Abstract

Pricing options on multiple underlying or on an underlying modeled with stochastic volatility may involve solving multi-dimensional Black-Scholes like partial differential equations (PDE). Computing several such options at once for various moneyness levels can be a numerical challenge. We investigate here the Kolmogorov equation and Dupire or “pre-Dupire” equations to solve the problem faster and we validate the approach numerically. The heart of the method is to use the adjoint of the PDE of the option at the discrete level and to use discrete duality identities to obtain Dupire-like relations. The method works on most linear models. Numerical results are given for a European call option on a basket of two assets.

1 Introduction

In back offices in bank it is often the case that one has to predict numerically the values of many financial derivatives which differ only by their maturity and/or pay-off.

For a simple derivative $U$, like European puts and calls on a single asset $S$ modeled by the Black-Scholes equation [2] (see also [3]), great numerical speed up can be obtained from Dupire’s observation that $U$ satisfies also a partial differential equation where the strike $K$ and the maturity $T$ are the variables, the so called Dupire equation [3].

There are several ways to derive Dupire’s equation, one using the Kolmogorov equation for the probability density of the underlying asset. This equation is also closely connected to the adjoint equation of Black-Scholes’
partial differential equation, as explained in [6], an observation which can perhaps push the method in fields where the Kolmogorov equation is harder to derive. However to be used numerically the Kolmogorov equation has to be integrated with a Dirac singularity as initial condition and it remains unclear whether numerical methods can handle that with the 0.1% precision required in finance.

After setting up the problem and recalling Kolmogorov’s equation and Dupire’s in the case of a basket option the paper investigates a numerical solution of the Kolmogorov approach. It is discretized by a finite element method with automatic mesh adaptation. The results are accurate though the computing time is still too large for the method to be used on a regular basis.

In many cases the Kolmogorov equation can be integrated into what could be called a pre-Dupire equation and, at the end, the paper presents such a case and reports on the numerical precision of the approach. We shall show that best is to use the discrete version of Kolmogorov’s equation which is adapted to the discretized Black-Scholes PDE; such a strategy is straightforward in the context of calculus of variation and adjoint operators and more difficult to interpret in the stochastic framework.

Let us begin by recalling for a simple call option the Black-Scholes model, the Kolmogorov equation of the process and Dupire’s equation.

If $S_t$ is the value of the underlying asset to the derivative $U$ which yields $(S_T - K)^+$ at maturity $T$, we have, according to Black and Scholes

$$dS_t = S_t(rdt + \sigma dW_t), \quad S_0 \text{ known}$$

$$U(t) = e^{-r(T-t)}\mathbb{E}(S_T - K)^+ \quad (1)$$

where $r$ is the current interest rate, $\sigma$ the volatility of $S$, $W$ a Gaussian random process and $\mathbb{E}$ the expected value.

Ito’s calculus allows $U$ to be computed by $U(t) = e^{-r(T-t)}u(S_t, t)$ where $u$ is the unique solution of

$$\frac{\partial u}{\partial t} + \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial x^2} + rx \frac{\partial u}{\partial x} = 0 \text{ in } \mathbb{R}^+ \times (0, T), \quad u(x, T) = (x - K)^+ \quad (2)$$

It can also be computed by

$$u(x, 0) = \int_{S_0}^\infty p(s, T) (s - K)^+ \, ds \quad (3)$$

where the probability density $p$ is the solution of the Kolmogorov equation

$$\frac{\partial p}{\partial t} - \frac{\sigma^2}{2} \frac{\partial^2 p}{\partial x^2} + xp(r)p = 0 \text{ in } \mathbb{R}^+ \times (0, T), \quad p(x, 0) = \delta(S_0 - x) \quad (4)$$
Finally one notices that if $v$ is a double primitive of $p$ in the sense that
\[ \partial_{xx}p = v \]
and a double integration by parts
\[ u(x,0) = \int_0^\infty \partial_{xx}v(s,T)(s-K)^+\,ds = \int_0^\infty v(s,T)\partial_{xx}(s-K)^+\,ds = v(K,T) \quad (5) \]
because $\partial_{xx}(s-K)^+$ is a Dirac mass at $s = K$ (see [3] for more details). As $v$ satisfies (4) integrated twice we come to the Dupire equation:
\[ \partial_t v - \frac{\sigma^2 x^2}{2} \partial_{xx}v - \partial_x(rxv) = 0 \quad \text{in } \mathbb{R}^+ \times (0,T), \quad v(x,0) = (x-S_0)^+ \quad (6) \]

On (6) it is clear that several calls can be computed by (5) while solving (6) once only. It is also true if (3) is used the numerical work is harder because the initial condition in (4) is numerically stiff. Note by the way that Dupire’s equation relies on the possibility of integrating by hand (4) twice.

