European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2004
P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer (eds.)
Jyväskylä, 24–28 July 2004

OPERATOR SPLITTING METHODS FOR AMERICAN OPTIONS WITH STOCHASTIC VOLATILITY

S. Ikonen* **and J. Toivanen***

*Department of Mathematical Information Technology
P.O. Box 35 (Agora), FIN-40014 University of Jyväskylä, Finland
e-mails: Samuli.Ikonen@mit.jyu.fi, Jari.Toivanen@mit.jyu.fi

Key words: American option pricing, stochastic volatility models, operator splitting methods, time discretization.

Abstract. *Option pricing models with a stochastic volatility are more realistic than the Black-Scholes model which uses a constant volatility. The prices of options based on such models can be obtained by solving a parabolic partial differential equation. Particularly, we consider the model presented by Heston. The variables in these problems are the time, the underlying asset value, and the volatility. Due to the early exercise possibility of American options the arising problems are free boundary problems. This paper considers the numerical solution of this type of option pricing models.*

We study operator splitting methods for performing time stepping after a finite difference space discretization is done. The idea is to decouple the early exercise constraint for the prices at the grid points and the solution of the system of linear equations to separate fractional time steps. With this approach we can use any efficient numerical method to solve the systems of linear equations in the first fractional step and then make a simple update to satisfy the constraint in the second fractional step. This leads to more simple and efficient solution procedures. This paper studies the accuracy of the operator splitting methods without considering efficient solution procedures which is the topic of our future research.

Numerical experiments compare the accuracy of our operator splitting methods with traditional unsplit implicit time discretizations. We demonstrate that the additional error due to the splitting does not essentially increase the time discretization error.

1 Introduction

The option prices obtained using the Black-Scholes model [2] with a constant volatility are not consistent with observed option prices. One possible remedy for this is to make the volatility to be a function of time and strike price. By calibrating this function, consistent option prices can be obtained. Another approach is to assume that also the volatility of the price process is stochastic. Such models have been considered for example in [1], [9], [12], and [13]. We will use the model introduced by Heston [12] for pricing American options. For example, in [8] it was shown that the time-dependent probability distribution of stock price changes generated by Heston's model is in good agreement with the Dow-Jones index after the calibration of the few parameters appearing in the model.

The price of an European option can be calculated using analytical expressions for Heston's model [12]. 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. No useful analytical methods are available for pricing American options and, thus, numerical methods are used to approximate the prices. A finite difference method for pricing American options for the Black-Scholes model leading to one dimensional parabolic partial differential inequality was considered 1977 by Brennan and Schwartz [4]. 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, that 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 pricing problems.

Clarke and Parrott [6], [7] 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. 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 used 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, Forsyth and Vetzal [21] 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 solution 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 [17] performed the space discretization of using central differences for the diffusion and second-order upwind differences for the convection. The time discretization was based on the second-order accurate backward finite difference. Oosterlee studied the PFAS multigrid for linear complementarity problems. The quality 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 [17] it seems that a robust and efficient PFAS multigrid has to use a rather involved smoother for linear complementarity problems arising from 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 [10]. Previously operators splitting methods have been applied for obstacle problems in [15], for example. In our earlier paper [14], we proposed operator splitting methods for pricing American options using the Black-Scholes model. The idea is divide each time step into two fractional time steps. The first step integrates a modified partial differential equation over the time step and the second step makes a simple correction. Such a splitting introduces additional error and, thus, it necessary to show that this error is sufficiently small so that this approach can be used to price option accurately and efficiently. Our numerical experiments in [14] showed that for the Black-Scholes model the accuracy with a splitting is essentially the same as without it.

The purpose of this paper is to study the accuracy of operator splitting methods for pricing American options using Heston's model. With stochastic volatility models it makes efficient solution procedures much more simple if the early exercise constraint can be treated in a separate fractional time step. For example, we can use any multigrid method which is suited for the underlying convection diffusion problems without making any modifications to it. In this study of accuracy we will use the SOR method with the operator splitting method and compare its accuracy to the projected SOR (PSOR) method. More efficient solution procedures are the topic of our future research. The numerical experiments in this paper demonstrate that the additional error due to the splitting is of the same order as the time discretization error of unsplit schemes.

The outline of the paper is the following. In the first section we describe Heston's model and a partial differential inequality for pricing American options. The second section introduces our finite difference space discretization and time discretizations which are the implicit Euler method, the Crank-Nicolson method, the second-order backward difference method and a second-order Runge-Kutta method. In the third section, we describe operator splitting schemes corresponding to all of these time discretizations. The next section presents our numerical experiments studying the accuracy of the splitting methods. In the end of this paper we give some conclusions and future directions for research.

2 Option pricing model

Our formulation and notations are based on the references [12], [17], and [21]. In the following we describe stock price and variance processes, a partial differential inequality, an initial value and boundary conditions. This section defines a problem whose numerical solution is studied in the consecutive sections.

In Heston's model, stochastical differential equations

$$\begin{aligned} dx_t &= \mu x_t dt + \sqrt{y_t} x_t dw_1, && (1) \\ dy_t &= \alpha(\beta - y_t)dt + \gamma\sqrt{y_t} dw_2, && (2) \end{aligned}$$

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 average rate of the deterministic growth of the stock price and $\sqrt{y_t}$ is the standard deviation (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 (volatility of volatility) and the variance will drift back to mean value $\beta > 0$ at a rate $\alpha > 0$. These two processes contain randomness, that is, w_1 and w_2 are Brownian motions and furthermore, the processes are linked together with the correlation factor $\rho \in [-1, 1]$ [7], [21].

