Space-time adaptive finite difference method for European multi-asset options

Per Lötstedt[1], Jonas Persson[1]*, Lina von Sydow[1]†, Johan Tysk[2]‡

[1] *Division of Scientific Computing, Department of Information Technology*
Uppsala University, SE-75105 Uppsala, Sweden
[2] *Department of Mathematics*
Uppsala University, SE-75106 Uppsala, Sweden
emails: perl, jonasp, lina @it.uu.se, Johan.Tysk@math.uu.se

Abstract

The multi-dimensional Black-Scholes equation is solved numerically for a European call basket option using *a priori–a posteriori* error estimates. The equation is discretized by a finite difference method on a Cartesian grid. The grid is adjusted dynamically in space and time to satisfy a bound on the global error at the expiry date. The discretization errors in each time step are estimated and weighted by the solution of the adjoint problem. Bounds on the local errors and the adjoint solution are obtained by the maximum principle for parabolic equations. The performance of the method is illustrated by examples in one, two, three, and four dimensions.

Keywords: Black-Scholes equation, finite difference method, space adaptation, time adaptation, maximum principle

AMS subject classification: 65M20, 65M50

1 Introduction

We are interested in the numerical solution of the multi-dimensional Black-Scholes equation

$$\begin{aligned} \frac{\partial F}{\partial \hat{t}} + \sum_{i=1}^{d} \bar{r} s_i \frac{\partial F}{\partial s_i} + \frac{1}{2}\sum_{i,j=1}^{d} [\bar{\sigma}\bar{\sigma}^*]_{ij} s_i s_j \frac{\partial^2 F}{\partial s_i \partial s_j} - \bar{r}F &= 0, \\ F(\hat{T}, s) &= \Phi(s), \end{aligned} \qquad (1)$$

to determine the arbitrage free price F of a European option expiring at $\hat{T}$ with contract function $\Phi(s)$. Here, $\bar{r} \in \mathbb{R}_+ = \{x | x \geq 0\}$ is the short rate of interest

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

†Partially supported by the Swedish Research Council (VR) under contract 621-2003-5280.

‡Partially supported by the Swedish Research Council (VR) under contract 621-2003-5267.

and $\bar{\sigma} \in \mathbb{R}^{d\times d}$ is the volatility matrix. Our numerical method allows $\bar{r}$ and $\bar{\sigma}$ to be both level and time dependent but some of the theoretical estimates are restricted to time dependent interest and volatility.

We will consider a European call basket option where the contract function is defined by

$$\Phi(s) = \left(\frac{1}{d} \sum_{i=1}^{d} s_i - K \right)^+ , \tag{2}$$

where $(x)^+ = \max(x, 0)$ and K is the so called strike price. Our method will work just as well for any contract function with vanishing second derivative across the boundary. This way of determining the arbitrage free price was introduced independently by F. Black and M. Scholes in [3] and R.C. Merton in [12], both in 1973.

Another way to determine this price is to solve a stochastic differential equation with a Monte Carlo method and use the Feynman-Kač formula, see e.g. [13]. However, this method is well-known to converge slowly in the statistical error. If we denote the number of simulations by M, the statistical error is proportional to $M^{-1/2}$. In [14], an adaptive finite difference method is developed with full control of the local discretization error which is shown to be very efficient. For more dimensions than five (or so), the solution with finite difference approximations on a grid suffers from the "curse of dimensionality" with an exponential growth in dimension d of the number of grid points and a Monte Carlo algorithm is presently perhaps the best alternative. However, we believe that the finite difference method is a better method for dimensions at most five due to the uncertainty in the convergence of Monte Carlo methods. Furthermore, only a finite difference solution is sufficiently smooth for calculation of derivatives of the solution such as the hedging parameters $\Delta_i = \partial F/\partial s_i, \Gamma_i = \partial^2 F/\partial s_i^2, \theta = \partial F/\partial t$ ("the Greeks"). Finite difference methods for option pricing are found in the books [18, 19] and in the papers [7, 11, 14, 17]. A Fourier method is developed in [4] and an adaptive finite element discretization is devised in [15] for American options. In this paper we will further develop the finite difference approach.

The Black-Scholes equation is here discretized by second order accurate finite difference stencils on a Cartesian grid. The time steps and the grid sizes are determined adaptively. Adaptive methods have the advantages of providing estimates of the numerical errors and savings in computing time and storage for the same accuracy in the solution. Moreover, there is no need for a user to initially specify a constant time step and a constant grid size for the whole solution domain.

Examples of algorithms for adaptivity in space and time are found in [10, 14, 20]. The grid and the time step may change at every discrete time point in [10]. In [14], a provisional solution is computed for estimation of the errors and then the fixed grid and the variable time step are chosen so that the local errors satisfy given tolerances in the final solution. The grid has a fixed number of points but the points move in every time step for optimal distribution of them in moving grid methods, see e.g. [20]. In this paper, the time step varies in every step but the grid is changed only in certain time steps so that a maximal number of points are located optimally or a requirement on the error is fulfilled.

For the adaptive procedure, an error equation is derived for the global error $E(\hat{t}, s)$ in the solution. The driving right hand side in this equation is the local discretization error. This error is estimated and the grid is adapted at selected time points so that the Cartesian structure of the grid is maintained and the time step is adjusted in every time step. The step sizes are chosen so that a linear functional of the solution error at $\hat{t} = 0$ satisfies an accuracy constraint ϵ

$$| \int g(s)E(0, s)\ ds| \leq \epsilon \tag{3}$$

for a non-negative g chosen to be compactly supported where the accuracy of the solution is most relevant. The weights for the local error bounds in each time interval are solutions of the adjoint equation of (1). The growth of the error in the intervals between the grid adaptations is estimated *a priori* by the maximum principle for parabolic equations. In the same manner, the solution of the adjoint equation is bounded. Furthermore our algorithm automatically chooses the discretization so that bounds on the errors of the type (3) above are satisfied also for multi-dimensional equations. The adaptation algorithm is first applied to a one-dimensional problem for comparison between the computed solution and the analytical solution. Two-, three-, and four-dimensional problems are then successfully solved with the adaptive algorithm.

The paper is organized as follows. The equation (1) is transformed by a change of variables and scaling in Section 2. The discretization in space and time is described in the following section. The adjoint equation and its relation to the discretization errors is the subject of Section 4. The adjoint solution is estimated with the maximum principle in Section 5. In Section 6, the local discretization errors are estimated and a simplification is derived based on the maximum principle. The algorithms for the space and time adaptivity are discussed in Sections 7 and 8. In Section 9, the adaptive algorithm is applied to the pricing of European call basket options with one, two, three, and four underlying assets. Conclusions are drawn in the final section.

