obtained using the Crank–Nicolson method and the operator splitting method based on it. We also compare numerically the accuracy of solutions when the operator splitting method and the projected SOR (PSOR) method are used. Our analysis and numerical experiments suggest that the additional error due to the splitting does not reduce the order of accuracy when compared to the corresponding unsplit scheme. The operator splitting method enables the use of efficient solution procedures for the system of linear equations since the early exercise constraint is treated in a separate fractional time step. In the numerical experiment, we apply a multigrid method at the first fractional step of the splitting.

The outline of this paper is the following. In the second section, we describe Heston's model and a partial differential inequality for pricing American options. The third section introduces finite difference discretizations applied for the parabolic partial differential equation. In the fourth section, we propose our operator splitting method in the case of four different time discretization schemes. The next section analyses the operator splitting method based on the Crank–Nicolson method. The multigrid method used in our experiments is described in the following section. The seventh section presents numerical experiments and the last section contains conclusions.

2 Option pricing model

In the following, we describe stock price and variance processes, a partial differential inequality, an initial value, and boundary conditions for the American option pricing model. This section defines a problem whose numerical solution is studied in the consecutive sections. Our formulation and notations are based on [17,25,30].

In Heston's model, stochastic differential equations

$$dx_t = \mu x_t dt + \sqrt{y_t} x_t dw_1, \tag{1}$$

$$dy_t = \alpha(\beta - y_t)dt + \gamma\sqrt{y_t} dw_2, \tag{2}$$

define the stock price process x_t and the variance process y_t. Equation (1) models the stock price process x_t. The parameter μ is the deterministic growth rate of the stock price and $\sqrt{y_t}$ is the standard deviation (the volatility) of the stock returns dx/x. The model for the variance process y_t is given by (2). The volatility of the variance process y_t is denoted by γ and the variance will drift back to a mean value $\beta > 0$ at a rate $\alpha > 0$. These two processes contain randomness as w_1 and w_2 are Brownian motions with a correlation factor $\rho \in [-1, 1]$ [9,30].

A two-dimensional parabolic partial differential inequality can be derived for the price of the American option using the previous stochastic volatility model; see for example [30] and references therein. We define Heston's operator

$$\begin{aligned} Lu = {} & \frac{\partial u}{\partial \tau} - \frac{1}{2} y x^2 \frac{\partial^2 u}{\partial x^2} - \rho\gamma y x \frac{\partial^2 u}{\partial x \partial y} - \frac{1}{2}\gamma^2 y \frac{\partial^2 u}{\partial y^2} - (r+s)x\frac{\partial u}{\partial x} \\ & - \left\{\alpha(\beta - y) - \vartheta\gamma\sqrt{y}\right\}\frac{\partial u}{\partial y} + ru, \end{aligned} \tag{3}$$

where $\tau = T - t$ is the time to expiry, r is a risk free interest rate, s is the yield of continuously paid dividends, and ϑ is a so-called market price of the risk. In the following, we assume ϑ to be zero as has been done in many previous studies like in [25]. The original option pricing problem is a final value problem, since the value of the option is known at the expiry. Similarly to [9,17,25], we have transformed this problem to be an initial value problem with the operator L in (3) which is a more common form.

The option pricing problem is defined in an unbounded domain $\{(x, y, \tau) |\ x \geq 0,\ y \geq 0,\ \tau \in [0, T]\}$. In order to use finite difference approximations for space variables, we truncate a finite computational domain

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

where X and Y are sufficiently large.

For a put option the payoff function is

$$g(x) = \max(K - x, 0) \tag{5}$$

and for a call option it is

$$g(x) = \max(x - K, 0), \tag{6}$$

where K is the exercise price. The value at the expiry gives the initial value for u, that is,

$$u(x, y, 0) = g(x) \quad (x, y) \in [0, X] \times [0, Y]. \tag{7}$$

On the truncation boundaries, we use the Neumann boundary conditions

$$\frac{\partial u}{\partial x}(X, y, \tau) = \frac{\partial g}{\partial x}(X), \quad (y, \tau) \in [0, Y] \times [0, T], \tag{8}$$

$$\frac{\partial u}{\partial y}(x, Y, \tau) = 0, \quad (x, \tau) \in [0, X] \times [0, T]. \tag{9}$$

Due to the early exercise possibility of the American option, we have to include the following early exercise constraint for the option price:

$$u(x, y, \tau) \geq g(x), \quad (x, y, \tau) \in \Omega \times [0, T]. \tag{10}$$

The price of the American option based on the stochastic volatility model can be obtained by solving a time-dependent linear complementarity problem (LCP) [18]

$$Lu \geq 0, \quad u \geq g, \quad (u - g)Lu = 0, \tag{11}$$

for $(x, y, \tau) \in \Omega \times [0, T]$ with the initial and boundary conditions (7)–(9).

On the boundary $x = 0$, the LCP has the form

$$\begin{aligned} Lu = \frac{\partial u}{\partial \tau} - \frac{1}{2}\gamma^2 y \frac{\partial^2 u}{\partial y^2} - \alpha(\beta - y)\frac{\partial u}{\partial y} + ru \geq 0, \\ u \geq g(0), \quad (u - g(0))Lu = 0. \end{aligned} \tag{12}$$

These inequalities and equation do not depend on the solution v in the interior of the domain. For put and call options, these are equivalent with the Dirichlet boundary condition

$$u(0, y, \tau) = g(0), \quad (y, \tau) \in [0, Y] \times [0, T]. \tag{13}$$

We will use this boundary condition in the numerical solution. On the boundary $y = 0$, the LCP has the form

$$\begin{aligned} Lu = \frac{\partial u}{\partial \tau} - (r + s)x\frac{\partial u}{\partial x} - \alpha\beta\frac{\partial u}{\partial y} + ru \geq 0, \\ u \geq g(x), \quad (u - g(x))Lu = 0. \end{aligned} \tag{14}$$

The parameters α and β are positive and, thus, the coefficient of $\frac{\partial u}{\partial y}$ is negative. The direction of the characteristic curve of the first-order operator L on the boundary $y = 0$ is outward from the domain which means that no boundary condition needs to be posed on this boundary.

In this paper, we propose operator splitting methods to solve the LCP (11). These splittings are based on the formulation with an auxiliary variable λ and, hence, the LCP is reformulated as

$$\begin{cases} Lu = \lambda, \\ \lambda \geq 0, \quad u \geq g, \quad (u - g)\lambda = 0, \end{cases} \tag{15}$$

for $(x, y, \tau) \in \Omega \times [0, T]$ with the initial and boundary conditions (7)–(9).

3 Discretization of the PDE

A numerical solution of the American option pricing problem (11) requires the discretization of Heston's operator (3). In the following, we perform the discretization of the spatial derivatives using a seven point finite difference stencil and, moreover, we consider four time discretization schemes.

The discretization is performed using a uniform space-time finite difference grid for the computational domain defined in (4). Let the number of grid steps to be m, n and l in the x-direction, in the y-direction, and in the τ-direction, respectively. The grid steps to these directions are denoted by $\Delta x = X/m$, $\Delta y = Y/n$, and $\Delta\tau = T/l$. The grid point values of a finite difference approximation are denoted by

$$u_{i,j}^{(k)} \approx u(x_i, y_j, \tau_k) = u(i\Delta x, j\Delta y, k\Delta\tau), \tag{16}$$

where $i = 0, \ldots, m$, $j = 0, \ldots, n$, and $k = 0, \ldots, l$.

3.1 Space discretization

In Heston's operator (3), all partial derivatives have variable coefficients. In part of the domain, a first-order derivative dominates the second-order one. The operator also contains a second-order cross-derivative term. Due to these reasons it is not easy to construct a discretization with good properties and accuracy. The discretization of these spatial derivatives is considered in [9,25,30], for example.

Usual finite difference approximations lead to some positive off-diagonal elements in a matrix due to the cross-derivative and dominating first-order derivative. These elements can lead to oscillations in the solution. A strictly diagonal dominant matrix with positive diagonal elements and non-positive off-diagonal elements has an M-matrix property. With such matrices the solutions are oscillation free with sufficiently well-behaving right-hand sides. For example, the Crank–Nicolson time discretization can lead to badly behaving right-hand sides.

In the following, we describe a seven point finite difference stencil which is based on some added diffusion in order to obtain an M-matrix. In order to simplify notations, the derivation of the space discretization scheme is described for a partial differential equation with general coefficients

$$\frac{\partial u}{\partial \tau} + a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = 0. \tag{17}$$

The first-order and second-order spatial derivatives, except the cross-derivative term, are approximated with the standard second-order accurate central finite differences. When a first-order derivative dominate the corresponding second-order derivative we increase the size of the coefficient of the second-order derivative in order to avoid positive off-diagonal elements. The discretization is not second-order accurate where diffusion has been added in this way. It is easy to show that our approach is equivalent with using a combination of first-order one-sided differences and the central differences for the first-order derivative. Another approaches are not to add diffusion [22], to use second-order one-sided differences [18,25], or employ a flux-limiter [30]. The influence of positive off-diagonal entries has been studied in [31].

