Multigrid Method for a Two Dimensional Fully Nonlinear Black-Scholes Equation with a Nonlinear Volatility Function

Aicha Driouch$^{1,2,*}$ and Hassan Al Moatassime$^2$

$^1$ LAMFA UMR 7352 CNRS, University of Picardie Jules Verne, Amiens, France;
$^2$ Department of Mathematics, Faculty of science and technology, University Cadi Ayyad, Marrakesh, Morocco.

Received May 31, 2019; Accepted October 22, 2019; Published online May 14, 2020.

Abstract. This paper deals with the task of pricing European basket options in the presence of transaction costs. We develop a model that incorporates the illiquidity of the market into the classical two-assets Black-Scholes framework. We perform a numerical simulation using finite difference method. We consider a nonlinear multigrid method in order to reduce computational costs. The objective of this paper is to investigate a deterministic extension for the Barles’ and Soner’s model and to demonstrate the effectiveness of multigrid approach to solving a fully nonlinear two dimensional Black-Scholes problem.

AMS subject classifications: 65N55, 65N06, 35K55, 65BXX

Key words: Fully nonlinear equation, multigrid method, black-scholes equation, finite difference method, FAS algorithm.

1 Introduction

Basket option is a financial derivative where the underlying asset is in the form of a sum or an average of different assets gathered together in a basket. Pricing basket options for nonlinear Black-Scholes equations is an interesting problem in financial mathematics, since an analytical solution of the basket option does not exist even in the linear case. Therefore, finding numerical solutions becomes an essential approach for solving the governing pricing equations (see [2]). In this work we are concerned with pricing European options in incomplete markets, more precisely, markets that are subjected to transaction costs. In this case the Black-Scholes theory given in [7] fails. To overcome this problem, several researchers have developed some extensions to the Black-Scholes equation.
by incorporating more realistic assumptions on financial markets (see [10, 11, 13, 20] for more details). We focus our attention on a nonlinear deterministic Black-Scholes extension given by Barles and Soner in [6]. They proposed an enlarged volatility to capture the influence of transaction costs,

\[
V_t + \frac{1}{2} \tilde{\sigma} \left( S, t, \frac{\partial^2 V}{\partial S^2} \right)^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \quad S > 0, \quad t \in (0, T),
\]

(1.1)

where

\[
\tilde{\sigma} \left( S, t, \frac{\partial^2 V}{\partial S^2} \right)^2 = \sigma^2 \left( 1 + \Psi \left( e^{\nu(T-t)}a^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) \right).
\]

(1.2)

The unknown \( V \) is the European call option, \( S \) is the underlying asset, \( T \) is the expiration date of the option \( V \), \( \sigma \) is the constant volatility given in the linear Black-Scholes model, \( r \geq 0 \) is the risk-free interest rate constant, \( a \) is a nonnegative parameter that summarizes risk aversion and transaction costs and \( \Psi \) denotes the solution to the nonlinear ordinary differential equation,

\[
\begin{cases}
\Psi'(x) = \frac{\Psi(x) + 1}{2\sqrt{x\Psi(x) - x}}, & x \neq 0, \\
\Psi(0) = 0.
\end{cases}
\]

(1.3)

