

Componentwise Splitting Methods for Pricing American Options Under Stochastic Volatility

Samuli Ikonen* Jari Toivanen*

8th November 2005

Abstract

Efficient numerical methods for pricing American options using Heston's stochastic volatility model are proposed. Based on this model the price of a European option can be obtained by solving a two-dimensional parabolic partial differential equation. For an American option the early exercise possibility leads to a lower bound for the price of the option. This price can be computed by solving a linear complementarity problem.

The idea of operator splitting methods is to divide each time step into fractional time steps with simpler operators. This paper proposes componentwise splitting methods for solving the linear complementarity problem. The basic componentwise splitting decomposes the discretized problem to three linear complementarity problems with tridiagonal matrices. These problems can be efficiently solved using the Brennan and Schwartz algorithm which was originally introduced for American options under the Black and Scholes model. The accuracy of the componentwise splitting method is increased by applying the Strang symmetrization. The good accuracy and computational efficiency of the proposed symmetrized splitting method are demonstrated by numerical experiments.

Keywords: American option pricing, stochastic volatility model, linear complementarity problem, componentwise splitting method, Strang symmetrization

1 Introduction

The Black and Scholes model [1] assumes constant volatility for the price of the underlying asset. For stock options this simple asset price model does not lead to prices which would be consistent with the observed market prices [35]. Many different generalizations giving more consistent prices have been proposed. One such approach is to assume the volatility to be also stochastic in addition to the stochastic return component of the simple asset price model. Particularly in this paper we use the stochastic

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

volatility model proposed by Heston [16]. Based on this model under suitable assumptions [35] a parabolic partial differential equation can be derived for the price function of the option with respect to time, the price of the underlying asset, and its volatility.

For European options the price is obtained by solving a parabolic partial differential equation (PDE). The spatial operator is of convection diffusion type with varying coefficients and a second-order cross-derivative. In order to have a numerical solution this two-dimensional spatial operator is discretized using a finite difference method, a finite volume method, or a finite element method, for example. An implicit time discretization leads to a sequence of systems of linear equations. American options can be exercised anytime during the life time of the option contract and therefore it is more flexible than the European one. This early exercise possibility of an American option leads to a lower bound for the option price. By combining this early exercise constraint with the PDE a time-dependent linear complementarity problem (LCP) is obtained for the prices of American options. Our aim is to develop an efficient numerical method for solving these LCPs.

Many methods have been proposed for solving LCPs. Maybe the simplest and most often used method is the projected successive over relaxation (PSOR) method [8]. A disadvantage of the PSOR method is that the number of iterations grows when the discretization is refined in order to have more accurate option prices. This feature makes the PSOR method computationally inefficient for finer discretizations. In [5], [23], a special multigrid method called the projected full approximation scheme (PFAS) [2] is employed. For the pricing of options with multiple underlying assets PFAS has been considered in [29]. This approach leads to an efficient numerical method, but the implementation is not simple. This is particularly the case for LCPs obtained from Heston's model [23]. In [38] a penalty method has been proposed to treat the early exercise constraint of the American option. The resulting system of nonlinear equations at each time step is solved using Newton's method. A Newton iteration requires the solution of a system of linear equations with the nonsymmetric Jacobian matrix. A preconditioned conjugate gradient (PCG) -like method is used for this in [38]. Thus, this approach leads to a quite involved implementation. Furthermore, the convergence of the PCG-like method can deteriorate without a multilevel preconditioner when the discretization is refined.

We adopt a different approach for time-dependent LCPs based on a componentwise splitting methods [19]. The classical Peaceman-Rachford [24] and Douglas-Rachford [9] alternating direction implicit (ADI) methods were introduced already in the 1950s. Since then a large number of generalizations have been proposed and successfully applied to many problems; see the books [14], [19], and [37], for example. Some applications lead to PDEs with a second-order cross-derivative term. For such problems ADI methods are studied in [20] and [21].

The basic idea of these operator splitting methods is to decompose the spatial operator into simpler operators and then fractional time steps are performed with these simpler operators [14], [19], [37]. At each time step these simpler operators can be applied in several ways. In the ADI method the simpler operators are applied both implicitly and explicitly, while in the componentwise splitting method the decomposed opera-

tors are applied only in the implicit manner. This componentwise splitting method is also known as the fractional step method and the Marchuk-Yanenko scheme [14], [19]. One typical decomposition decouples the operator to be a set of one-dimensional operators. This makes the numerical solution easier and more efficient, since only a sequence of one-dimensional problems has to be solved. When the problems has a constraint then another approach is to treat the constraint in its own fractional time step. This is particularly popular approach to enforce the incompressibility constraint in fluid dynamics; see [14] and references therein.

The ADI and operator splitting methods have been applied also to the option pricing and, for example, the books [6], [10], and [34] discuss this. In [18] an operator splitting method for pricing American options under stochastic volatility is proposed. It decouples the convection diffusion operator and the early exercise constraint to separate fractional steps. In [6], an ADI method is applied for pricing European options with the payoff depending on two assets. By making a change of variables they obtain a PDE without second-order cross-derivative and then it is solved using a basic ADI method. In [33] the same situation is considered for American options. Due to early exercise possibility they obtain one-dimensional LCPs which they solve using the Brennan and Schwartz algorithm [3].

In this paper we propose a componentwise splitting method for pricing American options under stochastic volatility assumption. The convection diffusion operator with the early exercise constraint is decomposed into a sequence of one-dimensional LCPs. Due to the second-order cross-derivative there are also one-dimensional problems to the diagonal direction and therefore we have three LCPs to solve at each time step. Similarly to [33] we use the Brennan and Schwartz algorithm [3] to solve these LCPs. Furthermore, we perform the Strang symmetrization [31] to the basic componentwise splitting method. This increases the accuracy of the method, but it also increases the number of fractional time steps to five. The advantage of the componentwise splitting method is that we do not have to solve LCPs with large and sparse matrix arising from the discretization of the generalized two-dimensional Black-Scholes PDE. In the proposed componentwise splitting method only LCPs with tridiagonal matrices are solved at each time step. In order to attain a stable discretization leading to an M -matrix we apply nonuniform grids together with a selective use of one-sided finite differences.

The outline of the paper is the following. We begin by describing Heston's stochastic volatility model and the resulting two-dimensional parabolic PDE. Then space and time discretizations are constructed. The next sections introduce the componentwise splitting methods for European and American options. Then we study the properties of the resulting discretization matrices. Based on this and accuracy considerations we propose nonuniform grids. After this we present numerical experiments demonstrating the accuracy and computational efficiency of the proposed splitting approach. We perform comparisons with the PSOR method. The last section gives our conclusions.

2 Option pricing models

A typical starting point for the pricing of option contracts is the model for the price behavior of the underlying asset. Particularly models based on stochastic differential equations (SDEs) are popular. For example, the famous Black-Scholes formula [1] assumes that the relative price of an asset follows a SDE given by a drift term and a Wiener process. Often it is possible to derive a partial differential equation (PDE) for the price of a European option using the SDE. Based on the PDE numerical methods can be developed for pricing options; see, for example, the books [17] and [35].

The Black-Scholes model assumes the volatility of the price of the asset to be a constant. In practice for stock options one has to use different values of the volatility in the Black-Scholes formula in order to obtain the market prices of options on the same stock with different exercise prices and expiry dates. Thus, usually the constant volatility assumption does not hold. There are several generalization leading to more realistic prices. For example, by choosing the volatility to be an appropriate function of time and stock price one can obtain market prices [35].

In this paper we consider a stochastic volatility model which is one way to generalize the Black-Scholes model leading to more compatible prices with the markets. Several such models exist; see [13] and references therein. Particularly, we use the model introduced by Heston in [16]. It assumes the variance of the price of the asset, that is, the square root of the volatility, to be a stochastic mean reverting square root process. Furthermore, the Wiener processes for the price of the asset and its variance can have a given correlation.

2.1 Model for European options

Although the aim of this paper is to develop numerical methods for American options it is more convenient to begin by considering European options. Under Heston's stochastic volatility model and suitable assumptions on the markets a two-dimensional parabolic PDE can be derived for the price of European type options; see [17], [38], for example. We denote the price of the asset by x and its variance by y . The price u of a European option is given by a payoff function g at the expiry date T and the price before the expiry can be obtained by integrating a parabolic PDE backwards in time. In the following we will change the direction of time so that the PDE becomes an initial value problem and integration is performed forward in time.

We define a generalized Black-Scholes operator

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

where the r is risk free interest rate, β is the mean level of the variance, α is the rate of reversion on the mean level, and γ is the volatility of the variance. The correlation between the price of the asset and its variance is ρ . Furthermore, ϑ is the so-called market price of the risk which we assume to be zero as has been done in [5], [23], and [38]. Also non zero values of ϑ can be treated in a straight forward manner in the following numerical methods.

Now the price u of a European option is given by a PDE

$$Lu = 0 \tag{2}$$

in an unbounded domain $\{(x, y, t) \mid x \geq 0, y \geq 0, t \in [0, T]\}$ with the initial value

$$u(x, y, 0) = g(x, y) \tag{3}$$

together with boundary conditions. For a put option giving the right to sell the asset with the exercise price E at the expiry T the payoff function is

$$g(x, y) = \max(E - x, 0). \tag{4}$$

2.2 Model for American options

An American option can be exercised at any time before the expiry date. Due to this the price u of the American option should be at least the same as the payoff function g . This leads to the early exercise constraint $u(x, y, t) \geq g(x, y)$. It is not known a priori where this constraint is active and this makes deriving analytical formulas for the price intractable. The most common way to obtain the price u is to formulate a linear complementarity problem for u and then solve it numerically; see [5], [18], [23], and [38], for example. The linear complementarity problem reads

$$\begin{cases} Lu \geq 0, & u \geq g, \\ (u - g)Lu = 0, \end{cases} \tag{5}$$

in $\{(x, y, t) \mid x \geq 0, y \geq 0, t \in [0, T]\}$ with the same initial condition as in (3).