2 Two Dimensional Problems

What follows works not only for basket options but also for stochastic volatility models which lead to multidimensional parabolic partial differential equations. For simplicity the method is explained on a basket option with two assets.

Consider two underlying assets $S_i, i = 1, 2$ with mean tendency $\mu_i$, volatility $\sigma_i$ and correlation $\rho$:
\[
\begin{align*}
    dS_1 &= \mu_1 S_1 \, dt + \sigma_1 S_1 \, dW_1 \\
    dS_2 &= \mu_2 S_2 \, dt + \sigma_2 S_2 \, dW_2 \\
    \rho \, dt &= d\langle W_1, W_2 \rangle
\end{align*}
\]

An option $U$ with pay-off $U_T(S_1, S_2)$ at maturity $T$ will satisfy the Black-Scholes partial differential equation in $Q = \mathbb{R}_+^2 \times [0,T]$:
\[
\begin{cases}
    \partial_t U + \sum_{i=1,2} (\mu_i S_i \partial_{S_i} U + \frac{1}{2} (\sigma_i S_i)^2 \partial_{S_i S_i} U) + \rho \sigma_1 S_1 \sigma_2 S_2 \partial_{S_1 S_2} U = 0 \\
    U(S_1, S_2, T) = U_T(S_1, S_2)
\end{cases} \quad (7)
\]

Notice that the price of the option still needs to be discounted at interest rate $r$ after solving (7).

Let $\Omega = \mathbb{R}_+^2$; let $(U, V)$ denote the $L^2(\Omega)$ scalar product of $U, V$ (i.e. the
integral of $UV$ on $\Omega$) then, in variational form, (7) is:

Find $U \in L^2(0, T; H^1(\Omega))$, such that

$$
(\partial_t U, V) - \sum_{i=1,2} \left( \frac{1}{2} (\partial_{S_i} (V \sigma_i^2 S_i^2), \partial_{S_i} U) - (\mu_i S_i \partial_{S_i} U, V) \right) \\
- (\partial_{S_1} (\rho \sigma_1 \sigma_2 S_2 V), \partial_{S_1} U) = 0, \quad \forall V \in H^1(\Omega), \quad \forall t \in (0, T)
$$

(8)

when $U(T) \in H^1(\Omega)$ is given. Equivalence between (8) and (7) is proved by applying Green’s formula to (7) multiplied by $V$ and integrated over $\Omega$ (see [1]).

**Proposition 1** Let $P$ be the solution to the adjoint equation for a given $P_0$:

$$
\begin{align*}
\partial_t P + \sum_{i=1,2} \left( \partial_{S_i} [\mu_i S_i P] - \frac{1}{2} \partial_{S_i} S_i [((\sigma_i S_i)^2 P)] - \rho \partial_{S_1 S_2} [\sigma_1 S_1 \sigma_2 S_2 P] \right) &= 0 \\
P(S_1, S_2, 0) &= P_0(S_1, S_2)
\end{align*}
$$

(9)

Then we have the duality relation, for any $t_0, t_1 \in [0, T]$:

$$
\int_{\mathbb{R}^2_+} P(S_1, S_2, t_0) U(S_1, S_2, t_0) = \int_{\mathbb{R}^2_+} P(S_1, S_2, t_1) U(S_1, S_2, t_1)
$$

(10)

**Proof**

Choosing $V = P$ in (8) with integration by part over $\Omega$ leads to

$$
(\partial_t U, P) + \sum_{i=1,2} \left( (U, \partial_{S_i} [\mu_i S_i P]) - \frac{1}{2} (U, \partial_{S_i} S_i [((\sigma_i S_i)^2 P)]) \right) - (U, \rho \partial_{S_1 S_2} [\sigma_1 S_1 \sigma_2 S_2 P]) = 0
$$

(11)