Figure 1 is a representation of the solution of ODE (1.3) using a fourth order Runge-Kutta method. The problem (1.1) has been studied theoretically in [6] and the existence of a unique continuous viscosity solution has been shown using the theory of stochastic optimal control [14]. Recently, a constructive theoretical approach for the problem (1.1) was given in [1]. For numerical simulations of option pricing problems, Monte-Carlo methods are widely used. They are easy to implement and can handle multiple dimensions but they suffer from slow convergence (see [15, 16]). Over the last two decades, numerical methods based on PDEs have also gained popularity since they allow an easy computation of Greeks which represent a basic instrument for risk management. However, PDE methods cannot compete with Monte-Carlo methods in two or more dimensions in terms of computing time, consequently, they are limited to one or two dimensions. The aim of our work is to push forward the limit of what can be done using PDE methods. There is a considerable literature that reviews numerical simulations of the option price when markets are subject to transaction costs using PDEs, we quote [4, 19, 22], where several nonlinear Black-Scholes models in the presence of transaction costs have been studied for European and American options, [30] presented a positivity preserving scheme for the Barles’ and Soner’s model and they studied its convergence. [21] used a grid stretching technique on non-uniform meshes for the computation and [24, 25] reviewed nonlinear parabolic equations arising from markets under transaction costs. However, there are only few works for the computation of nonlinear Black-Scholes equations depending on more than one underlying asset, among them we cite [29], they gave a numerical
analysis for two dimensional nonlinear Black-Scholes equation for Spread option in incomplete markets and [3] where the authors gave a theoretical and numerical study of multi-asset nonlinear Black-Scholes equation arising from models with general transaction costs. Another problem is the boundary conditions for Black-Scholes equations, they have a boundary condition at infinity, which proves problematic for numerical simulations. One can overcome this difficulty by choosing a far field boundary condition similar to conditions provided in [18]. The main scope of our work is to give an extended version of the Barles’ and Soner’s model for solving numerically the two dimensional deterministic pricing problem, the difficulty resides in the nonlinear term, since it depends on different parameters such as the volatility of the two assets, the correlation coefficient and second derivatives of the option price.

This paper is organized as follows. In Section 2 we present the two dimensional Barles’ and Soner’s model for a basket option, we give in Section 3 the discrete scheme and boundary conditions for our model. We review the multigrid full approximation storage technique for nonlinear problems in Section 4 then numerical results are given in the section thereafter. Finally a conclusion is in order.

2 2D nonlinear deterministic Black-Scholes model

The purpose of our study is to analyze the impact of transaction costs on the option price when dealing with two underlying assets. We assume that the stock values $S_1(t)$ and $S_2(t)$ are following a geometric Brownian motion governed by

$$dS_i(t) = \mu_i S_i(t) dt + \sigma_i S_i(t) dB_i(t),$$

where $\sigma_i$ and $\mu_i$ ($i = 1,2$) are nonnegative constants that represent the volatility and the drift for the $i^{th}$ asset respectively and $B_i(t)$ are correlated Brownian motions. Our aim is
to price the basket option, the payoff in this case is given by

\[ V(S_1(T), S_2(T), T) = \max \left( (a_1 S_1(T) + a_2 S_2(T)) - K, 0 \right), \tag{2.2} \]

where \( K \) is the strike price for \( S_1 \) and \( S_2 \) and \( a_1 + a_2 = 1 \). The literature on hedging portfolios with correlated risky assets is very scanty, in [5] the value of the option is obtained by solving optimization problems and using asymptotic analysis. In our work we use a completely deterministic approach, we give a generalization of the Barles’ and Soner’s model for pricing a European basket call,

\[
\begin{align*}
V_t + \frac{1}{2} \left( 1 + \Psi \left( a^2 e^{r(T-t)} \sum_{i=1}^{2} \sum_{j=1}^{2} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} \right) \right) \sum_{i=1}^{2} \sum_{j=1}^{2} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} \\
+ \sum_{i=1}^{2} r S_i \frac{\partial V}{\partial S_i} - r V = 0, \\
V(S_1, S_2, T) = \max \left( \frac{1}{2} (S_1 + S_2) - K, 0 \right). \tag{2.3} \end{align*}
\]

For \((S_1, S_2) \in \mathbb{R}_+^2, t \in (0, T)\), where \( \rho_{ij} = \rho \) if \( i \neq j \) and \( \rho_{ij} = 1 \) otherwise, \( \rho \) is the correlation coefficient between the two assets \( S_1 \) and \( S_2 \) and \( \Psi \) is the solution of (1.3). Barles and Soner stated that \( \Psi \) has an asymptotic behaviour similar to the identity for large \( x \). This propriety allows us to simulate Eq. (2.3) with this simplification.