The boundary conditions are described, for example, in [5] and [23]. On the boundaries $x = 0$ and $y = 0$ they are given by

$$u(0, y, t) = g(0, y) \quad \text{and} \quad u(x, 0, t) = g(x, 0), \tag{6}$$

respectively. On far-field the asymptotic behavior of u satisfies the conditions

$$\lim_{x \rightarrow \infty} \frac{\partial u(x, y, t)}{\partial x} = 0 \quad \text{and} \quad \lim_{y \rightarrow \infty} \frac{\partial u(x, y, t)}{\partial y} = 0. \tag{7}$$

3 Space and time discretization

We discretize the partial derivatives in (1) using finite difference methods. The spatial discretization is described in Section 3.1 and the time stepping methods are considered in Section 3.2. The option pricing problems are defined in the unbounded domain

$$\{(x, y, t) \mid x \geq 0, y \geq 0, t \in [0, T]\}. \tag{8}$$

We truncate this domain into a finite computational domain

$$(x, y, t) \in [0, X] \times [0, Y] \times [0, T] = \Omega \times [0, T], \tag{9}$$

where X and Y are large enough so that the error in the price u due to the truncation is negligible. For example, X could be two or three times larger than the exercise price E and Y could be several times larger than the mean level β of the variance.

We obtain approximate boundary conditions on the artificial boundaries $\{X\} \times [0, Y]$ and $[0, X] \times \{Y\}$ by assuming the asymptotic behavior of u given by (7) already on these boundaries. Thus, similarly to [23] we have the Neumann boundary conditions

$$\frac{\partial u(x, y, t)}{\partial x} = 0 \quad \text{on } \{X\} \times [0, Y] \quad (10)$$

and

$$\frac{\partial u(x, y, t)}{\partial y} = 0 \quad \text{on } [0, X] \times \{Y\}. \quad (11)$$

The number of grid steps is denoted by m , n and l in the x -direction, in the y -direction, and in the t -direction, respectively. Furthermore, the grid point values of the finite difference approximation are denoted by

$$u_{i,j}^{(k)} \approx u(x_i, y_j, t_k), \quad (12)$$

where $i = 0, \dots, m$, $j = 0, \dots, n$, and $k = 0, \dots, l$. Constant steps sizes are used in the y -direction and in the t -direction. In the x -direction we employ a nonuniform grid which is constructed in Section 6.

3.1 Space discretization

We use a non-standard finite difference method for the space discretization on nonuniform grids. For a general description of finite difference methods see [22], [30], [32], for example. We apply second-order accurate central finite differences for the second-order derivatives in (1). The first-order derivatives are mostly approximated using central finite difference schemes. In a small part of the domain Ω where the convection dominates the diffusion one-sided differences are used in order to attain non positive off-diagonal elements. The cross-derivative term is approximated using a second-order accurate scheme.

We introduce the discretization scheme for a reference grid point (x, y) . In the x -direction we use a nonuniform grid. Our main motivation for this is to avoid positive off-diagonal weights in the finite difference stencil which could lead to oscillations in the numerical solution. We have neighbor grid points at $(x - h_l, y)$ and at $(x + h_r, y)$, where h_l and h_r are the local left and right grid step sizes, respectively. The choice of these step sizes is considered in Section 6. In the y -direction we employ a uniform grid with the step size h and, hence, neighbor grid points are located at $(x, y - h)$ and at $(x, y + h)$.

For the rest of the paper we assume that the correlation ρ between the price of the asset and its variance is non negative. This dictates which grid point values we use to approximate the second-order cross-derivative in order to obtain non positive off-diagonal weights in the finite difference stencil. For a non negative ρ the proper choice is the values at the grids points $(x - h_l, y - h)$ and $(x + h_r, y + h)$. A similar

discretization to the following one can be constructed for the case $\rho < 0$, but we do not consider it here for brevity. The structure of our seven point finite difference stencil is shown in Figure 1. Similar stencils are constructed using a finite element method in [39].

The first-order and second-order derivatives in the x -direction are approximated in the following way. For the value of u at $(x - h_l, y)$ and at $(x + h_r, y)$ the Taylor series gives the approximations

$$u(x - h_l, y) \approx u - h_l \frac{\partial u}{\partial x} + \frac{1}{2} h_l^2 \frac{\partial^2 u}{\partial x^2} \quad (13)$$

and

$$u(x + h_r, y) \approx u + h_r \frac{\partial u}{\partial x} + \frac{1}{2} h_r^2 \frac{\partial^2 u}{\partial x^2}, \quad (14)$$

respectively. For simplicity, we have omitted the arguments of the function u and its derivatives when they are evaluated at the reference grid point (x, y) . The accuracies of the previous approximations are $\mathcal{O}(h_l^3)$ and $\mathcal{O}(h_r^3)$, respectively.

By taking a suitable linear combination of (13) and (14) we obtain the finite difference approximation

$$\frac{\partial u}{\partial x} \approx \frac{1}{h_l + h_r} \left(\frac{h_l}{h_r} u(x + h_r, y) - \left(\frac{h_l}{h_r} - \frac{h_r}{h_l} \right) u - \frac{h_r}{h_l} u(x - h_l, y) \right) \quad (15)$$

for the first-order derivative in the x -direction. When the convection term dominates the diffusion term the use of the approximation (15) can lead to positive off-diagonal elements. In this case the forward and backward finite difference approximations

$$\frac{\partial u}{\partial x} \approx \frac{u(x + h_r, y) - u(x, y)}{h_r} \quad \text{and} \quad \frac{\partial u}{\partial x} \approx \frac{u(x, y) - u(x - h_l, y)}{h_l} \quad (16)$$

are employed in order to achieve non positive off-diagonal elements.

Again by taking a suitable linear combination of (13) and (14) we obtain the approximation

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{2}{h_l + h_r} \left(\frac{1}{h_l} u(x - h_l, y) - \left(\frac{1}{h_l} + \frac{1}{h_r} \right) u + \frac{1}{h_r} u(x + h_r, y) \right) \quad (17)$$

for the second-order derivative in the x -direction.

In the y -direction we apply a uniform grid and hence the finite difference approximations are more familiar. The first-order derivatives are approximated by second-order and first-order accurate schemes:

$$\frac{\partial u}{\partial y} \approx \frac{u(x, y + h) - u(x, y - h)}{2h}, \quad (18)$$

$$\frac{\partial u}{\partial y} \approx \frac{u(x, y + h) - u(x, y)}{h}, \quad \text{and} \quad \frac{\partial u}{\partial y} \approx \frac{u(x, y) - u(x, y - h)}{h}. \quad (19)$$

The forward and backward schemes are again applied in a part of the domain so that we obtain a matrix with non positive off-diagonal elements. The second-order derivate in the y -direction is approximated by the central finite difference

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u(x, y - h) - 2u(x, y) + u(x, y + h)}{h^2}. \quad (20)$$

The last spatial derivative to be discretized in (1) is the second-order cross-derivative. By forming a Taylor series for the value of u at $(x + h_r, y + h_u)$ we obtain the approximation

$$u(x + h_r, y + h) \approx u + h_r \frac{\partial u}{\partial x} + h \frac{\partial u}{\partial y} + \frac{1}{2} h_r^2 \frac{\partial^2 u}{\partial x^2} + \frac{1}{2} h^2 \frac{\partial^2 u}{\partial y^2} + h_r h \frac{\partial^2 u}{\partial x \partial y}. \quad (21)$$

Dividing (21) by $h_r h$ gives the approximation

$$\frac{\partial^2 u}{\partial x \partial y} \approx \frac{1}{h_r h} \left(u(x + h_r, y + h) - u - h_r \frac{\partial u}{\partial x} - h \frac{\partial u}{\partial y} - \frac{1}{2} h_r^2 \frac{\partial^2 u}{\partial x^2} - \frac{1}{2} h^2 \frac{\partial^2 u}{\partial y^2} \right). \quad (22)$$

Similarly, we obtain the approximation

$$\frac{\partial^2 u}{\partial x \partial y} \approx \frac{1}{h_l h} \left(u(x - h_l, y - h) - u + h_l \frac{\partial u}{\partial x} + h \frac{\partial u}{\partial y} - \frac{1}{2} h_l^2 \frac{\partial^2 u}{\partial x^2} - \frac{1}{2} h^2 \frac{\partial^2 u}{\partial y^2} \right), \quad (23)$$

by using a Taylor series for the value of u at the grid point $(x - h_l, y - h)$.

In order to obtain a discretization with desired properties we use the convex combination

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} \approx & \frac{w}{h_l h} \left(u(x - h_l, y - h) - u + h_l \frac{\partial u}{\partial x} + h \frac{\partial u}{\partial y} - \frac{1}{2} h_l^2 \frac{\partial^2 u}{\partial x^2} - \frac{1}{2} h^2 \frac{\partial^2 u}{\partial y^2} \right) \\ & + \frac{1-w}{h_r h} \left(u(x + h_r, y + h) - u - h_r \frac{\partial u}{\partial x} - h \frac{\partial u}{\partial y} - \frac{1}{2} h_r^2 \frac{\partial^2 u}{\partial x^2} - \frac{1}{2} h^2 \frac{\partial^2 u}{\partial y^2} \right), \end{aligned} \quad (24)$$

of (22) and (23) for the second-order cross-derivative. The weighting parameter w has value between zero and one. For most of the grid points we use the value $w = 0.5$. On the boundaries $\{X\} \times [0, Y]$ and $[0, X] \times \{Y\}$ where the Neumann boundary conditions (10) and (11), respectively, are posed, we use the parameter value $w = 1$. This choice simplifies the componentwise splitting method introduced in Section 4. On the grid line $x = x_1$ we use the value $w = 0$ which helps to avoid positive off-diagonal elements.

