

Pricing American Options Using a Space-time Adaptive Finite Difference Method

Jonas Persson *

Abstract

American options are priced numerically using a space- and time-adaptive finite difference method. The generalized Black-Scholes operator is discretized on a Cartesian structured but non-equidistant grid in space. The space- and time-discretizations are adjusted such that a predefined tolerance level on the local discretization error is met. An operator splitting technique is used to separately handle the early exercise constraint and the solution of linear systems of equations from the finite difference discretization of the linear complementarity problem. In numerical experiments three variants of the adaptive time-stepping algorithm with and without local time-stepping are compared.

Keywords: Finite difference method, adaptive method, American option, stochastic volatility, local time-stepping

AMS subject classification: 65M06, 65M20, 65M50

1 Introduction

The Black-Scholes equation introduced 1973 by F. Black and M. Scholes in [2] and extended the same year by R.C. Merton in [19] is an often solved equation to determine the arbitrage free price of an option. Two common types of options are the European option that only can be exercised at the time of maturity T and the American option that can be exercised at any time $t \leq T$. This extra freedom in the American option will be reflected in the price. For American options this possibility of using the option before T is often referred to as the early exercise constraint. Because of this feature the problem of pricing American options will have an internal free boundary. The mathematical properties of the American option will be explained in more detail in Sec. 2.1.

European options can in some cases be priced using analytic formulas. If that is not possible, numerical methods like the Monte Carlo method [3] or partial

*Supported by FMB, the Swedish Graduate School in Mathematics and Computing Science.

differential equation (PDE) based methods like finite differences or finite elements, see [1, 28], can be used. American options require a somewhat different approach due to the early exercise feature in the problem and more involved numerical methods must be used. The problem in a PDE approach can be formulated as a linear complementarity problem (LCP) with inequalities involving the Black-Scholes PDE and some extra constraints. This will be described in more detail in Sec. 2.1. Below we will give some examples of the work of others that have used this formulation of the American option pricing problem and discuss examples of different ways of solving the LCP.

Penalty methods have been used by several authors, e.g. Zvan, Forsyth and Vetzal. In [29] they introduce a penalty parameter in the discretized equations that enforces the early exercise constraint. Nielsen, Skavhaug and Tveito takes a similar approach in [21] and [22] and introduce a penalty term in the continuous equations. Another technique is front-fixing methods, see e.g. Pantazopoulos, Houstis and Kortesis [24]. They use an expanding grid that keeps track of the free boundary by solving a nonlinear equation at each time-step. They admit that their method is difficult to use for more than one underlying asset and this means that e.g. multi-asset American options and American options with stochastic volatility cannot easily be priced using their method. One approach used by many researchers is to discretize with e.g. finite differences and reduce the problem to a sequence of LCPs (one for each time-step). One of the most popular methods for the solution of the LCPs in this category is to use projected successive overrelaxation, also known as PSOR. This approach is described by e.g. Tavella and Randall in [28] and Seydel in [26] and a refined approach is described by Koulisianis and Papatheodorou in [15]. Using Monte Carlo simulations for pricing the American option is a viable but fairly complicated method compared to pricing the European option. This is so because one must solve an optimal stopping problem to find the optimal exercise rule and then compute the expected discounted pay-off using this rule, see e.g. Glasserman [8]. For many underlying assets the Monte Carlo approach is still the only possible way to go but in other cases PDE methods are competitive.

Here we have adopted an operator splitting technique to solve the LCP. This technique has been used in other numerical fields and was implemented for financial problems by Ikonen and Toivanen in several articles and reports (see e.g. [12, 13]). This method is built upon a finite difference discretization (though it could be used also for other discretizations) and separates the solution of the linear systems of equations from the enforcing of the early exercise constraint. This means that the over all computational time needed is very low since no nonlinear systems of equations need to be solved. In the more recent paper [14] Ikonen and Toivanen introduce a component-wise operator splitting technique where special care is taken to the discretization such that only tri-diagonal LCPs have to be solved. For these problems they use the Brennan-Schwartz algorithm [4] together with a Strang symmetrization [27] to enhance the accuracy.

In this paper we will price the American option with constant and stochastic volatility and use the adaptive techniques developed in [16, 17, 25]. Three variants of an adaptive time-stepping scheme with the backward differentiation formula of order 2 (BDF2) are considered, the standard scheme, a scheme modified at the free boundary and one with local time-stepping at the free boundary. The three schemes will be denoted by x1a, x1b and x2 respectively where x should be replaced by either C (for constant volatility) or S (for stochastic volatility) depending on what case we study, see Tab. 1.

	Constant vol.	Stochastic vol.
Adaptive time-stepping	C1a	S1a
Adaptive time-stepping modified at the free boundary	C1b	S1b
Local time-stepping near the free boundary	C2	S2

Table 1: The different time-stepping methods used.

We also study the accuracy of the BDF2 approximation of the time-derivative near the free boundary. The case of stochastic volatility for the American option combined with adaptivity is also considered for the three schemes mentioned above. For stochastic volatility we have used Heston's model, see [11]. In that setting the volatility is modeled by a stochastic process instead of being constant or level dependent which is often the case in standard literature. From this a generalized Black-Scholes PDE can be derived.

Examples of others that have studied numerical pricing of American options under stochastic volatility include Ikonen and Toivanen in a series of papers e.g. [12, 13, 14], Zvan, Forsyth and Vetzal [29], Clarke and Parrott [7] and Oosterlee in [23]. The paper by Clarke and Parrott is interesting since they use an adaptive time-stepping algorithm with a technique different from the one used here. They find an approximation of the discretization error in time by comparing a first order approximation of the derivative to a fourth order approximation and use that information to adjust the time-steps.

The paper is organized as follows: In Sec. 2 we discuss the underlying theory of pricing American options and in Sec. 3 we talk about the discretization of the problem and the differences in the numerical solution of the constant and stochastic volatility problem. Sec. 4 introduces the operator splitting technique and Sec. 5 gives an overview of the space and time adaptivity used. In Sec. 6 we study the approximation of derivatives in the discretized problem near the free boundary. Sec. 7 then show some numerical results using our methods and Sec. 8 concludes the paper.

2 Pricing model for the American option

2.1 Constant volatility

The original problem is stated as a final value problem with a terminal condition at time T . As in most literature we have here used a transformation of the time-scale, $t = T - \tilde{t}$ (where \tilde{t} is the original time), so that the problem becomes a forward problem with a known initial condition. This makes standard texts on time-integration for an ordinary differential equation (ODE) applicable. Let us first give the transformed Black-Scholes equation for the arbitrage free price $F(x, t)$ of the European option,

$$\begin{aligned} \frac{\partial F}{\partial t} - rx \frac{\partial F}{\partial x} - \sigma^2 x^2 \frac{\partial^2 F}{\partial x^2} + rF &= 0, \\ F(x, 0) &= \Phi(x). \end{aligned} \tag{1}$$

Here r is the short rate of interest, σ is the volatility and $\Phi(x)$ is the contract function for the underlying stock x . To price an American option we need Eq. (1) and (2) but since the American option has the feature that it can be exercised at any time $t \leq T$ we also need some extra conditions. The value of the American option is e.g. always higher than the intrinsic value $\Phi(x)$ since it can be exercised at any time. The extra conditions can be collected in the following system of partial differential inequalities (see e.g. [28]),

$$\begin{aligned} \left(\frac{\partial F}{\partial t} - \mathcal{L}F \right) &\geq 0, \\ (F - \Phi) \left(\frac{\partial F}{\partial t} - \mathcal{L}F \right) &= 0, \\ F &\geq \Phi \end{aligned} \tag{3}$$

with the initial condition $F(x, 0) = \Phi(x)$ and the operator \mathcal{L} defined as

$$\mathcal{L} = rx \frac{\partial}{\partial x} + \sigma^2 x^2 \frac{\partial^2}{\partial x^2} - r. \tag{4}$$

The equations (3) is known as the LCP formulation or just the LCP for the American option. For simplicity we have here used the simplified notation $F(x, t) = F$ and $\Phi(x) = \Phi$. The American put option has the contract function

$$\Phi(x) = \max(K - x, 0),$$

where K is the strike-price.

2.2 Stochastic volatility

We have also studied the numerical pricing of American options under stochastic volatility. The model problem we have used is the American put option using Heston's model [11].