2 Model problem

We start by transforming (1) from a final value problem to an initial value problem with dimensionless parameters. The transformation of the time-scale has the advantage that standard texts on time-integrators are applicable. The following transformations give the desired properties:

$$\begin{array}{lll} Kx = s, & r = \bar{r}/\hat{\sigma}^2, & KP(t, x) = F(\hat{t}, s), \\ \sigma = \bar{\sigma}/\hat{\sigma}, & t = \hat{\sigma}^2(\hat{T} - \hat{t}), & K\ \Psi(x) = \Phi(s), \end{array} \tag{4}$$

where $\hat{\sigma}$ is a constant chosen as $\max_{ij} \sigma_{ij}$ in the solution domain. These transformations result in the following linear partial differential equation

$$\begin{aligned} P_t - r\sum_{i=1}^{d} x_i P_i - \sum_{i,j=1}^{d} a_{ij} x_i x_j P_{ij} + rP &= 0, \\ P(0,x) = \Psi(x) &= \left(\frac{1}{d} \sum_{i=1}^{d} x_i - 1 \right)^+, \end{aligned} \tag{5}$$

where $a_{ij} = \frac{1}{2}[\sigma\bar{\sigma}^*]_{ij}$. The coordinates of $\mathbb{R}^d$ are called the spatial variables and are denoted by $x_1, ..., x_d$. The subscripts i, j, and later also k, l, m, on a dependent variable denote differentiation with respect to x_i and x_j, e.g. P_{ij}. Subscripts on an independent variable denote components of a vector such as x_i, or entries of a matrix such as a_{ij}. The matrix $[a_{ij}]$ is assumed to be positive definite. Thus, (5) is a parabolic equation. The subscript t denotes differentiation with respect to normalized time.

We will solve (5) in a cylinder

$$C = D \times [0, T], \tag{6}$$

where D is a bounded computational domain in $\mathbb{R}^d_+$ with boundary ∂D.

3 Discretization

Let $\mathcal{L}$ be the operator

$$\mathcal{L} = r\sum_{i=1}^{d} x_i \frac{\partial}{\partial x_i} + \sum_{i,j=1}^{d} a_{ij} x_i x_j \frac{\partial^2}{\partial x_i \partial x_j} - r. \tag{7}$$

The partial differential equation (5) can then be written as

$$P_t = \mathcal{L}P. \tag{8}$$

We introduce a semi-discretization of (8) in space by using centered second order finite differences (FD) on a structured but non-equidistant grid, see Figure 1.

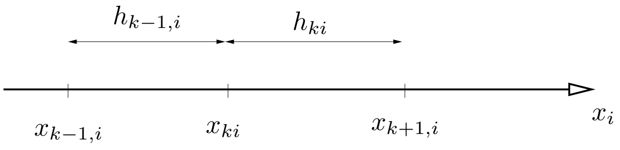

Figure 1: The x_i-axis. Here, x_{ki}, $k = 1 \ldots n_i$, denotes the k:th node of dimension i.

The number of grid-points in the i:th dimension is $n_i, i = 1, \ldots, d$. If we let P_h be a vector of the lexicographically ordered unknowns of length $\prod_{i=1}^{d} n_i$, then

$$\frac{dP_h}{dt} = A_h P_h, \tag{9}$$

where A_h is a matrix with the second order finite difference discretization of $\mathcal{L}$. The matrix A_h in (9) is a very large, sparse matrix with the number of non-zeros of each row depending on the number of space dimensions, i.e. the number of underlying assets. For a detailed derivation of the approximations of the derivatives, see [14].

There are several possible numerical boundary conditions that can be used for these problems. Here, the condition on a boundary where x_i is constant is that the numerical approximation of the second derivative P_{ii} is set to zero, which implies that the option price is nearly linear with respect to the spot price at the boundaries. These and other boundary conditions are discussed in [18].

For time-integration we use the backward differentiation formula of order two (BDF-2) [9], which is A-stable for constant time-steps. This method can be written

$$\begin{array}{l} \alpha_0^n P_h^n = \Delta t^n \mathcal{L}(P_h^n) - \alpha_1^n P_h^{n-1} - \alpha_2^n P_h^{n-2} \\ \alpha_0^n = (1+2\theta^n)/(1+\theta^n), \\ \alpha_1^n = -(1+\theta^n), \\ \alpha_2^n = (\theta^n)^2/(1+\theta^n) \end{array} \tag{10}$$

for variable time steps, where $\theta^n = \Delta t^n / \Delta t^{n-1}$, and $\Delta t^n = t^n - t^{n-1}$, see [9, 10].

4 Discretization errors and the adjoint equation

Let $\tilde{P}$ denote a smooth reconstruction of the discrete data in P_h^n so that they agree at $t = t^n$ and at the grid points. The solution error $E = \tilde{P} - P$ approximately satisfies the following boundary value problem ("the error equation")

$$E_t - r\sum_{i=1}^{d} x_i E_i - \sum_{i,j=1}^{d} a_{ij} x_i x_j E_{ij} + rE = E_t - \mathcal{L}E = \tau, \tag{11}$$
$$E(0,x) = 0, x \in D, \qquad E(t,x) = 0, x \in \partial D,$$

where τ is the local discretization or truncation error. By solving (11) we obtain the approximate global error $E_h^n = P_h^n - P$ at t^n at the grid points in $D \times [0,T]$.

The local discretization error consists of two parts, the temporal discretization error τ_k and the spatial discretization error τ_h

$$\tau = \tau_k + \tau_h. \tag{12}$$

The aim is to develop a method that estimates τ *a posteriori* at t^n and then estimates the evolution of τ_h *a priori* for $t > t^n$. Then we determine computational grids to control τ_h and the time steps are selected to control τ_k in order to obtain a final solution fulfilling predescribed error conditions on a functional of the global solution error E. Such methods have been developed for finite element discretizations of different PDEs, see e.g. [5, 1]. For this reason we introduce the adjoint equation to (11)

$$
\begin{aligned}
u_t + \mathcal{L}^* u = 0, \qquad (13)\\
\mathcal{L}^* u = -r \sum_{i=1}^{d} (x_i u)_i + \sum_{i,j=1}^{d} a_{ij} (x_i x_j u)_{ij} - ru, \qquad (14)\\
u(T,x) = g(x).
\end{aligned}
$$

The boundary conditions for the adjoint equation is $u = 0$ on ∂D. Note that the adjoint problem is a final value problem.

Using (11) and (13) we obtain

$$
\begin{aligned}
\int_0^T \int_D u\tau dx\, dt = \int_0^T \int_D uE_t dx\, dt - \int_0^T \int_D u\mathcal{L}E dx\, dt\\
= \int_D g(x)E(T,x)dx - \int_0^T \int_D u_t E dx\, dt - \int_0^T \int_D (\mathcal{L}^* u)E dx\, dt \qquad (15)\\
= \int_D g(x)E(T,x)dx.
\end{aligned}
$$

The function $g(x)$ should be chosen such that it is non-negative and has compact support in the domain where one is most interested in having an accurate solution. It is normalized such that

$$
\int_D g(x)dx = 1. \qquad (16)
$$

Partition the interval $[0,T]$ into L subintervals $\mathcal{I}_\ell = [t_\ell, t_{\ell+1})$ and take the absolute value of the left-hand side in (15) to arrive at

$$
\begin{aligned}
&| \int_0^T \int_D u(t,x)\tau(t,x)\; dxdt| \leq \sum_{\ell=0}^{L-1} | \int_{t_\ell}^{t_{\ell+1}} \int_D u(t,x)\tau(t,x)\; dxdt|\\
&\leq \sum_{\ell=0}^{L-1} \sup_{\substack{x\in D\\ t_\ell \leq t \leq t_{\ell+1}}} |\tau(t,x)| \int_{t_\ell}^{t_{\ell+1}} \int_D |u(t,x)|\; dxdt \qquad (17)\\
&= \sum_{\ell=0}^{L-1} \|u\|_\ell \sup_{\substack{x\in D\\ t_\ell \leq t \leq t_{\ell+1}}} |\tau(t,x)|,
\end{aligned}
$$

with the definition

$$
\|u\|_\ell = \int_{t_\ell}^{t_{\ell+1}} \int_D |u(t,x)|\; dxdt. \qquad (18)
$$

Our goal now is to generate a discretization of D and $[0,T]$ adaptively so that

$$
| \int_D g(x)E(T,x)dx| \leq \epsilon, \qquad (19)
$$

where ϵ is a prescribed error tolerance. From (15) and (17), it is clear that we can bound the integral from above by estimating $\sup|\tau|$ and $\|u\|_\ell$.