Using the approximation (24) for the cross-derivative in the partial differential equation (2), we obtain a form which contains only the partial derivatives in the x -direction

and in the y -direction. This intermediate form reads

$$\begin{aligned}
& \frac{\partial u}{\partial t} + \left[-\frac{1}{2}yx^2 + w\rho\gamma yx\frac{h_l}{2h} + (1-w)\rho\gamma yx\frac{h_r}{2h} \right] \frac{\partial^2 u}{\partial x^2} \\
& + \left[-\frac{1}{2}\gamma^2 y + w\rho\gamma yx\frac{h}{2h_l} + (1-w)\rho\gamma yx\frac{h}{2h_r} \right] \frac{\partial^2 u}{\partial y^2} \\
& + \left[-rx - w\rho\gamma yx\frac{1}{h} + (1-w)\rho\gamma yx\frac{1}{h} \right] \frac{\partial u}{\partial x} \\
& + \left[-\alpha(\beta - y) - w\rho\gamma yx\frac{1}{h_l} + (1-w)\rho\gamma yx\frac{1}{h_r} \right] \frac{\partial u}{\partial y} \\
& + \left[r + w\rho\gamma yx\frac{1}{h_l h} + (1-w)\rho\gamma yx\frac{1}{h_r h} \right] u \\
& - w\rho\gamma yx\frac{1}{h_l h} u(x - h_l, y - h) - (1-w)\rho\gamma yx\frac{1}{h_r h} u(x + h_r, y + h) = 0.
\end{aligned} \tag{25}$$

The seven point finite difference stencil is obtained by using the finite difference approximations introduced in this section for the spatial derivatives appearing in (25). We describe the use of the forward and backward differences in Section 6.

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

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}\mathbf{u} = 0, \tag{26}$$

where \mathbf{A} is a block tridiagonal $(m+1)(n+1) \times (m+1)(n+1)$ matrix and \mathbf{u} is a vector of length $(m+1)(n+1)$.

3.2 Time discretization

Here we consider the temporal discretization of the semi-discrete problem (26) using the implicit Euler scheme and the Crank-Nicolson method. These methods are commonly used for pricing options [30], [32], [34]. Furthermore, we consider the Rannacher time stepping [28] which uses these both method. It avoids the undesired oscillations caused by the Crank-Nicolson method.

The implicit Euler scheme reads

$$\left(\mathbf{I} + \Delta t \mathbf{A} \right) \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)}, \quad \text{for } k = 0, \dots, l-1, \tag{27}$$

where \mathbf{I} is the identity matrix. Once $\mathbf{u}^{(k)}$ is known the next finite difference approximation $\mathbf{u}^{(k+1)}$ is obtained by solving the system of linear equations given by (27). This scheme has good stability properties, that is, it is unconditionally stable and also L -stable [15], but it is only first-order accurate in time. Thus, a large number of time steps might be required to reach sufficiently high accuracy.

The well-known second-order accurate Crank-Nicolson method is given by

$$\left(\mathbf{I} + \frac{1}{2} \Delta t \mathbf{A} \right) \mathbf{u}^{(k+1)} = \left(\mathbf{I} - \frac{1}{2} \Delta t \mathbf{A} \right) \mathbf{u}^{(k)}, \quad \text{for } k = 0, \dots, l-1. \tag{28}$$

This method is also unconditionally stable, but it is not L -stable. Due to the lack of L -stability numerical solutions might have oscillations.

Rannacher concluded in [28] that the Crank-Nicolson method does not damp out fast enough oscillatory high frequency components in the initial value $\mathbf{u}^{(0)}$. He suggested to take a few first time steps using the implicit Euler scheme and then use the Crank-Nicolson method. This Rannacher time stepping scheme reduces oscillations, but it still maintains second-order accuracy. For option pricing it has been employed in [12], [27], for example. In Section 7 our numerical results with the Rannacher time stepping are computed by starting with two implicit Euler steps.

3.3 Discrete problems

After the discretization of the underlying parabolic partial differential equation with two space variables an approximate price of an European option can be obtained by solving a sequence of systems of linear equations

$$\mathbf{B}\mathbf{u}^{(k+1)} = \mathbf{C}\mathbf{u}^{(k)}, \quad (29)$$

for $k = 0, \dots, l-1$. The matrices \mathbf{B} and \mathbf{C} in (29) are defined by (27) for the implicit Euler scheme and by (28) for the Crank-Nicolson method. The initial value $\mathbf{u}^{(0)}$ is given by the discrete form \mathbf{g} of the payoff function g of the option.

Similarly an approximate price of an American option can be obtained by solving a sequence of linear complementarity problems

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

for $k = 0, \dots, l-1$. The inequalities in (30) for vectors are componentwise. The equation in (30) is called the complementarity condition. Again the initial value $\mathbf{u}^{(0)}$ is given by the vector \mathbf{g} .

4 Componentwise splitting method for European options

The solution of the linear systems (29) becomes time consuming when the number of grid points is increased in order to obtain a more accurate option price. Therefore it is worthwhile to develop efficient iterative or direct methods for solving these linear systems sufficiently accurately. We propose to approximate the solutions using componentwise splitting methods. These splitting methods are based on a decomposition of the matrix \mathbf{A} in (26) into three simpler ones:

$$\mathbf{A} = \mathbf{A}_x + \mathbf{A}_{xy} + \mathbf{A}_y. \quad (31)$$

The matrices \mathbf{A}_x , \mathbf{A}_{xy} , and \mathbf{A}_y contain the couplings of the finite difference stencil in the x -direction, in the xy -direction, and in the y -direction, respectively. This is

illustrated in Figure 1. For example, the row of \mathbf{A}_x corresponding to the grid point (i, j) has non zero values corresponding to the grid points $(i-1, j)$, (i, j) , and $(i+1, j)$. Thus, \mathbf{A}_x contains the discretized terms from (25) which have partial derivatives in the x -direction. The term ru contributing only to the diagonal is split into three equal parts in \mathbf{A}_x , \mathbf{A}_{xy} , and \mathbf{A}_y .

By performing different reorderings of unknowns for each of the matrices \mathbf{A}_x , \mathbf{A}_{xy} , and \mathbf{A}_y they can be made tridiagonal. For example, by first ordering the grid points from left to right and then from bottom to top the matrix \mathbf{A}_x will be tridiagonal. We will take an advantage of this in the following. When we refer to these matrices as tridiagonal ones the unknowns are assumed to be reordered accordingly.

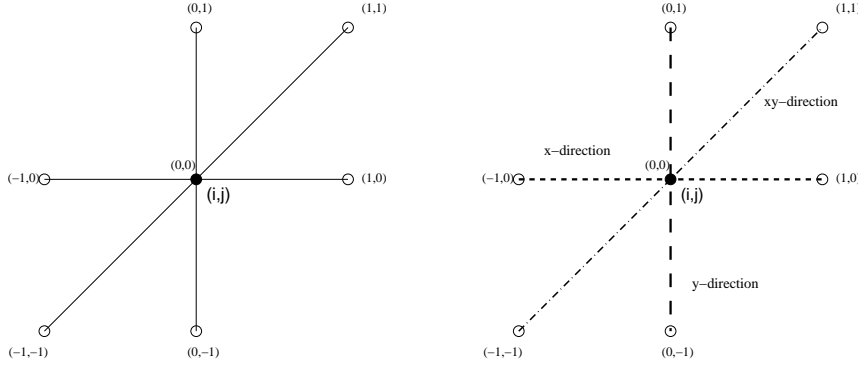


Figure 1: Discretization stencil (left) and its splitting (right).

We perform the componentwise splitting in the discrete level while it is more common to do it in the partial differential operator level. The reason for this is that the second-order cross-derivative term has to be decomposed in the discrete level in order to obtain a computationally feasible splitting.

There are also many other possible operator splittings. For example, the diffusion and convection terms can be handled separately in splitting schemes; see [14], [19], [37]. Sometimes other names are used for the componentwise splitting method. In [37] the method is called as the fractional step method and in [14] it is referred as the Marchuk-Yanenko scheme.

4.1 Basic splitting method

The motivation of the componentwise splitting method is to reduce the original two-dimensional problem into a sequence of one-dimensional problems which are easier to solve with a computer. We split the matrix \mathbf{A} into the matrices \mathbf{A}_x , \mathbf{A}_{xy} , and \mathbf{A}_y and then we perform a time step with each of them. The splitting method approximates the original problem (29) by three subproblems

$$\begin{aligned}
 \mathbf{B}_x \mathbf{u}^{(k+1/3)} &= \mathbf{C}_x \mathbf{u}^{(k)}, \\
 \mathbf{B}_{xy} \mathbf{u}^{(k+2/3)} &= \mathbf{C}_{xy} \mathbf{u}^{(k+1/3)}, \\
 \mathbf{B}_y \mathbf{u}^{(k+1)} &= \mathbf{C}_y \mathbf{u}^{(k+2/3)},
 \end{aligned} \tag{32}$$

for $k = 0, \dots, l-1$. Hence, we have to solve three problems instead of one at each time step. The solution of the first subproblem is the intermediate solution $\mathbf{u}^{(k+1/3)}$ which is also the initial value for the second subproblem. The intermediate solution $\mathbf{u}^{(k+2/3)}$ has a similar role between the second and third subproblems.