Now integrate over the time interval $(t_0, t_1)$:

$$
(U, P)|^{t_1}_{t_0} - \int_{t_0}^{t_1} (U, \partial_t P) \\
+ \int_{t_0}^{t_1} \left[ \sum_{i=1,2} \left( (U, \partial_{S_i} [\mu_i S_i P]) - \frac{1}{2} (U, \partial_{S_i} S_i [((\sigma_i S_i)^2 P)]) \right) \\
- (U, \rho \partial_{S_1 S_2} [\sigma_1 S_1 \sigma_2 S_2 P]) \right] = 0
$$

(12)

Everything vanishes due to (9) but the first term. ⋄

By choosing $t_0 = 0$ and

$$
P(S_1, S_2, 0) = \delta(S_1 - x_1) \delta(S_2 - x_2)
$$

(13)

one obtains a formula for $U$ at initial time in terms of $U$ at $t_1$

$$
U(x_1, x_2, 0) = \int_{\mathbb{R}^2_+} P(S_1, S_2, t_1) U(S_1, S_2, t_1)
$$

(14)
This indicates that $P$ can be interpreted as the probability density of $U$ at $t_1$.

With a maturity $T$ the formula gives a method to compute all European basket options based on $S_1, S_2$ with a single numerical integration of (9), the PDE for $P$, in the sense that

**Corollary 1** Equation (9) can be integrated once only with initial data (13) and independently of $U_T$ and

$$U(x_1, x_2, 0) = \int_{\mathbb{R}^2_+} P(S_1, S_2, T) U_T(S_1, S_2) \, dS_1 dS_2$$  \hspace{1cm} (15)

However the price to pay is the singularity of the initial condition (14).

### 2.1 Lognormal Approximation of the Dirac Masses

A Dirac mass at $z$ can be approximated by a lognormal density function in the sense that for all smooth function $w$

$$\lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi} \sigma} \int_{\mathbb{R}} \frac{1}{\sqrt{\sigma}} e^{-\frac{1}{2} \left[ \frac{\ln(S/S_0) - (\mu - \sigma^2 / 2) \sigma^2}{\sigma} \right]^2} w(S) \, dS = \int_{\mathbb{R}} \delta(S-S_0) w(S) \, dS = w(S_0)$$

In dimension $d$, a Dirac mass at point $z \in \mathbb{R}^d$ is the limit of

$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (\tilde{S} - M)^T \Sigma^{-1} (\tilde{S} - M) \right)$$

for any $\Sigma \in \mathbb{R}^{d \times d}$ positive definite and tending to zero with $\tilde{S}_i = \ln(S_i)$. This approximation has a probabilistic interpretation: $f$ is the density of a $d$-dimensional exponential Brownian process $S$ of mean $M$ and covariance matrix $(\Sigma)_{j,l}$.

In two dimensions we may rename the process $\{S_1, S_2\}$ and introduce

$$Z_{S_1} = \mathbb{E} S_1, \quad Z_{S_2} = \mathbb{E} S_2, \quad \sigma_{S_1} = \sqrt{\mathbb{E}|S_1 - Z_{S_1}|^2}, \quad \sigma_{S_2} = \sqrt{\mathbb{E}|S_2 - Z_{S_2}|^2}$$

$$\rho = \frac{\text{Cov}(S_1, S_2)}{\sigma_{S_1} \sigma_{S_2}} = \frac{\mathbb{E}((S_1 - Z_{S_1})(S_2 - Z_{S_2}))}{\sigma_{S_1} \sigma_{S_2}}$$ \hspace{1cm} (16)

Then we note $X_1 = \frac{\ln(S_1) - Z_{S_1}}{\sigma_{S_1}}$ et $X_2 = \frac{\ln(S_2) - Z_{S_2}}{\sigma_{S_2}}$

$$f(S_1, S_2) = \frac{1}{2\pi \sqrt{1 - \rho^2} \sigma_{S_1} \sigma_{S_2}} \exp \left( -\frac{X_1^2 + X_2^2 - 2\rho X_1 X_2}{2(1 - \rho^2)} \right)$$

Furthermore if $\{S_1, S_2\}$ is generated by

$$dS_i = \mu_i S_i dt + \sigma_i S_i dW^i \quad i = 1, 2$$ \hspace{1cm} (17)
where $W^i$, $i = 1, 2$ are Brownian motions with variance $\sigma_i^2$ and correlation $d\langle W^0, W^1 \rangle = \rho dt$, then

$$
\sigma_{S_1} = \sigma_1 \sqrt{t} \quad Z_{S_1} = \ln (K_1) + \left( \mu_1 - \frac{\sigma_1^2}{2} \right) t
$$

$$
\sigma_{S_2} = \sigma_2 \sqrt{t} \quad Z_{S_2} = \ln (K_2) + \left( \mu_2 - \frac{\sigma_2^2}{2} \right) t
$$