The stochastic volatility model by Heston is given by (the notation used here is the same as in [13])

$$dx_t = \mu x_t dt + \sqrt{y_t} x_t dW_1, \quad (5)$$

$$dy_t = \alpha(\beta - y_t)dt + \gamma\sqrt{y_t}dW_2, \quad (6)$$

where x_t is the stochastic stock process and y_t is a variance process. Furthermore, μ is the positive drift of the stock, γ is the volatility of the variance process and $\alpha, \beta > 0$ are parameters for the mean-reversion of the variance process. Note that the variance will tend to revert back to the value β at a rate determined by α . The Brownian motions W_1 and W_2 are correlated with a factor $\rho \in [-1, 1]$. From these stochastic differential equations a two-dimensional differential operator can be derived for the American put option with stochastic volatility, see e.g. [29] for the derivation. As most others we state it as a forward problem instead of the original backward problem, the same technique was used in Sec. 2.1.

$$\frac{\partial F}{\partial t} = rx \frac{\partial F}{\partial x} + \frac{1}{2}yx^2 \frac{\partial^2 F}{\partial x^2} + \rho\gamma xy \frac{\partial^2 F}{\partial x \partial y} + \{\alpha(\beta - y) - \theta\gamma\sqrt{y}\} \frac{\partial F}{\partial y} - rF := LF. \quad (7)$$

The parameter θ is the so called market price of risk. As Ikonen and Toivanen and others we will here and further on assume that $\theta = 0$. The problem is formulated and valid in the domain $(x, y, t) \in \{x|x \geq 0\} \times \{y|y \geq 0\} \times [0, T]$, a domain that must be truncated before numerical computations can be performed. The truncation and boundary conditions at all boundaries will be discussed in Sec. 3.

We can now give the LCP formulation of the American option pricing problem with stochastic volatility,

$$\begin{aligned} \left(\frac{\partial F}{\partial t} - LF \right) &\geq 0, \\ (F - \Phi) \left(\frac{\partial F}{\partial t} - LF \right) &= 0, \\ F &\geq \Phi, \end{aligned} \quad (8)$$

with very small differences compared to the formulation in Sec. 2.1. The only difference is the operator and that the contract function now is $\Phi = \Phi(x, y) = \max(K - x, 0)$. This LCP is reformulated in Sec. 4 using an auxiliary variable and solved using operator splitting which is a technique for separately handling the early exercise constraint and the systems of equations from the linear complementarity problem.

3 Discretization

For both the constant and the stochastic volatility cases a discretization of the operator in space and time must be performed. We have used the same discretization technique as was used in [25] and will not repeat any details here. All space derivatives are discretized using a centered second order finite difference (FD) approximation on a structured but not equidistant grid in each space dimension. The variable step-sizes will be needed in the space adaptive process described in Sec. 5. The time derivative is approximated using the second order backward differentiation formula (BDF2) [10] with variable time-steps.

Since the domain is unbounded it is necessary to use truncation for the numerical computations and we must impose additional boundary conditions. These will be described for the two cases of constant and stochastic volatility separately in the following.

The discretizations in space lead to the semi-discrete problem

$$\frac{\partial F_h}{\partial t} = AF_h + g \quad (9)$$

where A is the FD matrix of the spatial discretization of the operators \mathcal{L} and L respectively and F_h is a vector of unknowns for the spatial grid. Computations are only performed in the inner of the computational domain and boundary conditions are incorporated in the matrix A when possible (e.g. with Neumann conditions) and end up in g when it is not possible (Dirichlet conditions). Note that we do not perform any computations on the boundaries, only at the inner grid-points.

3.1 Constant volatility

The unbounded domain has in the constant volatility case been truncated at $x = X_{max}$ so the computational domain is $(x, t) \in [0, X_{max}] \times [0, T]$. The boundary condition

$$\frac{\partial^2 F}{\partial x^2} = 0,$$

has been imposed at $x = 0$ and $x = X_{max}$ boundaries assuming that the solution is close to linear there. This is a linear extrapolation at the boundary. At $x = X_{max}$ this is a good approximation as long as X_{max} is large enough, see [28], and at $x = 0$ the boundary condition is also a good approximation. Besides, the value at the boundary $x = 0$ will in any case be changed to the value of Φ because of the early exercise constraint.

At the free boundary the regularity of the solution is lower than in the rest of the domain, see Sec. 6. The first derivative in space and time is continuous over the free boundary but the second derivatives are not. This could reduce the second order accuracy to only first order accuracy there. In Sec. 6 we will study

the approximation of the time derivative near the free boundary. Ikonen and Toivanen report in their paper [13] that the second order accurate time-stepping methods used do not seem to keep the second order when refining the grid. For this reason we have compared three different but similar adaptive time-stepping schemes C1a, C1b and C2. See Tab. 1 in Sec. 1. The method C1b is identical to C1a with the exception that we remove the estimate of the local discretization error in a few grid-points in space close to the free boundary in each time-step. These estimates are often bad for a number of reasons, one being that the solution is not smooth over the free boundary (this will be discussed more further on) which causes problems in the estimation. When local time-stepping is used we use a one-step first order accurate time-stepping method, Euler backward, in a small region near the free-boundary. It will not be as sensitive to the discontinuity in the second derivative as the second order method since the leading order term in the error is of one order lower.

The small region is chosen as a few grid-points away from the free-boundary on each side. The location of the free-boundary is found by using the auxiliary variable λ introduced by the operator splitting. This will be discussed in Sec. 4 where the operator splitting technique is applied to the problem. In the left subfigure of Fig. 1 the bold lines show where the fine grid is used. The right subfigure show how the time-step from t_n to t_{n+1} is divided into several smaller steps where the lower order method is used. The idea here is to keep the error at a certain level without using the small steps all over the computational domain.

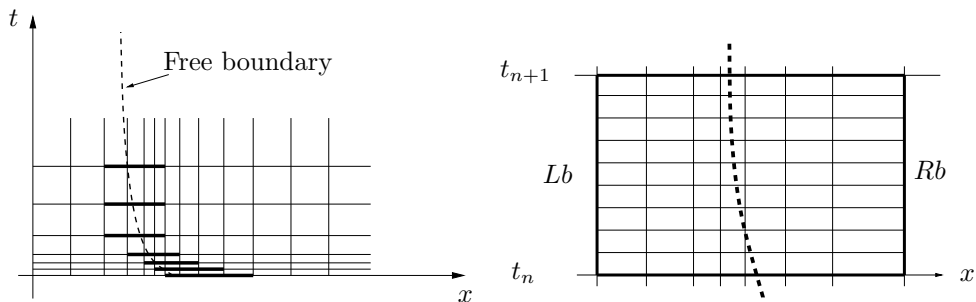


Figure 1: LEFT: Bold lines indicate where local time-steps are used in C2. RIGHT: A fine grid applied between two time-steps. Free boundary indicated by dashed line.

The algorithm to use these smaller time-steps is:

- After one step using BDF2, check where the free boundary is located.
- Decide where the fine grid in time is needed and generate a small FD matrix for only these points.
- The relevant values at the points needed along each vertical boundary, denoted Lb and Rb in Fig. 1, are interpolated from the corner points using

linear interpolation. These are used as Dirichlet boundary conditions for the local time-stepping.

As initial condition we use the values from time-level t_n and when we reach time-level t_{n+1} we update the values calculated with BDF2. How many steps that are needed on the fine time-grid is decided using an adaptive algorithm presented in Sec. 5.

3.2 Stochastic volatility

In the stochastic volatility case the unbounded domain of the problem has been truncated at $x = X_{max}$ and $y = Y_{max}$. The computational domain is thus defined as $(x, y, t) \in [0, X_{max}] \times [0, Y_{max}] \times [0, T]$. The boundary conditions used in the stochastic volatility case are

$$\frac{\partial^2 F(0, y, t)}{\partial x^2} = 0, \quad (y, t) \in [0, Y_{max}] \times [0, T], \quad (10)$$

$$F(x, 0, t) = \max(K - x, 0), \quad (x, t) \in [0, X_{max}] \times [0, T] \quad (11)$$

$$\frac{\partial F(X_{max}, y, t)}{\partial x} = 0, \quad (y, t) \in [0, Y_{max}] \times [0, T], \quad (12)$$

$$\frac{\partial F(x, Y_{max}, t)}{\partial y} = 0, \quad (x, t) \in [0, X_{max}] \times [0, T], \quad (13)$$

and are the same as was used in e.g. [7, 13, 29] except for the condition at $x = 0$ and $y \in [0, Y_{max}]$. They have $F(0, y, t) = K$ at this boundary but our condition is a good approximation and produces the same result because of the update step in the operator splitting method.