With the Crank-Nicolson method the matrices appearing in (32) are

$$\begin{aligned} \mathbf{B}_x &= \mathbf{I} + \frac{1}{2}\Delta t \mathbf{A}_x, & \mathbf{B}_{xy} &= \mathbf{I} + \frac{1}{2}\Delta t \mathbf{A}_{xy}, & \mathbf{B}_y &= \mathbf{I} + \frac{1}{2}\Delta t \mathbf{A}_y, \\ \mathbf{C}_x &= \mathbf{I} - \frac{1}{2}\Delta t \mathbf{A}_x, & \mathbf{C}_{xy} &= \mathbf{I} - \frac{1}{2}\Delta t \mathbf{A}_{xy}, & \text{and } \mathbf{C}_y &= \mathbf{I} - \frac{1}{2}\Delta t \mathbf{A}_y, \end{aligned} \quad (33)$$

and with the implicit Euler scheme they read as

$$\mathbf{B}_x = \mathbf{I} + \Delta t \mathbf{A}_x, \quad \mathbf{B}_{xy} = \mathbf{I} + \Delta t \mathbf{A}_{xy}, \quad \mathbf{B}_y = \mathbf{I} + \Delta t \mathbf{A}_y, \quad \text{and} \quad \mathbf{C}_x = \mathbf{C}_{xy} = \mathbf{C}_y = \mathbf{I}. \quad (34)$$

In both cases \mathbf{B}_x , \mathbf{B}_{xy} , and \mathbf{B}_y are tridiagonal matrices. Thus, the advantage of the splitting is that we need to solve problems with tridiagonal matrices instead of the block tridiagonal matrix \mathbf{B} . This makes the solution computationally much more easier. These problems can be solved efficiently using the \mathbf{LU} decomposition.

The subproblems with \mathbf{B}_x , \mathbf{B}_{xy} , and \mathbf{B}_y can be further divided into smaller tridiagonal problems. The problems with \mathbf{B}_x decouple into n tridiagonal problems of size $m+1$. Each of these problems corresponds to a horizontal grid line between the grid points $(0, j)$ and (m, j) for some j from 1 to n . Similarly, the problems with \mathbf{B}_y decouple into m tridiagonal problems of size $n+1$. The tridiagonal problems associated with \mathbf{B}_{xy} are of varying size along the diagonal grid lines.

The componentwise splitting method (32) is first-order accurate with respect to the time step size Δt even with the Crank-Nicolson method. This is shown, for example, in [14]. Thus, a large number of time steps might be needed to obtain a sufficiently accurate price of the European option.

4.2 Strang symmetrized splitting method

A second-order accurate splitting method can be achieved by performing a Strang symmetrization [31] for the splitting method (32) which uses the Crank-Nicolson method. For a threeway splitting the Strang symmetrization has been considered in [14]. With three operators there are several possible symmetrizations. We choose to perform first a half time step with \mathbf{A}_x and then with \mathbf{A}_y , a full time step with \mathbf{A}_{xy} , and finally a half time step with \mathbf{A}_y and then with \mathbf{A}_x .

In the following we use the notations

$$\begin{aligned} \mathbf{B}_{x/2} &= \mathbf{I} + \frac{1}{4}\Delta t \mathbf{A}_x, & \mathbf{B}_{y/2} &= \mathbf{I} + \frac{1}{4}\Delta t \mathbf{A}_y, \\ \mathbf{C}_{x/2} &= \mathbf{I} - \frac{1}{4}\Delta t \mathbf{A}_x, & \text{and } \mathbf{C}_{y/2} &= \mathbf{I} - \frac{1}{4}\Delta t \mathbf{A}_y. \end{aligned} \quad (35)$$

These matrices define half time steps based on the Crank-Nicolson method with \mathbf{A}_x

and \mathbf{A}_y . Now the Strang symmetrized componentwise splitting method reads

$$\begin{aligned}
\mathbf{B}_{x/2}\mathbf{u}^{(k+1/5)} &= \mathbf{C}_{x/2}\mathbf{u}^{(k)}, \\
\mathbf{B}_{y/2}\mathbf{u}^{(k+2/5)} &= \mathbf{C}_{y/2}\mathbf{u}^{(k+1/5)}, \\
\mathbf{B}_{xy}\mathbf{u}^{(k+3/5)} &= \mathbf{C}_{xy}\mathbf{u}^{(k+2/5)}, \\
\mathbf{B}_{y/2}\mathbf{u}^{(k+4/5)} &= \mathbf{C}_{y/2}\mathbf{u}^{(k+3/5)}, \\
\mathbf{B}_{x/2}\mathbf{u}^{(k+1)} &= \mathbf{C}_{x/2}\mathbf{u}^{(k+4/5)},
\end{aligned} \tag{36}$$

for $k = 0, \dots, l-1$. Hence, five tridiagonal problems have to be solved in order to perform one full time step.

5 Componentwise splitting method for American options

In order to obtain an approximate price of an American option the sequence (30) of linear complementarity problems (LCPs) have to be solved. The solution of a LCP is computationally more expensive than the solution of the corresponding system of linear equations. Furthermore, it is more difficult to develop fast solution methods for LCPs. In the following we propose componentwise splitting methods for LCPs based on the decomposition (31) of the matrix \mathbf{A} .

5.1 Basic splitting method

The componentwise splitting method for pricing American options has the same structure as the one (32) for pricing European options. The difference is that the systems of linear equations are replaced by LCPs. The splitting method approximates the original LCP (30) by three LCPs:

$$\begin{cases} \mathbf{B}_x\mathbf{u}^{(k+1/3)} \geq \mathbf{C}_x\mathbf{u}^{(k)}, & \mathbf{u}^{(k+1/3)} \geq \mathbf{g}, \\ \left(\mathbf{B}_x\mathbf{u}^{(k+1/3)} - \mathbf{C}_x\mathbf{u}^{(k)}\right)^T \left(\mathbf{u}^{(k+1/3)} - \mathbf{g}\right) = 0, \end{cases} \tag{37}$$

$$\begin{cases} \mathbf{B}_{xy}\mathbf{u}^{(k+2/3)} \geq \mathbf{C}_{xy}\mathbf{u}^{(k+1/3)}, & \mathbf{u}^{(k+2/3)} \geq \mathbf{g}, \\ \left(\mathbf{B}_{xy}\mathbf{u}^{(k+2/3)} - \mathbf{C}_{xy}\mathbf{u}^{(k+1/3)}\right)^T \left(\mathbf{u}^{(k+2/3)} - \mathbf{g}\right) = 0, \end{cases} \tag{38}$$

$$\begin{cases} \mathbf{B}_y\mathbf{u}^{(k+1)} \geq \mathbf{C}_y\mathbf{u}^{(k+2/3)}, & \mathbf{u}^{(k+1)} \geq \mathbf{g}, \\ \left(\mathbf{B}_y\mathbf{u}^{(k+1)} - \mathbf{C}_y\mathbf{u}^{(k+2/3)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \tag{39}$$

for $k = 0, \dots, l-1$. Each of (37)–(39) is a set of LCPs with tridiagonal matrices. In Section 5.3 we consider the solution of these LCPs using the Brennan and Schwartz algorithm [3].

5.2 Strang symmetrized splitting method

The symmetrized splitting method can be obtained from (36) by replacing the systems of linear equations by the corresponding LCPs. Now the original LCP (30) is approximated by five LCPs:

$$\begin{cases} \mathbf{B}_{x/2}\mathbf{u}^{(k+1/5)} \geq \mathbf{C}_{x/2}\mathbf{u}^{(k)}, & \mathbf{u}^{(k+1/5)} \geq \mathbf{g}, \\ \left(\mathbf{B}_{x/2}\mathbf{u}^{(k+1/5)} - \mathbf{C}_{x/2}\mathbf{u}^{(k)}\right)^T \left(\mathbf{u}^{(k+1/5)} - \mathbf{g}\right) = 0, \end{cases} \quad (40)$$

$$\begin{cases} \mathbf{B}_{y/2}\mathbf{u}^{(k+2/5)} \geq \mathbf{C}_{y/2}\mathbf{u}^{(k+1/5)}, & \mathbf{u}^{(k+2/5)} \geq \mathbf{g}, \\ \left(\mathbf{B}_{y/2}\mathbf{u}^{(k+2/5)} - \mathbf{C}_{y/2}\mathbf{u}^{(k+1/5)}\right)^T \left(\mathbf{u}^{(k+2/5)} - \mathbf{g}\right) = 0, \end{cases} \quad (41)$$

$$\begin{cases} \mathbf{B}_{xy}\mathbf{u}^{(k+3/5)} \geq \mathbf{C}_{xy}\mathbf{u}^{(k+2/5)}, & \mathbf{u}^{(k+3/5)} \geq \mathbf{g}, \\ \left(\mathbf{B}_{xy}\mathbf{u}^{(k+3/5)} - \mathbf{C}_{xy}\mathbf{u}^{(k+2/5)}\right)^T \left(\mathbf{u}^{(k+3/5)} - \mathbf{g}\right) = 0, \end{cases} \quad (42)$$

$$\begin{cases} \mathbf{B}_{y/2}\mathbf{u}^{(k+4/5)} \geq \mathbf{C}_{y/2}\mathbf{u}^{(k+3/5)}, & \mathbf{u}^{(k+4/5)} \geq \mathbf{g}, \\ \left(\mathbf{B}_{y/2}\mathbf{u}^{(k+4/5)} - \mathbf{C}_{y/2}\mathbf{u}^{(k+3/5)}\right)^T \left(\mathbf{u}^{(k+4/5)} - \mathbf{g}\right) = 0, \end{cases} \quad (43)$$

$$\begin{cases} \mathbf{B}_{x/2}\mathbf{u}^{(k+1)} \geq \mathbf{C}_{x/2}\mathbf{u}^{(k+4/5)}, & \mathbf{u}^{(k+1)} \geq \mathbf{g}, \\ \left(\mathbf{B}_{x/2}\mathbf{u}^{(k+1)} - \mathbf{C}_{x/2}\mathbf{u}^{(k+4/5)}\right)^T \left(\mathbf{u}^{(k+1)} - \mathbf{g}\right) = 0, \end{cases} \quad (44)$$