(18)

and in that case

$$
X_1(t) = \frac{\ln(S_1/K_1) - \left( \mu_1 - \frac{\sigma_1^2}{2} \right) t}{\sigma_1 \sqrt{t}} \text{ et } X_2(t) = \frac{\ln(S_2/K_2) - \left( \mu_2 - \frac{\sigma_2^2}{2} \right) t}{\sigma_2 \sqrt{t}}
$$

$$
f(S_1, S_2, t) = \frac{1}{2\pi \sigma_1 \sigma_2 t S_1 S_2 \sqrt{1 - \rho^2}} \exp \left( -\frac{X_1(t)^2 + X_2(t)^2 - 2\rho X_1(t)X_2(t)}{2(1 - \rho^2)} \right)
$$

(19)

As $t \to 0$ this formula approximates the bidimensional Dirac mass at 0 which is coherent in scale with the underlying asset of the basket option. It is also the solution of (9) with such approximate initial condition when the coefficients are constant.

**Proposition 2** If $\rho$, $\sigma_1$, $\sigma_2$, $\mu_1$, $\mu_2$ are constant, the solution of (9) at $t_0$ with $P_0$ given by (13) is equal to $f$ in (19).

### 2.2 Mesh Adaptation

Solving a PDE with a Dirac mass as initial condition requires a nonuniform mesh adapted near the singularity as illustrated by Figure 2.2. The difficulty relies on that, the singularity of $P$ at times zero induces very large derivatives on $P$ at later times in a region that moves with time. Mesh adaptivity has to be done at regular time intervals.

In [5] George, Hecht and Saltel noticed that Delaunay triangulations can be built with respect to any metric. Taking the Hessian of a convex function $f$ yields to a good way to compute one. The resulting Delaunay triangulation is adapted to the derivatives of $f$.

If the function is not convex one simply takes the absolute value of the Hessian. If $Q$ is a unitary matrix, $\Lambda$ a diagonal matrix and $H = Q^T \Lambda Q$, then $|H| := Q^T |\Lambda| Q$

These ideas are implemented in *freefem++* [3]; from the user’s point of one starts with a reasonable mesh, compute $f$ at the vertices then the software produces $|H|$, an approximation of the absolute value of the Hessian of $f$, which is used to build a new mesh by the Delaunay-Voronoi tesselation algorithm based on the metric defined by $|H|$. The process can be iterated.
Figure 1: Interpolation of a Gaussian function centered at (100,100) with variance 0.002 on the domain $[73,130] \times [68,160]$. Top left with a uniform mesh $21 \times 21$, the sum of the function on the domain is 0.7539. Top right: same with a mesh $41 \times 41$; the sum is then 1.00052. Bottom left: the same with an mesh $61 \times 61$ the error on the integral (whose exact value is 1) is $10 \ e^{-5}$; finally (bottom right) with an adapted mesh around the integral error is $10 \ e^{-8}$. 
### Data

<table>
<thead>
<tr>
<th>Data</th>
<th>$S_0$</th>
<th>$S_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest rate</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>Volatility</td>
<td>20%</td>
<td>30%</td>
</tr>
<tr>
<td>Correlation</td>
<td>−60%</td>
<td>−60%</td>
</tr>
<tr>
<td>Spot</td>
<td>98</td>
<td>105</td>
</tr>
<tr>
<td>Strike</td>
<td>130</td>
<td>110</td>
</tr>
<tr>
<td>points number</td>
<td>201</td>
<td>201</td>
</tr>
</tbody>
</table>

Table 1: Data range for the 2 years maturity options

### 2.3 Numerical Tests

We have used the finite element method of degree one on triangles to solve (9), as implemented in freefem++, with initial condition given by (19) at $t_0 = 0.1$ day and K=180. In Table 2 the numerical results of (14) are compared with analytical solutions given by extended Black-Scholes formulæ for a variety of pay-off $(\mathbb{E}[S_i], \mathbb{E}[S_i^2], \text{etc. } i = 1, 2)$. Table 2 compares both. That work for adapting