The unknown u is the solution to the adjoint problem (13) and thus $\|u\|_\ell$ cannot be adjusted in order to fulfil (19). However, we are able to adjust the discretization error τ by controlling h and Δt in the spatial and temporal discretization. Thus, we will require in each interval $\mathcal{I}_\ell$ that

$$\sup_{\substack{x\in D\\ t_\ell\le t\le t_{\ell+1}}} |\tau(t,x)| \le \epsilon_\ell. \tag{20}$$

We choose to equidistribute the errors in the intervals yielding

$$\epsilon_\ell = \epsilon/(L\|u\|_\ell). \tag{21}$$

Then from (17), (20) and (21) we find that

$$|\int_D g(x)E(T)dx| \quad \le \sum_{\ell=0}^{L-1} \|u\|_\ell \sup_{\substack{x\in D\\ t_\ell\le t\le t_{\ell+1}}} |\tau(t,x)| \le \epsilon.$$

To summarize this section we have a strategy to obtain the prescribed tolerance ϵ in (19):

(i) Compute $\|u\|_\ell,\ \ell=0,\dots,L-1$ in (18).

(ii) Compute ϵ_ℓ, $\ell=0,\dots,L-1$ using (21).

(iii) Generate computational grids Γ_ℓ, $\ell=0,\dots,L-1$, and choose time steps Δt^n for all n such that (20) is satisfied.

The time steps are adjusted in every step but the grids are changed only at L prespecified locations. The spatial error is estimated in the beginning of each interval with a constant grid and its growth in the interval is estimated (see Section 6). In this way, the expensive redistribution of the grid points and interpolation of the solution to a new grid are limited to $t=t_\ell, \ell=0,1,\dots,L-1$. In Section 5 we will estimate $\|u\|_\ell$ *a priori* and in Sections 6, 7, and 8 we will demonstrate how to estimate τ *a priori* and *a posteriori* and derive new computational grids and vary the time step.

5 Maximum principle for the solution of the adjoint equation

A bound on the solution of the adjoint equation (13) is derived assuming constant r and a_{ij} using the maximum principle for parabolic equations, see [6]. Performing the differentiation in (13) and transforming the adjoint equation to an initial value problem by substituting $\tilde{t} = T - t$ yields

$$\begin{aligned} u_{\tilde{t}} - \sum_{i,j=1}^{d} \left(2a_{ij}(1+\delta_{ij}) - \delta_{ij} r\right) x_j u_j - \sum_{i,j=1}^{d} a_{ij} x_i x_j u_{ij} \quad - & \\ \left(\sum_{i,j=1}^{d} a_{ij}(1+\delta_{ij}) - (d+1)r \right) u \quad = & \quad 0, \\ u(\tilde{t}, x) = 0, x \in \partial D, \qquad u(0,x) = g(x), x \in D. & \end{aligned} \tag{22}$$

The Kronecker delta function is denoted by δ_{ij}. We also have $t_\ell = \tilde{t}_{L-\ell}$, $\ell = 0, \ldots, L-1$.

We introduce the standard notion of parabolic boundary of the cylinder

$$\tilde{C}_\ell = D \times [\tilde{t}_\ell, \tilde{t}_{\ell+1}), \tag{23}$$

denoting it by $\partial\tilde{C}_\ell$ as the topological boundary of $\tilde{C}_\ell$ except $D \times \tilde{t}_\ell$. The standard maximum principle, see [6], says that in an equation of the type (22), in the absence of zero order terms, the maximum and minimum of u over $\tilde{C}_\ell$ are attained on $\partial\tilde{C}_\ell$. In our case there is a zero order term Ru where

$$R = \sum_{i,j=1}^{d} a_{ij}(1+\delta_{ij}) - (d+1)r.$$

However, the function $e^{-R\tilde{t}}u$ satisfies (22) without zero order terms. Thus, by the maximum principle

$$\inf_{\partial\tilde{C}_\ell} u e^{-R\tilde{t}} \leq u(\tilde{t}, x) e^{-R\tilde{t}} \leq \sup_{\partial\tilde{C}_\ell} u e^{-R\tilde{t}}. \tag{24}$$

Thus, using that $g \geq 0$ and the boundary condition on ∂D, the estimate

$$0 \leq u(\tilde{t}, x) \leq e^{R\tilde{t}} \sup_{\partial\tilde{C}_\ell} u e^{-R\tilde{t}} \leq \begin{cases} e^{R(\tilde{t}-\tilde{t}_\ell)} \sup_{\partial\tilde{C}_\ell} u & , \quad R \geq 0, \\ e^{R(\tilde{t}-\tilde{t}_{\ell+1})} \sup_{\partial\tilde{C}_\ell} u & , \quad R < 0. \end{cases} \tag{25}$$

holds for all $(\tilde{t}, x)$ in $\tilde{C}_\ell$.

Let $\Delta_\ell = t_{\ell+1} - t_\ell$. From the previous section we are interested in estimating

$$\begin{aligned} \sup_{\substack{x \in D \\ t_\ell \leq t \leq t_{\ell+1}}} |u(t,x)| = \sup_{\substack{x \in D \\ \tilde{t}_{L-\ell-1} \leq \tilde{t} \leq \tilde{t}_{L-\ell}}} |u(\tilde{t},x)| \leq e^{|R|\Delta_\ell} \sup_{\partial\tilde{C}_\ell} |u(\tilde{t},x)|, & \\ \ell = 0, \ldots, L-1. & \end{aligned} \tag{26}$$

Since $u(\tilde{t}, x) = 0$ on ∂D, $\sup_{\partial\tilde{C}\ell} |u(\tilde{t}, x)|$ is reached at $\tilde{t} = \tilde{t}_\ell$. In particular, with the initial data $g(x)$

$$\sup_{\substack{x \in D \\ t_\ell \leq t \leq t_{\ell+1}}} |u(t,x)| \leq \sup_{\substack{x \in D \\ 0 \leq t \leq t_L}} |u(t,x)| \leq e^{|R| t_L} \sup_{x \in D} g(x). \tag{27}$$

Finally, by (27) we have a bound on $\|u\|_\ell$ in (18)

$$\|u\|_\ell \leq |D|\Delta_\ell \sup_{\substack{x\in D \\ t_\ell \leq t \leq t_{\ell+1}}} |u| \leq |D|\Delta_\ell e^{|R|t_L} \sup_{x\in D} g(x). \tag{28}$$

From this upper bound, ϵ_ℓ, $\ell = 0, \ldots, L-1$, can be computed using (21).

These *a priori* estimates are in general not sufficiently sharp for the selection of ϵ_ℓ and an efficient adaptive procedure. Instead, (22) is solved numerically on a coarse grid in order to find $\|u\|_\ell$.