We denote the central finite difference operators for the first-order derivatives by

$$\delta_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \quad \text{and} \quad \delta_y u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}, \tag{18}$$

and for the second-order derivatives by

$$\delta_x^2 u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \quad \text{and} \quad \delta_y^2 u_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}. \tag{19}$$

Next, we consider the discretization of the second-order cross-derivative term; a similar discretization for this term is described in [22,24]. We assume here that the coefficient b for the cross-derivative in (17) is non-positive. From the Taylor series we obtain approximations

$$\begin{aligned} u(x_{i+1}, y_{j+1}) \approx\ & u + \Delta x\frac{\partial u}{\partial x} + \Delta y\frac{\partial u}{\partial y} \\ & + \frac{1}{2}\left(\Delta x^2\frac{\partial^2 u}{\partial x^2} + 2\Delta x\Delta y\frac{\partial u}{\partial x \partial y} + \Delta y^2\frac{\partial^2 u}{\partial y^2}\right) \quad \text{and} \\ u(x_{i-1}, y_{j-1}) \approx\ & u - \Delta x\frac{\partial u}{\partial x} - \Delta y\frac{\partial u}{\partial y} \\ & + \frac{1}{2}\left(\Delta x^2\frac{\partial^2 u}{\partial x^2} + 2\Delta x\Delta y\frac{\partial u}{\partial x \partial y} + \Delta y^2\frac{\partial^2 u}{\partial y^2}\right), \end{aligned} \tag{20}$$

where the value for u and its derivatives on the right side are evaluated at the grid point (x_i, y_j). If b would be positive then we would form similar approximations for

the grid point values $u(x_{i+1}, y_{j-1})$ and $u(x_{i-1}, y_{j+1})$, and use them in the derivation. By summing the equations in (20) and then dividing the result by $2\Delta x \Delta y$, we obtain

$$\frac{\partial u}{\partial x \partial y} \approx \frac{1}{2\Delta x \Delta y}\left[u(x_{i+1}, y_{j+1}) - 2u(x_i, y_j) + u(x_{i-1}, y_{j-1})\right] - \frac{\Delta x}{2\Delta y}\frac{\partial^2 u}{\partial x^2} - \frac{\Delta y}{2\Delta x}\frac{\partial^2 u}{\partial y^2}. \tag{21}$$

By performing the discretization of the cross-derivative using (21), we obtain an approximation for the second-order derivatives in (17) having the form

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} \approx \left[a - \frac{b\Delta x}{2\Delta y}\right]\frac{\partial^2 u}{\partial x^2} + \left[c - \frac{b\Delta y}{2\Delta x}\right]\frac{\partial^2 u}{\partial y^2} + \frac{b}{2\Delta x \Delta y}\left[u(x_{i+1}, y_{j+1}) - 2u(x_i, y_j) + u(x_{i-1}, y_{j-1})\right]. \tag{22}$$

Using the central finite differences (18) and (19) together with (22), we can approximate the partial differential equation (17) by the semi-discrete equation

$$\frac{\partial u}{\partial \tau} + \left[a - \frac{b\Delta x}{2\Delta y} + a_{\text{add}}\right]\delta_x^2 u_{i,j} + \left[c - \frac{b\Delta y}{2\Delta x} + c_{\text{add}}\right]\delta_y^2 u_{i,j} + d\delta_x u_{i,j} + e\delta_y u_{i,j} + \frac{b}{2\Delta x \Delta y}\left[u_{i+1,j+1} - 2u_{i,j} + u_{i-1,j-1}\right] + f u_{i,j} = 0, \tag{23}$$

where the coefficients a_{add} and c_{add} are due to additional diffusion in the x- and y-direction, respectively. We choose them so that all off-diagonal elements of the resulting matrix are non-positive. This leads to the formulas

$$\begin{aligned} a_{\text{add}} &= \min\left(-a + b\frac{\Delta x}{2\Delta y} + d\frac{\Delta x}{2}, -a + b\frac{\Delta x}{2\Delta y} - d\frac{\Delta x}{2}, 0\right), \\ c_{\text{add}} &= \min\left(-a + b\frac{\Delta y}{2\Delta x} + e\frac{\Delta y}{2}, -a + b\frac{\Delta y}{2\Delta x} - e\frac{\Delta y}{2}, 0\right). \end{aligned} \tag{24}$$

This defines a seven point discretization stencil. For Heston's partial differential operator (3) the coefficients in the above are

$$\begin{aligned} &a = -\frac{1}{2}y_j x_i^2, \quad b = -\rho\gamma y_j x_i, \quad c = -\frac{1}{2}\gamma^2 y_j, \\ &d = -(r+s)x_i, \quad e = -\alpha(\beta - y_j), \quad \text{and} \quad f = r. \end{aligned} \tag{25}$$

We can also use the finite difference stencil (23) on the boundary $y = 0$ where the operator has the form given in (14). The additional diffusion c_{add} in the y-direction zeroes the weights due to $\frac{\partial u}{\partial y}$ referring to the values outside the domain. This is equivalent with using an one-sided difference to approximate $\frac{\partial u}{\partial y}$.

We treat the Neumann boundary conditions (8) and (9) in the following way. Let us consider the boundary $x = X$. At a grid point (m, j), $j = 0, \ldots, n$, the boundary condition is $\partial u(X, y_j, \tau)/\partial x = 0$. We approximate it using the central finite difference stencil in (18) and we obtain

$$\delta_x u_{m,j} = \frac{u_{m+1,j} - u_{m-1,j}}{2\Delta x} = \frac{\partial g}{\partial x}(X). \tag{26}$$

From this it follows that the fictitious grid point value $u_{m+1,j}$ outside the computational domain has the value

$$u_{m+1,j} = u_{m-1,j} + 2\Delta x \frac{\partial g}{\partial x}(X). \tag{27}$$

We can use this formula to eliminate all fictitious grid point values $u_{m+1,j}$, $j = 0, \ldots, n$, appearing in the stencil (23) when it is applied on the boundary $x = X$. The other Neumann boundary condition on $y = Y$ can be handled in the same way.

For a put option, the space discretization leads to a semi-discrete equation which has the matrix representation

$$\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{A}\mathbf{u} = 0, \tag{28}$$

where $\mathbf{A}$ is an $m(n+1) \times m(n+1)$ matrix and $\mathbf{u}$ is a vector of length $m(n+1)$. For a call option, we have a non-zero right-hand side vector due to the non-homogeneous Neumann boundary condition (8).

3.2 Time discretization

In the following, we consider the discretization of the time derivative in (3). The stability properties of time discretization schemes are essential in the option pricing problems as the initial value (7) has discontinuous first derivative. We consider the implicit Euler method and three second-order accurate methods: the Crank–Nicolson method, the backward difference formula, and a Runge-Kutta scheme.

The first-order accurate implicit Euler scheme reads

$$(\mathbf{I} + \Delta\tau\mathbf{A})\,\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)}, \quad \text{for } k = 0, \ldots, l-1. \tag{29}$$

A well-known second-order accurate time discretization is the Crank–Nicolson method given by

$$\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k+1)} = \left(\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k)}, \quad \text{for } k = 0, \ldots, l-1. \tag{30}$$

This scheme is unconditionally stable which means that no restriction for the step size $\Delta\tau$ is needed [29]. Despite of this stability property, this method can cause oscillations in numerical solutions; see [14,26], for example.

The second-order accurate backward difference formula (BDF2) is

$$\left(\mathbf{I}+\frac{2}{3}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k+1)}=\frac{4}{3}\mathbf{u}^{(k)}-\frac{1}{3}\mathbf{u}^{(k-1)},\quad \text{for } k=1,\ldots,l-1, \tag{31}$$

where $\mathbf{u}^{(k+1)}$ is computed using the solutions at two previous time steps k and $k-1$. Hence, the solution $\mathbf{u}^{(1)}$ of the first time step has to be obtained using another method. The most typical choice for this is the implicit Euler method which we will also use in our numerical experiments. This scheme has better stability properties than the Crank–Nicolson method. This time discretization was applied to option pricing problems in [25].

The Runge-Kutta scheme considered here consists of the following two steps:

$$\begin{cases}(\mathbf{I}+\theta\Delta\tau\mathbf{A})\bar{\mathbf{u}}^{(k+1)}=(\mathbf{I}-(1-\theta)\Delta\tau\mathbf{A})\mathbf{u}^{(k)},\\ (\mathbf{I}+\theta\Delta\tau\mathbf{A})\mathbf{u}^{(k+1)}=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k)}-\left(\frac{1}{2}-\theta\right)\Delta\tau\mathbf{A}\bar{\mathbf{u}}^{(k+1)},\end{cases} \tag{32}$$

for $k=0,\ldots,l-1$. This scheme was proposed and shown to be L-stable in [7]. In order to obtain a second-order accuracy, the parameter θ has to be $\theta=1-1/\sqrt{2}$.