![Figure 2: Graph of the payoff of basket option \((K = 100, a_1 = a_2 = 0.5)\).](image)
3 Finite difference discretization

3.1 Boundary condition

It is known that the Black-Scholes equation is defined on an infinite domain \([0, +\infty) \times [0, +\infty)\). Therefore, in order to solve it numerically it is necessary to introduce artificial boundary conditions to make the computational domain bounded. Through rigorous error estimates, [18] shows that the optimal domain size for the far field boundary condition is given by

\[
S_{i,\text{max}} = Ke^{2\sqrt{2}T\ln(200)}
\]

where \(i = \{1, 2\}\). We denote our computational domain by \(\Omega = [0, S_{1\text{max}}] \times [0, S_{2\text{max}}]\). A very common type of boundary condition in finance is the linearity condition [12]; such a condition on a boundary \(\partial \Omega\) reads

\[
\frac{\partial^2 V}{\partial S_i^2} = 0, \quad \forall S_i \in \partial \Omega, \quad i \in \{1, 2\}.
\]

Eq. (3.2) means that the option price is nearly linear with respect to the asset price at a neighborhood of the infinity. However, a direct numerical simulation of Black-Scholes equation supplemented with this type of boundary conditions may cause some inaccuracies. Nevertheless, in [27, 28] the authors show that Black-Scholes problems with linearity conditions can be treated with sufficient accuracy using a theoretical stability result. In our work we use both linearity conditions and Dirichlet boundary conditions, i.e. if \(S_i = S_{i,\text{max}}\) for \((i = 1, 2)\) then we use the linearity condition, this leads us to the following equation

\[
V_t + \frac{1}{2}\sigma_i^2 \left( 1 + \Psi (a^2 e^{r(T-t)} \sigma_j^2 S_j^2 \frac{\partial^2 V}{\partial S_j^2}) S_i^2 \frac{\partial^2 V}{\partial S_i^2} + rS_j \frac{\partial V}{\partial S_j} + rS_{i,\text{max}} \frac{\partial V}{\partial S_i} - rV \right) = 0,
\]

with the Dirichlet boundary condition at \(S_{2\text{max}}\).

\[
V(0, S_{2\text{max}}, t) = \frac{1}{2} S_{2\text{max}} - Ke^{-r(T-t)}.
\]

If \(S_i = S_{i,\text{max}}\) for \((i = 1, 2)\) then we use the linearity condition, this leads us to the following equation

\[
V_t + \frac{1}{2}\sigma_i^2 \left( 1 + \Psi (a^2 e^{r(T-t)} \sigma_j^2 S_j^2 \frac{\partial^2 V}{\partial S_j^2}) S_i^2 \frac{\partial^2 V}{\partial S_i^2} + rS_i \frac{\partial V}{\partial S_i} + rS_{i,\text{max}} \frac{\partial V}{\partial S_i} - rV \right) = 0,
\]

for \(j \in \{1, 2\} \setminus i\). Finally, for the far field boundaries \(S_1 = S_{1\text{max}}\) and \(S_2 = S_{2\text{max}}\) we use the linearity condition,

\[
\frac{\partial^2 V}{\partial S_1^2} (S_{1\text{max}}, S_{2\text{max}}, t) = \frac{\partial^2 V}{\partial S_2^2} (S_{1\text{max}}, S_{2\text{max}}, t) = 0.
\]
### 3.2 The numerical scheme

In the following, we adopt the two dimensional Crank-Nicolson finite difference scheme for Eq. (2.3) with nine point stencil. We use a variable transformation in time in order to have a forward parabolic equation, then Eq. (2.3) becomes

$$
V_i - \frac{1}{2} \left[ 1 + \Psi \left( a^2 e^{rt} \left( \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial^2 x} + \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial^2 y} + 2 \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right) \right) \right] = \left( \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial^2 x} + \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial^2 y} + 2 \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right) - \frac{2}{N^2} \sum_{i=1}^{N} \frac{\partial V}{\partial S_i} + rV = 0,
$$

(3.6)

For the sake of simplicity we set $x = S_1$ and $y = S_2$. Equation (3.6) becomes

$$
V_i - \frac{1}{2} \left[ 1 + \Psi \left( a^2 e^{rt} \left( \sigma_1^2 x^2 \frac{\partial^2 V}{\partial^2 x^2} + \sigma_2^2 y^2 \frac{\partial^2 V}{\partial^2 y^2} + 2 \rho \sigma_1 \sigma_2 x y \frac{\partial^2 V}{\partial x \partial y} \right) \right) \right] = \left( \sigma_1^2 x^2 \frac{\partial^2 V}{\partial^2 x^2} + \sigma_2^2 y^2 \frac{\partial^2 V}{\partial^2 y^2} + 2 \rho \sigma_1 \sigma_2 x y \frac{\partial^2 V}{\partial x \partial y} \right) - r x \frac{\partial V}{\partial x} - r y \frac{\partial V}{\partial y} + rV = 0,
$$