<table>
<thead>
<tr>
<th>Computed</th>
<th>Kolmogorov</th>
<th>Analytical</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{E}[S_1]$</td>
<td>98.0006</td>
<td>98</td>
<td>0.00058%</td>
</tr>
<tr>
<td>$\mathbb{E}[S_1^2]$</td>
<td>11496.5</td>
<td>11499.5</td>
<td>0.02579%</td>
</tr>
<tr>
<td>$\mathbb{E}[S_2]$</td>
<td>105</td>
<td>105</td>
<td>0.00007%</td>
</tr>
<tr>
<td>$\mathbb{E}[S_2^2]$</td>
<td>14583.7</td>
<td>14590.3</td>
<td>0.01021%</td>
</tr>
<tr>
<td>$\mathbb{E}[S_1 S_2]$</td>
<td>10581.3</td>
<td>10582.4</td>
<td>0.01021%</td>
</tr>
<tr>
<td>$\mathbb{E}[(S_2 - K_1)^+]$</td>
<td>13.0321</td>
<td>13.0326</td>
<td>0.00384%</td>
</tr>
<tr>
<td>$\mathbb{E}[(K_2 - S_1)^+]$</td>
<td>11.8914</td>
<td>11.8903</td>
<td>0.00925%</td>
</tr>
<tr>
<td>$\mathbb{E}[(S_1 + S_2 - K)^+]$</td>
<td>41.6291</td>
<td></td>
<td>%</td>
</tr>
<tr>
<td>$\mathbb{E}[(K_1 - S_1)^+ (K_2 - S_2)^+]$</td>
<td>190.284</td>
<td>190.24996</td>
<td>0.01789%</td>
</tr>
</tbody>
</table>

Table 2: Numerical results of (14) are compared with analytical solutions of the Black-Scholes equation (7).

...the mesh needs to be iterated many times to finally give a good accuracy. This method is so time expensive.

### 3 Discrete Adjoint

We now propose a different implementation which reduces the computing time of the previous approach. In view of cost of the previous approach we propose another implementation of the same idea.
Figure 2: Solution of the 2D Black-Scholes model for a basket put.
Left: Initial solution: The bilognormal function which approximates the Diracs: its integral is 1.00005. Right: The probability density of the final solution; its integral is 1.00003

The first order triangular Finite element method replaces $H^1(\Omega)$ by $H_h$ in \[[5]\], the space of piecewise linear functions on a triangulation of $\Omega_h = (0, L) \times (0, L)$, continuous and vanishing at $S_i = L$ (see \[[1]\] for more details).

With self explanatory notations the semi-discrete problem is of the form:
Find $U_h \in H_h$ such that
\[
(\partial_t U_h, V_h) - a(U_h, V_h) = 0 \quad \forall t \in (0, T), \quad U_h(T) = U_{T_h}, \quad \forall V_h \in H_h \tag{20}
\]
where
\[
a(U, V) = \sum_{i=1,2} \left( \frac{1}{2} \left( \partial S_i (V \sigma_i^2 S_i^2), \partial S_i U \right) - (\mu_i S_i \partial S_i U, V) \right) - \left( \partial S_i (\rho \sigma_1 \sigma_2 S_i S_j), \partial S_i U \right) \tag{21}
\]
A simple basis for $H_h$ is the so called “hat functions” $W^i$ defined by
\[
W^i(q_1, q_2) = \delta_{ij}, \quad W^i \in H_h
\]
i.e. $W^i$ is piecewise linear and takes value 1 at vertex $q_i$ and zero at all other vertices of the triangulation of $\Omega_h$.

Let $P_h \in H_h$ be the solution to the semi-discrete adjoint equation
\[
(\partial_t P_h, W_h) + a(W_h, P_h) = 0 \quad \forall t \in (0, T), \quad P_h(0) = P_0 \quad \forall W_h \in H_h \tag{22}
\]
By taking $W_h = U_h$ in \[[22]\] and $V_h = P_h$ in \[[20]\] we find that
\[
(\partial_t U_h, P_h) + (\partial_t P_h, U_h) = 0 \tag{23}
\]
Therefore
\[
\int_{\Omega_h} U_h(T) P_h(T) = \int_{\Omega_h} U_h(0) P_h(0) \tag{24}
\]
Take \( P(0) = W^j \). If \( U(0) \) is smooth in the sense that it does not vary too much on \( D_j \), the support of that hat function, then (24) gives

\[
U(S_1, S_2, 0) \approx \frac{\int_\Omega P_h(T)U_{Th}}{\int_\Omega W^j} = 3\frac{\int_\Omega P_h(T)U_{Th}}{|D_j|} \tag{25}
\]