4 Operator splitting methods

Once the space and time discretizations are performed, the solution of (11) is obtained by solving the sequence of discrete LCPs

$$\begin{cases}\mathbf{B}\mathbf{u}^{(k+1)}\geq\mathbf{C}\mathbf{u}^{(k)},\quad \mathbf{u}^{(k+1)}\geq\mathbf{g},\\ \left(\mathbf{B}\mathbf{u}^{(k+1)}-\mathbf{C}\mathbf{u}^{(k)}\right)^{T}\left(\mathbf{u}^{(k+1)}-\mathbf{g}\right)=0,\end{cases} \tag{33}$$

for $k=0,\ldots,l-1$. For the alternative formulation (15) the discrete problems read

$$\begin{cases}\mathbf{B}\mathbf{u}^{(k+1)}=\mathbf{C}\mathbf{u}^{(k)}+\Delta\tau\boldsymbol{\lambda}^{(k+1)},\\ \boldsymbol{\lambda}^{(k+1)}\geq 0,\quad \mathbf{u}^{(k+1)}\geq\mathbf{g},\quad \left(\boldsymbol{\lambda}^{(k+1)}\right)^{T}\left(\mathbf{u}^{(k+1)}-\mathbf{g}\right)=0,\end{cases} \tag{34}$$

for $k=0,\ldots,l-1$. Next, we present the operator splitting methods based on the formulation (34) for solving the time-dependent LCPs.

Our operator splitting methods have two fractional time steps. In the first fractional step, a system of linear equations is solved and, in the second step, an intermediate solution and the auxiliary variable $\boldsymbol{\lambda}$ are updated in such a way that they satisfy their constraints. According to (34), the solution is required to satisfy the early exercise constraint of the option pricing problem while the components of $\boldsymbol{\lambda}$ are required to be non-negative. Since the form of the operator splitting methods depends on the underlying time discretization, we describe the operator splitting methods for all previously described time discretizations.

The operator splitting method in the case of the implicit Euler scheme reads

$$(\mathbf{I} + \Delta\tau\mathbf{A})\,\tilde{\mathbf{u}}^{(k+1)} = \mathbf{u}^{(k)} + \Delta\tau\tilde{\boldsymbol{\lambda}}^{(k+1)}, \tag{35}$$

$$\begin{cases} \mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)} - \Delta\tau\left(\boldsymbol{\lambda}^{(k+1)} - \tilde{\boldsymbol{\lambda}}^{(k+1)}\right) = 0, \\ \boldsymbol{\lambda}^{(k+1)} \geq 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g}, \quad \left(\boldsymbol{\lambda}^{(k+1)}\right)^T\left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{36}$$

for $k = 0, \ldots, l-1$. The vector $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ is an approximation of $\boldsymbol{\lambda}^{(k+1)}$ in (34). The simplest choice is to assume it to stay the same, that is,

$$\tilde{\boldsymbol{\lambda}}^{(k+1)} = \boldsymbol{\lambda}^{(k)}. \tag{37}$$

An alternative is to use the linear extrapolation which yields

$$\tilde{\boldsymbol{\lambda}}^{(k+1)} = 2\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)} \tag{38}$$

for $k = 1, \ldots, l-1$.

At each time step k, the above operator splitting method requires the solution of the system of linear equations (35) and the update for the intermediate solution $\tilde{\mathbf{u}}^{(k+1)}$ and the variable $\tilde{\boldsymbol{\lambda}}^{(k+1)}$. The update to satisfy (36) can be performed componentwise using a simple formula. The computational cost of this step is negligible compared to the cost of the solution of the system of linear equations (35) and, thus, the speed of the operator splitting method (35) and (36) is determined by the solution method of the system of linear equations.

Similarly the operator splitting method based on the Crank–Nicolson method is

$$\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\tilde{\mathbf{u}}^{(k+1)} = \left(\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k)} + \Delta\tau\tilde{\boldsymbol{\lambda}}^{(k+1)}, \tag{39}$$

$$\begin{cases} \mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)} - \Delta\tau\left(\boldsymbol{\lambda}^{(k+1)} - \tilde{\boldsymbol{\lambda}}^{(k+1)}\right) = 0, \\ \boldsymbol{\lambda}^{(k+1)} \geq 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g}, \quad \left(\boldsymbol{\lambda}^{(k+1)}\right)^T\left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{40}$$

for $k = 0, \ldots, l-1$. The proposed operator splitting based on the BDF2 time discretization reads

$$\left(\mathbf{I} + \frac{2}{3}\Delta\tau\mathbf{A}\right)\tilde{\mathbf{u}}^{(k+1)} = \frac{4}{3}\mathbf{u}^{(k)} - \frac{1}{3}\mathbf{u}^{(k-1)} + \frac{2}{3}\Delta\tau\tilde{\boldsymbol{\lambda}}^{(k+1)}, \tag{41}$$

$$\begin{cases} \mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)} - \frac{2}{3}\Delta\tau\left(\boldsymbol{\lambda}^{(k+1)} - \tilde{\boldsymbol{\lambda}}^{(k+1)}\right) = 0, \\ \boldsymbol{\lambda}^{(k+1)} \geq 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g}, \quad \left(\boldsymbol{\lambda}^{(k+1)}\right)^T\left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{42}$$

for $k = 1, \ldots, l-1$. Again, the intermediate solution $\tilde{\mathbf{u}}^{(k+1)}$ is obtained by solving (41) and the update for this solution and $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ are performed using Eq. (42).

The Runge-Kutta scheme is a two step time discretization method and, thus, the operator splitting method based on this Runge-Kutta scheme requires two solutions of linear systems of equations. After these, the intermediate solution $\tilde{\mathbf{u}}^{(k+1)}$ and the variable $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ are updated to satisfy their constraints. The splitting method is

$$(\mathbf{I} + \theta \Delta\tau \mathbf{A})\bar{\mathbf{u}}^{(k+1)} = (\mathbf{I} - (1-\theta)\Delta\tau \mathbf{A})\mathbf{u}^{(k)} + (1-\theta)\Delta\tau\tilde{\boldsymbol{\lambda}}^{(k+1)}, \tag{43}$$

$$(\mathbf{I}+\theta\Delta\tau\mathbf{A})\tilde{\mathbf{u}}^{(k+1)} = \left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k)} - \left(\frac{1}{2}-\theta\right)\Delta\tau\mathbf{A}\bar{\mathbf{u}}^{(k+1)} + (1-\theta)\Delta\tau\tilde{\boldsymbol{\lambda}}^{(k+1)}, \tag{44}$$

$$\begin{cases} \mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)} - \Delta\tau\left(\boldsymbol{\lambda}^{(k+1)} - \tilde{\boldsymbol{\lambda}}^{(k+1)}\right) = 0, \\ \boldsymbol{\lambda}^{(k+1)} \geq 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g}, \quad \left(\boldsymbol{\lambda}^{(k+1)}\right)^T\left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{45}$$

for $k = 0, \ldots, l-1$.

5 Accuracy considerations for operator splitting methods

In the following, we study the accuracy of the operator splitting method based on the Crank–Nicolson method with $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ chosen as in (37). These considerations can be easily extended also for the operator splitting method based on the implicit Euler method. We compare the solutions of discrete LCPs obtained using the Crank–Nicolson method and the operator splitting method based on it. Under some assumptions we give an estimate for this difference in Theorem 1.

For the analysis, we need a version of the Crank–Nicolson method which uses the auxiliary variable λ. It is given by

$$\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k+1)} = \left(\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{u}^{(k)} + \Delta\tau\,\boldsymbol{\lambda}^{(k+1)}, \tag{46}$$

$$\boldsymbol{\lambda}^{(k+1)} \geq 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g}, \quad \left(\boldsymbol{\lambda}^{(k+1)}\right)^T\left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0. \tag{47}$$

The operator splitting method is the same as in (39) and (40). In order to compare the solutions, we denote the approximations for $\mathbf{u}$ and $\boldsymbol{\lambda}$ in the operator splitting method by $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\lambda}}$, respectively. Thus, the method reads

$$\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\tilde{\mathbf{u}}^{(k+1)} = \left(\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}\right)\hat{\mathbf{u}}^{(k)} + \Delta\tau\hat{\boldsymbol{\lambda}}^{(k)}, \tag{48}$$

$$\hat{\mathbf{u}}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)} - \Delta\tau\left(\hat{\boldsymbol{\lambda}}^{(k+1)} - \hat{\boldsymbol{\lambda}}^{(k)}\right) = 0, \tag{49}$$

$$\hat{\boldsymbol{\lambda}}^{(k+1)} \geq 0, \quad \hat{\mathbf{u}}^{(k+1)} \geq \mathbf{g}, \quad \left(\hat{\boldsymbol{\lambda}}^{(k+1)}\right)^T\left(\hat{\mathbf{u}}^{(k+1)} - \mathbf{g}\right) = 0. \tag{50}$$

The following lemma gives two algebraic relations for the differences of the solution vectors generated by the methods.