(3.7)

$$
V(x,y,0) = \max \left( \frac{1}{2}(x+y) - K, 0 \right).
$$

We uniformly discretize the time interval $[0,T]$ by $N+1$ points and the space domain $\Omega = [0,S_{1\text{max}}] \times [0,S_{2\text{max}}]$ by $N_1+1$ and $N_2+1$ points respectively. The discrete solution will be denoted by $V^n_{ij} = V(x_i,y_j,n\Delta t)$ where $\Delta t = t^n + 1 - t^n$, $n \in \{0,\ldots,N\}$, $0 \leq i \leq N_1$ and $0 \leq j \leq N_2$. We denote by

$$
D_{xx} V^n_{ij} = \frac{V^n_{i+1,j} - 2V^n_{ij} + V^n_{i-1,j}}{\Delta x^2},
$$

$$
D_{yy} V^n_{ij} = \frac{V^n_{i,j+1} - 2V^n_{ij} + V^n_{i,j-1}}{\Delta y^2}, \quad D_{x} V^n_{ij} = \frac{V^n_{i+1,j} - V^n_{i-1,j}}{2\Delta x},
$$

$$
D_{xy} V^n_{ij} = \frac{V^n_{i+1,j+1} - V^n_{i+1,j-1} - V^n_{i-1,j+1} - V^n_{i-1,j-1}}{4\Delta x \Delta y}, \quad D_{y} V^n_{ij} = \frac{V^n_{i+1,j} - V^n_{i,j-1}}{2\Delta y}.
$$

**Remark 3.1.** One can notice that since we deal with nonsmooth initial data some inaccuracies may occur when approximating the derivatives using finite difference method. The accuracy can be improved using the grid stretching technique presented in [23]. For the sake of simplicity of exposure, we do not use this technique and we restrict ourselves to uniform grids.
We set
\[
\Delta \rho V_{ij}^n = \sigma_1^2 x_i^2 D_{xx} V_{ij}^n + \sigma_2^2 y_j^2 D_{yy} V_{ij}^n + 2 \rho \sigma_1 \sigma_2 x_i y_j D_{xy} V_{ij}^n.
\]

Assuming that \( V_{ij}^n \) is known, we discretize our Equation (3.7) explicitly for the nonlinear term (See [13]), and with a Crank-Nicolson method for the other terms

\[
\frac{V_{ij}^{n+1} - V_{ij}^n}{\Delta t} = -\frac{1}{4} \left( 1 + \Psi (a^2 e^{rn \Delta t} \Delta \rho V_{ij}^n) \right) (\Delta \rho V_{ij}^{n+1} + \Delta \rho V_{ij}^n)
- \frac{r x}{2} (D_x V_{ij}^{n+1} + D_x V_{ij}^n) - \frac{r y}{2} (D_y V_{ij}^{n+1} + D_y V_{ij}^n) + \frac{r}{2} (V_{ij}^{n+1} + V_{ij}^n) = 0. \tag{3.8}
\]

4 The multigrid method

In this section, we introduce the concepts of the multigrid method established in [8]. The idea of this method is based on the fact that, low frequencies of components of the error on a fine grid become high frequencies on a coarser grid, therefore, we can accelerate the convergence of the solution of a fine grid problem by computing corrections on a coarser grid and then interpolating them back to the fine grid problem. In the following, we consider uniform two dimensional grids.

There are different multigrid approaches. The most known are the V-cycle, W-cycle, and full multigrid cycle [26]. In our numerical simulations, we use the V-cycle method, which consists on recursively solving a system of equations on a sequence of grids of different levels from finest to coarsest. Each iteration transfers information through a series of restrictions (from a fine to a coarser grid) and prolongations (from a coarse to a finer grid).