**Example**  Consider a basket call option with the same data range (1).

Figure 3-left shows the results of a finite element simulation using Freefem++ when time step 0.02 and computational domain \((0, 400) \times (0, 900)\). The value of the option is recomputed at \( S_0 = 98, S_1 = 105 \) by integrating (22) with initial condition \( P_0 = 1 \) at the nearest mesh point and zero at other vertices (figure 3-center). Figure 3-right shows \( P(T) \). According to (25), we have \( U(98, 105, 0) = 41.722 \), while (20) gives 41.684. The method seems precise enough; it is far superior to the previous one because it is fast but is restricted to areas where the option is relatively constant on each triangle closed to the money. That can always be obtained by refining the mesh.

![Figure 3: Left: iso-lines of the put option. Center: initial condition for the adjoint equation. Right solution of the adjoint equation at time T.](image)

### 3.1 Adjoint at the level of Linear Algebra

It remains to explain why the method is not affected by the numerical discretization in time.

Let us use the Euler implicit time scheme with time step \( \delta t \). Then a time approximation of (20) requires solving

\[
(B + A)U^n - BU^{n+1} = 0 \quad U^N = U_T \tag{26}
\]

where \( U^n \) is the vector of values of \( U_h(q^i, n\delta t) \) and \( A, B \) are the matrices

\[
B_{ij} = \frac{1}{\delta t} \int_{\Omega_h} W^i W^j, \quad A_{ij} = a(W^i, W^j) \tag{27}
\]
Introduce $P^{n+1}$ solution of:

\[(A + B)^TP^{n+1} - B^TP^n = 0 \quad P^0 = P_0\]  

(28)

Now notice that (26) multiplied by $P^T$ gives

\[0 = P^T(A + B)U^n - P^TBU^{n+1} = P^{n-1}^TBU^n - P^TBU^{n+1}\]  

(29)

where the last equality has used (28). Summing up over all $n$ gives

\[P^0^TBU^1 = P^{N-1}^TBU^N\]  

(30)

Choosing $P_j^0 = \delta_{ij}$, $j = 1 \ldots I$ gives the discrete equivalent of (25)

\[\left(\sum_{j=1}^I b_{ij}\right)U_1^i \approx (BU)^1_i = P^{N-1}^TBU^N\]  

(31)

where $I$ is the total number of vertices.

**Remark 1** If an extra level is introduced in (28) by setting $P^{-1} = P^0$ then (30) becomes

\[P^{-1}^TBU^0 = P^{N-1}^TBU^N\]  

(32)

**Results** In search for a fast numerical solver, we implemented the finite element method in C++ and solved the linear system with the superLU library (see [7]) which is a general purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations on high performance machines. The use of the sequential package is sufficient to challenge the execute time of the most efficient Finite Difference Methods.

We present in Table 5 some results showing the efficiency of superLU versus a LU scheme. The first two lines are with mesh adaptation and the rest is with uniform log-distributed meshes. With a $200 \times 200$ mesh log-distributed. We obtain an error of $1.47e-9$ with the forward equation (30).

### 4 A Dupire-like Equation

So far we have not made any use of the explicit form of the pay-off of the basket option,

\[U_T(S_2, S_2) = (S_1 + S_2 - K)^+\]

to remove the Dirac singularities in (9).

Let us seek first a $w$ such that $P = \partial S_1S_2 w$ and let us seek a primitive
<table>
<thead>
<tr>
<th>Mesh size</th>
<th>LU [s]</th>
<th>Relative error</th>
<th>superLU [s]</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>101 x 101</td>
<td>10.094</td>
<td>3.532 %</td>
<td>2.39</td>
<td>3.076 %</td>
</tr>
<tr>
<td>126 x 126</td>
<td>14.547</td>
<td>1.338 %</td>
<td>4.016</td>
<td>1.797 %</td>
</tr>
<tr>
<td>151 x 151</td>
<td>22.313</td>
<td>0.751 %</td>
<td>6.203</td>
<td>0.489 %</td>
</tr>
<tr>
<td>176 x 176</td>
<td>31.985</td>
<td>1.131 %</td>
<td>8.735</td>
<td>0.790 %</td>
</tr>
<tr>
<td>201 x 201</td>
<td>43.938</td>
<td>0.432 %</td>
<td>12.109</td>
<td>0.670 %</td>
</tr>
</tbody>
</table>