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 [21] and references therein. We define a generalized Black-Scholes operator

$$Lu := \frac{\partial u}{\partial t} - \frac{1}{2}yx^2\frac{\partial^2 u}{\partial x^2} - \rho\gamma yx\frac{\partial^2 u}{\partial x \partial y} - \frac{1}{2}\gamma^2 y\frac{\partial^2 u}{\partial y^2} - rx\frac{\partial u}{\partial x} - \left\{\alpha(\beta - y) - \vartheta\gamma\sqrt{y}\right\}\frac{\partial u}{\partial y} + ru, \quad (3)$$

where the parameter ϑ is a so-called market price of the risk. In the following we assume ϑ to be zero as have been done in many previous studies like [17]. The original option pricing problem is an final value problem, since the value of the option is known at the expiry. Similarly to [7], [12], and [17], we have transformed this problem to be an initial value problem with the operator L in (3) which is more common form for such problems.

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

$$(x, y, t) \in [0, X] \times [0, Y] \times [0, T] =: \Omega \times [0, T], \quad (4)$$

where X and Y are sufficiently large.

For the American put option an initial value and boundary conditions are described in [17], for example. The initial value is

$$u(x,y,0) = \max(E-x,0), \tag{5}$$

where E is the exercise price. The boundary conditions are

$$u(0,y,t) = E, \quad (y,t) \in [0,Y] \times [0,T], \tag{6}$$

$$u(x,0,t) = \max(E-x,0), \quad (x,t) \in [0,X] \times [0,T], \tag{7}$$

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

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

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

$$u(x,y,t) \geq \max(E-x,0) =: g(x), \quad (x,y,t) \in \Omega \times [0,T]. \tag{10}$$

Finally, the price of the American option based on the stochastic volatility model can be obtained by solving a time dependent complementarity problem

$$\left\{ \begin{array}{c} Lu \geq 0, \\ u \geq g, \\ \big(u-g\big)Lu = 0, \end{array} \right. \tag{11}$$

for $(x,y,t) \in \Omega \times [0,T]$ with the initial and boundary conditions (5) – (9).

In this paper, we present operator splitting methods for the linear complementarity problem (11) based on the following formulation with an auxiliary variable λ:

$$\left\{ \begin{array}{c} Lu = \lambda, \\ \lambda \geq 0, \quad u \geq g, \\ \big(u-g\big)\lambda = 0, \end{array} \right. \tag{12}$$

for $(x,y,t) \in \Omega \times [0,T]$ with the initial and boundary conditions (5) – (9).

3 Discretization of PDE

The solution of the American option pricing problem requires a numerical approximation of the two-dimensional partial differential operator (3). In this section, we consider the space and time discretizations of this generalized Black-Scholes operator. The discretization of spatial derivatives is performed using a seven point finite difference stencil and several time discretization schemes are considered.

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

The grid point values of a finite difference approximation are denoted by

$$u_{i,j}^{(k)} \approx u(x_i, y_j, t_k) = u(i\Delta x, j\Delta y, k\Delta t), \tag{13}$$

where $i = 1, \ldots, n$, $j = 1, \ldots, m$, and $k = 1, \ldots, l$.

3.1 Space discretization

The space discretization is based on central finite difference schemes and on a special approximation of the second-order cross-derivative term. In order to simplify notations, the derivation of the space discretization scheme is described for a partial differential equation with general coefficients

$$\frac{\partial u}{\partial t} + a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = 0. \tag{14}$$

The second-order and first-order spatial derivatives are approximated with standard second-order accurate central finite differences. For this purpose, we denote the finite difference operators for the first-order derivatives by

$$\delta_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \qquad \text{and} \qquad \delta_y u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}, \tag{15}$$

and for the second-order derivatives they are denoted by

$$\delta_x^2 u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \qquad \text{and} \qquad \delta_y^2 u_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}. \tag{16}$$

We consider next the discretization of the second-order cross-derivative term; a similar discretization for this term is described in [16]. In the following, we assume that the coefficient b for the cross-derivative in (14) is non positive. From the Taylor series, we obtain approximations

$$\begin{cases} u(x_{i+1}, y_{j+1}) \approx u + \Delta x \frac{\partial u}{\partial x} + \Delta y \frac{\partial u}{\partial y} + \frac{1}{2}\Big(\Delta x^2 \frac{\partial^2 u}{\partial x^2} + 2\Delta x \Delta y \frac{\partial u}{\partial x \partial y} + \Delta y^2 \frac{\partial^2 u}{\partial y^2}\Big), \\ u(x_{i-1}, y_{j-1}) \approx u - \Delta x \frac{\partial u}{\partial x} - \Delta y \frac{\partial u}{\partial y} + \frac{1}{2}\Big(\Delta x^2 \frac{\partial^2 u}{\partial x^2} + 2\Delta x \Delta y \frac{\partial u}{\partial x \partial y} + \Delta y^2 \frac{\partial^2 u}{\partial y^2}\Big), \end{cases} \tag{17}$$

where the value for u and its derivatives on the right side are evaluated at the grid point (x_i, y_j). The order of the accuracy of these approximations is $\mathcal{O}(\max(\Delta x, \Delta y)^3)$. 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 following. By summing the equations in (17), we get

$$2\Delta x \Delta y \frac{\partial u}{\partial x \partial y} \approx u(x_{i+1}, y_{j+1}) - 2u(x_i, y_j) + u(x_{i-1}, y_{j-1}) - \Delta x^2 \frac{\partial^2 u}{\partial x^2} - \Delta y^2 \frac{\partial^2 u}{\partial y^2}, \tag{18}$$

and dividing this by $2\Delta x \Delta y$, we obtain the approximation

$$\frac{\partial u}{\partial x \partial y} \approx \frac{1}{2\Delta x \Delta y} \Big[u(x_{i+1}, y_{j+1}) - 2u(x_i, y_j) + u(x_{i-1}, y_{j-1}) \Big] - \frac{\Delta x}{2\Delta y} \frac{\partial^2 u}{\partial x^2} - \frac{\Delta y}{2\Delta x} \frac{\partial^2 u}{\partial y^2}. \tag{19}$$