![Multigrid sketch on uniform 2D meshes.](image)
4.1 Prolongation and restriction in two dimensions

4.1.1 Prolongation

We use a bilinear interpolation in order to have the solution values on the fine grid. We denote by \(i, j\) the position indices in the coarse grid. The bilinear interpolation is given vertically by

\[
\begin{align*}
    u_{2i,2j}^f &= u_{i,j}^c, \\
    u_{2i,2j+1}^f &= \frac{1}{2} \left( u_{i,j}^c + u_{i,j+1}^c \right),
\end{align*}
\]

(4.1)

and horizontally,

\[
\begin{align*}
    u_{2i+1,2j}^f &= \frac{1}{2} \left( u_{i,j}^c + u_{i+1,j}^c \right), \\
    u_{2i+1,2j+1}^f &= \frac{1}{4} \left( u_{i,j}^c + u_{i,j+1}^c + u_{i+1,j}^c + u_{i+1,j+1}^c \right),
\end{align*}
\]

(4.2)

where \(u^c\) and \(u^f\) are the solutions on the coarse grid and the fine grid respectively.

4.1.2 Restriction

We use a combination of restriction using only the identity in some points of the grid and the full weighting method [9]. For example, we can choose our restriction operator vertically as the following

\[
    u_{i,j}^c = \frac{1}{8} u_{2i,2j}^f + \frac{1}{4} u_{i,2j}^f + \frac{1}{2} u_{i,2j+1}^f,
\]

(4.3)

and horizontally,

\[
    u_{i,j}^c = \frac{1}{4} u_{2i,j}^f + \frac{1}{2} u_{i,j}^f + \frac{1}{4} u_{2i+1,j}^f,
\]

(4.4)

for interior points,

\[
    u_{i,j}^c = \frac{1}{2} u_{2i,2j}^f + \frac{1}{8} \left( u_{2i,2j-1}^f + u_{2i,2j+1}^f + u_{2i-1,2j}^f + u_{2i+1,2j}^f \right).
\]

(4.5)

Without loss of generality we use the identity operator when restricting points with dirichlet condition

\[
    u^c = u^f \quad \text{if} \quad \begin{cases} 
        S_1 = S_{1\text{max}} \quad \text{and} \quad S_2 = 0, \\
        S_2 = S_{2\text{max}} \quad \text{and} \quad S_1 = 0, \\
        S_1 = S_2 = 0. 
    \end{cases}
\]

(4.6)
4.2 The full approximation storage algorithm

An effective approach of the multigrid method to nonlinear problems is the FAS algorithm which is well established as a fast solver for nonlinear PDEs. In fact FAS algorithm is a generalization of the coarse-grid correction principle to nonlinear problems [17]. Let \((Ω_k)\) \(1 ≤ k ≤ n\) be a sequence of \(n\) grids \(Ω_k\) of the domain \(Ω\) where \(Ω_n\) is the finest grid and \(Ω_1\) is the coarsest grid. Suppose that on every grid we have an iterative method \(G_k\) for solving the problem \(A_k(U_k) = f_k\), where \(A_k\) is the nonlinear operator. We denote \(G_k^{ν_k}(U_k, f_k)\) the final result after \(ν_k\) iterations. The FAS algorithm is described as follows

**Algorithm 4.1.** Full Approximation Storage.

**Step 1: Pre-smoothing.**
Perform nonlinear relaxation on the fine grid problem with \(ν_n\) iterations to obtain \(U_n, k:= n\).

**Step 2: Coarse grid correction**
(a) Compute \(U_k = G_k^{ν_k}(U_k, f_k)\) with \(ν_k\) iterations on the grid \(Ω_k\).
(b) Restrict \(U_k\) and the residual \(f_k - A_k(U_k)\).
(c) Compute the right hand side \(f_{k-1} = R_k^{k-1}(f_k - A_k(U_k)) + A_{k-1}(R_k^{k-1}(U_k))\) and set \(U_{k-1} = R_k^{k-1}(U_k)\), (where \(R_k^{k-1}\) is the restriction operator).
\(k:= k - 1\).
If \(k > 1\) we go to (a), otherwise we are on the coarsest grid and we compute \(U_1 = G_k^{ν_1}(U_1, f_1)\) with \(ν_1\) iterations.

(d) Compute the correction \(e_k = U_k - R_k^{k+1}(U_{k+1})\).
(e) Correction of the solution with the prolongation of \(e_k\): \(U_{k+1} = U_{k+1} + P_k^{k+1}(e_k)\), (where \(P_k^{k+1}\) is the prolongation operator).
\(k:= k + 1\).