As with constant volatility we have compared three variants of the adaptive time-stepping algorithm, S1a, S1b and S2, as described for constant volatility. When the variant S2 of the adaptive scheme, with local time-stepping, is used the idea from Sec. 3.1 could not easily be used since it would be problematic to perform computations locally only close to the free boundary. Such a domain would be irregular and instead we choose a computationally less complicated region by using a smaller time-step in a region of space totally surrounding the free boundary for all (x, y, t) . This area was found using λ (the auxiliary variable introduced in Sec. 4) from the first solve. In this region we have used the same technique as in the constant volatility case for the small interval in space, i.e., solutions at two time-levels are used to interpolate values at Lb and Rb for all time-steps t . Linear interpolation has been used between the time-levels. At the boundaries $y = 0$ and $y = Y_{max}$ the same boundary conditions as for the rest of the problem was used. Another approach is to surround the free boundary with a box that expands over time. However, we found that this did not improve efficiency to any noticeable extent. Note that as $y \rightarrow 0$ the dissipative term in Eq. (7) disappears. This have the effect that approximating derivatives in the x -direction is difficult close to the strike-price for small volatilities when using

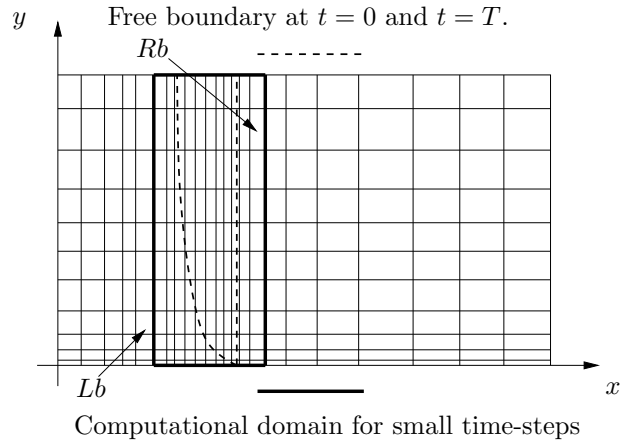


Figure 2: Location of the free boundary at $t = 0$ and $t = T$ plus the surrounding box where smaller time-steps are used.

centered difference approximations. This problem was only noted for very fine discretizations. For realistic settings giving three to four “correct” digits there were no problems.

4 Operator splitting

Here we explain in short the operator splitting technique used by Ikonen and Toivanen in [12, 13]. Assume that we have the LCP problem formulated as in (3). The operator splitting is based on a reformulation of the problem using an auxiliary variable λ that keeps track of the free boundary. The variable λ can be used to find the location of the free boundary during or after the computations. We have used λ to find where local time-stepping should be used.

Omitting boundary conditions we get

$$\left(\frac{\partial}{\partial t} - \mathcal{M}\right) F = \lambda, \quad (14)$$

$$(F - \Phi) \lambda = 0, \quad (15)$$

$$F \geq \Phi, \quad (16)$$

$$\lambda \geq 0. \quad (17)$$

Note that here F and Φ will depend on (x, t) in the constant volatility case and on (x, y, t) using stochastic volatility. Furthermore, \mathcal{M} is the operators \mathcal{L} and L respectively, see Eqs. (4) and (7).

Ikonen and Toivanen introduce operator splittings for Euler backward, BDF2,

Crank-Nicolson and a Runge-Kutta time-stepping scheme with constant time-steps. Here we will use only the BDF2 scheme but allow for changing time-steps. This is because we will combine the splitting technique with the adaptive time-stepping algorithm we have previously used in [17] and [25].

The time-stepping scheme BDF2 with variable time-steps for a discrete vector u is

$$\alpha_0^{n+1}u^{n+1} = k_{n+1}Au^{n+1} - \alpha_1^{n+1}u^n - \alpha_2^{n+1}u^{n-1} \quad (18)$$

$$\alpha_0^{n+1} = (1 + 2\theta^{n+1})/(1 + \theta^{n+1}), \quad (19)$$

$$\alpha_1^{n+1} = -(1 + \theta^{n+1}), \quad (20)$$

$$\alpha_2^{n+1} = (\theta^{n+1})^2/(1 + \theta^{n+1}). \quad (21)$$

Here $\theta^{n+1} = k_{n+1}/k_n$ where k_{n+1} and k_n are the time-steps used. The proposed operator splitting for the BDF2 scheme with non-equidistant time-steps using equation (14) to (17) and (18) to (21) is then given by

$$\alpha_0^{n+1}\tilde{u}^{n+1} = k_{n+1}A\tilde{u}^{n+1} - \alpha_1^{n+1}u^n - \alpha_2^{n+1}u^{n-1} + k_{n+1}\lambda^n \quad (22)$$

$$u^{n+1} - \tilde{u}^{n+1} - \frac{k_{n+1}}{\alpha_0^{n+1}}(\lambda^{n+1} - \lambda^n) = 0, \quad (23)$$

$$(\lambda^{n+1})^T(u^{n+1} - \Phi) = 0, \quad u^{n+1} \geq \Phi, \quad \lambda^{n+1} \geq 0. \quad (24)$$

where A is considered to be the space discretization of the continuous operator \mathcal{L} or L respectively. After the intermediate step (22) has been calculated Eqs. (23) and (24) are used during the update step to make sure that the solution at time $n + 1$ fulfills the constraints. This can be done pointwise which makes this method efficient compared to more standard solution methods of the LCP.

5 Space and time adaptivity

The space adaptivity used in this paper is directly based on the method used by the author in [17] and [25]. The ideas will be repeated here but for the details and formulas the reader is referred to the other papers. The time adaptivity is also similar to the techniques used in the above mentioned papers but adjusted and changed to better handle the problems arising from the discontinuities near the internal free-boundary, see Sec. 3. The main idea is to first solve the problem on a very coarse mesh in space and time and estimate the discretization error in space and keep track of the free boundary using the auxiliary variable λ . The variable λ is positive on the continuation region and zero on the stopping region so it can easily be used to find the free boundary. From the estimates of the discretization errors we can create new grids in space that will keep the local discretization errors below a pre-defined level ε during the second solve. During the second solve we use an adaptive time-stepping method. In [25] we create an adaptive time discretization during the first solve but here we do not do that since we want to be able to closely follow the free boundary. The first solve would only give a rough estimate of the location of the free boundary.

5.1 Space adaptivity

The space adaptivity is based on a technique similar to Richardson extrapolation. In [26] Seydel proposes the use of two grids with step-lengths h and $2h$ to estimate the local discretization error. He then uses it to find a suitable constant step-length. Here we use a similar technique that assumes that the error in the discretization is dominated by its leading term. We estimate the local discretization error τ_{h_i} (for dimension i and with step-sizes h_i) in space on a coarse grid using two different step-sizes. From that estimate of the discretization error a new grid is computed using an assumption that the discretization error has the form $\tau_{h_i} = h_i^2 \eta(x) + \mathcal{O}(\text{higher order terms})$ for our second order discretization. Given a discretization and the estimated discretization error using this discretization it is possible to construct a new grid on which the estimated discretization errors are below a certain tolerance level ε . This works in both one and several space dimensions. In more than one dimension, though, the theoretical derivations are simplified by the assumption that the coefficients and thus the discretization error from the mixed derivative is small and can be neglected. This adaptive process is carried out in each dimension separately to create a tensor product mesh of structured but non-equidistant space discretizations. That this technique works has been proven in the papers [16, 17, 25] by the author. A difference compared to the technique used in the references is that for constant volatility (one-dimensional problem) we estimate the discretization error using one-sided differences close to the free boundary. For stochastic volatility this would be more difficult, especially for the mixed derivative. In this case we instead use the normal centered approximations and replace the values of τ near the free boundary with values of τ a few grid-points away.