6 Estimating the spatial discretization error

The spatial error is estimated *a priori* in this section by applying the maximum principle to equations satisfied by terms in the discretization error. A simplifying assumption concerning the spatial error in the analysis here and in the implementation of the adaptive scheme is

Assumption 1 The dominant error terms in the approximations of the second derivatives in (5) are due to the diagonal terms $a_{ii}x_i^2P_{ii}$.

The assumption is valid if $a_{ij} \ll a_{kk}$ for $i \neq j$ and all k.

The following assumption is necessary for the analysis below to be valid. The adaptive procedure works well for an x-dependent interest rate and volatility but the *a priori* analysis is much more complicated.

Assumption 2 The interest rate r and the volatility matrix $[a_{ij}]$ are level (i.e. space) independent.

If Assumption 1 is valid, then the dominant terms in the discretization error in space of the operator (5) is

$$\tau_h = \sum_{k=1}^{d} \tau_{hk} = r\sum_{k=1}^{d} x_k\tau_{1k} + \sum_{k=1}^{d} a_{kk}x_k^2\tau_{2k}, \tag{29}$$

where τ_{1k} is the error in the approximation of P_k and τ_{2k} is the error in P_{kk}. With the centered difference schemes in Section 3, the leading terms in τ_{1k} and τ_{2k} in the step size h_k in the k-direction are

$$\tau_{1k} = -\tfrac{1}{6}h_k^2P_{kkk} + O(h_k^3),\ \tau_{2k} = -\tfrac{1}{12}h_k^2P_{kkkk} + O(h_k^3). \tag{30}$$

The derivatives of P satisfy parabolic equations similar to the equation for P (5). These equations are derived in the following lemma.

Lemma 1 *Let $G = \partial^K P/(\partial x_k)^K$ and let Assumption 2 be valid. Then G fulfills*

$$G_t = \sum_{i,j=1}^{d} a_{ij}x_ix_jG_{ij} + \sum_{j=1}^{d}(\alpha_K a_{jk}x_j + r)G_j + (\beta_K a_{kk} + r\gamma_K)G, \tag{31}$$

where $\alpha_K = 2K,\ \beta_K = \sum_{j=1}^{K-1}\alpha_j,\ \gamma_K = K-1.$

Proof. The result follows from induction starting with (5) for $K = 0$. □

In order to estimate the error terms in each separate coordinate direction in (29) and (30) a parabolic equation is derived for fG, where f depends on only one coordinate.

Lemma 2 *Let $G = \partial^K P/(\partial x_k)^K, f = f(x_k)$, and let $[a_{ij}]$ be symmetric and let Assumption 2 be satisfied. Then*

$$\begin{array}{rcl} (fG)_t & = & \sum_{i,j=1}^d a_{ij}x_ix_j(fG)_{ij} \\ & + & \sum_{j=1}^d((\alpha_K a_{jk}+r)x_j - 2a_{jk}x_jx_k(f_k/f))(fG)_j \\ & + & (\beta_K a_{kk} + r\gamma_K - a_{kk}x_k^2(f_{kk}/f) - (\alpha_K a_{kk}+r)x_k(f_k/f) \\ & & +2a_{kk}(x_kf_k/f)^2)(fG). \end{array} \tag{32}$$

Proof. Multiply (31) by f, replace fG_j and fG_{ij} by

$$\begin{array}{rcl} fG_j & = & (fG)_j - \delta_{jk}(f_k/f)(fG), \\ fG_{ij} & = & (fG)_{ij} - \delta_{ik}\delta_{jk}(f_{kk}/f)(fG) + 2\delta_{ik}\delta_{jk}(f_k/f)^2(fG) \\ & & -\delta_{ik}(f_k/f)(fG)_j - \delta_{jk}(f_k/f)(fG)_i, \end{array}$$

and we have the equation (32). □

We are now able to obtain a bound on the spatial discretization error in (29) by letting $f = x_k^q h(x_k)^2,\ q = 1, 2$, and $G = P_{kkk}$ and P_{kkkk} in Lemma 2.

Theorem 1 *Let $[a_{ij}]$ be symmetric and let Assumption 2 be satisfied. Then the spatial error τ_h in (29) in $C_\ell = D \times [t_\ell, t_{\ell+1})$ is bounded by*

$$\begin{array}{rl} |\tau_h| & \leq \sum_{k=1}^d \frac{1}{6} r \exp(z_{3k}\Delta_\ell) \sup_{\partial C_\ell} x_k h^2(x_k)|P_{kkk}| \\ & + \sum_{k=1}^d \frac{1}{12} a_{kk} \exp(z_{4k}\Delta_\ell) \sup_{\partial C_\ell} x_k^2 h^2(x_k)|P_{kkkk}|. \end{array} \tag{33}$$

The constants z_{3k} and z_{4k} are the upper bounds

$$\begin{array}{ll} \beta_K a_{kk} + \gamma_K r - (\alpha_K a_{kk} + r)x_kf_k/f & \\ +2a_{kk}(x_kf_k/f)^2 - a_{kk}x_k^2 f_{kk}/f & \leq z_{Kk},\ K = 3, 4. \end{array} \tag{34}$$

Proof. With $f = x_kh^2(x_k)$ and $G = P_{kkk}$ the nonconstant part of the leading term of τ_{1k} in (29) and (30) satisfies (32). By the maximum principle, see [6], applied to (32) using the same type of argument as in Section 5 we obtain

$$|x_kh^2(x_k)P_{kkk}| \leq \exp(z_{3k}(t - t_\ell)) \sup_{\partial C_\ell} x_kh^2(x_k)|P_{kkk}|.$$

Then the error due to the first derivatives is inferred from (29). The error caused by the second derivatives is derived in the same manner. □

The factors $z_{Kk}, K = 3, 4$, in the theorem depend on the smoothness of the step sizes. With $\partial_k h = \partial h/\partial x_k$ the terms in (34) with $q = 1, 2$, are

$$\begin{array}{rcl} \dfrac{x_kf_k}{f} & = & q + 2\dfrac{x_k\partial_k h}{h}, \\ \dfrac{x_k^2 f_{kk}}{f} & = & q(q-1) + 4q\dfrac{x_k\partial_k h}{h} + 2\dfrac{x_k^2\partial_k^2 h}{h} + 2(\dfrac{x_k\partial_k h}{h})^2. \end{array}$$

If the successive steps vary so that $h(x_k) = h_{0k}\exp(cx_k)$ for some constant c, then

$$\frac{\partial_k h}{h} = c,\ \frac{\partial_k^2 h}{h} = c^2,$$

and with a small c, $x_k f_k/f$ and $x_k^2 f_{kk}/f$ are small in (34).