(f) Compute \(U_k = G_k^{µ_k}(U_k, f_k)\) with \(µ_k\) iterations on \(Ω_k\).
If \(k < n\) we go to (d), otherwise we are on the finest grid.

**Step 3: Post-smoothing**
Perform \(µ_n\) post-smoothing iterations and perform a convergence test.

It is worth noting that FAS reduces to the multigrid algorithm when \(A_k\) is a linear operator [8, 17]. However, FAS requires storing the values of the solutions over the entire grids.
4.2.1 Application to the Barles’ and Soner’s model

**Algorithm 4.2.** Computation of a European basket option for the Barles’ and Soner’s model using FAS.

**Input data:** $S_{1\text{max}}, S_{2\text{max}}, \sigma_1, \sigma_2, K, r, \rho, a, T$.
- Solve the ODE (1.3) using an explicit fourth order Runge-Kutta method.
- Interpolate the solution of (1.3) using a cubic spline method.

**For n=0...T:**
- Initialise $V^n$ and compute the nonlinear term.
- Compute $V^{n+1}$ using FAS algorithm 4.1.
- Copy $V^{n+1}$ in $V^n$.

end for

5 Numerical results

5.1 The multigrid cycle

We begin our simulations with a demonstration of the effectiveness of the multigrid method. All the numerical experiments are performed on uniform grids. For the computations, we used the following parameters $S_{1\text{max}} = 300$, $S_{2\text{max}} = 300$, $T = 1$ (one year), $a = 0.02$, $K = 100$, $\sigma_1 = 0.2$, $\sigma_2 = 0.2$, $\rho = 0.5$, $r = 0.1$. We used 3 levels of grids with $\nu = 2$ for pre-smoothing and $\nu = 3$ for coarse grid corrections.

Figure 4 represents a rate of convergence comparison for both the multigrid and monogrid (standard Gauss-Seidel method on fine grids) in the linear case, Figure 5 and 6 represent the rate of convergence comparison for multigrid and monogrid when treating $\Psi$ as the identity and in the case of Barles’ and Soner’s model respectively. We have compared the total CPU time elapsed using the multigrid approach on uniform meshes and monogrid. All the numerical experiments were implemented in Mac OS X, 2.7GHz Intel core i5 processor.

In Table 1 we compute the total CPU time elapsed for the two methods (monogrid and multigrid). The multigrid approach becomes significantly faster than monogrid (s-

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Nonlinear case ($\Psi (x) = x$)</th>
<th>Nonlinear case (Barles and Soner)</th>
<th>Linear case ($a = 0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128×128</td>
<td>Multigrid 0.8s 9.52s</td>
<td>Multigrid 0.8s 9.6s</td>
<td>Multigrid 0.8s 9.36s</td>
</tr>
<tr>
<td>256×256</td>
<td>4.3s 2.41min</td>
<td>5.0s 2.1min</td>
<td>3.1s 1.9min</td>
</tr>
<tr>
<td>512×512</td>
<td>25.5s 19.21min</td>
<td>31s 20.6min</td>
<td>14s 15min</td>
</tr>
<tr>
<td>1024×1024</td>
<td>2min10s 2h14min</td>
<td>2min30s 2h28min</td>
<td>1.62min 1h40min</td>
</tr>
</tbody>
</table>
Figure 4: A comparison of rate of convergence for multigrid and monogrid methods in the linear case on different mesh sizes (128², 256², and 512²) for 3 levels of grids.

Figure 5: A comparison of rate of convergence for multigrid and monogrid when Ψ(x) = x on different mesh sizes (128², 256², and 512²) for 3 levels of grids.

standard Gauss-Seidel method) for higher mesh resolutions. In the case of meshes having a discretisation sizes 512² and 1024², we found that our implementation of the multigrid solver (Barles’ and Soner’s equation) becomes 40 and 59 faster than a standard Gauss-Seidel method respectively.

5.2 Back to the Barles’ and Soner’s 2D model: the impact of transaction costs

We end our numerical experiments by demonstrating the effect of the transaction cost parameter a on the option price. In Figure 7 we illustrate the influence of transaction costs on the numerical solution, by plotting the difference between the solution obtained with
Figure 6: A comparison of rate of convergence for the multigrid and monogrid in the case of Barles’ and Soner’s model on different mesh sizes \(128^2, 256^2\) and \(512^2\) for 3 levels of grids.