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

5.3 Brennan and Schwartz algorithm

Brennan and Schwartz [3] introduced a direct algorithm for solving LCPs with tridiagonal matrices resulting from the pricing of American options under the Black and Scholes model. A similar algorithm was considered in [11] for more general class of free boundary problems and they also described sufficient conditions for the applicability. One of the conditions is that the matrices are M -matrices which is satisfied in our case. They also give a sign pattern condition for the right-hand side vector of the LCP. The verification of this condition seems to be intractable for the LCPs considered here. Instead of this we justify the use of the Brennan and Schwartz algorithm by studying the behavior of the solutions. In Figure 2 we show typical free boundaries between the active and inactive sets defined by

$$\left\{ (i, j) \mid \mathbf{u}_{(i,j)}^{(k)} = \mathbf{g}_{(i,j)} \right\} \quad \text{and} \quad \left\{ (i, j) \mid \mathbf{u}_{(i,j)}^{(k)} \geq \mathbf{g}_{(i,j)} \right\}, \quad (45)$$

respectively, at different times $k\Delta t$ for an American put option. In (45) the subscript (i, j) refers to the element of the vector corresponding to the grid point (i, j) . The LCPs arising from the splitting methods correspond to horizontal, diagonal from bottom-left to top-right, and vertical grid lines. Figure 2 shows that the free boundary divides

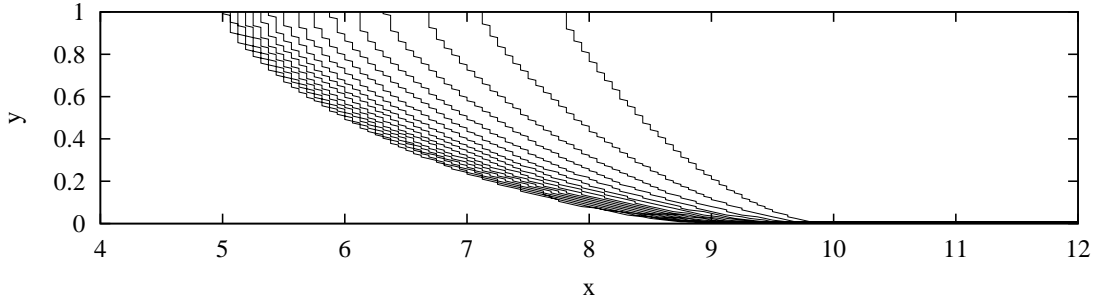


Figure 2: Typical free boundaries between the active and inactive sets at different times for an American put option with the exercise price 10.

each of these grids lines into a continuous active set and a continuous inactive set. The Brennan and Schwartz algorithm is applicable exactly in such cases.

We formulate the Brennan and Schwartz algorithm using the standard \mathbf{LU} decomposition. As an example we consider a LCP with a tridiagonal matrix along, say, a horizontal grid line defined by the grid points (i, j) , $i = 0, \dots, m$, and j given. The LCPs along the diagonal and vertical grid lines can be treated in the same way. The solution is known at the grid point $(0, j)$ due to the Dirichlet boundary condition and, thus, it is sufficient to consider a LCP with a $m \times m$ tridiagonal matrix. When the \mathbf{LU} decomposition is used the unknowns have to be ordered in such a way that the ones corresponding to the inactive set appear first. Thus, for an American put option the order is from right to left.

In order to be more precise we consider the LCP in (37) as our example. For describing the Brennan and Schwartz algorithm we define auxiliary vectors for solution, right-hand side, and payoff as

$$\mathbf{x}_{m-i+1} = \mathbf{u}_{(i,j)}^{(k+1/3)}, \quad \mathbf{b}_{m-i+1} = \left(\mathbf{C}_x \mathbf{u}^{(k)} \right)_{(i,j)}, \quad \text{and} \quad \mathbf{c}_{m-i+1} = \mathbf{g}_{(i,j)}, \quad (46)$$

for $i = 1, \dots, m$, respectively. Furthermore, the tridiagonal matrix \mathbf{T} for the LCP is defined by

$$\mathbf{T}_{m-i+1, m-p+1} = (\mathbf{B}_x)_{(i,j), (p,j)}, \quad (47)$$

for $i = 1, \dots, m$, $p = 1, \dots, m$. In (47) the subscript $(i, j), (p, j)$ refers to the matrix element at the row corresponding to the grid point (i, j) and at the column corresponding to the grid point (p, j) .

The Brennan and Schwartz algorithm can be described as the standard solution procedure using the \mathbf{LU} decomposition with an additional projection for the solution in the backward substitution stage. The matrix \mathbf{L} is lower triangular with only diagonal and one codiagonal having non zero elements. Furthermore, the diagonal elements of \mathbf{L} are chosen to be ones. The matrix \mathbf{U} is upper triangular with only diagonal and one codiagonal having non zero elements. Under these assumptions the condition $\mathbf{T} = \mathbf{LU}$ defines \mathbf{L} and \mathbf{U} uniquely. The following pseudocode gives the Brennan and Schwartz algorithm:

Brennan and Schwartz algorithm with the LU decomposition

The computation of the LU decomposition and forward substitution:

$$\mathbf{U}_{1,1} = \mathbf{T}_{1,1}$$

$$\mathbf{y}_1 = \mathbf{b}_1$$

Do $i = 2, \dots, m$

$$\mathbf{L}_{i,i-1} = \mathbf{T}_{i,i-1} / \mathbf{U}_{i-1,i-1}$$

$$\mathbf{U}_{i-1,i} = \mathbf{T}_{i-1,i}$$

$$\mathbf{U}_{i,i} = \mathbf{T}_{i,i} - \mathbf{L}_{i,i-1} \mathbf{U}_{i-1,i}$$

$$\mathbf{y}_i = \mathbf{b}_i - \mathbf{L}_{i,i-1} \mathbf{y}_{i-1}$$

End Do

Backward substitution with a projection:

$$\mathbf{x}_m = \mathbf{y}_m / \mathbf{U}_{m,m}$$

$$\mathbf{x}_m = \max\{\mathbf{x}_m, \mathbf{c}_m\}$$

Do $i = m - 1, \dots, 1$

$$\mathbf{x}_i = (\mathbf{y}_i - \mathbf{U}_{i,i+1} \mathbf{x}_{i+1}) / \mathbf{U}_{i,i}$$

$$\mathbf{x}_i = \max\{\mathbf{x}_i, \mathbf{c}_i\}$$

End Do

6 M -matrix property and nonuniform grids

The first part of this section shows that the matrices arising from the discretization and splitting methods have the M -matrix property under certain conditions for the grid step sizes. The M -matrix property guarantees that there are no oscillations in numerical solutions due to the spatial discretization. To our experience the lack of this property causes often the numerical solutions computed using the splitting methods to blow up. The M -matrix property is also one part of the sufficient conditions for the applicability of the Brennan and Schwartz algorithm.

In the second part of this section we propose a way to generate nonuniform grids in the x -direction satisfying the conditions for the grid step sizes derived in the first part. Our procedure allows the grids to be refined near the exercise price. This enables to reduce the number of grid points and, thus, computational time without increasing the discretization error in the computed price of the option.

6.1 The M -matrix property of the discretization matrices

Here we derive conditions for the grid step sizes in the x -direction which guarantee that the matrices arising from the discretization and splitting methods are M -matrices. Sufficient conditions for a matrix to be an M -matrix are that it is strictly diagonally dominant with positive diagonal elements and it has non positive off-diagonal elements [36].

We show the M -matrix property for the matrices \mathbf{A}_x , \mathbf{A}_y , and \mathbf{A}_{xy} appearing in the decomposition (31) of \mathbf{A} . Then it follows easily that the matrices \mathbf{B}_x , \mathbf{B}_y , \mathbf{B}_{xy} , $\mathbf{B}_{x/2}$, and $\mathbf{B}_{y/2}$ defined by (33)–(35) are also M -matrices, since they are obtained by

scaling with a positive number and then adding the identity matrix. Furthermore, \mathbf{A} is an M -matrix, since summing maintains diagonal dominance and non positivity of the off-diagonal elements. Thus, \mathbf{B} in (29) has also the M -matrix property by the previous scaling and adding arguments.

In Section 3.1 we did not specify when we use the forward or backward finite differences instead of the central finite differences for the first-order partial derivative terms in (25). Our approach is the following: First we discretize all terms with the second-order derivatives in (25) using the central differences. Under the grid step size limitations given in Sections 6.1.1 and 6.1.2 this leads to a matrix with non positive off-diagonal elements. Then we choose the discretization for the first-order derivatives in such a way that the non positiveness of off-diagonal elements is maintained. The preferred discretization for the term with $\frac{\partial u}{\partial x}$ in (25) is the central finite difference (15), but if it leads to a positive off-diagonal element then the forward or backward finite differences (16) is used. The term with $\frac{\partial u}{\partial y}$ in (25) is treated in the same manner.

In the following (x, y) is a grid point (x_i, y_j) with $i \geq 1$ and $j \geq 1$. As in Section 3.1 we denote the grid step size to left and right by h_l and h_r , respectively. Thus, we have $x_{i-1} = x_i - h_l$ and $x_{i+1} = x_i + h_r$. Furthermore, h is the grid step size to up and down which is independent of j , since the grid is uniform in the y -direction.

6.1.1 Matrix \mathbf{A}_x

First we consider the approximation of the term

$$\left[-\frac{1}{2}yx^2 + w\rho\gamma yx\frac{h_l}{2h} + (1-w)\rho\gamma yx\frac{h_r}{2h} \right] \frac{\partial^2 u}{\partial x^2} \quad (48)$$

from (25) using the finite difference (17). We observe that the off-diagonal elements are non positive when

$$-yx^2 + \rho\gamma yxw\frac{h_l}{h} + \rho\gamma yx(1-w)\frac{h_r}{h} \leq 0. \quad (49)$$

For a general weight $w \in [0, 1]$ the upper bounds

$$h_l \leq \frac{1}{\rho\gamma}xh \quad \text{and} \quad h_r \leq \frac{1}{\rho\gamma}xh \quad (50)$$

for h_l and h_r give sufficient conditions for the inequality (49) to hold. In order to have more freedom to choose the location of the grid line $x = x_1$ we use the weight $w = 0$ on this grid line. This removes the upper bound for h_l and, thus, the upper bound for x_1 .