5.2 Time adaptivity

The time adaptive method is built on the one used in [17] which follows the work by Söderberg et al. in [18]. The local discretization error in time, here denoted τ_k , with the BDF2 method is estimated at each time-step by comparing the solution using BDF2 and the solution from an explicit solver. Knowledge about the leading terms in each method makes it possible to estimate the discretization error in BDF2 as

$$\tau_k = c_1(F_{\text{BDF2}} - F_{\text{Expl.}})/(kc_2),$$

where c_i , $i = 1, 2$ are known parameters and F_{name} are the numerical solutions from each method. After a solution has been computed and τ_k has been calculated the solution is discarded and recomputed with a smaller k if τ_k was too large. If τ_k is small enough the solution is accepted and k is adjusted for the next step. In this way the time-steps are dynamically adjusted during the computations. All details about this methodology and how to accept/reject the time-steps can be found in [17]. Here we have the additional problem of losing regularity near the free boundary. That has been handled differently for the two cases of constant and stochastic volatility and is described in Sec. 3.

The time-adaptivity on the fine grid with the local time-stepping schemes has been performed almost as with the BDF2 method and will only be described here in short. Recall that a small part of the spatial domain is selected and a local time-stepping algorithm is used there. To calculate the number of small time-steps needed we do as follows. On this small domain we take one step of length k using the one-step methods Euler backward (EB) and Euler forward (EF). Let us here denote the solutions by F_{EB} and F_{EF} . The two solutions are then used to approximate the local discretization error in the EB method,

$$\tau_{EB} = -\frac{1}{2k}(F_{EB} - F_{EF}).$$

The estimated discretization error is then compared to the tolerance level ε for the time discretization,

$$\eta = |\varepsilon/\tau_{EB}|,$$

and the new small local time-step is computed as

$$k_{fine} = 0.9\eta k.$$

The number of local time-steps with EB on each time-steps k is then $N_{fine} = k/k_{fine}$ and adjusted to be an integer. This methodology follows the one used in [25] very closely but has here been adapted to the first order methods and the local time-stepping procedure.

6 Approximation of the time derivative near the free boundary

The behavior of the solution of the American put option has been studied by many researchers over the years. It is of great interest because the location and behavior of the free boundary, here denoted $b^*(t)$, is important for traders. It turns out that it is optimal to exercise the American option when the stock price hits the free boundary. See e.g. [20] for analytical results on the free boundary. In this section we will denote partial derivatives by subscripts for ease of notation.

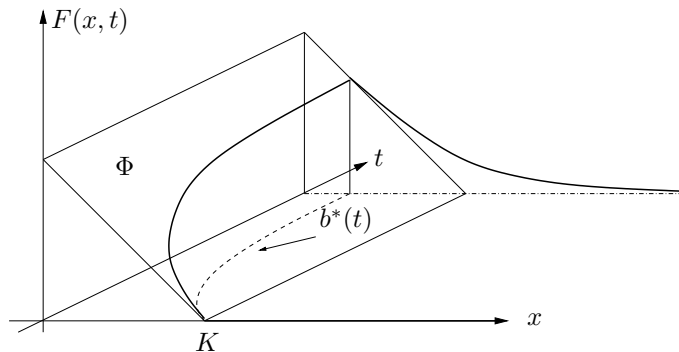


Figure 3: The free boundary in the x - t -plane.

Fig. 3 shows the free boundary as the projection of the curve where the value of the American option meets the contract function into the x - t -plane.

An analysis of the order of accuracy near the free boundary is motivated by the result showed in Fig. 4. In the figure we plot the quotient, point-wise in space, between errors where a large number of grid-points in space have been used and a small number of time-steps with step-lengths k and $k/2$. The quotient should be 4 for second order convergence but the figure shows that it is only 2 in parts of the domain indicating first order convergence there. To find the error we have compared our numerical solutions with a fine reference solution. The same analysis for the space accuracy show that we have second order accuracy in space. This does not seem possible considering that the second derivative in space is not continuous over the free boundary, which will soon show. For this we have no explanation. The investigation of the free boundary is also motivated by observations by Ikonen and Toivanen in [13] that state that the second order time-stepping schemes and the operator-splitting methods based upon them cannot maintain a second order convergence as the number of time-steps is increased. They also say that the cause of this could be that the solution is not regular enough at the free boundary.

At the stopping region, i.e. at the left of the free boundary, we know that $F = K - x$. This means that $F_t = 0$ and $F_x = -1$ in this region. In the following analysis we will use this, what is known about the value on the free boundary and the equation to study the continuity over the free boundary. Derivatives taken on the solution F should here be considered as derivatives (on the free boundary where indicated) from the continuation region and up to and including the free boundary. If the derivative to the left (on the stopping region) and the derivative from the right coincide we can conclude continuity of the derivative.

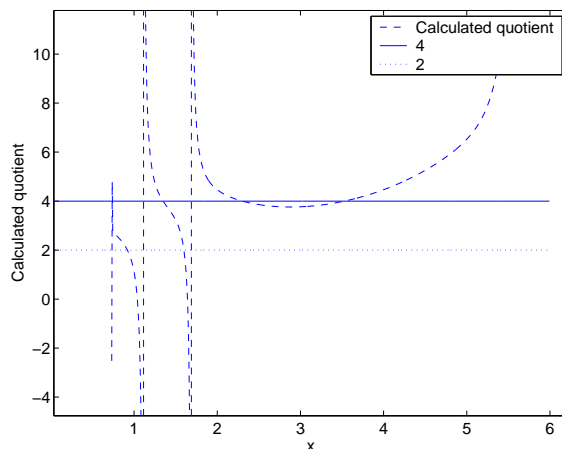


Figure 4: Estimate of order of accuracy in time as a function of the transformed space coordinate. Transformed strike-price is at $x = 1$.

Here we will use a result concerning the continuity of the first derivative in space of the American put price over the free boundary, the “smooth fit” condition. It states (see e.g. [20]) that

$$F_x(b^*(t), t) = -1. \quad (25)$$

Since $F_x = -1$ in the stopping region the first space derivative is continuous over the free boundary. In the following analysis we will also need the condition

$$F_t(b^*(t), t) = 0, \quad (26)$$

at the free boundary. This result can be derived by some simple calculations. The following holds at the free boundary

$$F(b^*(t), t) = K - b^*(t), \quad (27)$$

and differentiation with respect to time gives

$$F_t(b^*(t), t) + b_t^* F_x(b^*(t), t) = -b_t^*, \quad (28)$$

and we can conclude that since $F_x(b^*(t), t) = -1$ on $b^*(t)$ we are only left with $F_t(b^*(t), t) = 0$ at the free boundary. Since $F_t = 0$ holds at the stopping region we conclude that the time derivative is continuous. Here we must assume that the free boundary as a function of time is at least C^1 and that the volatility is constant. For constant volatility it is actually C^∞ [5]. Furthermore, taking another derivative with respect to t in Eq. (28) and use the results from above we find that

$$F_{tt}(b^*(t), t) = (b_t^*)^2 F_{xx}(b^*(t), t). \quad (29)$$

Since F_{xx} is zero at the stopping region we need to study the value at the continuation region. By the simple calculation of solving for the second derivative with respect to x in (1) and using (25), (26) and (27) we find that

$$F_{xx}(b^*(t), t) = \frac{rK}{\sigma^2(b^*(t))^2} \quad (30)$$

From Eq. (29) and (30) we can conclude that the second derivative in time will not be continuous over the free boundary since b_t^* is not zero. We also note that the second derivative in space is not continuous either over the free boundary.

Since the error in the approximation of the time derivative include second order derivatives that are not continuous near the free boundary approximating the time derivative using BDF2 will be problematic there. The method is multi-leveled and uses values from two old time-steps, which give us two possible cases when the stencil ranges across the boundary. These will be studied below. In the following of this section we will denote the discrete values of F by u and indicate time-steps by superscripts n . The t -value at the free boundary, between two grid-points in time, will be denoted by t^α which is the point in time where, for a fixed x , the BDF2 stencil crosses the free boundary. See Fig. 5 and 6 below.

In the two problematic cases that can occur, either u^{n-1} is in the stopping region or both u^{n-1} and u^n are. These two cases will be studied here. Note that \tilde{u}^{n-1} is the value used in the numerical computations and u^{n-1} is the value on the imagined smooth continuation into the stopping region. In Fig. 5 we see the first case with only one unknown in the stopping region.