Lemma 1 *The vectors in the Crank–Nicolson method* (46) *and* (47) *and the operator splitting method* (48)–(50) *satisfy*

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\left(\mathbf{d}^{(1)}-\Delta\tau\boldsymbol{\delta}^{(1)}\right)=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{d}^{(0)}-\frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\boldsymbol{\lambda}^{(1)}-\hat{\boldsymbol{\lambda}}^{(0)}\right) \tag{51}$$

and

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\left(\mathbf{d}^{(l)}-\Delta\tau\boldsymbol{\delta}^{(l)}\right)=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{d}^{(l-1)}-\frac{1}{2}(\Delta\tau)^2\times\mathbf{A}\left(\boldsymbol{\delta}^{(l-1)}+\boldsymbol{\lambda}^{(l)}-\boldsymbol{\lambda}^{(l-1)}\right) \tag{52}$$

for $l\geq 2$, *where*

$$\mathbf{d}^{(k)}=\mathbf{u}^{(k)}-\hat{\mathbf{u}}^{(k)}\quad\textit{and}\quad\boldsymbol{\delta}^{(k)}=\boldsymbol{\lambda}^{(k)}-\hat{\boldsymbol{\lambda}}^{(k)}. \tag{53}$$

Proof Let the index k be an arbitrary integer such that $0\leq k\leq l-1$. We start by eliminating the intermediate solution $\tilde{\mathbf{u}}^{(k+1)}$ in the splitting method (48)–(50). The Eq. (49) gives us $\tilde{\mathbf{u}}^{(k+1)}=\hat{\mathbf{u}}^{(k+1)}-\Delta\tau(\hat{\boldsymbol{\lambda}}^{(k+1)}-\hat{\boldsymbol{\lambda}}^{(k)})$ and by substituting it to (48) we get

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\hat{\mathbf{u}}^{(k+1)}=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\hat{\mathbf{u}}^{(k)}+\Delta\tau\hat{\boldsymbol{\lambda}}^{(k+1)}+\frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\hat{\boldsymbol{\lambda}}^{(k+1)}-\hat{\boldsymbol{\lambda}}^{(k)}\right). \tag{54}$$

By subtracting this from (46) we obtain

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{d}^{(k+1)}=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{d}^{(k)}+\Delta\tau\boldsymbol{\delta}^{(k+1)}-\frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\hat{\boldsymbol{\lambda}}^{(k+1)}-\hat{\boldsymbol{\lambda}}^{(k)}\right). \tag{55}$$

Subtracting $(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A})\Delta\tau\boldsymbol{\delta}^{(k+1)}$ from both sides and using $\boldsymbol{\delta}^{(k+1)}+\hat{\boldsymbol{\lambda}}^{(k+1)}-\hat{\boldsymbol{\lambda}}^{(k)}=\boldsymbol{\lambda}^{(k+1)}-\hat{\boldsymbol{\lambda}}^{(k)}$ gives us

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\left(\mathbf{d}^{(k+1)}-\Delta\tau\boldsymbol{\delta}^{(k+1)}\right)=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{d}^{(k)}-\frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\boldsymbol{\lambda}^{(k+1)}-\hat{\boldsymbol{\lambda}}^{(k)}\right). \tag{56}$$

From this we get the Eq. (51) by choosing $k=0$.

By subtracting from (56) the same equation with one smaller indices ($k \rightarrow k-1$) we get

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\left[\mathbf{d}^{(k+1)}-\Delta\tau\boldsymbol{\delta}^{(k+1)}-\left(\mathbf{d}^{(k)}-\Delta\tau\boldsymbol{\delta}^{(k)}\right)\right]=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)$$
$$\times\left(\mathbf{d}^{(k)}-\mathbf{d}^{(k-1)}\right)-\frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\boldsymbol{\lambda}^{(k+1)}-\boldsymbol{\lambda}^{(k)}-\hat{\boldsymbol{\lambda}}^{(k)}+\hat{\boldsymbol{\lambda}}^{(k-1)}\right) \tag{57}$$

for $1 \leq k \leq l-1$. Calculating the sum of these equations from $k=1$ to $k=l-1$ gives us

$$\left(\mathbf{I}+\frac{1}{2}\Delta\tau\mathbf{A}\right)\left[\mathbf{d}^{(l)}-\Delta\tau\boldsymbol{\delta}^{(l)}-\left(\mathbf{d}^{(1)}-\Delta\tau\boldsymbol{\delta}^{(1)}\right)\right]=\left(\mathbf{I}-\frac{1}{2}\Delta\tau\mathbf{A}\right)$$
$$\times\left(\mathbf{d}^{(l-1)}-\mathbf{d}^{(0)}\right)-\frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\boldsymbol{\lambda}^{(l)}-\boldsymbol{\lambda}^{(1)}-\hat{\boldsymbol{\lambda}}^{(l-1)}+\hat{\boldsymbol{\lambda}}^{(0)}\right). \tag{58}$$

By adding the Eq. (51) to this and using $\boldsymbol{\lambda}^{(l)}-\hat{\boldsymbol{\lambda}}^{(l-1)}=\boldsymbol{\delta}^{(l-1)}+\boldsymbol{\lambda}^{(l)}-\boldsymbol{\lambda}^{(l-1)}$ we get the Eq. (52). □

In order to make the relations in Lemma 1 to be useful, we need to be able to estimate the norms of associated vectors. Using the constraints (47) and (50) we obtain the following result.

Lemma 2 *Let $\|\cdot\|_\infty$ be the l_∞-norm and $\mathbf{d}^{(l)}$ and $\boldsymbol{\delta}^{(l)}$ are defined as in Lemma 1. Then the equation*

$$\|\mathbf{d}^{(l)}\|_\infty+\Delta\tau\|\boldsymbol{\delta}^{(l)}\|_\infty=\|\mathbf{d}^{(l)}-\Delta\tau\boldsymbol{\delta}^{(l)}\|_\infty \tag{59}$$

holds for all $\Delta\tau>0$.

Proof We consider the ith component of the vectors. The index i belongs to one of the following four sets:

Set 1: i is such that $\lambda_i^{(l)}=\hat{\lambda}_i^{(l)}=0$. We have $\delta_i^{(l)}=\lambda_i^{(l)}-\hat{\lambda}_i^{(l)}=0$ and, thus, $|d_i^{(l)}|+\Delta\tau|\delta_i^{(l)}|=|d_i^{(l)}|=|d_i^{(l)}-\Delta\tau\delta_i^{(l)}|$.

Set 2: i is such that $\lambda_i^{(l)}>0$ and $\hat{\lambda}_i^{(l)}=0$. We have $\delta_i^{(l)}=\lambda_i^{(l)}-\hat{\lambda}_i^{(l)}=\lambda_i^{(l)}>0$. Due to (47) and (50) we have $u_i^{(l)}=g_i$ and $\hat{u}_i^{(l)}\geq g_i$. Thus, we get $|d_i^{(l)}|+\Delta\tau|\delta_i^{(l)}|=-d_i^{(l)}+\Delta\tau\delta_i^{(l)}=|-d_i^{(l)}+\Delta\tau\delta_i^{(l)}|=|d_i^{(l)}-\Delta\tau\delta_i^{(l)}|$.

Set 3: i is such that $\lambda_i^{(l)}=0$ and $\hat{\lambda}_i^{(l)}>0$. We have $\delta_i^{(l)}=\lambda_i^{(l)}-\hat{\lambda}_i^{(l)}=-\hat{\lambda}_i^{(l)}<0$. Due to (47) and (50) we have $u_i^{(l)}\geq g_i$ and $\hat{u}_i^{(l)}=g_i$. Thus, we get $|d_i^{(l)}|+\Delta\tau|\delta_i^{(l)}|=d_i^{(l)}-\Delta\tau\delta_i^{(l)}=|d_i^{(l)}-\Delta\tau\delta_i^{(l)}|$.

Set 4: i is such that $\lambda_i^{(l)}>0$ and $\hat{\lambda}_i^{(l)}>0$. Due to (47) and (50) we have $u_i^{(l)}=\hat{u}_i^{(l)}=g_i$. Thus, we get $|d_i^{(l)}|+\Delta\tau|\delta_i^{(l)}|=\Delta\tau|\delta_i^{(l)}|=|d_i^{(l)}-\Delta\tau\delta_i^{(l)}|$.