Figure 7: Comparison of the difference between the linear and the nonlinear case for different transaction costs parameters \(a\).

The underlying asset

![](image)

The difference between the nonlinear and the linear case

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a=0.02)</td>
<td>2.5</td>
</tr>
<tr>
<td>(a=0.05)</td>
<td>2.2</td>
</tr>
<tr>
<td>(a=0.2)</td>
<td>1.8</td>
</tr>
<tr>
<td>(a=0.4)</td>
<td>1.5</td>
</tr>
</tbody>
</table>

In what follows, we analyze different aspects of the proposed transaction cost model.
Table 2: The chosen parameters for the numerical tests.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_1, \sigma_2$</td>
<td>0.3</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.5</td>
<td>-0.6</td>
<td>0.2</td>
</tr>
<tr>
<td>$T$</td>
<td>one year</td>
<td>one year</td>
<td>one year</td>
</tr>
<tr>
<td>$r$</td>
<td>0.05</td>
<td>0.02</td>
<td>0.1</td>
</tr>
<tr>
<td>$a$</td>
<td>0.02</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>100</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>$dS_1, dS_2$</td>
<td>2.3</td>
<td>2.3</td>
<td>2.3</td>
</tr>
<tr>
<td>$dt$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
</tr>
</tbody>
</table>

(two dimensional model of Barles and Soner). For numerical tests we used the following parameters.

We denote by $V_{TC}$, the option price using the Barles’ and Soner’s model, $V_{Linear}$ the option price using the linear Black-Scholes model and $\Gamma = \Delta V_{TC}$, the Gamma of the option that is a financial measure used by traders for hedging tasks. The Gamma represents the convexity of the option price with respect to the underlying asset, the higher it is, the greater the convexity and the price of the option varies rapidly in value.

Figures 8, 9 and 10 illustrate the impact of transaction costs and the option price at $t = 0$. It should be noted that transaction costs are influenced by several factors, including the volatility of each underlying and the second derivatives of the assets. In Figure 8 (Test 1), we observe that the maximum of these transaction costs is reached at the neighborhood of $2K$, this is justified by the behaviour of Gamma (The ‘Gamma’ is maximum for values close to $2K$). Moreover, the volatility effect on the transaction costs is observed in Figures 8 and 9. The ‘Gamma’ has an opposite behaviour with transaction costs, as it tends to decrease when the volatility increases. Finally, in Figure 10 (Test 3), we notice a symmetry in (10(a) - 10(c)) due to the fact that the two assets have the same volatility, again, the maximum of the transaction costs is reached when the two assets are close to $2K$ and these costs are canceled when $S_1$ and $S_2$ are far from $2K$.

6 Conclusion

The major contribution of this work focuses on two aspects, a deterministic extension of the Barles’ and Soner’s model for two assets and a numerical investigation of the effect of the Multigrid method. The obtained results confirmed the relevance of the extension proposed. In fact, we have shown numerically that these costs depend primarily on asset prices, the magnitude of the second derivatives (i.e. the ‘Gamma’) and the volatility of each asset. We also examined the effectiveness of the multigrid method when dealing with a two dimensional problem, we showed that it gives better results in terms of the execution time in contrast to the standard iterative method on a fine grid.
(a) The influence of transaction costs at $t=0$.

(b) The option price at $t=0$.

(c) The 'Gamma' of the option at $t=0$.

Figure 8: Test 1: Asset $S_1$ is more volatile than $S_2$. 
(a) The influence of transaction costs at $t=0$.

(b) The option price at $t=0$.

(c) The 'Gamma' of the option at $t=0$.

Figure 9: Test 2: Assets $S_1$ and $S_2$ are low volatiles.
Figure 10: Test 3: Assets $S_1$ and $S_2$ having the same volatility.
Acknowledgments

This work was supported by the ARC-Mathematics research federation FR3399. We would like to thank Olivier Goubet, Professor at university of Picardie Jules Verne and Mostafa Abounouh, Professor at Cadi Ayyad university for their insights and expertise that substantially improved the manuscript. We also would like to thank the anonymous reviewers for their precious advices and comments.

References