By performing the discretization of the cross-derivative using (19), we get for the second-order derivatives in (14)

$$\begin{aligned} a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} \approx &\Big[a - \frac{b\Delta x}{2\Delta y} \Big] \frac{\partial^2 u}{\partial x^2} + \Big[c - \frac{b\Delta y}{2\Delta x} \Big] \frac{\partial^2 u}{\partial y^2} \\ &+ \frac{b}{2\Delta x \Delta y} \Big[u(x_{i+1}, y_{j+1}) - 2u(x_i, y_j) + u(x_{i-1}, y_{j-1}) \Big]. \end{aligned} \tag{20}$$

Using the central finite differences (15) and (16) together with (20), we can approximate the partial differential equation (14) by the semi-discrete equation

$$\begin{aligned} \frac{\partial u}{\partial t} + \Big[a - \frac{b\Delta x}{2\Delta y} \Big] \delta_x^2 u_{i,j} + \Big[c - \frac{b\Delta y}{2\Delta x} \Big] \delta_y^2 u_{i,j} + d\delta_x u_{i,j} + e\delta_y u_{i,j} + f u_{i,j} \\ + \frac{b}{2\Delta x \Delta y} \Big[u_{i+1,j+1} - 2u_{i,j} + u_{i-1,j-1} \Big] = 0. \end{aligned} \tag{21}$$

After using the definitions (15) and (16) and rearranging terms, this equation has the form

$$\begin{aligned} \frac{\partial u}{\partial t} &+ \left(\frac{b}{2\Delta x \Delta y} \right) u_{i-1,j-1} + \left(\frac{1}{\Delta y^2} \Big[c - \frac{b\Delta y}{2\Delta x} \Big] - \frac{e}{2\Delta y} \right) u_{i,j-1} \\ &+ \left(\frac{1}{\Delta x^2} \Big[a - \frac{b\Delta x}{2\Delta y} \Big] - \frac{d}{2\Delta x} \right) u_{i-1,j} \\ &+ \left(-\frac{2}{\Delta x^2} \Big[a - \frac{b\Delta x}{2\Delta y} \Big] - \frac{2}{\Delta y^2} \Big[c - \frac{b\Delta y}{2\Delta x} \Big] - \frac{b}{\Delta x \Delta y} + f \right) u_{i,j} \\ &+ \left(\frac{1}{\Delta x^2} \Big[a - \frac{b\Delta x}{2\Delta y} \Big] + \frac{d}{2\Delta x} \right) u_{i+1,j} \\ &+ \left(\frac{1}{\Delta y^2} \Big[c - \frac{b\Delta y}{2\Delta x} \Big] + \frac{e}{2\Delta y} \right) u_{i,j+1} + \left(\frac{b}{2\Delta x \Delta y} \right) u_{i+1,j+1} = 0. \end{aligned} \tag{22}$$

Finally, we apply this finite difference scheme for the generalized Black-Scholes partial differential operator (3). After replacing the coefficients of (14) by the coefficients of the Black-Scholes equation, the space discretization scheme reads

$$\begin{aligned}
&\frac{\partial u}{\partial t} - \left(\frac{\rho\gamma}{2\Delta x\Delta y}x_i y_j\right)u_{i-1,j-1} - \left(\frac{1}{\Delta y^2}\left[\frac{1}{2}x_i^2 y_j - \frac{\rho\gamma\Delta y}{2\Delta x}x_i y_j\right] - \frac{\alpha(\beta - y_j)}{2\Delta y}\right)u_{i,j-1} \\
&\quad - \left(\frac{1}{\Delta x^2}\left[\frac{1}{2}x_i^2 y_j - \frac{\rho\gamma\Delta x}{2\Delta y}x_i y_j\right] - \frac{r}{2\Delta x}x_i\right)u_{i-1,j} \\
&\quad + \left(\frac{2}{\Delta x^2}\left[\frac{1}{2}x_i^2 y_j - \frac{\rho\gamma\Delta x}{2\Delta y}x_i y_j\right] + \frac{2}{\Delta y^2}\left[\frac{1}{2}\gamma^2 y_j - \frac{\rho\gamma\Delta y}{2\Delta x}x_i y_j\right] + \frac{\rho\gamma}{\Delta x\Delta y}x_i y_j + r\right)u_{i,j} \\
&\quad - \left(\frac{1}{\Delta x^2}\left[\frac{1}{2}x_i^2 y_j - \frac{\rho\gamma\Delta x}{2\Delta y}x_i y_j\right] + \frac{r}{2\Delta x}x_i\right)u_{i+1,j} \\
&\quad - \left(\frac{1}{\Delta y^2}\left[\frac{1}{2}\gamma^2 y_j - \frac{\rho\gamma\Delta y}{2\Delta x}x_i y_j\right] + \frac{\alpha(\beta - y_j)}{2\Delta y}\right)u_{i,j+1} - \left(\frac{\rho\gamma}{2\Delta x\Delta y}x_i y_j\right)u_{i+1,j+1} = 0,
\end{aligned} \tag{23}$$

for $i = 2, \ldots, n-1$ and $j = 2, \ldots m-1$. This defines a seven point discretization stencil for the generalized Black-Scholes equation.

The Dirichlet boundary conditions (6) and (7) are posed on the boundaries $x = 0$ and $y = 0$, respectively. These can be implemented in a straightforward manner with the finite difference stencil (23).