Hence, the equation $|d_i^{(l)}|+\Delta\tau|\delta_i^{(l)}|=|d_i^{(l)}-\Delta\tau\delta_i^{(l)}|$ holds for all i. The Eq. (59) follows from this. □

Using the previous lemmas we obtain the following main result which states that under given assumptions the difference between the solutions is $\mathcal{O}((\Delta\tau)^2)$.

Theorem 1 *Let the initial value for the operator splitting method* (48)–(50) *be* $\hat{\mathbf{u}}^{(0)} = \mathbf{u}^{(0)}$ *and* $\hat{\boldsymbol{\lambda}}^{(0)} = 0$. *Assume that the sequence* $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \ldots, \boldsymbol{\lambda}^{(l)}$ *in the Crank–Nicolson method* (46) *and* (47) *satisfies the condition*

$$\|\boldsymbol{\lambda}^{(1)}\|_\infty + \sum_{k=2}^{l} \|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\|_\infty \leq C_1, \tag{60}$$

where C_1 *is a positive constant independent of* $l = T/\Delta\tau$. *Furthermore assume that* $\mathbf{A}$ *is diagonally dominant matrix with positive diagonal entries, that is, for the entries* a_{ij} *of* $\mathbf{A}$ *it is assumed that*

$$\sum_{j \neq i} |a_{ij}| \leq a_{ii} \quad \text{for all } i. \tag{61}$$

Then the inequality

$$\|\mathbf{u}^{(l)} - \hat{\mathbf{u}}^{(l)}\|_\infty + \Delta\tau \|\boldsymbol{\lambda}^{(l)} - \hat{\boldsymbol{\lambda}}^{(l)}\|_\infty \leq C_2 (\Delta\tau)^2 \tag{62}$$

holds for $\Delta\tau \leq 1/(\max_i a_{ii})$ *and a positive constant* C_2 *independent of* $\Delta\tau$.

Proof We begin by showing that for any vector $\mathbf{x}$ the inequality

$$\|\mathbf{x}\|_\infty \leq \left\| \left(\mathbf{I} + \frac{1}{2}\Delta\tau \mathbf{A} \right) \mathbf{x} \right\|_\infty \tag{63}$$

holds. Let i be such that $|x_i| \geq |x_j|$ for all j. Due to the inequalities $a_{ii} - \sum_{j \neq i} |a_{ij}| \geq 0$, $-\sum_{j \neq i} |a_{ij}||x_i| \leq -\sum_{j \neq i} |a_{ij} x_j|$, and $|b| - |c| \leq |b + c|$ for any real numbers b, c, we obtain

$$\begin{aligned}
|x_i| &\leq |x_i| + \frac{1}{2}\Delta\tau \left(a_{ii} - \sum_{j \neq i} |a_{ij}| \right) |x_i| \\
&\leq |x_i| + \frac{1}{2}\Delta\tau \left(|a_{ii} x_i| - \sum_{j \neq i} |a_{ij} x_j| \right) \\
&\leq \left| x_i + \frac{1}{2}\Delta\tau a_{ii} x_i \right| - \frac{1}{2}\Delta\tau \left| \sum_{j \neq i} a_{ij} x_j \right| \\
&\leq \left| x_i + \frac{1}{2}\Delta\tau \sum_{j} a_{ij} x_j \right|.
\end{aligned} \tag{64}$$

From this the inequality (63) follows.

Since $\hat{\mathbf{u}}^{(0)} = \mathbf{u}^{(0)}$, we have $\mathbf{d}^{(0)} = 0$. Due to this and $\hat{\boldsymbol{\lambda}}^{(0)} = 0$ Lemma 1 gives us

$$\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\left(\mathbf{d}^{(1)} - \Delta\tau\boldsymbol{\delta}^{(1)}\right) = -\frac{1}{2}(\Delta\tau)^2\mathbf{A}\boldsymbol{\lambda}^{(1)}. \tag{65}$$

From this we get the inequality

$$\begin{aligned}\|\mathbf{d}^{(1)} - \Delta\tau\boldsymbol{\delta}^{(1)}\|_\infty &\le \left\|\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\left(\mathbf{d}^{(1)} - \Delta\tau\boldsymbol{\delta}^{(1)}\right)\right\|_\infty \\ &\le \frac{1}{2}(\Delta\tau)^2\|\mathbf{A}\|_\infty\|\boldsymbol{\lambda}^{(1)}\|_\infty,\end{aligned} \tag{66}$$

where $\|\mathbf{A}\|_\infty = \max_i \sum_j |a_{ij}|$.

Due to (61) and $\Delta\tau \le 1/(\max_i a_{ii})$ we obtain the inequality

$$\begin{aligned}\frac{1}{2}\Delta\tau\|\mathbf{A}\|_\infty &= \frac{1}{2}\Delta\tau\max_i\sum_j |a_{ij}| \le \frac{1}{2}\Delta\tau\max_i 2a_{ii} = \Delta\tau\max_i a_{ii} \\ &\le \frac{1}{\max_i a_{ii}}\max_i a_{ii} = 1.\end{aligned} \tag{67}$$

The inequality $1 - \frac{1}{2}\Delta\tau a_{ii} \ge 0$ follows from $\Delta\tau \le 1/(\max_i a_{ii})$ and $a_{ii} - \sum_{j\ne i}|a_{ij}| \ge 0$ follows from (61). Using these we get the inequality

$$\begin{aligned}\left\|\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}\right\|_\infty &= \max_i\left(\left|1 - \frac{1}{2}\Delta\tau a_{ii}\right| + \frac{1}{2}\Delta\tau\sum_{j\ne i}|a_{ij}|\right) \\ &= \max_i\left(1 - \frac{1}{2}\Delta\tau a_{ii} + \frac{1}{2}\Delta\tau\sum_{j\ne i}|a_{ij}|\right) \\ &= \max_i\left[1 - \frac{1}{2}\Delta\tau\left(a_{ii} - \sum_{j\ne i}|a_{ij}|\right)\right] \le 1.\end{aligned} \tag{68}$$

Let k be an integer such that $k \ge 2$. Using Lemma 1, the previous inequalities (67), (68), and Lemma 2 we get

$$\begin{aligned}\|\mathbf{d}^{(k)} - \Delta\tau\boldsymbol{\delta}^{(k)}\|_\infty &\le \left\|\left(\mathbf{I} + \frac{1}{2}\Delta\tau\mathbf{A}\right)\left(\mathbf{d}^{(k)} - \Delta\tau\boldsymbol{\delta}^{(k)}\right)\right\|_\infty \\ &\le \left\|\left(\mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A}\right)\mathbf{d}^{(k-1)} - \frac{1}{2}(\Delta\tau)^2\mathbf{A}\left(\boldsymbol{\delta}^{(k-1)} + \boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\right)\right\|_\infty\end{aligned}$$

$$\begin{aligned}
&\leq \left\| \mathbf{I} - \frac{1}{2}\Delta\tau\mathbf{A} \right\|_\infty \|\mathbf{d}^{(k-1)}\|_\infty + \frac{1}{2}\Delta\tau\|\mathbf{A}\|_\infty \Delta\tau\|\boldsymbol{\delta}^{(k-1)}\|_\infty \\
&\quad + \frac{1}{2}(\Delta\tau)^2\|\mathbf{A}\|_\infty\|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\|_\infty \\
&\leq \|\mathbf{d}^{(k-1)}\|_\infty + \Delta\tau\|\boldsymbol{\delta}^{(k-1)}\|_\infty + \frac{1}{2}(\Delta\tau)^2\|\mathbf{A}\|_\infty\|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\|_\infty \\
&= \|\mathbf{d}^{(k-1)} - \Delta\tau\boldsymbol{\delta}^{(k-1)}\|_\infty + \frac{1}{2}(\Delta\tau)^2\|\mathbf{A}\|_\infty\|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\|_\infty. \qquad (69)
\end{aligned}$$

Choosing $k = l$ we have

$$\|\mathbf{d}^{(l)} - \Delta\tau\boldsymbol{\delta}^{(l)}\|_\infty \leq \|\mathbf{d}^{(l-1)} - \Delta\tau\boldsymbol{\delta}^{(l-1)}\|_\infty + \frac{1}{2}(\Delta\tau)^2\|\mathbf{A}\|_\infty\|\boldsymbol{\lambda}^{(l)} - \boldsymbol{\lambda}^{(l-1)}\|_\infty. \qquad (70)$$

Using (69) from $k = l - 1$ to $k = 2$ and then (66) we get

$$\|\mathbf{d}^{(l)} - \Delta\tau\boldsymbol{\delta}^{(l)}\|_\infty \leq \frac{1}{2}\|\mathbf{A}\|_\infty(\Delta\tau)^2\left(\|\boldsymbol{\lambda}^{(1)}\|_\infty + \sum_{k=2}^{l} \|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\|_\infty \right). \qquad (71)$$

The assumption (60) gives us

$$\|\mathbf{d}^{(l)} - \Delta\tau\boldsymbol{\delta}^{(l)}\|_\infty \leq \frac{1}{2}\|\mathbf{A}\|_\infty C_1(\Delta\tau)^2. \qquad (72)$$

By choosing $C_2 = \frac{1}{2}\|\mathbf{A}\|_\infty C_1$ and using Lemma 2 we get the inequality (62). □

Remark 1 It is easy to show that the assumption (60) in Theorem 1 is fulfilled in the case that $\|\boldsymbol{\lambda}^{(k)}\|_\infty$ is bounded for all k and the sequence is $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \ldots$, is increasing or decreasing componentwise. Based on our numerical investigations $\boldsymbol{\lambda}^{(k)}$s seem to be bounded and they are decreasing componentwise in the problems considered in Sect. 7.