7 Space adaptivity

The computational domain D is a d-dimensional cube $[0, x_{max}]^d$ covered by a Cartesian grid with the step sizes $h_{ij}, i = 1, \ldots, n_j, j = 1, \ldots, d$. The grid points, the outer boundary x_{max} and the step sizes are related by (cf. Figure 1)

$$\begin{array}{lll} x_{ij} & = & x_{i-1,j} + h_{ij}, i = 2, \ldots, n_j, \\ \sum_{i=1}^{n_j} h_{ij} & = & x_{max}, j = 1, \ldots, d. \end{array}$$

Suppose that the time step Δt is constant in $[t_\ell, t_{\ell+1})$ and that the spatial step h_j is constant in the j:th dimension of D. If w_0 is the computational work per grid point and time step, then the total computational work in C_ℓ is

$$w = w_0 \frac{\Delta_\ell}{\Delta t} \prod_{j=1}^{d} \frac{x_{max}}{h_j}. \tag{35}$$

The discretization error according to (20), (29), and (30) satisfy

$$|\tau| \le |\tau_k| + |\tau_h| \le |\tau_k| + \sum_{j=1}^{d} |\tau_{hj}| \le c_t \Delta t^2 + \sum_{j=1}^{d} c_j h_j^2 \le \epsilon_\ell, \tag{36}$$

for all t and x for some positive constants c_t and c_j in a second order method. The step sizes Δt and h_j should be chosen such that w in (35) is minimized subject to the accuracy constraint (36). Since c_t and c_j are positive, the minimum of w is attained when the right part of (36) is satisfied as an equality. Then w is

$$w = w_0 \frac{\sqrt{c_t}\Delta_\ell}{\sqrt{\epsilon_\ell - \sum_{j=1}^{d} c_j h_j^2}} \prod_{j=1}^{d} \frac{x_{max}}{h_j},$$

and a stationary point with respect to h_i is at

$$\frac{\partial w}{\partial h_i} = w \left(\frac{c_i h_i}{\epsilon_\ell - \sum_{j=1}^{d} c_j h_j^2} - \frac{1}{h_i} \right) = 0.$$

Hence,

$$c_i h_i^2 = \epsilon_\ell - \sum_{j=1}^{d} c_j h_j^2,\ i = 1, \ldots, d,$$

with the solution

$$c_i h_i^2 = \epsilon_\ell/(d+1),\ i = 1, \dots, d. \tag{37}$$

The optimal bound on the time steps is obtained from (36) and (37)

$$c_t \Delta t^2 = \epsilon_\ell/(d+1). \tag{38}$$

Thus, it is optimal under these conditions to equidistribute the discretization errors in time and the dimensions.

As in [14], the spatial error τ_h is estimated *a posteriori* from the numerical solution by comparing the result of the fine grid space operator B_h with a coarse grid operator B_{2h} using every second grid point. Both B_h and B_{2h} approximate B to second order. Suppose that P_{hi} approximates the analytical solution $P(x)$ at x_i to second order in one dimension so that

$$P_{hi} = P(x_i) + c(x_i)h_i^2 + O(h_i^3),$$

where $c(x)$ is a smooth function and h_i has a slow variation. Then

$$\begin{array}{rcl} (B_h P_h)_i & = & (B_h P)(x_i) + (B_h c)(x_i)h_i^2 + O(h_i^3) \\ & = & (BP)(x_i) + (Bc)(x_i)h_i^2 + \eta_{hi} + O(h_i^3), \\ (B_{2h} P_h)_i & = & (B_{2h} P)(x_i) + (B_{2h} c)(x_i)h_i^2 + O(h_i^3) \\ & = & (BP)(x_i) + (Bc)(x_i)h_i^2 + \eta_{2hi} + O(h_i^3). \end{array}$$

Subtract $B_h P_h$ from $B_{2h} P_h$ at every second grid point and use the the second order accuracy in the discretization error to obtain $\eta_{2hi} = 4\eta_{hi} + O(h_i^3)$ and

$$\eta_{hi} = \frac{1}{3}((B_{2h} P_h)_i - (B_h P_h)_i) + O(h_i^3). \tag{39}$$

The leading term in the spatial error is given by the first term in the right hand side of (39).

The sequence h_{ij} in each dimension j is chosen so that (cf. Theorem 1 and (33))

$$\max_i h_{ij}^2 (\tfrac{1}{6} r \exp(z_{3j}\Delta_\ell) x_{ij} |P_{jjj}| + \tfrac{1}{12} a_{jj} \exp(z_{4j}\Delta_\ell) x_{ij}^2 |P_{jjjj}|) \leq \epsilon_\ell/(d+1) \tag{40}$$

in each coordinate direction j where the maximum for i is taken over all the other dimensions. By changing the step size in each dimension separately, the Cartesian grid structure is maintained. The derivative P_{jjj} is estimated by computing η_{hi} in (39) with B_h being the centered difference approximation of the first derivative of P. Then $\eta_{hi} = h_i^2 P_{jjj}/6$. With B_h approximating the second derivative of P to second order, we have $\eta_{hi} = h_i^2 P_{jjjj}/12$.

The spatial error τ_h at t_ℓ is estimated as in (40) with the solution P^ℓ at t_ℓ and the step size sequences $h_{ij},\ i = 1, \dots, n_j, j = 1, \dots, d$. The new sequence $\bar{h}_{ij}$ for $t > t_\ell$ is chosen locally at x_{ij} such that

$$\bar{h}_{ij} = h_{ij} \sqrt{\frac{\epsilon_\ell}{(d+1)(\epsilon_\ell \chi_h + |\tau_{hj}(x_{ij})|)}}. \tag{41}$$

Then the new error $\tau_{\bar{h}}$ is expected to be

$$|\tau_{\bar{h}j}(x_{ij})| = \bar{h}_{ij}^2 |\tau_{hj}(x_{ij})| / h_{ij}^2 = \epsilon_\ell/(d+1). \tag{42}$$

The small parameter χ_h in (42) ensures that h_{ij} is not too large when τ_h is very small. Since τ_h occasionally is non-smooth we apply a filter on these approximations of the local discretization errors to avoid an oscillatory sequence h_{ij}.

For multi-dimensional problems, the storage requirements may be the limiting factor and as an option the number of grid points can be restricted to a predefined level. The grid will be optimized for a small error within the limits of the available memory. By choosing a maximum number of grid points N_{max} in each direction j the method will still distribute the points so that $|\tau_{hj}(x_{ij})|$ is minimized. Suppose that the numerically computed discrete distribution of the grid points is $\bar{h}(x)$ determined by τ_h and that this distribution induces that $\bar{N}$ grid points are used. The new distribution will then place the grid points according to the scaled function

$$h_{new} = \frac{\bar{N}}{N_{max}} \bar{h}(x). \tag{43}$$

In several dimensions this simple technique can reduce the number of grid points in each interval so that larger problems can be solved, but it can also be used to ensure that not too many points are used in the first interval. Experiments have shown that limiting the number of grid points, especially in the first interval, does not destroy the end-time accuracy in (19).

8 Time adaptivity

The discretization error in space is estimated by comparing a fine grid operator with a coarse grid operator. For the adaption of the time steps we compare an explicit predictor and an implicit corrector (BDF-2), both of second order accuracy, to find an approximation of the local error in BDF-2 in the same way as in [10]. The predictor is the explicit method

$$\begin{array}{l} \tilde{\alpha}_0^n \tilde{P}^n = \Delta t^n \mathcal{L}(P^{n-1}) - \tilde{\alpha}_1^n P^{n-1} - \tilde{\alpha}_2^n P^{n-2}, \\ \tilde{\alpha}_0^n = 1/(1+\theta^n),\ \tilde{\alpha}_1^n = \theta^n - 1, \\ \tilde{\alpha}_2^n = -(\theta^n)^2/(1+\theta^n), \end{array} \tag{44}$$

with the local discretization error

$$\begin{array}{l} P(t^n) - \tilde{P}^n = C_p(\theta^n)(\Delta t^n)^3 P_{ttt} + O(\Delta t^4), \\ C_p(\theta^n) = (1 + 1/\theta^n)/6, \end{array} \tag{45}$$

and θ^n defined by $\theta^n = \Delta t^n / \Delta t^{n-1}$ as in (10). The solution at t^n is determined by the implicit method BDF-2 defined in (10) with the predicted value $\tilde{P}^n$ from (44) as initial guess in an iterative solver. The local error of BDF-2 is

$$\begin{array}{l} P(t^n) - P^n = C_i(\theta^n)(\Delta t^n)^3 P_{ttt} + O(\Delta t^4), \\ C_i(\theta^n) = -(1+\theta^n)^2/(6\theta^n(1+2\theta^n)). \end{array} \tag{46}$$

The integration is initialized at $t = 0$ with the Euler backward method with $\alpha_0^1 = 1$, $\alpha_1^1 = -1$, and $\alpha_2 = 0$ in (10).