In the following we describe the treatment of the Neumann boundary conditions (8) and (9). Let us consider the boundary $x = X$. The other boundary condition on $y = Y$ can be handled in the same way. At a grid point (n, j), $j = 1, \ldots, m$, the boundary condition is $\partial u(X, y_j, t)/\partial x = 0$. We approximate this using the central finite difference operator in (15) and get

$$\delta_x u_{n,j} = \frac{u_{n+1,j} - u_{n-1,j}}{2\Delta x} = 0. \tag{24}$$

From this it follows that the fictitious grid point value $u_{n+1,j}$ outside the computational domain has to be the same as the grid point value $u_{n-1,j}$. We can use this knowledge to eliminate all fictitious grid point values $u_{n+1,j}$, $j = 1, \ldots, m$, appearing in the stencil (23) when it is used on the boundary $x = X$.

The space discretization leads to the semi-discrete equation which has the matrix representation

$$\frac{\partial u}{\partial t} + \mathbf{A}\mathbf{u} = 0, \tag{25}$$

where $\mathbf{A}$ is an $nm \times nm$ matrix and $\mathbf{u}$ is a vector of length nm.

3.2 Time discretization

In addition to the space discretization, the first-order time derivative in (25) needs to be approximated. In the option pricing problems the stability properties of time discretization schemes are essential, since the initial value (5) is nonsmooth. We consider three second-order accurate methods, which are the backward difference formula, a Runge-Kutta scheme and the Crank-Nicolson method, as well as the implicit Euler method.

The first-order accurate implicit Euler scheme is

$$\Big(\frac{1}{\Delta t}\mathbf{I}+\mathbf{A}\Big)\mathbf{u}^{(k+1)}=\Big(\frac{1}{\Delta t}\mathbf{I}\Big)\mathbf{u}^{(k)},\qquad \text{for } k=1,\dots,l-1. \tag{26}$$

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

$$\Big(\frac{1}{\Delta t}\mathbf{I}+\frac{1}{2}\mathbf{A}\Big)\mathbf{u}^{(k+1)}=\Big(\frac{1}{\Delta t}\mathbf{I}-\frac{1}{2}\mathbf{A}\Big)\mathbf{u}^{(k)},\qquad \text{for } k=1,\dots,l-1. \tag{27}$$

This scheme is unconditionally stable which means that no restriction for the step size Δt is needed [19]. Despite of this stability property, this method can lead to numerical solutions which oscillates; see for example [18].

The second-order backward difference formula (BDF2) is

$$\Big(\frac{2}{3}\mathbf{A}-\frac{1}{\Delta t}\mathbf{I}\Big)\mathbf{u}^{(k+1)}=\frac{1}{3}\frac{1}{\Delta t}\mathbf{u}^{(k-1)}-\frac{4}{3}\frac{1}{\Delta t}\mathbf{u}^{(k)},\qquad \text{for } k=2,\dots,l-1, \tag{28}$$

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 experiments. This scheme is L-stable which is a stronger property than the unconditional stability [11]. Due to this the solution obtained with the BDF2 does not have oscillations similar to those in the solution obtained with the Crank-Nicolson method.

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

$$\begin{cases} (\mathbf{I}+\theta\Delta t\mathbf{A})\overline{\mathbf{u}}^{(k+1)}=(\mathbf{I}-(1-\theta)\Delta t\mathbf{A})\mathbf{u}^{(k)}, \\[2ex] (\mathbf{I}+\theta\Delta t\mathbf{A})\mathbf{u}^{(k+1)}=(\mathbf{I}-\tfrac{1}{2}\Delta t\mathbf{A})\mathbf{u}^{(k)}-(\tfrac{1}{2}-\theta)\Delta t\mathbf{A}\overline{\mathbf{u}}^{(k+1)}, \end{cases} \tag{29}$$

for $k=1,\dots,l-1$. This scheme was proposed and shown to be L-stable in [5]. 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 linear complementarity problems

$$\begin{cases} \mathbf{B}\mathbf{u}^{(k+1)} \geq \mathbf{C}\mathbf{u}^{(k)}, \\ \mathbf{u}^{(k+1)} \geq \mathbf{g}, \\ \left(\mathbf{B}\mathbf{u}^{(k+1)} - \mathbf{C}\mathbf{u}^{(k)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{30}$$

for $k = 1, \ldots, l-1$. The discrete form of the formulation (12) with the auxiliary variable λ reads

$$\begin{cases} \mathbf{B}\mathbf{u}^{(k+1)} = \mathbf{C}\mathbf{u}^{(k)} + \boldsymbol{\lambda}^{(k+1)}, \\ \boldsymbol{\lambda}^{(k+1)} \geq 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g}, \\ \left(\boldsymbol{\lambda}^{(k+1)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{31}$$

for $k = 1, \ldots, l-1$. The following operator splitting methods are based on this formulation.

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, the solution and the variable $\boldsymbol{\lambda}$ are updated in such a way that they satisfy the constraints for them. Since the operator splitting depends on the underlying time discretization, we describe operator splitting methods for all previously considered time discretizations.

