To appear in *Quantitative Finance*, Vol. 00, No. 00, Month 20XX, 1–21

A parallel wavelet-based pricing procedure for Asian options

S. CORSARO†§, D. MARAZZINA*‡ and Z. MARINO†

†Università di Napoli "Parthenope", Dipartimento di Studi Aziendali e Quantitativi, Napoli, Italy
§ ICAR - National Research Council of Italy, Napoli, Italy
‡Politecnico di Milano, Dipartimento di Matematica, Milano, Italy

(*Received 00 Month 20XX; in final form 00 Month 20XX*)

In this paper we present a parallel pricing algorithm for Asian options based on the Discrete Wavelet Transform (DWT). The computational kernel of the pricing model is the solution of integral equations. We obtain a sparse and accurate representation of the kernel of such equations in wavelet functions bases. It is worth to point out that the execution time of our procedure is almost constant with respect to the number of monitoring dates. Thus, our pricing procedure is particularly competitive when the number of monitoring dates is large. We moreover discuss the parallelization of the algorithm. Numerical results which show the accuracy and efficiency of the procedure are reported in the paper.

Keywords: Asian options, Option pricing, Wavelet transform, High performance computing

JEL Classification: C63, G12

1. Introduction

Asian options are path-dependent contracts whose value depends on the average stock price. This family of contracts is popular in the financial industry since they are less expensive than their plain-vanilla counterparts and less sensitive to changes in the underlying asset price. Moreover, these options enable investors to eliminate losses from movements in the underlying asset without the need for continuous re-hedging.

The payoff of a Fixed Strike Asian option depends on the difference between the average of the underlying and a fixed strike. The payoff of a Floating Strike option depends on the difference between the average of the underlying and the value of the asset at maturity. Moreover, most Asian options are traded in a discrete monitoring framework, i.e., the average is computed considering the underlying asset values at prefixed dates, like the end of each day, week, or of each month (daily, weekly or monthly monitoring respectively). The common form of the average could be either arithmetic or geometric. In the first case, no analytical formulae are available, thus numerical valuation methods are required. Several methods are proposed in literature to deal with this kind of contracts: we refer to Chang and Tsao (2011), Fusai *et al.* (2011), Kim and Wee (2011) for a review of numerical methods for Asian contracts.

In this paper, we refer to recent contributions given in this field, which assume that the underlying assets evolve according to an exponential Lévy process, as in Fusai and Meucci (2008), Černý and Kyriakou (2011) and Fusai *et al.* (2011), among the others, dealing with both

*Corresponding author. Email: daniele.marazzina@polimi.it

Floating and Fixed Strike Asian options. Moreover, we consider the discrete monitoring case, much more relevant in practice than the continuous one, since, as stated above, the average is usually computed at prefixed dates.

We present a pricing algorithm based on the Discrete Wavelet Transform (DWT). Wavelets are functions that are localized both in space and frequency domain. Wavelet bases are thus suitable for adaptive approximation, since the coefficients of the representation contain local information.

Earliest applications of wavelets to economic and financial problems concerned time series analysis. In Gençay *et al.* (2002) and Ramsey (2002) it is shown that wavelet techniques can provide insight beyond other approaches in variance-covariance estimation, denosing, identification of long memory processes, jump detection in high-frequency data.

Successively, wavelet bases have been recognized as effective tools in different financial topics, among which foreign exchanges, financial asset pricing, hedges and futures. In and Kim (2012) provides a comprehensive introduction to the application of wavelet analysis to financial markets.

In particular, the widely recognized wavelets potential to obtain sparse computational domains in the solution of problems modeled by Partial Differential Equation (PDE) has been exploited in finance. In Carton De Wiart *et al.* (2011) a PDE method, that makes use of wavelets, for evaluating financial derivatives is introduced. Mariani *et al.* (2008) take advantage of the same property to efficiently represent the kernel of an integral operator. More precisely, authors use wavelet expansions to approximate the kernel of an integral equation arising in the context of Heston stochastic volatility model calibration.

Our starting point is the pricing procedure described in Fusai *et al.* (2011); it is based on a randomization technique, according to which the expiry date of the option is modeled as a random variable distributed as geometric. The computational kernel is the solution of integral equations; as authors point out, this procedure turns out to be accurate but computational demanding. Our contribution in this framework is twofold. On one hand, we integrate the DWT into the pricing algorithm in order to speed up the process while preserving accuracy; on the other, we parallelize the developed algorithm in order to benefit from advanced architectures. The integral equations involved in the model are Fredholm integral equations of the second kind; the kernel of such equations fall within the class of the aforementioned operators for which projection onto wavelet bases is particularly effective. For this reason, we apply the DWT to the linear systems which arise from the discrete operators. Thanks to the wavelets localization property, we transform the linear systems matrices into sparse matrices, obtaining a fast algorithm that preserves the accuracy of the original method. Numerical results reported in this paper confirm the effectiveness of our approach. It is worth to point out that the execution time of our procedure is almost constant with respect to the number of monitoring dates. This is only partially due to the use of Euler summation technique, which bounds the number of linear systems to be solved, so to avoid severe increase of the execution with respect to the number of monitoring dates. Fundamental issues are the sparsity of the involved matrices, which is obtained applying the DWT, and the use of a suitably preconditioned iterative solver. Thus, our pricing procedure is particularly competitive when the number of monitoring dates is large.