The leading term $C_i(\theta^n)(\Delta t^n)^3 P_{ttt}$ in the local error in time in (46) is estimated by computing the difference between the numerical solution P^n in (10) and $\tilde{P}^n$ in (44)

$$\tau_k(t^n) = -\alpha_0^n C_i (\Delta t^n)^2 P_{ttt} \approx \alpha_0 C_i (P^n - \tilde{P}^n)/(\Delta t^n (C_i - C_p)). \qquad (47)$$

The maximum $|\tau_k|$ of the estimate $\tau_k(t^n)$ in (47) over all grid points in D is compared to the accuracy requirement $\epsilon_\ell/(d+1)$ by computing

$$\zeta^n = \sqrt{\frac{\epsilon_\ell}{(d+1)(\epsilon_\ell \chi_k + |\tau_k|)}}, \qquad (48)$$

where χ_k is a small parameter to avoid large time steps when τ_k is small (cf. (41)). If ζ^n is too large, then the time step is rejected and P^n is recomputed with a smaller Δt. Otherwise, P^n is accepted and a new Δt^{n+1} is determined. If $0.8 \le \zeta^n \le 1.15$, then we accept the time step and let $\Delta t^{n+1} = \Delta t^n$. If $\zeta^n < 0.8$, then the time step to t^n is rejected and P^n is recomputed with $\Delta t^n := 0.9\zeta^n \Delta t^n$. If $\zeta^n > 1.15$, then the step is accepted and the next time step is increased to $\Delta t^{n+1} = \min(0.9\zeta^n \Delta t^n, 2\Delta t^n)$ with the upper bound $2\Delta t^n$ introduced to avoid instabilities.

Since BDF-2 is an implicit method in time, we must solve large, linear, sparse systems of equations in each time step. These systems are solved with `GMRES` iterations [16]. The `GMRES` iterations are terminated when the relative residual norm is sufficiently small. To be efficient and memory lean, the iterative method is restarted after six iterations. The system of equations is preconditioned by the incomplete LU factorization [8]. The same factorization computed in the first time step is applied in all time steps after t_ℓ in each interval.

9 Numerical results

The transformed Black-Scholes equation (5) is solved in one, two, three and four space dimensions with our adaptive method. Several different tests have been performed examining the method and its performance. Our method is compared to the standard method with a uniform grid in space and adaptivity in time and we also study how the memory can be efficiently used by restricting the number of grid points.

Since the precision of the estimates of the derivatives was investigated in [14] we mainly focus on the estimates of the linear functional (19) in this paper. In one space dimension the true numerical error can be calculated so that the functional (19) can be determined. In higher dimensions this is not possible. However, in all tests the upper bound (17) of the leftmost integral in (15) is computed. This estimate will be denoted by Υ^ϵ and the adaptive process controls this value.

As a standard setup we have used the following parameters: the local mean rate of return $\bar{r}$ has been set to 0.05 and the volatility matrix $\bar{\sigma}$ has the value

0.3 on the diagonal and 0.05 in the sub- and super-diagonals. All other entries are zero. In the examples that follow, the volatility matrix is neither level nor time dependent but it could be chosen to be so without causing any difficulty in the adaptive method. In all computations we solve the transformed PDE (5) in forward time from 0 to $T = 0.1$. The computational domain D is a d-dimensional cube truncated at $x_{\max} = 4dK$ in every dimension, using a generalization of the common 'rule of thumb', i.e., the computational domain is four times the strike price multiplied by the number of dimensions d. The reason for multiplying by d is to have the far-field boundary at four times the location of the discontinuity of the derivative of the initial function $\Phi(s)$ in each dimension.

9.1 Estimating the functional

To evaluate the method, the functional (19) is estimated in numerical experiments. In one space dimension, the exact solution for the European call option is found in [2, 13] and is used to calculate the true error $E(x, T)$. The product $g(x)E(x, T)$ is integrated numerically with the second order trapezoidal method. The integral is denoted by $\int_D^* g(x)E(x, T)dx$.

The estimate Υ^ϵ defined by

$$\Upsilon^\epsilon = \sum_{\ell=0}^{L-1} \|u\|_\ell \sup^*_{\substack{x\in D \\ t_\ell \le t \le t_{\ell+1}}} |\tau(t, x)| \tag{49}$$

has been used in one and multiple space dimensions. This is the most interesting quantity since it is used to generate the grids in space and to select the time steps, see Section 4. The supremum of τ in (49) is denoted by a $*$ since it is not truly the supremum but has been estimated as follows.

The adjoint solution (13) is computed on a coarse equidistant grid with only a few time steps. Then $\|u\|_\ell$ in (18) is computed numerically. Theoretically the supremum of τ_h should be measured on the parabolic cylinder C_ℓ, see Theorem 1, but the errors are small on ∂D and we measure only τ on D after a few time steps from the start t_ℓ of each interval. The reason is that, when interpolating the solution from one grid to the next additional errors are introduced making the estimates of τ_h at t_ℓ unreliable. The initial condition is not sufficiently smooth for the adaptive procedure to work properly. Hence, in the first interval, we measure τ towards the end of the interval instead since the approximations of the derivatives P_{kkk} and P_{kkkk} blow up close to $t = 0$ and the algorithm would then use an excessive amount of grid points and very small time steps in the vicinity of $t = 0$. In Section 9.3 we show that the method actually can produce good results even with a restricted number of points in the first interval.

The *a priori* spatial error estimate in Theorem 1 contains the two factors $\exp(z_{3k}\Delta_\ell)$ and $\exp(z_{4k}\Delta_\ell)$. These coefficients in front of the third and fourth derivatives of P are typically of the size 1 to 3 indicating that the local discretization errors can grow that much in each interval. However, all our results show that these are really overestimates of the growth. The discretization errors do not increase with time in the intervals. On the contrary, they decay. This implies that Υ^ϵ will be overestimated in each interval.

The d-dimensional function $g(x)$ has been chosen as the product of Gaussian functions

$$g(x) = c \prod_{i=1}^{d} \exp(-5(x_i - 1)^2), \tag{50}$$

scaled by c to satisfy (16).

9.2 A one-dimensional numerical example

In the first one-dimensional example we have studied two different levels of ϵ. The estimate Υ^ϵ is compared with the numerically integrated (19) and the desired tolerance level ϵ for $L = 8$. The results are presented in Table 1.

ϵ	Υ^ϵ	$\lvert \int_D^* gE\ dx \rvert$	# grid points
10^{-3}	0.001088	0.000083	[81 61 49 45 41 37 33 53]
10^{-4}	0.000152	0.000009	[233 173 137 121 109 101 93 157]

Table 1: The estimate Υ^ϵ, the error functional in (19) and the number of grid points used in each interval for different tolerances.