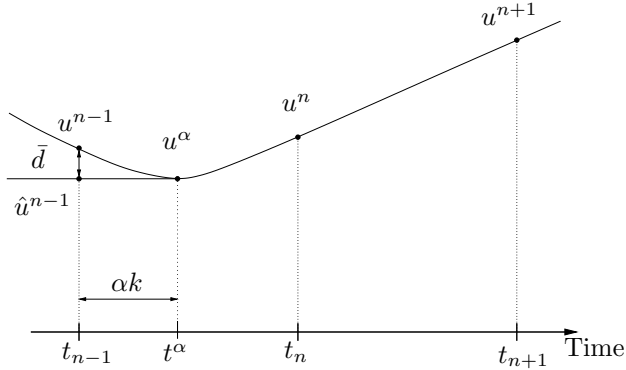


Figure 5: Approximation of the time derivative near the free boundary, with the grid-point \hat{u}^{n-1} on the stopping region.

Study now the BDF2 approximation of the time derivative using the value \tilde{u} in

the grid-point at t_{n-1} .

$$F_t \approx \frac{1}{k} \left[\frac{3}{2} u^{n+1} - 2u^n + \frac{1}{2} \hat{u}^{n-1} \right] = \frac{1}{k} \left[\frac{3}{2} u^{n+1} - 2u^n + \frac{1}{2} u^{n-1} \right] - \frac{\bar{d}}{2k}. \quad (31)$$

Here \bar{d} is defined as the difference between the value u^{n-1} and \hat{u}^{n-1} ,

$$\bar{d} = u^{n-1} - \hat{u}^{n-1}.$$

Inserting a continuous function with $u^{n-1} = u(t_{n-1})$ and then making a Taylor expansion around t^α shows that \bar{d} is (note that $\hat{u}^{n-1} = u^\alpha$)

$$\bar{d} = u^\alpha - \alpha k u_t^\alpha + \frac{1}{2} \alpha^2 k^2 u_{tt}^\alpha - \mathcal{O}(k^3) - u^\alpha = -\frac{1}{2} \alpha^2 k^2 u_{tt}^\alpha - \mathcal{O}(k^3)$$

where the last equality holds since $u_t^\alpha = 0$ according to (26). Inserting this in (31) we see that we get the known error term that is of order $\mathcal{O}(k^2)$ for BDF2 and the extra term

$$-\bar{d}/2k = \frac{1}{4} k \alpha^2 u_{tt}^\alpha.$$

Hence we see that we lose the second order accuracy due to the extra term that is of order $\mathcal{O}(k)$.

The next case is when both u^{n-1} and u^n in the time direction lie on the stopping region. See Fig. 6.

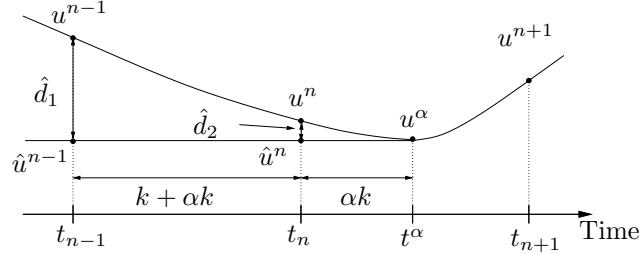


Figure 6: Approximation of the time derivative near the free boundary, with two grid-points on the stopping region.

This means that we will approximate the time derivative with BDF2 with the discretization

$$F_t \approx \frac{1}{k} \left[\frac{3}{2} u^{n+1} - 2\hat{u}^n + \frac{1}{2} \hat{u}^{n-1} \right]. \quad (32)$$

Note that

$$\hat{u}^{n-1} = u^{n-1} - \bar{d}_1 \quad \text{and that} \quad \hat{u}^n = u^n - \bar{d}_2$$

so that the approximation (32) can be written

$$\begin{aligned} \frac{1}{k} \left[\frac{3}{2} u^{n+1} - 2 \underbrace{(\hat{u}^n - u^n + u^n)}_{=-\bar{d}_2} + \frac{1}{2} \underbrace{(\hat{u}^{n-1} - u^{n-1} + u^{n-1})}_{=-\bar{d}_1} \right] = \\ = \frac{1}{k} \left[\frac{3}{2} u^{n+1} - 2u^n + \frac{1}{2} u^{n-1} \right] - \frac{\bar{d}_1}{2k} + \frac{2\bar{d}_2}{k}. \end{aligned} \quad (33)$$

To get an idea of how large errors the extra terms introduce we again insert the continuous function u at the points t_n and t_{n-1} and make Taylor expansions of the expressions for \bar{d}_1 and \bar{d}_2 around t^α . Here we use that $|t_{n-1} - t^\alpha| = k(1 + \alpha)$ and that $\hat{u}^{n-1} = \hat{u}^n = u^\alpha$.

$$\bar{d}_1 = u^{n-1} - \hat{u}^{n-1} = [\text{Taylor}] = \frac{k^2(1 + \alpha)^2}{2} u_{tt}^\alpha + \mathcal{O}(k^3) \quad (34)$$

$$\bar{d}_2 = u^n - \hat{u}^n = [\text{Taylor}] = \frac{k^2\alpha^2}{2} u_{tt}^\alpha + \mathcal{O}(k^3). \quad (35)$$

Eq. (34) and (35) hold since $u_t^\alpha = 0$ according to (26). From equation (33) we see that the extra term will be of order $\mathcal{O}(k)$ since

$$-\frac{\bar{d}_1}{2k} + \frac{2\bar{d}_2}{k} = k(\alpha^2 - \frac{1}{4}(1 + \alpha)^2) u_{tt}^\alpha.$$

As in the first case this means that the overall $\mathcal{O}(k^2)$ error from BDF2 is reduced to $\mathcal{O}(k)$ in the grid-points near the free boundary if the BDF2 method is used.

These results motivate the use of a one-step method in a region near the free boundary, as described in Sec. 3. Any method of higher order will be reduced to a first order method in time near the free boundary.

7 Numerical experiments

Here we will show some results from our experiments with the space and time adaptive methods presented in Sec. 5 for pricing the American put option. The tests have been divided into the two cases constant and stochastic volatility. For both these cases we have studied the adaptive time-stepping algorithm with and without local time-stepping. We also consider a slightly modified method without local time-stepping. These methods are denoted C1a, C1b and C2, for constant volatility and S1a, S1b and S2 for stochastic volatility, as was described earlier. For constant volatility we have used the operator splitting technique proposed by Ikonen and Toivanen in [12] to solve the LCPs and for stochastic volatility we have used the operator splitting technique used in [13]. These two techniques are basically the same. For the two different volatility cases respectively the adaptive space discretization is very similar to what was used in our previous work [25, 17, 16] but with modifications at the free boundary, these

are described in Sec. 5.1. Since the space adaptivity used is similar to the one used in e.g. [25] we will not show any results regarding space adaptivity here. From our experiments with the method we see that we get the discretization errors we require except at the free boundary (which is what can be expected with this method).

Note that we require a certain tolerance ε on the local discretization errors in these experiments but we sometimes present the errors. The size of these errors can be of a different magnitude than the local discretization errors, and this is not strange. The tolerance ε will put restrictions on the discretization error but only indirectly on the error.

7.1 Constant volatility - Results

All the experiments in this section have been performed with the transformed PDE (1) with an additional scaling of the space coordinate by K . All methods have been implemented in Matlab. For the experiments with constant volatility we have used the same experimental setup that was used by Seydel in [26]. Since no analytical formula is available our results can be compared to results by Seydel to confirm the correctness of the method. For most computations though we used a fine reference solution computed with our method using very small time and space steps. The parameters used by Seydel was: Time to maturity $T = 0.4166$ (corresponding to about 5 months), interest rate $r = 0.1$, strike-price $K = 50$ and volatility $\sigma = 0.4$. At $x = 50$ Seydel gives the value $F = 4.284$. We have truncated our computational domain at $X_{max} = 300$, i.e. at $6K$, which should be sufficient for our far-field boundary condition to be a good approximation.

Since the BDF2 time-stepping method is implicit in time and the operator splitting method separates the LCP into an intermediate time-step and an update step we have to solve a linear system of equations for each time-step before updating. Since BDF2 needs two old time-steps, the implicit one-step method backward Euler has been used for the first time-step. For the linear systems of equations we use the preconditioned restarted **GMRES(6)** [9] as iterative solver, where 6 denotes the number of iterations before restart. As preconditioner we use an incomplete LU factorization (ILU) that is not recomputed as the time-steps are changing. The same technique was used successfully in [25]. The number of iterations needed for convergence by the **GMRES** method will increase as the time-steps increase but not in a way that motivates recomputing the ILU factors continuously. The small systems on the fine grid have been solved using Matlab's `\`-operator.