Under the assumptions in Theorem 1 the accuracy of the operator splitting method based on the Crank–Nicolson method has the same order of accuracy as the Crank–Nicolson method which is at most $\mathcal{O}((\Delta\tau)^2)$. The possible irregularity of the solution with respect to time might reduce these both orders of accuracy. For one-dimensional American option pricing problems this reduction has been studied in [12], for example.

6 Multigrid

Optimal multigrid methods are scalable iterative methods in the sense that their convergence rate does not deteriorate when the grid step size decreases. For example, the convergence of the (projected) SOR method slows down as the grid step size is made smaller. Here, we describe only briefly the multigrid method which we have used in the numerical experiments. For details we refer to the books [5,16,27,28].

The basic idea of multigrid methods is to reduce low frequency error using coarser grids while high frequency error can be typically reduced using a standard iterative method on the fine grid. A multigrid method uses a restriction operation to transfer the solution or residual to a coarser grid. We have chosen to use the full weighting restriction. For coarser grids we construct matrices by discretizing Heston's operator on them. The solution or correction to it is transferred to a finer grid using a prolongation (interpolation) operation. We use the bilinear prolongation operator.

In a multigrid, the iterative method which reduces the high frequency error is called a smoother. For the Laplace operator, the (point) Gauss-Seidel or SOR method can be used as a smoother. Heston's operator requires a more involved smoother. Clarke and Parrott used a x-line Gauss-Seidel type smoother in their multigrid [8]. Oosterlee performed Fourier analysis for smoothers for Heston's operator [25]. According to the analysis the point Gauss-Seidel method is not good smoother while the x-line Gauss-Seidel smoother is efficient when the grid is fine in the y-direction. An alternating direction line Gauss-Seidel method was found out to have good smoothing properties for all grids. Thus, we chose to use the alternating direction line Gauss-Seidel smoother.

One cycle of the multigrid method is much more expensive than one (P)SOR iteration. In the alternating direction line Gauss-Seidel smoother, the x-line and y-line Gauss-Seidel sweeps can be both considered to be comparable in computational cost to a matrix vector multiplication. We use one presmooth iteration before moving to a coarser grid and one postsmooth iteration after adding the correction from the coarser grid. Thus, the computational cost of the smoothing is similar to four matrix vector multiplications. The multigrid also performs two matrix vector multiplication in order to compute the residual twice during one cycle. Thus, one cycle requires six matrix vector multiplication type operations on the finest grid. By summing operations over all grids we get that one multigrid cycle corresponds approximately eight matrix vector multiplications. Since one (P)SOR iteration is comparable to a matrix vector multiplication, one multigrid cycle is roughly eight times more expensive than one (P)SOR iteration. On the other hand a large number of SOR iterations are required to reduce the residual by the same amount as one multigrid cycle does.

7 Numerical experiments

In the following experiments, we study the properties of the operator splitting methods numerically. The prices of American put options are computed using the model (11) with the parameter values:

$$\alpha = 5, \quad \beta = 0.16, \quad \gamma = 0.9, \quad \rho = 0.1, \quad r = 0.1, \quad \text{and} \quad K = 10. \tag{73}$$

In order to compare computed option prices to the reference values, we chose the parameter values (73) to be the same as in [8,25,30]. Furthermore, we use the same computational domain $[0, X] \times [0, Y] \times [0, T] = [0, 20] \times [0, 1] \times [0, 0.25]$ as in [25].

In the first numerical experiment, we compare the option prices obtained by an operator splitting method with the prices computed using the PSOR method and the

prices presented in the references. The time convergence rates of four operator splitting methods are studied in the second experiment, and in the last numerical experiment, we compare the efficiency of the operator splitting algorithm with the multigrid method to the efficiency of the PSOR method.

7.1 Option prices computed using the operator splitting method

We computed the prices of the American put options using the operator splitting (OS) method (43)–(45) based on the Runge-Kutta scheme. These prices are presented in Table 1 for the asset values $x = 8.0, 9.0, 10.0, 11.0, 12.0$, and for the variance values $y = 0.0625$ and $y = 0.25$. We have used three different discretization grids in order to study the accuracy of numerical solutions. In the tables, the grids are defined by the triplets (m, n, l) where m, n, and l are the numbers of steps in the x-direction, in the y-direction, and in the τ-direction, respectively.

For the comparison, the option prices computed using the PSOR method are also given in these tables as well as the prices reported in [8,25,30]. As one can see from Table 1, the operator splitting method produces prices which are in good agreement with the prices obtained by the PSOR method. Only one price differs at the third decimal while other prices have three, four or even five same decimals. Furthermore, the prices obtained with the finest grid are fairly similar to the ones given in [25,30].

We study the accuracy of the finite difference discretizations used to compute the American option prices by computing the prices of the corresponding European options. Table 2 reports these approximate prices, their errors, and the exact prices. The error is the l_2 norm of the vector containing the errors at the ten points. The exact prices have been computed using an analytical formula found in [17], for example. The ratios of consecutive errors are 3.91 and 3.88 which suggest that the discretization scheme is nearly second-order accurate.

7.2 Time convergence

Next, we study the time convergence rates of the operator splitting methods based on the implicit Euler (IE) scheme, the Crank–Nicolson (CN) method, the BDF2 formula, and the Runge-Kutta (RK) scheme. The same American put option pricing problem is considered as before. In this experiment, we compare the errors and the convergence rates obtained with the operator splitting methods and with the PSOR method. Moreover, we have computed the errors and their ratios also for the explicit payoff method.

The spatial grid was chosen to be $(m, n) = (80, 32)$ while the number of time steps varies from 4 to 256. In Tables 3 and 4, we report the l_2 error based on the ten points used in the previous experiments. Our reference solution was computed with the PSOR method using the Runge-Kutta time discretization on the (80, 32, 8192) grid. The ratios in the tables were calculated by dividing the error of the previous coarser time discretization by the error of the current discretization. The PSOR method with the implicit Euler scheme is referred as PSOR-IE and the operator splitting method

Table 1 Option prices for American put options

Method	(m, n, l)	y	Asset value				
			8.0	9.0	10.0	11.0	12.0
	(80,32,16)	0.0625	2.00000	1.10612	0.51784	0.21217	0.08147
		0.25	2.07783	1.33232	0.79438	0.44697	0.24200
OS	(160,64,32)	0.0625	2.00000	1.10728	0.51948	0.21326	0.08187
		0.25	2.07830	1.33335	0.79558	0.44793	0.24259
	(320,128,64)	0.0625	2.00000	1.10761	0.51987	0.21353	0.08197
		0.25	2.07847	1.33361	0.79587	0.44816	0.24272
	(80,32,16)	0.0625	2.00000	1.10599	0.51787	0.21222	0.08150
		0.25	2.07754	1.33219	0.79435	0.44699	0.24204
PSOR	(160,64,32)	0.0625	2.00000	1.10718	0.51946	0.21327	0.08188
		0.25	2.07815	1.33325	0.79554	0.44793	0.24260
	(320,128,64)	0.0625	2.00000	1.10749	0.51985	0.21354	0.08198
		0.25	2.07829	1.33351	0.79583	0.44815	0.24273
Ref. [8]		0.0625	2.0000	1.1080	0.5316	0.2261	0.0907
		0.25	2.0733	1.3290	0.7992	0.4536	0.2502
Ref. [25]		0.0625	2.00	1.107	0.517	0.212	0.0815
		0.25	2.079	1.334	0.796	0.449	0.243
Ref. [30]		0.0625	2.0000	1.1076	0.5202	0.2138	0.0821
		0.25	2.0784	1.3337	0.7961	0.4483	0.2428

Table 2 Option prices for European put options

(m, n, l)	y	Asset value					
		8.0	9.0	10.0	11.0	12.0	Error
(80,32,16)	0.0625	1.83864	1.04717	0.49961	0.20690	0.07995	
	0.25	1.97672	1.27882	0.76833	0.43496	0.23662	3.42e-3
(160,64,32)	0.0625	1.83879	1.04804	0.50100	0.20785	0.08030	
	0.25	1.97716	1.27970	0.76935	0.43577	0.23710	8.74e-4
(320,128,64)	0.0625	1.83885	1.04827	0.50135	0.20810	0.08039	
	0.25	1.97727	1.27992	0.76960	0.43598	0.23722	2.25e-4
Exact	0.0625	1.83887	1.04835	0.50147	0.20819	0.08043	
	0.25	1.97731	1.27800	0.76969	0.43605	0.23726	

based on the implicit Euler scheme is referred as OS-IE. Methods based on other time discretizations are referred in the same way.