Table 3: Comparison of CPU time for the LU algorithm and superLU for a product put option on a uniform mesh and 200 step time.

of (3) by two integrations, one in $S_1$ and one in $S_2$. In (3) the term $\frac{\partial}{\partial S_i}(\sigma_i^2 S_i^2 \partial S_i S_j w)$ can be integrated in $S_j$, $j \neq i$ only if $\sigma_i$ does not depend on $S_j$. With this assumption the following equation holds for $w$:

$$
\partial_t w - \frac{1}{2} \sum_{i=1}^{2} \frac{\partial}{\partial S_i}(\sigma_i^2 S_i^2 \partial S_i w) + 2q\sigma_1 \sigma_2 S_1 \partial S_1 S_2 w + r S_1 \partial S_1 w + r S_2 \partial S_2 w = 0 \\
w(S_1, S_2, t_0) = (1 - H(S_1 - S_{01}))(1 - H(S_2 - S_{02}))
$$  (33)

It corresponds to a special choice of the integration constant giving a fast decay at infinity of $w$; $H$ is the Heaviside function, a primitive of the Dirac mass. Recall that

$$
\partial_1 S_2 w(S_1, S_2, t_0) = \partial_1(S_1 - H(S_1 - S_{01})) \partial_2(S_2 - H(S_2 - S_{02})) = (-\delta(S_1 - S_{01}))(\delta(S_2 - S_{02})) = P(S_1, S_2, t_0)
$$  (34)

Formula (15) can be integrated by part

$$
U(S_{01}, S_{02}, t_0) = \int_{\mathbb{R}_+^2} (\partial_1 S_2 w)(S_1 + S_2 - K)^+ dS_1 dS_2 \\
= \int_{\mathbb{R}_+^2} w \partial_1 S_2 (S_1 + S_2 - K)^+ dS_1 dS_2 \\
+ \int_{\partial\mathbb{R}_+^2} ((S_1 + S_2 - K)^+ \partial_1 w - w \partial_1 (S_1 + S_2 - K)^+) \\
= \int_{\{S_1 + S_2 = K, S_i > 0\}} \frac{w}{\sqrt{2}} + \int_K w(S_1, T) dS_1 + \int_K w(T, S_2) dS_2
$$  (35)

**Proposition 3** If $r$ is function of $t$ only and each $\sigma_i$ is a function of $S_i$ and $t$ only, $i = 1, 2$, then the basket option solution of (7) is given by

$$
U(S_{01}, S_{02}, t_0) = \int_{\mathbb{R}_+^2} \frac{w(S_1, S_2, T)}{\sqrt{2}} \\
+ \int_K w(S_1, 0, T) dS_1 + \int_K w(0, S_2, T) dS_2
$$  (36)
$H(z)$ being the Heaviside function ($= z^+/z$), $w$ the solution of

$$\partial_t w - \sum_1^2 \left[ \partial_{S_i} \left( \frac{\sigma_i^2}{2} S_i^2 \partial_{S_i} w \right) - r S_i \partial_{S_i} w \right] + 2qS_1S_2\partial_{S_1S_2} w = 0$$

$$w(S_1, S_2, t_0) = [1 - H(S_1 - S_{01})][1 - H(S_2 - S_{02})] \quad (37)$$

If $q\sigma_1\sigma_2$ is constant or depends on $t$ only then it is possible to integrate further by setting $w = \partial_{S_1S_2} v$.