As a test we have used our adaptive method with tolerance levels $\varepsilon \in [10^{-4}, 10^{-2}]$ for the space and time discretization errors respectively. The maximum absolute errors have been measured and plotted in Fig. 7 vs the computational time that was needed to solve the problem with the three time-stepping algorithms

considered. Recall that the method with adaptive time-stepping and no local time-stepping (LT) is denoted by C1a and the method with a small modification is denoted by C1b. The scheme with LT is denoted C2. We see that if C1a is used the computational times are longer for approximately the same error. Using C1b or C2 shortens the computational times. Since the same space discretization was used for all schemes this gives an indication that it is possible to improve the standard adaptive time-stepping scheme C1a using any of the other two approaches.

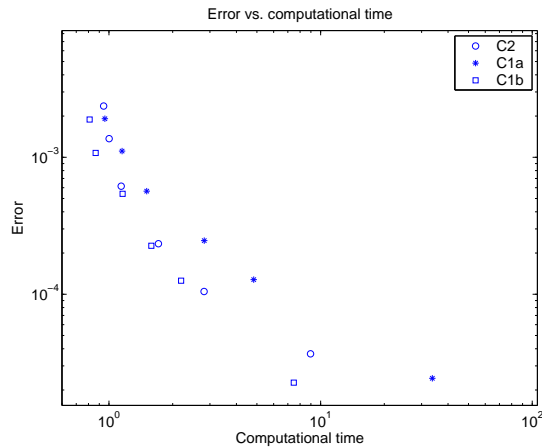


Figure 7: Error with C1a, C1b and C2 plotted vs. computational time using the same space discretization.

Fig. 8 shows the time-steps that were selected in an experiment with the tolerance level $\varepsilon = 10^{-3}$ using all three schemes C1a, C1b and C2. From the results we see that using C1b we get a time-step selection similar to the one using C2 with LT. However, omitting some estimates in τ_k as we do in C1b, will of course not let us control the local discretization errors as in C2.

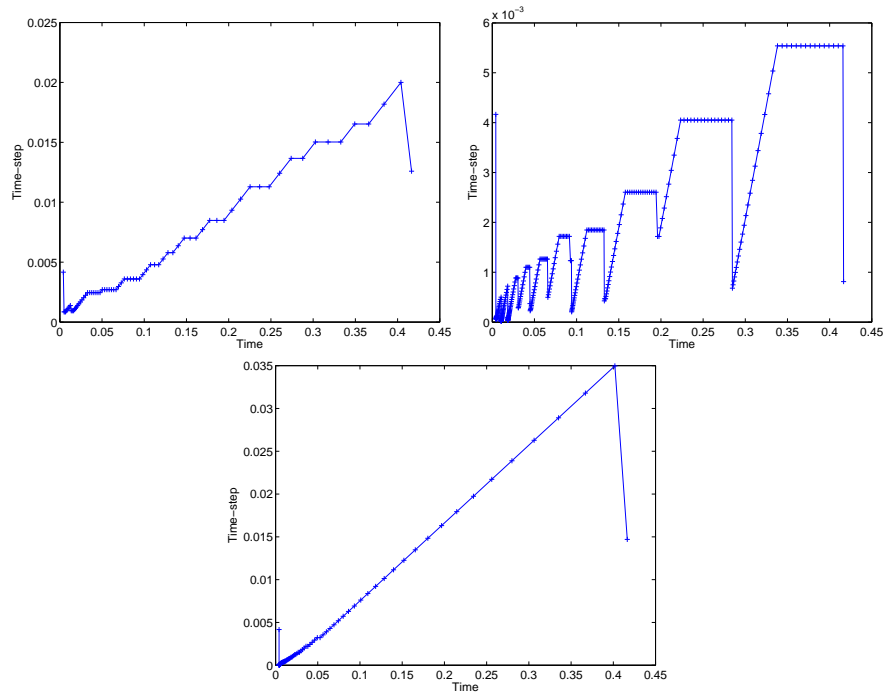


Figure 8: Time-steps used with the three methods. LEFT: C2, RIGHT: C1a and BELOW: C1b.

Estimating the discretization errors in time on the part where LT is used and on the part where it is not used with the method C2 we can see (Fig. 9 (LEFT)) that the criterion 10^{-3} used in this example is clearly met. In both domains the discretization error is below ε for (almost) all time-steps. We also see that the local discretization error in time using C1a will almost meet the desired criterion. When a large τ_k is encountered the time-step is adjusted down as seen in Fig. 8 (LEFT). The figure (RIGHT) also shows that using C1b, τ_k will be too large and the ε criterion is not met. However, this is what can be expected since some of the estimated τ_k is disregarded when choosing time-steps. The resulting error does not seem to increase which indicates that the large τ_k measured are only bad estimates close to the free boundary rather than abrupt changes in the solution, see Fig. 7. Remember that the estimates close to the free boundary using C2 are computed using the backward and forward Euler which does not seem to be so sensitive close to the free boundary. The measured τ_k are of the expected size.

The resulting space-time adaptive grid in these tests is depicted in Fig. 10. We see that the space adaptive method concentrates all grid-points around the

free boundary and only very few points are used far away from it. To the right in Fig. 10 we show a close-up of the region including the free boundary.

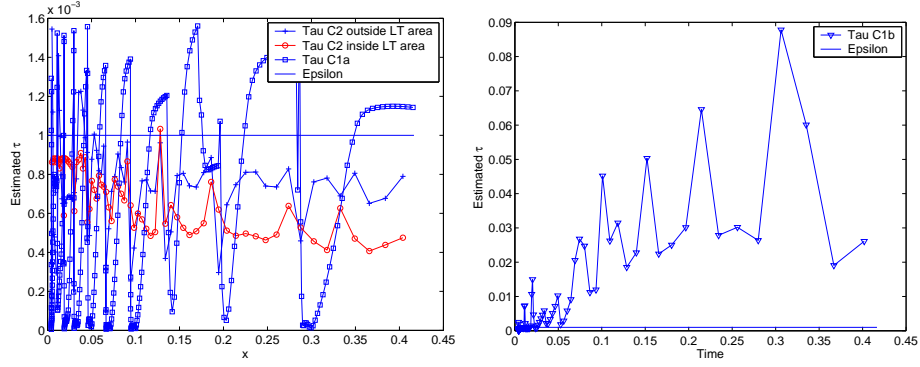


Figure 9: LEFT: The estimates of the discretization error τ_k using C2 in the areas with and without LT. Estimated discretization error with C1a. RIGHT: Estimated discretization error with C1b. For both figures $\varepsilon = 10^{-3}$ was used.

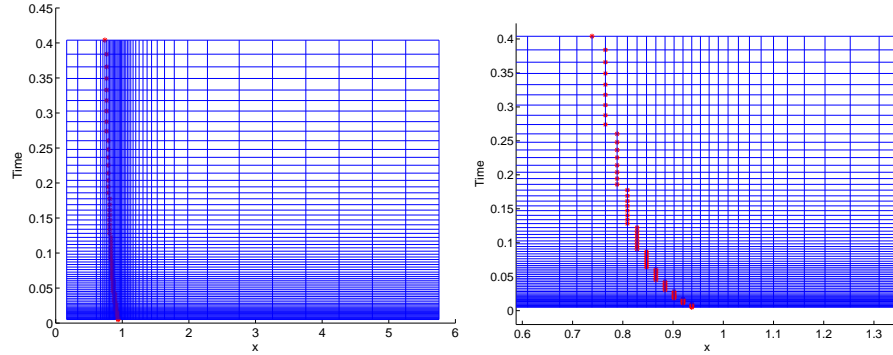


Figure 10: LEFT: Space and time adaptive grid. RIGHT: A close-up of the area close to the strike price. Free boundary indicated by stars '*'.

7.2 Stochastic volatility

The experiments with stochastic volatility have been performed using Eq. (7) and was implemented in Matlab. The experimental setup was the same as was used in e.g. [13] and [23]. The following parameter values were used: time to maturity $T = 0.25$ years, strike price $K = 10$, interest rate $r = 0.1$. From the stochastic differential equation we have the correlation parameter $\rho = 0.1$ and the variance process parameters $\alpha = 5.0$, $\beta = 0.6$ and $\gamma = 0.9$. As the references

we have truncated the computational domain at $X_{max} = 20$ and $Y_{max} = 1$. We will compare our prices with the values presented in [6, 13, 23, 29] for the same parameters.