Under the same upper bounds (50) the matrix given by the discretization of (48) is diagonally dominant with a positive diagonal. By adding the term $ru/3$ to the diagonal makes the matrix strictly diagonal dominant assuming that the interest rate r is positive.

By analyzing the effects of the different possible discretizations for the term with $\frac{\partial u}{\partial x}$ in (25) to the off-diagonal elements we obtain that the following choice guarantees

non positive off-diagonal elements. We use the forward finite difference in (16) when

$$-yx + w\rho\gamma y \frac{h_l + h_r}{h} + rh_r \geq 0 \quad (51)$$

and the backward finite difference in (16) when

$$-yx - (w - 1)\rho\gamma y \frac{h_l + h_r}{h} - rh_r \geq 0. \quad (52)$$

When neither of the conditions (51) and (52) hold we use the central finite difference (15).

6.1.2 Matrix \mathbf{A}_y

The non positiveness of the off-diagonal elements of \mathbf{A}_y can be analysed in the same manner as for \mathbf{A}_x in Section 6.1.1. We obtain the condition

$$-\gamma^2 y + \rho\gamma y x w \frac{h}{h_l} + \rho\gamma y x (1 - w) \frac{h}{h_r} \leq 0 \quad (53)$$

for the off-diagonal elements to be non positive. For a general weight $w \in [0, 1]$ the lower bounds

$$h_l \geq \frac{\rho}{\gamma} x h \quad \text{and} \quad h_r \geq \frac{\rho}{\gamma} x h \quad (54)$$

for h_l and h_r give sufficient conditions for the inequality (53) to hold. The matrix resulting from the discretization of the term with the second-order partial derivative in the y -direction in (25) and the term $ru/3$ with positive r is strictly diagonally dominant under the lower bounds (54).

The following choice between the central finite difference and one-sided differences for the first-partial derivative in the y -direction leads to non positive off-diagonal elements. We use the forward finite difference in (19) when

$$-\gamma^2 y \frac{1}{h} + 2\rho\gamma y x \frac{w}{h_r} + \alpha(\beta - y) \geq 0 \quad (55)$$

and the backward finite difference in (19) when

$$-\gamma^2 y \frac{1}{h} - 2\rho\gamma y x \frac{w}{h_l} + 2\rho\gamma y x \frac{1}{h_r} - \alpha(\beta - y) \geq 0. \quad (56)$$

When neither of the conditions (55) and (56) hold we use the central finite difference (18).

6.1.3 Matrix \mathbf{A}_{xy}

In the diagonal direction the off-diagonal elements

$$-w\rho\gamma y x \frac{1}{h_l h} \quad \text{and} \quad -(1 - w)\rho\gamma y x \frac{1}{h_r h} \quad (57)$$

given by (25) are clearly non positive for any $w \in [0, 1]$. Furthermore, the diagonal element is the sum of the terms in (57) with the minus sign and the term $ru/3$. Thus, the matrix is strictly diagonally dominant with a positive diagonal assuming that r is positive.

6.2 Nonuniform grids

The use of a nonuniform grid is one way to improve the efficiency of the solution algorithm. In some parts of the computational domain the finite difference grid can be sparse while in the other parts more density is required. The advantage of nonuniform grids is that a fewer grid points are needed to obtain the solution with a desired accuracy. Here we use a predetermined grid generating function which refines the grid near the exercise price E . More advanced approach would be to use an error estimate to define the grid step sizes; see [25] and [26], for example.

We use a nonuniform grid in the x -direction and a uniform grid in the y -direction. Moreover constant time steps are employed. When the correlation ρ between the price of the asset and its variance is positive it is generally necessary to use a nonuniform grid in the x -direction in order to satisfy the lower and upper bounds for the grid step sizes in Sections 6.1.1 and 6.1.2. Particularly with larger values of ρ uniform grids lead to matrices without M -matrix property and unstable numerical solutions with the componentwise splitting methods.

6.2.1 Grid generating function

We use a grid generating function $h_r(x)$ to define the step size at the point x . We choose this function to be a parabola

$$h_r(x) = ax^2 + bx + c, \quad x \in [0, X]. \quad (58)$$

The coefficients a , b , and c are chosen in such a way that $h_r(x)$ has desired properties. First we want to have about m_t grid points in the interval $[0, X]$. This condition is approximately satisfied when

$$\int_0^X \frac{1}{h_r(x)} dx = m_t - 1. \quad (59)$$

Furthermore, we prefer that the grid step size has the smallest value at the exercise price E which leads to the condition

$$h'_r(E) = 0. \quad (60)$$

The last requirement is that the ratio between the grid step sizes at X and at E has a given value κ . This is satisfied when

$$\kappa = \frac{h_r(X)}{h_r(E)}. \quad (61)$$

Using the conditions (59)–(61) the coefficients a , b , and c can be determined uniquely. We remark that when the bounds in Section 6.2.2 for the grid step sizes are applied the used grid step sizes do not usually satisfy some of the conditions (59)–(61).

6.2.2 Lower and upper bounds for grid steps

We want the grid step sizes to satisfy the conditions (50) and (54) so that we obtain an M -matrix. Thus, we have the inequalities

$$\frac{\rho}{\gamma}xh \leq h_r(x) \leq \frac{1}{\rho\gamma}xh \quad (62)$$

and

$$\frac{\rho}{\gamma}xh \leq h_l(x) \leq \frac{1}{\rho\gamma}xh, \quad (63)$$

where the grid step size $h_l(x)$ to left is defined by $h_l(x + h_r(x)) = h_r(x)$ and h is the grid step size in the y -direction. Using the definition of $h_l(x)$ and (63) we obtain the inequalities

$$\frac{\rho}{\gamma}(x + h_r(x))h \leq h_r(x) \leq \frac{1}{\rho\gamma}(x + h_r(x))h. \quad (64)$$

By rearranging the terms in (64) we get

$$\frac{\rho}{\gamma}hx \leq \left(1 - \frac{\rho}{\gamma}h\right)h_r(x) \quad \text{and} \quad \left(1 - \frac{1}{\rho\gamma}h\right)h_r(x) \leq \frac{1}{\rho\gamma}hx. \quad (65)$$

The first inequality of (65) can be satisfied only if $1 - (\rho h)/\gamma > 0$, that is, the mesh step size in the y -direction has to be such that $h < \gamma/\rho$. Furthermore, by comparing the inequalities in (62) and (65) we obtain that they are satisfied when the inequalities

$$\left(1 - \frac{\rho}{\gamma}h\right)^{-1} \frac{\rho}{\gamma}hx \leq h_r(x) \leq \frac{1}{\rho\gamma}hx \quad (66)$$

hold.

6.2.3 Grid generation

Here we combine the grid generating function $h_r(x)$ defined in Section 6.2.1 with the lower and upper bounds defined in Section 6.2.2. In order to be well within the bounds we add 10 % margin to them. We use the modified grid generating function

$$\bar{h}_r(x) = \min \left\{ \max \left\{ h_r(x), 1.1 \left(1 - \frac{\rho}{\gamma}h\right)^{-1} \frac{\rho}{\gamma}hx \right\}, 0.9 \frac{1}{\rho\gamma}hx \right\}. \quad (67)$$

The first grid point is $x_0 = 0$ and the second one is chosen to be $x_1 = h/10$, where h is the grid step size in the y -direction. Then the following grid points are given by the formula

$$x_{i+1} = x_i + \bar{h}_r(x_i) \quad (68)$$

from $i = 1$ until $x_{i+1} \geq X$. The last grid point is denoted by x_m .

When the correlation ρ is close to one the bounds for the grid step sizes are tight. In such cases the actual number m of grid points in the x -direction is usually much larger than the desired number m_t of grid points and also a large number of grid points are located near the boundary $x = 0$. Unfortunately this accumulation of grid points cannot be avoided if the matrices are required to have the M -matrix property.

7 Numerical experiments

The following numerical experiments demonstrate the accuracy and efficiency of the proposed componentwise splitting methods for pricing American options under Heston's stochastic volatility model. First we show that the computed prices are in good agreement with the prices computed using the PSOR method and also with the prices reported in the scientific literature. Then the time convergence of the splitting method is studied for the three implicit time discretizations given in Section 3.2. Finally the efficiency of the componentwise splitting methods is demonstrated by comparing the required CPU times for different grid sizes. All numerical experiments were performed on an HP J5600 workstation.

7.1 American option prices

We compare the computed prices of American options using the componentwise splitting methods and the PSOR method. Moreover, the option prices given in three references are presented. For these experiments the parameter values

$$\alpha = 5.0, \quad \beta = 0.16, \quad \gamma = 0.9, \quad r = 0.1, \quad \text{and} \quad E = 10.0, \quad (69)$$

are used with the correlations $\rho = 0.1$ and $\rho = 0.5$. Similarly to [23] the computational domain is chosen to be

$$[0, X] \times [0, Y] \times [0, T] = [0, 20] \times [0, 1] \times [0, 0.25]. \quad (70)$$

The basic componentwise splitting method (37)–(39) with the implicit Euler scheme is referred as CS-IE. The Strang symmetrized componentwise splitting method (40)–(44) with the Crank-Nicolson method is referred as CS-S-CN. The option prices obtained using these two methods are compared with the prices computed using the PSOR method with the implicit Euler scheme (PSOR-IE) and with the Crank-Nicolson method (PSOR-CN).