The operator splitting method for the implicit Euler scheme reads

$$\left(\tfrac{1}{\Delta t}\mathbf{I} + \mathbf{A}\right)\tilde{\mathbf{u}}^{(k+1)} = \left(\tfrac{1}{\Delta t}\mathbf{I}\right)\mathbf{u}^{(k)} + \boldsymbol{\lambda}^{(k)}, \tag{32}$$

$$\begin{cases} \tfrac{1}{\Delta t}\left(\mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)}\right) - \left(\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^{(k)}\right) = 0, \\ \left(\boldsymbol{\lambda}^{(k+1)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g} \quad \text{and} \quad \boldsymbol{\lambda}^{(k+1)} \geq 0, \end{cases} \tag{33}$$

for $k = 1, \ldots, l-1$. At each time step k the splitting method requires the solution of the system of linear equations (32) and the update for $\boldsymbol{\lambda}^{(k+1)}$ and $\mathbf{u}^{(k+1)}$ using the equations (33). In a similar manner, the operator splitting method for the Crank-Nicolson method is

$$\left(\tfrac{1}{\Delta t}\mathbf{I} + \tfrac{1}{2}\mathbf{A}\right)\tilde{\mathbf{u}}^{(k+1)} = \left(\tfrac{1}{\Delta t}\mathbf{I} - \tfrac{1}{2}\mathbf{A}\right)\mathbf{u}^{(k)} + \boldsymbol{\lambda}^{(k)}, \tag{34}$$

$$\begin{cases} \tfrac{1}{\Delta t}\left(\mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)}\right) - \left(\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^{(k)}\right) = 0, \\ \left(\boldsymbol{\lambda}^{(k+1)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g} \quad \text{and} \quad \boldsymbol{\lambda}^{(k+1)} \geq 0, \end{cases} \tag{35}$$

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

The operator splitting for the BDF2 discretization is

$$\Big(\tfrac{2}{3}\mathbf{A} - \tfrac{1}{\Delta t}\mathbf{I}\Big)\tilde{\mathbf{u}}^{(k+1)} = \tfrac{1}{3}\tfrac{1}{\Delta t}\mathbf{u}^{(k-1)} - \tfrac{4}{3}\tfrac{1}{\Delta t}\mathbf{u}^{(k)} + \tfrac{2}{3}\boldsymbol{\lambda}^{(k)}, \tag{36}$$

$$\left\{\begin{array}{l} \tfrac{1}{\Delta t}\Big(\mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)}\Big) - \tfrac{2}{3}\Big(\,\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^{(k)}\Big) = 0, \\ \Big(\,\boldsymbol{\lambda}^{(k+1)}\Big)^T\Big(\mathbf{u}^{(k+1)} - \mathbf{g}\Big) = 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g} \quad \text{and} \quad \boldsymbol{\lambda}^{(k+1)} \geq 0, \end{array}\right. \tag{37}$$

for $k = 2, \dots, l-1$. Again, the solution $\tilde{\mathbf{u}}^{(k+1)}$ is obtained by solving (36) and updates are performed using equations (37).

The Runge-Kutta scheme is a two step time discretization method. Thus, the operator splitting method based on the Runge-Kutta scheme has three fractional time steps instead of two. Two systems of linear equations are solved at each time steps and then the solution and $\boldsymbol{\lambda}$ are updated to satisfy the constraints. These three fractional steps are given by

$$(\mathbf{I} + \theta\Delta t\mathbf{A})\overline{\mathbf{u}}^{(k+1)} = (\mathbf{I} - (1-\theta)\Delta t\mathbf{A})\mathbf{u}^{(k)} + (1-\theta)\Delta t\ \boldsymbol{\lambda}^{(k)}, \tag{38}$$

$$(\mathbf{I} + \theta\Delta t\mathbf{A})\tilde{\mathbf{u}}^{(k+1)} = (\mathbf{I} - \tfrac{1}{2}\Delta t\mathbf{A})\mathbf{u}^{(k)} - (\tfrac{1}{2} - \theta)\Delta t\mathbf{A}\overline{\mathbf{u}}^{(k+1)} + (1-\theta)\Delta t\boldsymbol{\lambda}^{(k)}, \tag{39}$$

$$\left\{\begin{array}{l} \tfrac{1}{\Delta t}\Big(\mathbf{u}^{(k+1)} - \tilde{\mathbf{u}}^{(k+1)}\Big) - \Big(\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^{(k)}\Big) = 0, \\ \Big(\,\boldsymbol{\lambda}^{(k+1)}\Big)^T\Big(\mathbf{u}^{(k+1)} - \mathbf{g}\Big) = 0, \quad \mathbf{u}^{(k+1)} \geq \mathbf{g} \quad \text{and} \quad \boldsymbol{\lambda}^{(k+1)} \geq 0, \end{array}\right. \tag{40}$$

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

5 Numerical experiments

In the following experiments we study the properties of the operator splitting methods numerically. We compare the solutions obtained with the splitting methods with the ones obtained without a splitting using the PSOR method. A put option price is computed with the same parameter values as in [6], [17] and [21]. These parameters are

$$\alpha = 5.0, \quad \beta = 0.16, \quad \gamma = 0.9, \quad \rho = 0.1, \quad r = 0.1, \quad E = 10.0, \tag{41}$$

and $T = 0.25$. We use the computational domain

$$\Omega \times [0, T] = [0, 20] \times [0, 1] \times [0, 0.25] \tag{42}$$

in our experiments. This same domain was used also in [17].

In the first numerical experiment, we compare the put option prices obtained by an operator splitting method with the option prices computed using the PSOR method and the prices presented in the references. The time convergence is studied in the second experiment, and in the third experiment, we report the CPU-times for different grids. In the case of the operator splitting method, the system of linear of equations were solved by the SOR method at each time steps.

5.1 Option prices obtained by the operator splitting method

We compute put option prices with the operator splitting method (38) – (40) based on the Runge-Kutta time discretization. These prices are presented in Tables 1 and 2 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 grids in order the study the accuracy of numerical solutions. The triplets in the grids definitions are the number of steps in the x-direction, in the y-direction, and in the t-direction in this order. The option prices computed using the PSOR method are also given in the tables as well as the prices reported by Clarke and Parrott [6], Oosterlee [17], and Zvan, Forsyth and Vetzal [21].