Similar errors can be observed for the operator splitting method and the PSOR method for all time discretizations. Actually, for many discretizations the errors for the operator splitting method are smaller. First-order convergence rate can be observed for the implicit Euler scheme and the operator splitting method based on it. The second-

Table 3 l_2 errors and their ratios for the PSOR method

l	PSOR-IE		PSOR-CN		PSOR-BDF2		PSOR-RK	
	Error	Ratio	Error	Ratio	Error	Ratio	Error	Ratio
4	5.93e-2		4.53e-2		2.15e-2		2.52e-3	
8	3.20e-2	1.85	8.68e-3	5.22	6.58e-3	3.26	1.00e-3	2.52
16	1.69e-2	1.89	4.40e-4	19.74	2.29e-3	2.88	4.25e-4	2.36
32	8.77e-3	1.92	1.78e-4	2.47	8.48e-4	2.70	1.74e-4	2.44
64	4.51e-3	1.94	7.07e-5	2.52	3.18e-4	2.66	6.55e-5	2.65
128	2.30e-3	1.96	2.73e-5	2.59	1.19e-4	2.69	2.49e-5	2.63
256	1.17e-3	1.97	9.38e-6	2.91	4.29e-5	2.77	8.49e-6	2.94

Table 4 l_2 errors and their ratios for the operator splitting methods

l	OS-IE		OS-CN		OS-BDF2		OS-RK	
	Error	Ratio	Error	Ratio	Error	Ratio	Error	Ratio
4	5.87e-2		4.61e-2		2.52e-2		7.82e-3	
8	3.03e-2	1.94	8.37e-3	5.51	6.60e-3	3.83	2.42e-3	3.23
16	1.56e-2	1.94	9.43e-4	8.88	1.93e-3	3.42	8.48e-4	2.85
32	8.11e-3	1.93	4.06e-4	2.32	6.25e-4	3.09	3.42e-4	2.48
64	4.23e-3	1.92	1.34e-4	3.03	1.99e-4	3.14	1.18e-4	2.89
128	2.20e-3	1.92	3.89e-5	3.44	7.24e-5	2.75	3.46e-5	3.41
256	1.14e-3	1.94	1.00e-5	3.88	2.76e-5	2.62	1.07e-5	3.23

order accurate time discretization schemes and the operator splitting methods based on them cannot maintain second-order convergence rate when the number of time steps is increased. The Crank–Nicolson method is not L-stable and due to this the convergences of the PSOR and operator splitting methods based on it are a bit erratic.

For a comparison, we also computed the prices with the explicit payoff (EP) method which has been considered for an one-dimensional American option pricing problem in [10,11,20]. In this method, at each time step the system of linear equations resulting from the discretization of the PDE for the corresponding European option is solved and then the maximum between the solution and the payoff function is taken componentwise. In the operator splitting method and in the explicit payoff method, a system of linear equations is solved and a simple update is performed at each time step and, hence, the computational costs of these methods are essentially the same. According to Tables 4 and 5, the proposed operator splitting methods are much more accurate than the explicit payoff methods. Only first-order convergence rates can be observed with the explicit payoff method even when the second-order accurate time discretization schemes are used.

Tables 3 and 4 show that the second-order time discretizations cannot maintain second-order time convergence. This was also observed in [12] with the Black–

Table 5 l_2 errors and their ratios for the explicit payoff methods

l	EP-IE		EP-CN		EP-BDF2		EP-RK	
	Error	Ratio	Error	Ratio	Error	Ratio	Error	Ratio
4	2.28e-2		5.49e-2		2.82e-2		1.27e-2	
8	1.20e-3	19.09	1.51e-2	3.63	1.11e-2	2.54	5.95e-3	2.14
16	3.63e-4	3.30	5.30e-3	2.85	5.35e-3	2.08	2.51e-3	2.37
32	8.94e-5	4.06	2.69e-3	1.97	2.74e-3	1.96	1.45e-3	1.73
64	2.37e-5	3.78	1.37e-3	1.97	1.52e-3	1.80	7.03e-4	2.06
128	5.50e-6	4.30	6.86e-4	1.99	7.99e-4	1.90	3.51e-4	2.01
256	1.31e-3	1.97	3.43e-4	2.00	4.00e-4	2.00	1.75e-4	2.01

Scholes model. The free boundary moves faster near the expiry which makes the solution less regular for small τ. In [12], adaptively chosen time steps restored second-order convergence. To study the influence of nonuniform time steps, we choose them according to

$$\Delta\tau^{(k+1)} = \left[\left(\frac{k+1}{l}\right)^2 - \left(\frac{k}{l}\right)^2\right] T \tag{74}$$

for $k = 0, \ldots, l-1$. Table 6 compares the time convergence of methods based the Crank–Nicolson time discretization. EP refers to the explicit payoff method, OS refers to the operator splitting method with $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ chosen according to (37), and OS2 refers to the operator splitting method with $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ chosen according the generalization of (38) to nonuniform time steps given by

$$\tilde{\boldsymbol{\lambda}}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \frac{\Delta\tau^{(k+1)}}{\Delta\tau^{(k)}}\left(\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\right). \tag{75}$$

Based on Table 6 the nonuniform time steps given by (74) lead to second-order convergence with the PSOR method and the operator splitting methods. With these methods the use of nonuniform time steps become clearly beneficial only with fairly larger number of time steps. The explicit payoff method is first-order accurate and it has similar accuracy with uniform and nonuniform time steps.

7.3 CPU-time comparison

In our last numerical experiment, we compare the efficiency of the PSOR method and the operator splitting method. We report the CPU-times required to solve the previously described American put option pricing problem on a PC with 3.8 GHz Xeon processor when different discretization grids are used. The time discretization is based

Table 6 l_2 errors and their ratios for nonuniform time steps

l	EP-CN		PSOR-CN		OS-CN		OS2-CN	
	Error	Ratio	Error	Ratio	Error	Ratio	Error	Ratio
4	3.45e-2		2.28e-2		3.52e-2		8.52e-2	
8	1.33e-2	2.59	1.20e-3	19.09	4.88e-3	7.20	1.40e-2	6.09
16	7.13e-3	1.87	3.63e-4	3.30	2.21e-3	2.21	2.55e-3	5.49
32	3.62e-3	1.97	8.94e-5	4.06	1.01e-3	2.19	5.02e-4	5.08
64	1.84e-3	1.97	2.37e-5	3.78	3.33e-4	3.03	6.58e-5	7.63
128	9.25e-4	1.99	5.50e-6	4.30	9.40e-5	3.54	9.61e-6	6.84
256	4.66e-4	1.98	1.37e-6	4.02	2.45e-5	3.84	1.67e-6	5.76
512	2.34e-4	1.99	3.49e-7	3.93	6.21e-6	3.94	3.69e-7	4.53
1024	1.17e-4	2.00	8.69e-8	4.01	1.56e-6	3.97	8.89e-8	4.15

on the Crank–Nicolson method with uniform time steps and in the operator splitting method $\tilde{\boldsymbol{\lambda}}^{(k+1)}$ chosen according to (38).

For the PSOR method the stopping criterion and the relaxation parameter have to be chosen. The iteration was stopped when the norm of the residual was reduced below 10^{-5} times the norm of the right-hand side vector. Based on numerical tests, the error due to the termination of iterations is smaller than the discretization error. On the other hand, the criterion is not unnecessary tight which would increase the CPU-time without increasing accuracy. The relaxation parameter ω was optimised for the different grids as the choice of ω has significant effect to the efficiency of the PSOR method. For example, the value $\omega = 1.5$ for the grid $(m, n, l) = (320, 128, 64)$ leads the CPU-time to be 8.23 seconds while with the choice $\omega = 1.8$ it is 3.73 seconds. These parameters and also the average numbers of iterations are presented in Table 7.

With the operator splitting method many efficient solution method for the system of linear equations can be used. In this experiment, we employ the multigrid (MG) method described in Sect. 6 with the operator splitting method. According our tests, the multigrid method is not so sensitive to the choice of the stopping criterion as the PSOR method, and the value 10^{-5} for the reduction of the norm of the residual is also suitable for the multigrid method. It is easier to use the multigrid method than the PSOR method, since the PSOR method requires optimisation of the relaxation parameter in order to achieve fast convergence.