We see that the algorithm produces a solution with a bound on the error close to the desired tolerance. As expected the estimate Υ^ϵ is larger than $\int_D^* gE\ dx$. A sharper estimate is obtained by increasing the number of intervals implying more frequent changes of the grid. We seek a balance between accurate estimates and many regridding operations (as in moving grid methods [20]) and coarser estimates with fewer changes of the grid (as we prefer here).

9.3 Restricting the number of grid points

An upper bound on the number of grid points is introduced in this one-dimensional example. Either this bound or the error tolerance determines the number of points. The distribution of points still depends on the spatial error estimate, see Section 7 and (43).

N_{max}	Υ^ϵ	# grid points
–	0.001088	[81 61 49 45 41 37 33 53]
61	0.001161	[65 61 49 45 41 37 33 53]
53	0.001246	[57 53 49 45 41 37 33 53]

Table 2: The bound on the number of points, the upper bound on the error functional, and number of grid points in the eight intervals.

The limit has been set to unlimited, 61 and 53 grid points in Table 2 and $\epsilon = 0.001$. We see that by restricting the number of grid points we might still achieve quite accurate results. The method sometimes has to add a few extra points (maximum of 4) since the number of points n_j must be $n_j \mod (4) = 1$ to be suitable for the error estimates.

9.4 Comparison with uniform grids in one dimension

A solution on an equidistant grid in space is compared to a solution with our adaptive method in Table 3. The maximal number of grid points used by the adaptive algorithm with tolerance 0.001 is distributed equidistantly.

	Υ^ϵ	$\int_D^* gE\, dx$	# grid points
Ad. grid	0.001088	-0.000083	[81 61 49 45 41 37 33 53]
Equi. grid	0.002229	-0.000178	[81]
Equi. grid	0.001086	-0.000075	[121]

Table 3: Estimates of the functionals are compared for adaptive grid and two uniform grids.

The results show that by redistributing the grid points adaptively, the error functional can be reduced significantly with fewer points. Counting the total number of points in the intervals, more than twice as many points are needed to reduce the error with an equidistant grid to the same level as the adapted grid (400 vs. 968). The price is more administration for the adaptivity but this overhead cost drops with increasing number of dimensions.

9.5 Two-dimensional numerical example

In the first two-dimensional example, two tolerance levels $\epsilon = 0.01$ and 0.001 are tested. In this case, an exact solution is not available. Therefore, only the estimate Υ^ϵ is presented together with the number of grid points used in each dimension in Table 4.

ϵ	Υ^ϵ	# grid points
10^{-2}	0.007710	$[61^2\ 45^2\ 33^2\ 29^2\ 29^2\ 25^2\ 25^2\ 29^2]$
10^{-3}	0.001417	$[193^2\ 113^2\ 81^2\ 65^2\ 57^2\ 53^2\ 45^2\ 73^2]$

Table 4: The error tolerances, the estimate of the functional (19), and the number of grid points in two dimensions.

As in the one-dimensional case in Table 1 we find that our method produces a result that almost fulfills the desired accuracy.

9.6 A second two-dimensional example

The one dimensional numerical example from Section 9.4 is repeated here in two dimensions. The result on an adapted grid with $\epsilon = 0.001$ is compared to the results on two equidistant grids in space in Table 5. The same number of points in space is used in one uniform grid as the largest number in an interval of the adapted grid. The other uniform grid is chosen so that Υ^ϵ is approximately the same.

	Υ^ϵ	# grid points
Ad. grid	0.007710	$[61^2\, 45^2\, 33^2\, 29^2\, 29^2\, 25^2\, 25^2\, 29^2]$
Equi. grid	0.014202	$[61^2]$
Equi. grid	0.007874	$[81^2]$

Table 5: Estimates of the functional for two uniform grids and the adaptive grid.

From the table we observe that the equidistant grid results in lower bound on the error even though 61^2 grid points were used in all time steps. The equidistant grid uses 81^2 grid points to achieve the same level of accuracy as our adaptive method. However, as remarked in Section 9.4, the adaptive method introduces a certain overhead and computation time is sometimes longer.

9.7 A three-dimensional example

	ϵ	Υ^ϵ	# grid points
Ad. grid	0.1	0.055996	$[41^3\, 29^3\, 29^3\, 25^3\, 29^3\, 25^3\, 29^3\, 29^2 \times 25]$
Equi. grid	-	0.082717	$[41^3]$
Ad. grid-N_{max}	0.1	0.084784	$[33^3\, 29^3\, 29^3\, 29^3\, 29^3\, 29^3\, 29^3\, 29^3]$
Ad. grid	0.05	0.039529	$[53^3\, 37^3\, 29^3\, 29^3\, 29^3\, 29^3\, 29^3\, 29^3]$

Table 6: The estimate of the functional (19) Υ^ϵ with two adaptive grids, an equidistant grid and an adaptive grid with a maximal number of grid points.

In this three-dimensional example we combine two of the experiments in the previous examples. First we solve with our adaptive metod and the error tolerance $\epsilon = 0.1$. Then we solve with an equidistant grid with the same number of grid points as the maximal number used by the adaptive method. In the next experiment, the number of grid points is restricted as in (43) such that the maximal number of points is lower in the first interval ($N_{max} = 33$). Finally, the solution is computed with a halved ϵ. The results are displayed in Table 6. The conclusion is also here that adaptive distribution achieves a lower error bound for the same number of points compared to a uniform distribution or the same error with fewer points.

9.8 A four-dimensional example

In four dimensions, an upper bound on the number of grid points in each dimension is introduced. The tolerance is $\epsilon = 0.01$ but it is effective only for the controlling the time step in this example. Comparison is made with two equidistant grids in Table 7.

	Υ^ϵ	# grid points
Ad. grid	0.149585	$[29^4,\ 29^4,\ 29^2 \times 25^2,\ 29^2 \times 25^2,\ 29^2 \times 25^2,\ 25^4,\ 25^4,\ 25^4]$
Equi. grid	0.275625	$[25^4]$
Equi. grid	0.263027	$[29^4]$

Table 7: The estimate Υ^ϵ with an adaptive and two equidistant grids.

From Table 7 we see that the adaptive method is more accurate than the solution on the equidistant grid with the same number of grid points. The computing time is measured to be approximately proportional to the total number of grid points in all three cases in the table, i.e. the small equidistant grid requires the shortest time, the adapted grid is in the middle, and the large equidistant grid requires the longest time. The time step is adapted to satisfy the error tolerance also for the equidistant grids. The time step varies by orders of magnitude in $[0, T]$, cf. the next subsection. Our adaptive method would be even more efficient compared with integration with a constant small time step on an equidistant grid.

9.9 Time-stepping and iterations

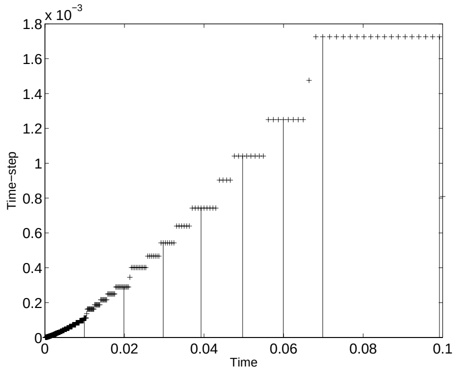

Figure 2: The time steps as a function of time.