		Asset price				
method	grid	8.0	9.0	10.0	11.0	12.0
O-S	(80,32,16)	2.00000	1.10442	0.50736	0.20446	0.07900
	(160,64,32)	2.00000	1.10695	0.51651	0.21106	0.08121
	(320,128,64)	2.00000	1.10751	0.51904	0.21294	0.08181
PSOR	(80,32,16)	2.00000	1.10435	0.50755	0.20462	0.07909
	(160,64,32)	2.00000	1.10675	0.51653	0.21112	0.08125
	(320,128,64)	2.00000	1.10738	0.51902	0.21296	0.08183
ref. [6]		2.0000	1.1080	0.5316	0.2261	0.0907
ref. [17]		2.00	1.107	0.517	0.212	0.0815
ref. [21]		2.0000	1.1076	0.5202	0.2138	0.0821

Table 1: Option prices for five different asset values at $y = 0.0625$.

		Asset price				
method	grid	8.0	9.0	10.0	11.0	12.0
O-S	(80,32,16)	2.07798	1.33205	0.79378	0.44634	0.24157
	(160,64,32)	2.07845	1.33333	0.79543	0.44775	0.24246
	(320,128,64)	2.07846	1.33360	0.79585	0.44813	0.24271
PSOR	(80,32,16)	2.07744	1.33192	0.79388	0.44650	0.24170
	(160,64,32)	2.07811	1.33317	0.79541	0.44779	0.24250
	(320,128,64)	2.07829	1.33350	0.79581	0.44813	0.24272
ref. [6]		2.0733	1.3290	0.7992	0.4536	0.2502
ref. [17]		2.079	1.334	0.796	0.449	0.243
ref. [21]		2.0784	1.3337	0.7961	0.4483	0.2428

Table 2: Option prices for five different asset values at $y = 0.25$.

As one can see from Tables 1 and 2 the operator splitting method produces prices which are in good agreement with the prices obtained without a splitting. Also, the prices obtained with the finest grid are similar to the ones given in [17] and [21].

5.2 Time convergence

In the following we study the time convergence of time discretization schemes. Again, we compare the numerical results obtained with the operator splitting methods and with the PSOR method. We use the same put option pricing problem as in the previous section.

The spatial grid was chosen to be $(80, 32)$ while the number of time steps varies from 4 to 256. Our reference solution is computed with the PSOR method using the Runge-Kutta time discretization and the grid defined by $(80, 32, 2048)$. The errors reported in the following are differences between computed solutions and the reference solution.

In Tables 3 and 4, we have given the errors at the point $(x, y) = (10.0, 0.25)$ while in Tables 5 and 6, we have reported the maximum errors over the whole computational domain. The associated convergence rates in the tables are calculated by dividing the error of the previous coarser time discretization by the error of the current discretization.

	Implicit Euler		Crank-Nicolson		BDF2		Runge-Kutta	
l	error	ratio	error	ratio	error	ratio	error	ratio
4	3.38464E-2		4.40538E-2		7.07421E-3		3.29540E-4	
8	1.77726E-2	1.90	8.61392E-3	5.11	2.38319E-3	2.97	2.63010E-4	1.25
16	9.24041E-3	1.92	1.77330E-4	48.58	9.04874E-4	2.63	1.41138E-4	1.86
32	4.75863E-3	1.94	6.12462E-5	2.90	3.48284E-4	2.60	6.92868E-5	2.04
64	2.43523E-3	1.95	2.85582E-5	2.14	1.36203E-4	2.56	2.85396E-5	2.43
128	1.23919E-3	1.96	1.37991E-5	2.07	5.30703E-5	2.57	1.31462E-5	2.17
256	6.28172E-4	1.97	6.87001E-6	2.00	2.13241E-5	2.49	6.82420E-6	1.93

Table 3: Errors and time convergence rates at $(x, y) = (10.0, 0.25)$ for the PSOR method.

The errors for all time discretizations are similar between the operator splitting method and the PSOR method. Actually for many discretizations the error for the splitting method is smaller. This suggest that the splitting error has the opposite sign to the error of the time discretization method which leads to the cancellation of a part of the error.

A first-order convergence rate can be observed for the implicit Euler method. None of the second-order accurate methods can maintain a second-order convergence when the time step is decreased. This suggest the solution is not regular enough to obtain a second-order convergence. For a small number of time steps, the Runge-Kutta scheme is the most accurate. The convergence of the Crack-Nicolson method is quite erratic. This due to lack of L-stability.

In Figure 1, we present the absolute value of the error for the PSOR method and the operator splitting method for the Runge-Kutta and Crank-Nicolson methods. It can be

	Implicit Euler		Crank-Nicolson		BDF2		Runge-Kutta	
l	error	ratio	error	ratio	error	ratio	error	ratio
4	3.41519E-2		4.41581E-2		8.33098E-3		2.35974E-3	
8	1.72542E-2	1.98	8.18615E-3	5.39	2.18976E-3	3.80	8.35683E-4	2.82
16	8.78909E-3	1.96	1.17041E-4	69.94	6.11676E-4	3.58	2.35564E-4	3.55
32	4.50501E-3	1.95	1.04473E-4	1.12	1.78201E-4	3.43	4.19806E-5	5.61
64	2.31843E-3	1.94	3.74388E-5	2.80	5.73676E-5	3.11	6.66921E-6	6.29
128	1.19649E-3	1.94	9.58596E-6	3.91	2.54857E-5	2.25	4.47167E-6	1.49
256	6.14664E-4	1.95	1.91744E-7	50.00	1.27642E-5	2.00	5.19710E-6	0.86

Table 4: Errors and time convergence rates at $(x, y) = (10.0, 0.25)$ for the operator splitting methods.