The space adaptivity described in Sec. 5.1 has been combined with three versions of the time-adaptivity described in Sec. 5.2. The adaptive algorithm in time without LT has been tested both with and without a small modification that omits the estimate of the local discretization error at the free boundary. These two schemes are denoted by S1a and S1b. The scheme with a first order LT around the free boundary is denoted by S2.

Tab. 2 and Tab. 3 show respectively the computed option prices at five different stock values $x = 8.0, 9.0, 10.0, 11.0, 12.0$ and volatility $y = 0.0625$ and $y = 0.25$ with the three variations of the time-stepping algorithm. Since the grid-points in space are not placed exactly at these x - and y -values we have used interpolation to find them. This explains why we in Tab. 2 do not get exactly 2 as the references for $(x, y) = (8.0, 0.0625)$. In these tables the numbers $[m, n, l]$ denote the number of grid-points, adaptively placed, used to discretize x , y and t . OS and PSOR denotes the values given by operator splitting and projected successive over-relaxation in [13]. We see that even with very few grid-points we get values with good correspondence to the reference values.

	[m,n,l]	8.0	9.0	10.0	11.0	12.0
S1a	[21,9,62]	1.99615	1.09804	0.50694	0.20597	0.08188
	[81,21,1975]	1.99976	1.10794	0.51938	0.21495	0.08219
S1b	[21,9,35]	1.99628	1.09831	0.50702	0.20596	0.08180
	[81,21,329]	1.99977	1.10853	0.51945	0.21495	0.08217
S2	[21,9,14]	1.99618	1.09728	0.504978	0.20513	0.08234
	[81,21,121]	1.99976	1.10768	0.51873	0.21424	0.08193
ref. [13] ^a	[320,128,64]	2	1.10751	0.51904	0.21294	0.08181
ref. [13] ^b	[320,128,64]	2	1.10738	0.51902	0.21296	0.08183
ref. [6]		2	1.1080	0.5316	0.2261	0.0907
ref. [23]		2	1.107	0.517	0.212	0.0815
ref. [29]		2	1.1076	0.5202	0.2138	0.0821

Table 2: Option values for the three time-stepping variants S1a, S1b and S2 compared with references at variance $y = 0.0625$.

^a(OS)

^b(PSOR)

	[m,n,l]	8.0	9.0	10.0	11.0	12.0
S1a	[21,9,62]	2.07070	1.32784	0.79154	0.44675	0.24362
	[81,21,1975]	2.07798	1.33298	0.79533	0.44784	0.24259
S1b	[21,9,35]	2.07112	1.32809	0.79181	0.44691	0.24367
	[81,21,329]	2.07839	1.33325	0.79550	0.44794	0.24262
S2	[21,9,14]	2.06969	1.32540	0.78821	0.44378	0.24147
	[81,21,121]	2.07770	1.33219	0.79377	0.44621	0.24170
ref. [13] ^a	[320,128,64]	2.07846	1.33360	0.79585	0.44813	0.24271
ref. [13] ^b	[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. [23]		2.079	1.334	0.796	0.449	0.243
ref. [29]		2.0784	1.3337	0.7961	0.4483	0.2428

Table 3: Option values for the three time-stepping variants S1a, S1b and S2 compared with references at variance $y = 0.25$.

^a(OS)

^b(PSOR)

Note that the number of time-steps displayed in Tab. 2 and Tab. 3 for the method S2 do not include the local time-steps. For the two examples given the method S2 used 97 and 1362 local steps respectively. These are only used on part of the domain but can, especially for small values of ε , be quite many. For these tests we have limited the number of local time-steps to 10 for each “normal” time-step. Often 10 local steps are too few in the beginning, when the problems coming from the first time-step (remember that the initial solution is not smooth) is not yet damped out, but suffice at the end, at least for larger ε . An automatic way of choosing the maximum number of local steps would be useful but we have only used a predefined constant in these experiments. In Fig. 11 we see how the adaptive time-stepping methods S1a and S1b without LT and how the method S2 with LT chooses the time-steps. Here $\varepsilon = 0.01$ was used. Recall that the difference between S1a and S1b is that some troublesome points near the free boundary have been removed in S1b when estimating the local discretization error for the time-step control. This gives it a behavior very similar to S2. Remember that S2 also uses smaller steps in parts of the domain.

Figure 12 show an example of the estimated τ_k using the schemes S1a, S1b and S2. S1a uses all grid-points in the estimate of τ_k and will adjust the time-step down when it encounters a large τ_k , as was mentioned earlier. Since S1b omits the time discretization errors estimated close to the free boundary it will give estimated τ_k larger than ε there. This could however just be a effect of how they are estimated. For S2 we show both the local discretization errors estimated where local time-stepping has and has not been used, this is referred to in the figure as inside and outside the LT area. The discretization errors estimated outside the LT area are smaller than can be expected, but that is because the method can not increase the step-size fast enough due to a limitation

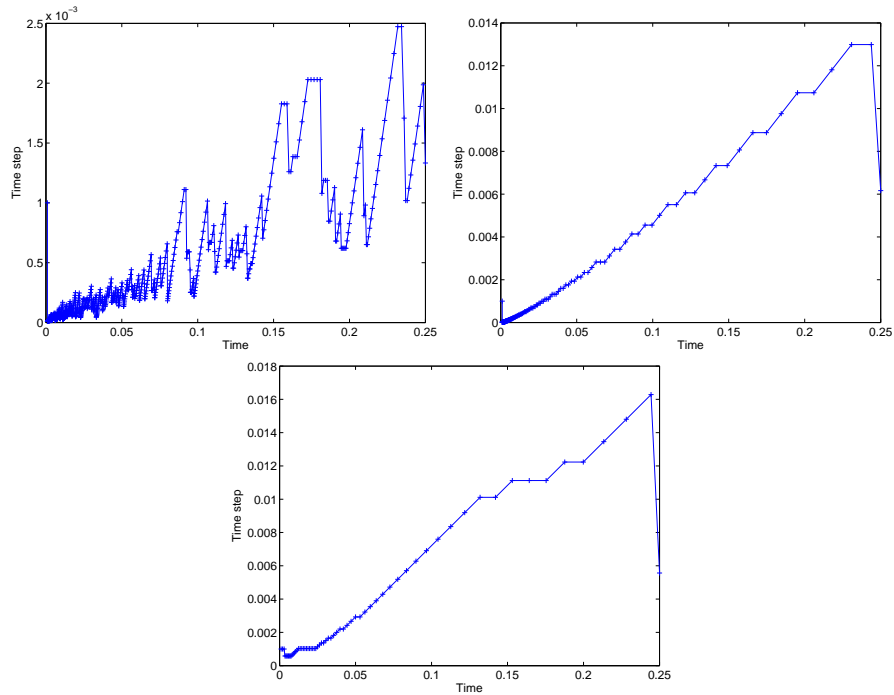


Figure 11: Time-steps used with the three methods. LEFT: S1a, RIGHT: S1b, BELOW: S2.

in the step-size increase. We force the methods to not increase the step-size more than 10% of the previous step. As before the maximum number of local time-steps was set to 10, which was not sufficient for the first steps. But after a few steps the value of τ_k goes down to the specified ε also inside the LT area. In this example $\varepsilon = 0.05$ was used. A numerical reference solution was computed using the adaptive method without local time-steps and a small ε in time and space. This solution was then compared with our numerical solution with S1a, S1b and S2 in the x and y -values as above for two different ε . The results are shown in Tab. 4 and 5. We see that S1a and S1b show similar errors but S1b is faster. S2 that uses a local time-stepping algorithm shows larger errors and longer computation times. The reason for the larger errors is that it uses a first order accurate time-stepping algorithm in parts of the domain where we could get second order convergence with BDF2 and that we in this example have restricted the number of steps that are allowed in that area. Allowing for more local time-steps though would lead to even longer computation times.