Table 1 contains the results for the correlation $\rho = 0.1$. These prices are computed using the grid $(m, n, l) = (179, 32, 64)$. As described in Section 6.2, n and l are chosen while the number m depends on the lower and upper bounds for the grid step sizes and on the grid generating function. The parameter values for the grid generating function were $m_t = 161$ and $\kappa = 4$. With these choices the upper bound limits the grid step sizes only very near the boundary $x = 0$.

We have reported results for five asset prices. The prices of options computed using the componentwise splitting methods are in good agreement with the prices computed using the PSOR method with the same time discretization method. This shows that the additional error due to the splitting is not large when compared to the error of the time discretization method. In Table 1 we give also the option prices reported in [4], [23], and [38]. Our prices agree particularly well with the ones reported in [38].

In Table 2 the correlation is increased to $\rho = 0.5$. Again the option prices are reported for the five asset prices. Also in this case the prices computed with the componentwise splitting methods are in good agreement with the prices computed

with the PSOR method. We used the grid $(m, n, l) = (183, 32, 64)$ which was obtained using the parameter values $m_t = 81$ and $\kappa = 4$. The lower and upper bounds for the step sizes are now more tight than with the correlation $\rho = 0.1$. Due to this the bounds often limit the grid step size. Figure 3 shows how the grid step sizes are chosen between the lower and upper bounds. A part of the parabola function is visible in the plot and also the 10 % margins around the bounds can be seen.

7.2 Time convergence rates

Here we study time convergence rates using the same American option pricing problem as in Section 7.1. In order to obtain convergence rates for different methods we computed a reference solution using the grid $(179, 32, 32768)$. The number of time step l is large so that the error due to the time discretization is essentially eliminated. The reference solution was computed using the PSOR method with the Rannacher time stepping. The grid generating function was based on the parameter values $m_t = 161$ and $\kappa = 4$. In Tables 3 and 4 we present the errors and the convergence rates. In this section the error is the absolute value of the difference between the computed price and the reference price at the point $(10, 0.25)$

In Table 3 we report the errors and convergence rates for the basic component-wise splitting method when the implicit Euler scheme (CS-IE) and the Crank-Nicolson method (CS-CN) method are used. The time convergence ratio is calculated by dividing the error computed using l time steps by the error computed using $2l$ time steps. For comparison we report also the results computed using the PSOR method with the implicit Euler scheme. The most accurate solutions were obtained by the component-wise splitting method with the Crank-Nicolson method. The implicit Euler scheme leads to the expected first-order convergence rate. In this experiment the splitting method was slightly more accurate than the PSOR method.

In Table 4 we present the errors and convergence rates for the methods which are expected to have a higher than first-order convergence rate. The Strang symmetrized componentwise splitting method is applied with the Crank-Nicolson method (CS-S-CN) and the Rannacher time stepping (CS-S-R). The errors and convergence rates are also computed using the PSOR method with the Crank-Nicolson method (PSOR-CN) and the Rannacher time stepping (PSOR-R). The Rannacher time stepping is used in order to obtain numerical solutions without undesired oscillations. We observe that the splitting method is more accurate with the Crank-Nicolson method than with the Rannacher time stepping. However the convergence rates are higher with the Rannacher time stepping. With the PSOR method the Rannacher time stepping improves the accuracy for larger time steps. Also the convergence rates for the Rannacher time stepping are stable while with the Crank-Nicolson method they are erratic. The results in Table 4 show clearly that none of the methods lead to a second-order time convergence. The probable reason for this is the reduced regularity of the solution due to the early exercise constraint.

Table 4 demonstrates that the Strang symmetrized componentwise splitting with the Crank-Nicolson method is very well suited for pricing American options. Moreover

method	asset price				
	8.0	9.0	10.0	11.0	12.0
CS-IE	2.07768	1.33032	0.79076	0.44380	0.24048
CS-S-CN	2.07836	1.33286	0.79440	0.44681	0.24201
PSOR-IE	2.07730	1.33072	0.79185	0.44470	0.24076
PSOR-CN	2.07830	1.33279	0.79434	0.44676	0.24198
ref. [4]	2.0733	1.3290	0.7992	0.4536	0.2502
ref. [23]	2.079	1.334	0.796	0.449	0.243
ref. [38]	2.0784	1.3337	0.7961	0.4483	0.2428

Table 1: The option prices for five different asset prices at $y = 0.25$ when $\rho = 0.1$.

method	asset price				
	8.0	9.0	10.0	11.0	12.0
CS-IE	2.10572	1.35635	0.79348	0.41790	0.19932
CS-S-CN	2.10565	1.35754	0.79598	0.42094	0.20161
PSOR-IE	2.10408	1.35506	0.79325	0.41883	0.20052
PSOR-CN	2.10532	1.35726	0.79577	0.42081	0.20153

Table 2: The option prices for five different asset prices at $y = 0.25$ when $\rho = 0.5$.

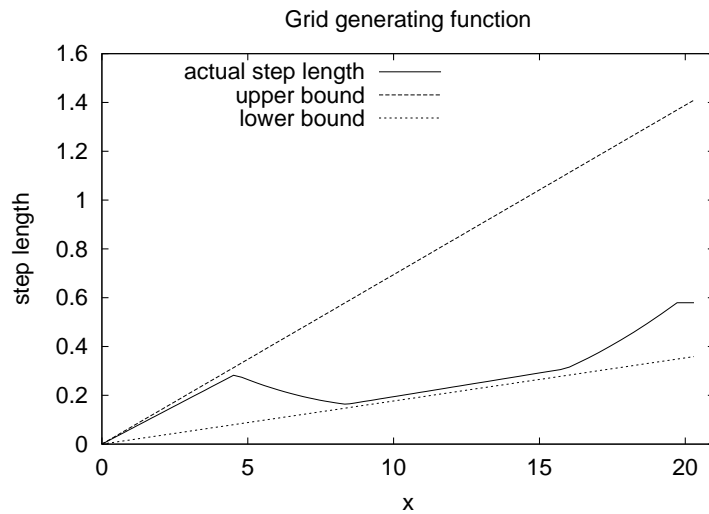


Figure 3: The upper and lower bounds for the grid step sizes and the modified grid generating function $\bar{h}_r(x)$.

l	CS-IE		CS-CN		PSOR-IE	
	error	ratio	error	ratio	error	ratio
4	0.051634		0.031118		0.034710	
8	0.026988	1.91	0.013583	2.29	0.018287	1.90
16	0.013906	1.94	0.006756	2.01	0.009540	1.92
32	0.007110	1.96	0.003383	2.00	0.004937	1.93
64	0.003619	1.96	0.001687	2.01	0.002538	1.94
128	0.001835	1.97	0.000840	2.01	0.001298	1.96
256	0.000928	1.98	0.000418	2.01	0.000661	1.96
512	0.000468	1.98	0.000208	2.01	0.000335	1.97

Table 3: The errors and time convergence rates for the componentwise splitting method and the PSOR method.

l	CS-S-CN		CS-S-R		PSOR-CN		PSOR-R	
	error	ratio	error	ratio	error	ratio	error	ratio
4	0.000413		0.017623		0.058573		0.017031	
8	0.000438	0.94	0.005383	3.27	0.027033	2.17	0.005620	3.03
16	0.000123	3.55	0.001803	2.99	0.009514	2.84	0.001979	2.84
32	0.000040	3.06	0.000635	2.84	0.001224	7.77	0.000727	2.72
64	0.000019	2.14	0.000230	2.76	0.000046	26.67	0.000276	2.63
128	0.000010	1.87	0.000083	2.76	0.000019	2.37	0.000106	2.59
256	0.000006	1.68	0.000030	2.79	0.000008	2.32	0.000041	2.58
512	0.000003	1.75	0.000010	2.89	0.000003	2.57	0.000016	2.60

Table 4: The errors and time convergence rates for the componentwise splitting method and the PSOR method.

the Strang symmetrization proved to be very useful when compared with the results computed using the basic splitting method. This conclusion can be made by comparing the results in Tables 3 and 4.

7.3 CPU time comparison

The componentwise splitting method is efficient because only linear complementarity problems with tridiagonal matrices need to solve at each time step. In the following numerical experiments we compare the efficiency of the Strang symmetrized componentwise splitting method and the PSOR method. We report results for three spatial grids and varying number of time steps. The parameter values (69) and (70) with $\rho = 0.1$ are used. The Rannacher time stepping method is used for all computations.

Again the error is the absolute value of the differences between the computed price and the reference price at the point $(10, 0.25)$. The reference solution was computed using the componentwise splitting method with the Rannacher time stepping on the grid $(m, n, l) = (5960, 1025, 4096)$ generated using the parameter value $m_t = 5120$ and $\kappa = 4$. This lead to the reference price 0.79595573.

(m, n, l)	PSOR-R				CS-S-R	
	error	iter ave	ω	CPU	error	CPU
(90,16,8)	0.010602	92.8	1.7	0.17	0.010472	0.02
(90,16,16)	0.006910	76.3	1.6	0.28	0.006789	0.03
(90,16,32)	0.005675	56.4	1.5	0.42	0.005604	0.05
(90,16,64)	0.005239	41.1	1.4	0.60	0.005203	0.10
(90,16,128)	0.005080	29.5	1.3	0.90	0.005061	0.22
(90,16,256)	0.005020	21.9	1.2	1.35	0.005011	0.40
(179,32,8)	0.007193	234.0	1.8	1.78	0.006955	0.09
(179,32,16)	0.003551	159.8	1.8	2.38	0.003376	0.15
(179,32,32)	0.002300	120.0	1.7	3.41	0.002207	0.29
(179,32,64)	0.001848	86.9	1.6	4.93	0.001802	0.53
(179,32,128)	0.001679	65.7	1.5	7.45	0.001656	1.05
(179,32,256)	0.001613	47.2	1.4	10.70	0.001602	2.12
(359,64,8)	0.006084	428.3	1.9	15.62	0.005839	0.44
(359,64,16)	0.002485	330.0	1.9	23.48	0.002297	0.80
(359,64,32)	0.001237	325.6	1.8	44.46	0.001136	1.52
(359,64,64)	0.000783	202.8	1.8	57.41	0.000733	3.02
(359,64,128)	0.000611	142.1	1.7	78.50	0.000586	5.93
(359,64,256)	0.000544	110.2	1.6	120.38	0.000532	11.80

Table 5: The CPU times for the PSOR method with an optimized relaxation parameter and for the componentwise splitting method.