**Proposition 4** If $q\sigma_1\sigma_2$ is not a function of $S_1, S_2$ then

$$U(S_{01}, S_{02}, t_0) = \int_{S_1 + S_2 = K} \frac{\partial_{S_1S_2} v(S_1, S_2, T)}{\sqrt{2}} - \partial_{S_2} v(K, 0, T) - \partial_{S_1} v(0, K, T) \quad (38)$$

where $v$ is the solution of

$$\partial_t v - \sum_1^2 \left( \frac{\sigma_i^2}{2} S_i^2 \partial_{S_i} v - (r + 2q\sigma_1\sigma_2)S_i \partial_{S_i} v \right) - 2q\sigma_1\sigma_2 S_1 S_2 \partial_{S_1S_2} v - 2q\sigma_1\sigma_2 v = 0$$

$$v(S_1, S_2, t_0) = (S_{01} - S_1)^+ (S_{02} - S_2)^+ \quad (39)$$

**Numerical Results** Two programs are written in the *freefem* language [4], one to solve (7) and one for (37). Both use a finite element method of order 1 on triangles and Euler’s implicit time scheme. Mesh adaptivity is used for (7) and the numerical scheme is applied to $\tilde{U} = U - S_1 - S_2 + Ke^{-r(T-t)}$ so as to have a decaying function at infinity. The data range is (1). The computational domain is chosen to be $(0, 300) \times (0, 300)$ and the mesh is shown on figure 4. The time step is 0.01. The level lines of $\tilde{U}$ are also shown on figure 4.

Figure 4: Iso-value lines of $U$ computed by (7) on the adapted mesh shown on the right.
Equation (37) was solved by the same method with various values for $S_{01}, S_{02}$ shown in Table 4. Figure 5 shows the level lines at $T = 1$ and the mesh used. We also compute the option of Payoff $(S_1 + S_2 - K)^+$ with the data of (1) to compare this with all the other methods. The Value is $U(Spot_0, Spot_1, 0) = 40.989$

Figure 5: Iso-value lines of $w$ computed by (37) on the mesh shown on the right. The yellow line is $S_1 + S_2 = K$.

<table>
<thead>
<tr>
<th>$S_1 \backslash S_2$</th>
<th>20</th>
<th>50</th>
<th>80</th>
<th>110</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>20: c</td>
<td>61.49</td>
<td>31.57</td>
<td>5.61</td>
<td>-0.089</td>
<td>-0.097</td>
</tr>
<tr>
<td>20: d</td>
<td>61.67</td>
<td>32.11</td>
<td>5.88</td>
<td>0.0065</td>
<td>1.1e-07</td>
</tr>
<tr>
<td>50: c</td>
<td>31.57</td>
<td>91.49</td>
<td>61.49</td>
<td>31.50</td>
<td>4.32</td>
</tr>
<tr>
<td>50: d</td>
<td>32.11</td>
<td>91.44</td>
<td>61.81</td>
<td>31.52</td>
<td>4.39</td>
</tr>
<tr>
<td>80: c</td>
<td>5.61</td>
<td>61.49</td>
<td>121.49</td>
<td>91.49</td>
<td>61.49</td>
</tr>
<tr>
<td>80: d</td>
<td>5.88</td>
<td>61.81</td>
<td>121.62</td>
<td>91.99</td>
<td>61.70</td>
</tr>
<tr>
<td>110: c</td>
<td>-0.089</td>
<td>31.50</td>
<td>91.5</td>
<td>151.49</td>
<td>121.49</td>
</tr>
<tr>
<td>110: d</td>
<td>0.0065</td>
<td>31.52</td>
<td>91.99</td>
<td>151.86</td>
<td>122.23</td>
</tr>
<tr>
<td>140: c</td>
<td>-0.097</td>
<td>4.3</td>
<td>61.49</td>
<td>121.5</td>
<td>181.49</td>
</tr>
<tr>
<td>140: d</td>
<td>0</td>
<td>4.3</td>
<td>61.69</td>
<td>122.2</td>
<td>181.60</td>
</tr>
</tbody>
</table>

Table 4: Comparison between direct calculation of $U$, the basket call on $S_1, S_2$, based on (7) –lines :c– and $U$ computed by solving the Dupire equation (37) –lines :d–. Here the Data are $T=1$, $\sigma_1 = \sigma_2 = 1$, rate $= \mu_1 = \mu_2 = 5\%$, $K=100$
\begin{tabular}{|l|c|c|c|c|}
\hline
Pay-off & Backward & Dirac approximation & Discrete & Linear algebra & Dupire-Like \\
\hline
$E[(S_1 + S_2 - K)^+]$ & 41.663 & 41.629 & 41.722 & 41.663 & 40.989 \\
\hline
\end{tabular}

Table 5: Comparison of every numerical methods.

References


[4] Frédéric Hecht, Olivier Pironneau, Antoine Le Yaric, Koji Ohtsuka; *freefem++ documentation*. www.freefem.org