	Implicit Euler		Crank-Nicolson		BDF2		Runge-Kutta	
l	error	ratio	error	ratio	error	ratio	error	ratio
4	6.15884E-2		1.11281E-1		1.25347E-2		1.71188E-3	
8	3.19686E-2	1.93	4.47692E-2	2.49	3.16622E-3	3.95	1.03731E-3	1.65
16	1.64541E-2	1.94	8.68625E-3	5.15	1.00550E-3	3.15	3.16830E-4	3.27
32	8.41289E-3	1.96	1.47103E-4	59.05	4.09668E-4	2.45	7.18950E-5	4.41
64	4.28124E-3	1.97	3.16917E-5	4.64	1.66863E-4	2.46	3.27586E-5	2.19
128	2.16936E-3	1.97	1.60005E-5	1.98	6.69046E-5	2.49	1.48919E-5	2.20
256	1.09595E-3	1.98	7.79188E-6	2.05	2.67999E-5	2.50	7.12583E-6	2.09

Table 5: Maximum errors and time convergence rates for the PSOR method.

observed that the contour lines of the error are quite similar for the PSOR and splitting methods. The error for the Crank-Nicolson method is much more oscillatory. It can be also seen that the splitting error cancels a part of the time discretization error. These errors are for the numerical solutions computed using the grid defined by $(80, 32, 16)$ and the previously mentioned reference solution.

5.3 CPU-time comparison

In our last example, we report the CPU-times required to solve the previous described put option pricing problem on a J5600 HP workstation. Table 7 shows times for the PSOR method and the operator splitting method which uses the SOR method to solve the arising systems of linear equations. The time discretizations are based on the Runge-Kutta scheme. The relaxation parameter for the PSOR and SOR methods was 1.5 and the stopping criterion was that a weighted l^1-norm of the change is less than a fixed value. Based on these CPU-times, we see that the operator splitting method requires slightly less time.

When the number of steps are double in all directions the CPU-time increases almost 20 fold while the number of the space-time grid points becomes 8 times larger. Hence,

	Implicit Euler		Crank-Nicolson		BDF2		Runge-Kutta	
l	error	ratio	error	ratio	error	ratio	error	ratio
4	6.20757E-2		1.11083E-1		1.76833E-2		2.58122E-2	
8	3.14090E-2	1.98	4.42568E-2	2.51	4.56060E-3	3.88	8.25943E-3	3.13
16	1.59583E-2	1.97	8.32375E-3	5.32	1.45107E-3	3.14	2.32439E-3	3.55
32	8.11941E-3	1.97	5.03790E-4	16.52	4.66273E-4	3.11	5.52405E-4	4.21
64	4.13625E-3	1.96	1.13952E-4	4.42	1.37406E-4	3.39	2.16980E-4	2.55
128	2.11098E-3	1.96	3.18526E-5	3.58	3.81111E-5	3.61	4.56701E-5	4.75
256	1.07588E-3	1.96	7.83804E-6	4.06	1.57298E-5	2.42	1.07787E-5	4.24

Table 6: Maximum errors and time convergence rates for the operator splitting methods.

more scalable methods like multigrid methods could make solution much faster. When the number of time steps is increased for a given space grid the CPU-times do not change much. The iterative methods converge faster with smaller times steps, since the matrices become more diagonal dominant. Another observation is that the CPU-time increases by the factor 8 when we move from the grid (320,128,64) to the grid (640,128,64). Hence, the convergence of the SOR methods deteriorates very much when the grid is refined only in the x-direction.

	CPU-times	
grid	PSOR	O-S&SOR
(80,32,16)	7.71	6.00
(80,32,32)	10.27	8.26
(80,32,64)	17.89	14.43
(80,32,128)	31.51	24.96
(160,64,16)	164.27	117.05
(160,64,32)	152.96	116.06
(160,64,64)	150.06	116.49
(160,64,128)	152.35	131.52
(320,128,32)	3094.97	2429.71
(320,128,64)	2990.03	2303.22
(640,128,64)	23489.25	18774.22

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

6 Conclusion

We proposed operator splitting methods for pricing American options using Heston's stochastic volatility model. In these methods each time step is divided into two fractional