The time steps are selected at every t^n following Section 8 such that the estimated τ_k satisfies $\max_D |\tau_k| \leq \epsilon_\ell/(d+1)$.

The time history of the time steps in the one-dimensional example with $\epsilon = 0.0001$ is displayed in Figure 2. The horizontal lines indicate the interval boundaries t_ℓ where a new grid is determined. At t_ℓ the estimate of the time discretization error is not always reliable and three steps with a constant Δt are taken there. The time step increases rapidly after $t = 0$ where higher derivatives of P are large due to the discontinuous initial data in (5).

The two-dimensional problem is solved in four intervals with $\epsilon = 0.005$. The variation of $|\tau_k|$ in the intervals is smooth in Figure 3. The error tolerance $\epsilon_\ell/(d+1)$ is not satisfied in the first steps after $t = 0$ where the integration is advanced with a minimal time step Δt_{min}. The number of GMRES iterations in each time step is found in Figure 4. It is about 10 in the whole interval with a small increase at the end when Δt is longer.

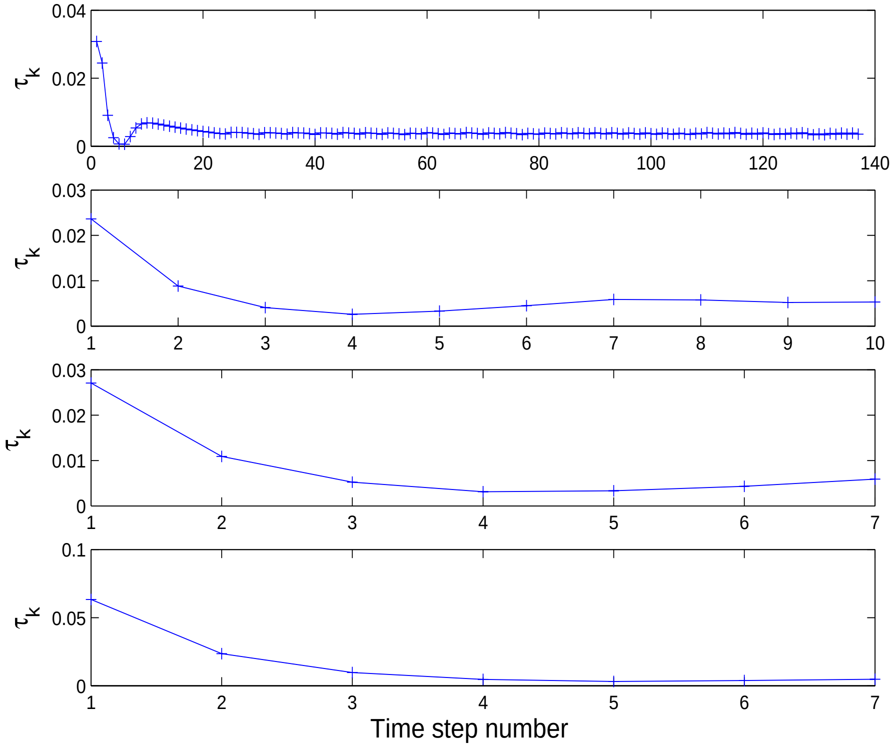

Figure 3: The measured local discretization error in time $|\tau_k|$ in four intervals.

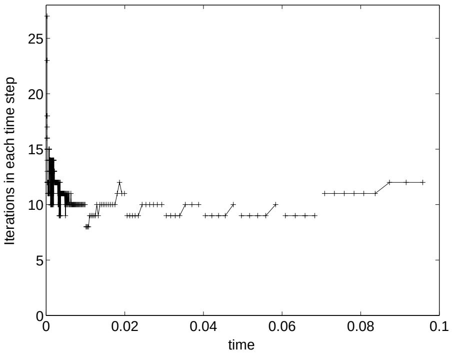

Figure 4: The number of GMRES iterations in each time step.

10 Conclusions

An adaptive method has been developed for efficient integration of the Black-Scholes equation for European multi-asset call options. The multi-dimensional computational grid and the time step are chosen such that a tolerance on the final global error is satisfied by the solution. The temporal discretization error is estimated *a posteriori* in every step but the spatial grid is constant in intervals of the time domain. In each interval, the error due to the space discretization is first determined *a posteriori* based on the solution and then its growth is estimated *a priori*. The grid is adjusted in each dimension separately so that its Cartesian structure is maintained. The user has to supply the error tolerance and a maximal number of grid points in each dimension. The algorithm automatically selects the grid and the time steps and provides an upper bound on the numerical error at the final time. The method has been tested successfully for problems with up to four dimensions corresponding to four underlying assets. Comparisons between adapted and equidistant grids with time step control show that lower bounds on the solution error are obtained with the same number of grid points with adaptation or we have the same bounds with fewer grid points. Since the time step increases rapidly from a low level, important gains in efficiency are achieved with a variable, adapted time step compared to a fixed, small time step.

References

[1] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[2] T. Björk. *Arbitrage Theory in Continuous Time*. Oxford University Press, New York, 1998.

[3] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Pol. Econ.*, 81:637–659, 1973.

[4] B. Engelmann and P. Schwender. The pricing of multi-asset options using a Fourier grid method. *J. Comput. Finance*, 1(4):53–61, 1998.

[5] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta Numerica*, 4:105–158, 1995.

[6] A. Friedman. *Partial Differential Equations of Parabolic Type.* Prentice-Hall, Englewood Cliffs, NJ 1964.

[7] M. Gilli, E. Këllezi, and G. Pauletto. Solving finite difference schemes arising in trivariate option pricing. *J. Econ. Dyn. Contr.*, 26:1499–1515, 2002.

[8] A. Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, Philadelphia, 1997.

[9] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations, Nonstiff problems.* Springer-Verlag, 2nd edition, Berlin, 1993.

[10] P. Lötstedt, S. Söderberg, A. Ramage, and L. Hemmingsson-Frändén. Implicit solution of hyperbolic equations with space-time adaptivity. *BIT*, 42:128–153, 2002.

[11] B. J. McCartin and S. M. Labadie. Accurate and efficient pricing of vanilla stock options via the Crandall-Douglas scheme. *Appl. Math. Comput.*, 143:39–60, 2003.

[12] R. C. Merton. Theory of rational option pricing. *Bell J. Econom. Manag. Sci.*, 4:141–183, 1973.

[13] M. Musiela and M. Rutkowski. *Martingale Methods in Financial Modelling.* Springer-Verlag, Heidelberg, 1997.

[14] J. Persson and L. von Sydow. Pricing European multi-asset options using a space-time adaptive FD-method. Report 2003-059, Dept. of Information Technology, Uppsala University, Uppsala, Sweden, 2003.

[15] O. Pironneau and F. Hecht. Mesh adaption for the Black & Scholes equations. *East-West J. Numer. Math.*, 8(1):25–35, 2000.

[16] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

[17] E. S. Schwartz. The valuation of warrants: implementing a new approach. *J. Finan. Econ.*, 4:79–93, 1977.

[18] D. Tavella and C. Randall. *Pricing Financial Instruments - The Finite Difference Method.* John Wiley & Sons, Chichester, 2000.

[19] P. Wilmott. *Option pricing: Mathematical Models and Computation.* Oxford Financial Press, Oxford, 1993.

[20] A. Vande Wouwer, P. Saucez, and W. E. Schiesser. *Adaptive Method of Lines.* Chapman & Hall, Boca Raton, 2001.