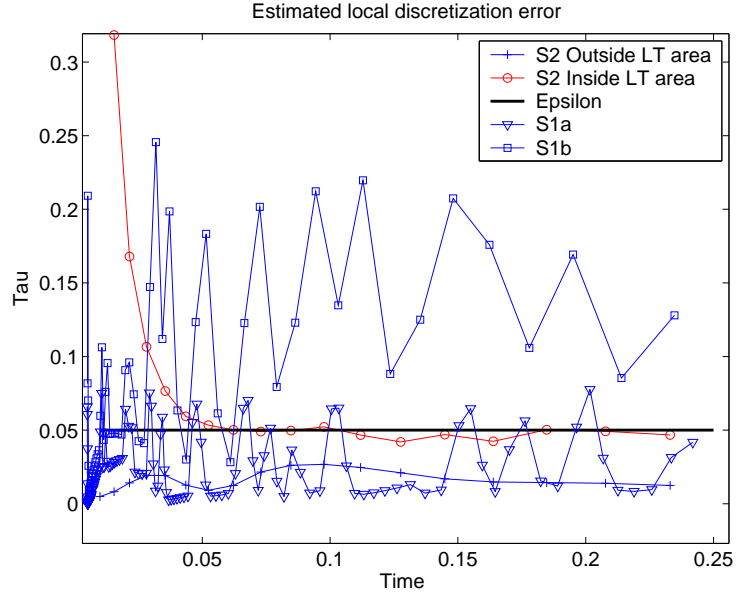


Figure 12: Absolute maximum of the estimated τ_k in the regions with and without local time-stepping, $\varepsilon = 0.05$ in both space and time.

$y = 0.0625$						
	$\#k$	$x=8.0$	9.0	10.0	11.0	12.0
S1a	128	0.00070	0.00691	0.01193	0.00710	0.00186
S1b	60	0.00065	0.00632	0.01166	0.00712	0.00199
S2	20*	0.00069	0.00739	0.01319	0.00754	0.00121
$y = 0.25$						
S1a	128	0.00338	0.00481	0.00589	0.00318	0.00064
S1b	60	0.00247	0.00402	0.00531	0.00285	0.00053
S2	20*	0.00392	0.00635	0.00808	0.00544	0.00307

Table 4: Errors with S1a, S1b and S2. All methods use [61 17] adaptively placed grid-points in space. S1a uses 3.26s, S1b 2.88s and S2 uses 3.71s. Here space and time $\varepsilon = 0.05$. '*' - 165 local time-steps was used.

$y = 0.0625$						
	$\#k$	$x=8.0$	9.0	10.0	11.0	12.0
S1a	1975	0.00024	-0.00034	0.00059	-0.00131	-0.00016
S1b	329	0.00023	-0.00094	0.00053	-0.00131	-0.00013
S2	121*	0.00024	-0.000082	0.00124	-0.00060	0.00010
$y = 0.25$						
S1a	1975	0.00038	0.00063	0.00062	0.00041	0.00021
S1b	329	-0.000036	0.00037	0.00046	0.00032	0.00017
S2	121*	0.00066	0.00142	0.00218	0.00204	0.00109

Table 5: Errors with S1a, S1b and S2. All methods use [81 21] adaptively placed grid-points in space. S1a uses 58.8s, S1b 20.4s and S2 uses 80.1s. Here space and time $\varepsilon = 0.005$. '*' - 1329 local time-steps was used.

8 Conclusions

For constant volatility we study continuity of derivatives of the solution over the free boundary. We conclude that the second derivatives are not continuous. Studying the numerical approximation of the time derivative near the free boundary, when BDF2 is used, we find that the order of accuracy is reduced from second to first order which motivates the use of a first order method there. With constant volatility we see that the time-stepping algorithm C1b gives similar time-steps and performance as C2 but without the control of the local discretization error C2 has. With stochastic volatility S1a and S1b give similar errors but S1b is faster due to smoother time-step selection. S2 is slower than both S1a and S1b and produces larger errors. However, it seems to control the local discretization errors better. The reason for the larger errors is probably due to the restriction in the number of steps with local time-stepping and the fact that local time-stepping will be used in a small area away from the free boundary with the technique used. The most efficient method for both constant and stochastic volatility is the method where estimated discretization errors in time are disregarded close to the free boundary in the time-stepping selection. The results indicate that the adaptive time-stepping method used have problems estimating the discretization error close to the free boundary, this explains why omitting them there gives such a good result.

References

- [1] Y. Achdou and O. Pironneau. *Computational methods for option pricing*. SIAM, 2005. In the series Frontiers in applied mathematics.
- [2] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

- [3] P.P. Boyle, M. Broadie, and P. Glasserman. Monte Carlo methods for security pricing. *Journal of Economic Dynamics & Control*, 21:1267–1321, 1997.
- [4] M.J. Brennan and E.S. Schwartz. The valuation of American put options. *The Journal of Finance*, 32(2):449–462, 1977.
- [5] X. Chen and J. Chadam. A mathematical analysis for the optimal exercise boundary of American put option. Preprint., 2000.
- [6] N. Clarke and K. Parrott. The multigrid solution of two-factor American put options. Technical Report 96-16, Oxford Comp. Lab, 1996.
- [7] N. Clarke and K. Parrott. Multigrid for American option pricing with stochastic volatility. *Applied Mathematical Finance*, 6:177–195, 1999.
- [8] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Applications of mathematics: Stochastic modeling and applied probability. Springer Verlag, New York, 2004.
- [9] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 3rd edition, Baltimore, Maryland 1996.
- [10] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I -Non-stiff problems*. Springer-Verlag, 2nd edition, Berlin 1993.
- [11] S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.
- [12] S. Ikonen and J. Toivanen. Operator splitting methods for pricing American options. *Applied Mathematical Letters*, 17:809–814, 2004.
- [13] S. Ikonen and J. Toivanen. Operator splitting methods for pricing American options with stochastic volatility. Technical report B11/2004, Department of Mathematical Information Technology, 2004. University of Jyväskylä, Finland.
- [14] S. Ikonen and J. Toivanen. Componentwise splitting methods for pricing American options under stochastic volatility. Technical report B7/2005, Department of Mathematical Information Technology, 2005. University of Jyväskylä, Finland.
- [15] M.D. Koulisianis and T.S. Papatheodorou. Improving projected successive overrelaxation method for linear complementarity problems. *Applied Numerical Mathematics*, 45:29–40, 2003.
- [16] G. Linde, J. Persson, and L. von Sydow. High-order adaptive space-discretizations for the Black–Scholes equation. Technical report, Dept. of Information Technology, Uppsala University, Uppsala, Sweden, 2006. Technical report 2006-021.

- [17] P. Lötstedt, J. Persson, L. von Sydow, and Johan Tysk. Space-time adaptive finite difference method for European multi-asset options. Technical report 2004-055, Dept. of Information Technology, Uppsala University, Uppsala, Sweden, 2004.
- [18] P. Lötstedt, S. Söderberg, A. Ramage, and L. Hemmingsson-Frändén. Implicit solution of hyperbolic equations with space-time adaptivity. *BIT*, 42:128–153, 2002.
- [19] R.C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.
- [20] M. Musiela and M. Rutkowski. *Martingale Methods in Financial Modeling*. Springer-Verlag, Berlin, 1997.
- [21] B. F. Nielsen, O. Skavhaug, and A. Tveito. Penalty methods for the numerical solution of American multi-asset option problems. Preprint 2000-289, Department of Informatics, University of Oslo, 2000.
- [22] B. F. Nielsen, O. Skavhaug, and A. Tveito. Penalty and front-fixing methods for the numerical solution of American option problems. *Journal of Computational Finance*, 5(4):69–97, 2002.
- [23] C.W. Oosterlee. On multigrid for linear complementarity problems with application to American-style options. *Electronic Transactions on Numerical Analysis*, 15:165–185, 2003.
- [24] K.N. Pantazopoulos, E.N. Houstis, and S. Kortesis. Front-tracking finite difference methods for the valuation of American options. *Computational Economics*, 12:255–273, 1998.
- [25] J. Persson and L. von Sydow. Pricing European multi-asset options using a space-time adaptive FD-method. To appear in *Computing and Visualization in Science*, Dept. of Information Technology, Uppsala University, Uppsala, Sweden, 2003.
- [26] R. Seydel. *Tools for Computational Finance*. Springer Verlag, Berlin, 2003.
- [27] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5:506–517, 1968.
- [28] D. Tavella and C. Randall. *Pricing Financial Instruments - The Finite Difference Method*. John Wiley & Sons, Inc., Chichester, 2000.
- [29] R. Zvan, P.A. Forsyth, and K.R. Vetzal. Penalty methods for American options with stochastic volatility. *Journal of Computational and Applied Mathematics*, 91:199–218, 1998.