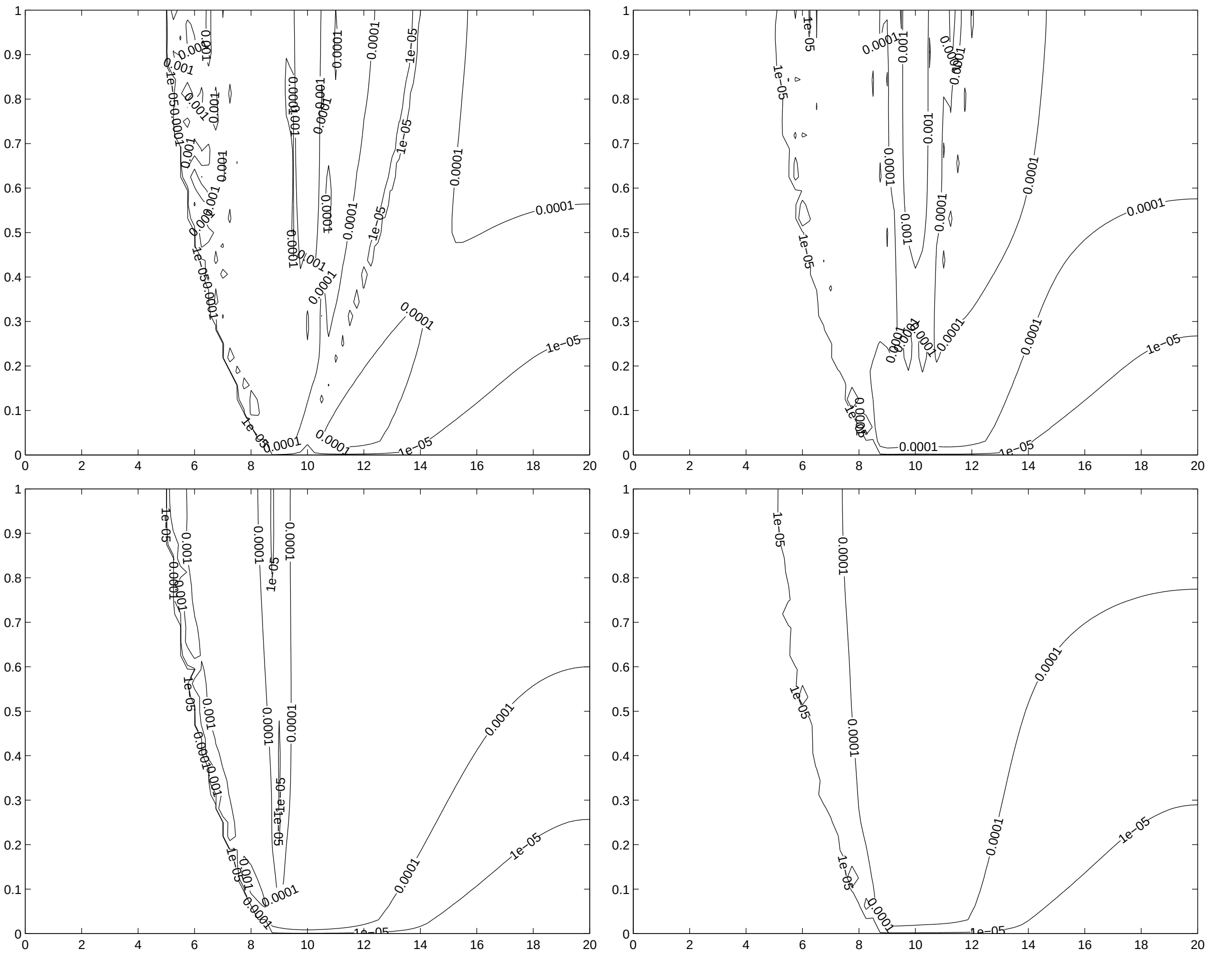

Figure 1: The absolute value of the errors for the operator splitting methods (left) and the PSOR methods (right) when the Crank-Nicolson method (top) and the Runge-Kutta method (bottom) are applied.

time steps. In the first step a system of linear equations is solved while in the second step the early exercise constraint is enforced by performing a simple update. With this approach any efficient solution method for a convection diffusion equation like a multigrid method can be used without any modifications.

We studied the accuracy of the operators splitting methods in the numerical experiments and we found out that their accuracy is similar to the accuracy of the PSOR method. Thus, the splitting does not essentially increase the error. Also, the computed prices were in good agreement with the prices published in the literature. The time convergence of the Crank-Nicolson method was somewhat erratic while the time convergence for the *L*-stable BDF2 and Runge-Kutta methods was more steady. None of the methods could maintain a second-order time convergence with uniform time steps.

In this paper, we did not consider efficient solution method for arising systems of linear equations. The study of multigrid methods for this is a future research topic. To our knowledge it is open question what is the regularity of the solution and what kind of convergence rates can be obtained for space and time discretizations. With this knowledge one could design optimal adaptive space and time discretizations which would lead to large reductions in computing times. Further analysis of the operator splitting methods is one possible topic of future research. Also, it would be interesting to apply these methods for pricing more complicated American options.

REFERENCES

[1] C.A. Ball and A. Roma. Stochastic volatility option pricing. *J. Financial Quant. Anal.*, **29**, 589–607, 1994.

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

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

[4] M.J. Brennan and E.S. Schwartz. The valuation of American put options. *J. Finance*, **32**, 449–462, 1977.

[5] J.R. Cash. Two new finite difference schemes for parabolic equations. *SIAM. J. Numer. Anal.*, **21**, 433–446, 1984.

[6] N. Clarke and K. Parrott. *The multigrid solution of two-factor American put options.* Res. Report 96-16, Oxford Comp. Lab, Oxford, 1996.

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

[8] A.A. Drăgulescu and V.M. Yakovenko. Probability distribution of returns in the Heston model with stochastic volatility. *Quant. Finance*, **2**, 443–453, 2002.

[9] J.-P. Fouque, G. Papanicolaou and K.R. Sircar. *Derivatives in financial markets with stochastic volatility.* Cambridge University Press, 2000.

[10] R. Glowinski. Finite element methods for incompressible viscous flow, *Handbook of numerical analysis*, Vol. IX, North-Holland, 2003.

[11] E. Hairer and G. Wanner. *Solving ordinary differential equations. II, Stiff and differential-algebraic problems*, 2nd ed., Springer-Verlag, 1996.

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

[13] J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *J. Finance*, **42**, 281–300, 1987.

[14] S. Ikonen and J. Toivanen. Operator splitting methods for American option pricing. *Appl. Math. Lett.*, To appear.

[15] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.*, **16**, 964–979, 1979.

[16] A.R. Mitchell and D.F. Griffiths. *The finite difference method in partial differential equations*, John Wiley & Sons Ltd., 1980.

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

[18] D.M. Pooley, K.R. Vetzal and P.A. Forsyth. Convergence remedies for non-smooth payoffs in option pricing. *J. Computational Finance*, **6**, 25–40, 2003.

[19] P. Wilmott, S. Howison and J. Dewynne. *The mathematics of financial derivatives.* Cambridge University Press, 1995.

[20] U. Trottenberg, C.W. Oosterlee and A. Schüller. *Multigrid*, With contributions by A. Brandt, P. Oswald and K. Stüben, Academic Press Inc., 2001.

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