The errors for the prices computed using the PSOR method are given in the column PSOR-R of Table 5. In order to achieve smallest possible CPU times we optimized the relaxation parameter ω for each grid. The PSOR iterations were terminated when the l_1 -norm of an approximate residual was reduced by the factor of 10^{-7} . Moreover, we report the average number of iterations. The column CS-S-R contains the errors and the required CPU times for the Strang symmetrized componentwise splitting method. We have used the same space discretizations for both methods.

Table 5 shows that the componentwise splitting method is much faster than the PSOR method even with the optimized relaxation parameter. Moreover, the use of the PSOR method requires to choose the relaxation parameter and the stopping criterion while the componentwise splitting method do not require any parameters. The optimization of the relaxation parameter in the PSOR method reduces the required CPU time a lot. This can be seen by comparing the CPU times with the optimized parameter in Table 5 and with two fixed parameter values $\omega = 1.0$ and $\omega = 1.6$ in Table 6.

8 Conclusion

We have proposed a componentwise splitting method for pricing American options under stochastic volatility. The method is based on a linear complementarity formu-

(m, n, l)	PSOR-R				CS-S-R	
	error	iter ave	ω	CPU	error	CPU
(179,32,8)	0.007193	2690.0	1.0	19.05	0.006955	0.09
(179,32,16)	0.003551	1461.9	1.0	19.95	0.003376	0.15
(179,32,32)	0.002300	779.9	1.0	21.36	0.002207	0.29
(179,32,64)	0.001848	411.7	1.0	24.55	0.001802	0.53
(179,32,128)	0.001679	217.3	1.0	23.72	0.001656	1.05
(179,32,256)	0.001613	116.4	1.0	25.30	0.001602	2.12
(179,32,8)	0.007193	665.9	1.6	4.71	0.006955	0.09
(179,32,16)	0.003551	354.1	1.6	5.37	0.003376	0.15
(179,32,32)	0.002300	181.4	1.6	5.30	0.002207	0.29
(179,32,64)	0.001848	86.9	1.6	4.93	0.001802	0.53
(179,32,128)	0.001679	73.8	1.6	8.39	0.001656	1.05
(179,32,256)	0.001613	69.6	1.6	15.75	0.001602	2.12

Table 6: The CPU times for the PSOR method with two fixed values of the relaxation parameter and for the componentwise splitting method.

lation with a two-dimensional parabolic partial differential operator. For the spatial discretization we introduced a special seven point finite difference stencil on nonuniform grids. With a suitable choice of grid step sizes together with a selective use of one-sided finite differences we showed that the discretization leads to an M -matrix.

At each time step the basic form of the proposed componentwise splitting method decomposes the discrete problem into three linear complementarity problems with tridiagonal matrices. We increased the accuracy of this splitting by performing the Strang symmetrization. In the symmetrized form five linear complementarity problems with tridiagonal matrices need to be solve at each time step. By solving these problems using the Brennan and Schwartz algorithm we obtain a very efficient method.

In the numerical experiments the option prices computed using the componentwise splitting method were in good agreement with the prices presented in the literature. Our experiments showed that the accuracy of the componentwise splitting method is similar to the PSOR method. Furthermore, the splitting method was between three and 30 times faster than the PSOR method with an optimized relaxation parameter.

The implementation of the componentwise splitting method using the Brennan and Schwartz algorithm is only applicable for pricing American call and put options. By replacing the Brennan and Schwartz algorithm with a generic solver for linear complementarity problems with tridiagonal matrices the same componentwise splitting method can be applied also for option pricing problems with other payoff functions. For example, the solver in [7] can be used in the general case.

Acknowledgments

The authors thank Prof. Tommi Kärkkäinen for many suggestions improving the paper.

References

- [1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Political Economy*, 81:637–654, 1973.
- [2] Achi Brandt and Colin W. Cryer. Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems. *SIAM J. Sci. Statist. Comput.*, 4(4):655–684, 1983.
- [3] M. J. Brennan and E. S. Schwartz. The valuation of American put options. *J. Finance*, 32:449–462, 1977.
- [4] N. Clarke and K. Parrott. The multigrid solution of two-factor American put options. Technical Report 96-16, Oxford Computing Laboratory, Oxford, 1996.
- [5] N. Clarke and K. Parrott. Multigrid for American option pricing with stochastic volatility. *Appl. Math. Finance*, 6:177–195, 1999.
- [6] L. Clewlow and C. Strickland. *Implementing derivatives models*. John Wiley & Sons Ltd., Chichester, 1999.
- [7] C. W. Cryer. The efficient solution of linear complementarity problems for tridiagonal Minkowski matrices. *ACM Trans. Math. Software*, 9(2):199–214, 1983.
- [8] Colin W. Cryer. The solution of a quadratic programming problem using systematic overrelaxation. *SIAM J. Control*, 9:385–392, 1971.
- [9] Jim Douglas, Jr. and H. H. Rachford, Jr. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.
- [10] D. J. Duffy. *Financial instrument pricing using C++*. John Wiley & Sons Ltd., Chichester, 2004.
- [11] C. M. Elliott and J. R. Ockendon. *Weak and variational methods for moving boundary problems*, volume 59 of *Research Notes in Mathematics*. Pitman, Boston, MA, 1982.
- [12] P. A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing American options using a penalty method. *SIAM J. Sci. Comput.*, 23(6):2095–2122, 2002.
- [13] Jean-Pierre Fouque, George Papanicolaou, and K. Ronnie Sircar. *Derivatives in financial markets with stochastic volatility*. Cambridge University Press, Cambridge, 2000.
- [14] Roland Glowinski. Finite element methods for incompressible viscous flow. In *Handbook of numerical analysis, Vol. IX*. North-Holland, Amsterdam, 2003.

- [15] E. Hairer and G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996.
- [16] S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review Financial Stud.*, 6:327–343, 1993.
- [17] J. C. Hull. *Options, futures, and other derivatives*. Prentice Hall, Upper Saddle River, NJ, third edition, 1997.
- [18] S. Ikonen and J. Toivanen. Operator splitting methods for pricing American options with stochastic volatility. Technical Report B11/2004, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland, 2004. Submitted.
- [19] G. I. Marchuk. Splitting and alternating direction methods. In *Handbook of numerical analysis, Vol. I*, pages 197–462. North-Holland, Amsterdam, 1990.
- [20] S. McKee and A. R. Mitchell. Alternating direction methods for parabolic equations in two space dimensions with a mixed derivative. *Comput. J.*, 13:81–86, 1970.
- [21] S. McKee, D. P. Wall, and S. K. Wilson. An alternating direction implicit scheme for parabolic equations with mixed derivative and convective terms. *J. Comput. Phys.*, 126(1):64–76, 1996.
- [22] A. R. Mitchell and D. F. Griffiths. *The finite difference method in partial differential equations*. John Wiley & Sons Ltd., Chichester, 1980.
- [23] C. W. Oosterlee. On multigrid for linear complementarity problems with application to American-style options. *Electron. Trans. Numer. Anal.*, 15:165–185, 2003.
- [24] D. W. Peaceman and H. H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.
- [25] J. Persson and L. von Sydow. Pricing European options using a space-time adaptive FD-method. *Comput. Vis. Sci.*, 2005. To appear.
- [26] O. Pironneau and F. Hecht. Mesh adaption for the Black and Scholes equations. *East-West J. Numer. Math.*, 8(1):619–643, 2000.
- [27] D. M. Pooley, K. Vetzal, and P. A. Forsyth. Remedies for non-smooth payoffs in option pricing. *J. Comput. Finance*, 6:25–40, 2003.
- [28] Rolf Rannacher. Finite element solution of diffusion problems with irregular data. *Numer. Math.*, 43(2):309–327, 1984.
- [29] Christoph Reisinger and Gabriel Wittum. On multigrid for anisotropic equations and variational inequalities: pricing multi-dimensional European and American options. *Comput. Vis. Sci.*, 7(3-4), 2004.

- [30] Rüdiger Seydel. *Tools for computational finance*. Universitext. Springer-Verlag, Berlin, 2002.
- [31] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.
- [32] Domingo Tavella and Curt Randall. *Pricing financial instrument*. John Wiley & Sons, Chichester, 2000.
- [33] Stephane Villeneuve and Antonino Zanette. Parabolic ADI methods for pricing American options on two stocks. *Math. Oper. Res.*, 27(1):121–149, 2002.
- [34] Paul Wilmott. *Derivatives*. John Wiley & Sons Ltd., Chichester, 1998.
- [35] Paul Wilmott, Sam Howison, and Jeff Dewynne. *The mathematics of financial derivatives*. Cambridge University Press, Cambridge, 1995.
- [36] Günther Windisch. *M-matrices in numerical analysis*, volume 115 of *Teubner Texts in Mathematics*. BSB B. G. Teubner Verlagsgesellschaft, Leipzig, 1989.
- [37] N. N. Yanenko. *The method of fractional steps. The solution of problems of mathematical physics in several variables*. Springer-Verlag, New York, 1971.
- [38] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Penalty methods for American options with stochastic volatility. *J. Comput. Appl. Math.*, 91(2):199–218, 1998.
- [39] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Negative coefficients in two-factor option pricing models. *J. Comput. Finance*, 7(1):37–73, 2003.