We report the l_2 error for each numerical solution in Table 7. We have used four different time step sizes and several space discretization grids. The most coarse grid is (40, 16, 16) and the most fine grid is (640, 256, 128). The errors for the PSOR method and the operator splitting method differ for each discretization grid due different time discretizations of the LCP. We can observe that the errors are smaller for the operator splitting method except on the (80, 64, 32) grid. Here, the reference solution was computed using the operator splitting method on the (2560, 1024, 512) grid.

Table 7 shows CPU-times in seconds required for the PSOR method and for the operator splitting method. For coarser grids the CPU-times are quite similar for these

Table 7 CPU-times for the PSOR method and the operator splitting method

(m, n, l)	PSOR				OS2 with MG		
	Error	Iter	ω	CPU	Error	Iter	CPU
(40,16,16)	1.55e-2	10.4	1.4	0.00	1.49e-2	2.0	0.00
(40,32,16)	1.61e-2	12.9	1.4	0.00	1.55e-2	2.0	0.00
(80,16,16)	4.57e-3	22.1	1.6	0.01	4.06e-3	2.0	0.01
(80,32,16)	4.44e-3	22.1	1.6	0.02	3.69e-3	2.1	0.02
(80,32,32)	4.19e-3	14.4	1.5	0.04	3.77e-3	2.0	0.04
(80,64,32)	1.72e-3	16.1	1.6	0.09	3.83e-3	2.0	0.09
(160,32,32)	1.36e-3	32.9	1.7	0.17	9.79e-4	2.0	0.08
(160,64,32)	1.30e-3	32.0	1.7	0.33	8.78e-4	2.1	0.17
(160,64,64)	1.15e-3	20.1	1.7	0.42	9.19e-4	2.0	0.33
(160,128,64)	1.16e-3	21.9	1.7	0.92	9.13e-4	2.0	0.75
(320,64,64)	4.22e-4	49.9	1.8	2.07	2.43e-4	2.0	0.74
(320,128,64)	4.05e-4	45.4	1.8	3.73	2.17e-4	2.1	1.62
(320,128,128)	3.28e-4	26.9	1.7	4.46	2.24e-4	1.7	2.68
(320,256,128)	3.30e-4	29.8	1.8	9.84	2.46e-4	1.7	5.48
(640,128,128)	1.43e-4	65.3	1.8	21.34	6.88e-5	2.0	6.63
(640,256,128)	1.42e-4	69.4	1.8	45.21	6.71e-5	2.0	13.63

two solution methods. The fairly good performance of the PSOR method is due to optimised relaxation parameter ω and strongly diagonally dominant matrices leading to faster convergence. We see that the operator splitting method with the multigrid method requires much less time when finer grids are used.

8 Conclusion

We have described operator splitting methods for numerically pricing American options under Heston's stochastic volatility model. In these splitting methods, each time step is divided into two fractional steps. In the first step, a system of linear equations is solved while in the second step the constraint due to the early exercise possibility of the American option is enforced by performing a simple update. With this approach efficient solution methods for a system of linear equations can be applied without modifications.

Our analysis suggest that the Crank–Nicolson method with the implicit treatment of complementarity conditions (for example, the PSOR method) and the operator splitting method based on it have the same order of accuracy. In the numerical experiments, the accuracy of solutions obtained using any of the tested four implicit time discretizations with the PSOR method and the corresponding operator splitting method were similar. The second-order time discretizations and the operator splitting methods based on them could not maintain second-order time convergence with uniform time steps. Using refined time steps near the expiry we observed second-order time convergence

with the methods based on the Crank–Nicolson time discretization excluding the explicit payoff method. The explicit treatment of complementarity conditions in the explicit payoff methods leads to reduced accuracy and first-order time convergence.

The operator splitting method with a multigrid method for solving the systems of linear equations was faster than the PSOR method. On finer grids the operator splitting approach was several times faster while on coarser grids the CPU-times were similar. Also in a recent comparison of several numerical methods in [22] the operator splitting method was competitive.

Acknowledgments This research has been supported by the University of Jyväskylä and by the Academy of Finland, grants #53588 and #207089. We thank Dr. Roland Glowinski for introduction and discussions on operator splitting methods. We also thank Dr. Peter Forsyth for fruitful discussions on option pricing and a referee for suggestions improving this paper.

References

1. Ball, C.A., Roma, A.: Stochastic volatility option pricing. J. Financ. Quant. Anal. **29**, 589–607 (1994)
2. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Polit. Econ. **81**, 637–654 (1973)
3. Brandt, A., Cryer, C.W.: Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems. SIAM Sci. Stat. Comput. **4**, 655–684 (1983)
4. Brennan, M.J., Schwartz, E.S.: The valuation of American put options. J. Finance **32**, 449–462 (1977)
5. Briggs, W.L., Henson, V.E., McCormick, S.F.: A Multigrid Tutorial, 2nd edn. SIAM, Philadelphia (2000)
6. Broadie, M., Chernov, M., Johannes, M.: Model specification and risk premia: evidence from futures options. J. Finance **62**, 1453–1490 (2007)
7. Cash, J.R.: Two new finite difference schemes for parabolic equations. SIAM J. Numer. Anal. **21**, 433–446 (1984)
8. Clarke, N., Parrott, K.: The multigrid solution of two-factor American put options. Technical Report 96-16, Oxford Comp. Lab, Oxford (1996)
9. Clarke, N., Parrott, K.: Multigrid for American option pricing with stochastic volatility. Appl. Math. Finance **6**, 177–195 (1999)
10. Coleman, T.F., Li, Y., Verma, A.: A Newton method for American option pricing. J. Comput. Finance **5**, 51–78 (2002)
11. Duffie, D.: Dynamic Asset Pricing Theory, 2nd edn. Princeton University Press, Princeton (1996)
12. Forsyth, P.A., Vetzal, K.R.: Quadratic convergence for valuing American options using a penalty method. SIAM J. Sci. Comput. **23**, 2095–2122 (2002)
13. Fouque, J.-P., Papanicolaou, G., Sircar, K.R.: Derivatives in Financial Markets with Stochastic Volatility. Cambridge University Press, Cambridge (2000)
14. Giles, M.B., Carter, R.: Convergence analysis of Crank–Nicolson and Rannacher time-marching. J. Comput. Finance **9**, 89–112 (2006)
15. Glowinski, R.: Finite element methods for incompressible viscous flow. Handbook of Numerical Analysis, vol. IX. North-Holland, Amsterdam (2003)
16. Hackbusch, W.: Multigrid methods and applications. Springer Series in Computational Mathematics, vol. 4. Springer, Berlin (1985)
17. Heston, S.: A closed-form solution for options with stochastic volatility with applications to bond and currency options. Rev. Finance Stud. **6**, 327–343 (1993)
18. Huang, J., Pang, J.-S.: Option pricing and linear complementarity. J. Comput. Finance **2**, 31–60 (1998)
19. Hull, J., White, A.: The pricing of options on assets with stochastic volatilities. J. Finance **42**, 281–300 (1987)
20. Hull, J.C.: Options, Futures, and Other Derivatives, 3rd edn. Prentice Hall, Upper Saddle River (1997)
21. Ikonen, S., Toivanen, J.: Operator splitting methods for American option pricing. Appl. Math. Lett. **17**, 809–814 (2004)

22. Ikonen, S., Toivanen, J.: Efficient numerical methods for pricing American options under stochastic volatility. Numer. Methods Partial Differ. Equ. **24**, 104–126 (2007)
23. Lions, P.-L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**, 964–979 (1979)
24. Mitchell, A.R., Griffiths, D.F.: The Finite Difference Method in Partial Differential Equations. Wiley, Chichester (1980)
25. Oosterlee, C.W.: On multigrid for linear complementarity problems with application to American-style options. Electron. Trans. Numer. Anal. **15**, 165–185 (2003)
26. Pooley, D.M., Vetzal, K.R., Forsyth, P.A.: Convergence remedies for non-smooth payoffs in option pricing. J. Comput. Finance **6**, 25–40 (2003)
27. Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid. Academic Press Inc., San Diego (2001)
28. Wesseling, P.: An Introduction to Multigrid Methods. Wiley, Chichester (1992)
29. Wilmott, P., Howison, S., Dewynne, J.: The Mathematics of Financial Derivatives. Cambridge University Press, Cambridge (1995)
30. Zvan, R., Forsyth, P.A., Vetzal, K.R.: Penalty methods for American options with stochastic volatility. J. Comput. Appl. Math. **91**, 199–218 (1998)
31. Zvan, R., Forsyth, P.A., Vetzal, K.R.: Negative coefficients in two factor option pricing models. J. Comput. Finance **7**, 37–73 (2003)