We furthermore introduce parallelism in the pricing wavelet-based algorithm, both in the DWT computation and in the linear systems solution process, to speed-up the evaluation process. It is now widely recognized the key role of High Performance Computing (HPC) in finance. Many projects at present involve academics and financial institutions, with the aim of both developing HPC hardware and software solutions for financial problems and training young people to meet the future requirements of financial industry. The interest has been particularly pushed in the last years by the financial crisis; new regulations clearly establish requirements that concern the statistical quality standards of the models for valuation and risk management. Therefore frequent, large-scale analyses of financial data have to be performed to develop early warning

systems. It is recognized that "the crux of the most challenging banking reform problems lays now more than ever in the realm of computer engineering" (Albanese *et al.* (2011)). Of special importance is the concept of parallelism, since today microprocessors have multiprocessor support, thus high-performance computers are actually parallel computers.
The literature on applications of HPC to finance is very rich; in particular, we mention some related work on the development of parallel option pricing procedures. In Sak *et al.* (2007) authors discuss the use of parallel computing in Asian option pricing and evaluate the efficiency of various algorithms. In Fusai *et al.* (2010) authors show the use of a grid computing architecture for the solution of a complex Asian option pricing problem. In Bronstein *et al.* (2010) parallel implementations of a quantization tree algorithm, applied to price American multi-asset options and swing options, are presented. In Leentvaar and Oosterlee (2008) authors show a parallel Fourier-based sparse grid method for pricing multi-asset options. Finally Surkov (2010) discusses the implementation on a GPU of algorithms, based on the Fourier space time-stepping method, for pricing single- and multi-asset options.

Preliminary results of this research activity were presented in Corsaro *et al.* (2011), where a first analysis of the DWT benefits has been addressed. In that paper, the pricing procedure is based on a recursive relation defined on the transition probability density. Fixed Strike options are considered only. Also in this paper we refer to a pricing procedure which is based on a randomization technique, but the recursion considered here is a price recursion. Moreover, we here consider both Fixed and Floating Strike Asian options.

The paper is organized as follows: in Section 2. we describe the pricing model we refer to; in Section 3. we recall basic elements of wavelet theory; in Section 4. we describe the wavelet-based pricing method; in Section 5. we discuss the parallel implementation. Finally, in Section 6. we give some conclusions.

2. The randomization pricing method

In this section we briefly recall the pricing model considered in this paper, addressing reader to Fusai *et al.* (2011) for details. More precisely, we assume that the risk-neutral process for the stock price $S(t)$ is described by

$$S(t) = S_0 e^{(r-d+g)t+L(t)}$$

where r is the short rate, d is the dividend yield, g is the *compensator*, chosen to ensure that the discounted price process is a martingale. $L(t)$ is a Lévy process, identified by its characteristic exponent $\psi(\omega) = \log \mathbb{E}(e^{i\omega L(1)})$. Let us consider M equidistant monitoring dates, with amplitude of the interval Δ, such that $t_0 = 0$, $t_1 = \Delta, \cdots, t_n = n\Delta, \cdots, t_M = M\Delta = T$. The log-return on each time interval has the following characteristic function

$$\phi(\omega) = e^{(\psi(\omega)+i\omega(r-d+g))\Delta}. \tag{1}$$

The density f of the log-return is obtained computing the Fast Fourier Transform (FFT) of (1). Let us denote with S_n the price of the underlying at time $n\Delta$, i.e., $S_n = S(n\Delta)$; the payoff of an arithmetic Asian option is given by

$$(I_M - cS_M)^+$$

where

$$I_M := \sum_{n=0}^{M} \lambda_n S_n$$

is a path dependent random variable.
By suitable choices of values λ_n and c, it is possible to describe a wide class of Asian options. Standard cases are

$$\lambda_0 = \frac{\gamma}{M+\gamma} - \frac{K}{S_0}; \quad \lambda_n = \lambda = \frac{1}{M+\gamma}, \quad n = 1, \ldots, M; \quad c = 0 \tag{2}$$

for Fixed Strike call options, and

$$\lambda_0 = -\frac{\gamma}{M+\gamma}; \quad \lambda_n = \lambda = -\frac{1}{M+\gamma}, \quad n = 1, \ldots, M; \quad c = -1 \tag{3}$$

for Floating Strike calls; if S_0 is included in the average, we set $\gamma = 1$, otherwise $\gamma = 0$.
The following recursion holds for the option price:

$$\begin{aligned} V(S_M, I_M, M) &= (I_M - cS_M)^+ \\ V(S_n, I_n, n) &= e^{-r\Delta} \int_{-\infty}^{+\infty} f(s)\, V(S_n e^s, I_n + \lambda_{n+1} S_n e^s, n+1)\, ds \\ &\qquad\qquad\qquad\qquad n = M-1, \ldots, 0. \end{aligned} \tag{4}$$

The randomization technique consists in modeling the expiry date T as a random variable distributed as geometric of parameter q; if one defines

$$H(x, q) := (1-q) \sum_{k=0}^{+\infty} q^k v(x, k) \tag{5}$$

with $v(x, k) := V(1, x, M-k)$, the option price is given by $S_0 v(\lambda_0, M)$.
Considering recursion (4) and definition (5), Fusai *et al.* (2011) show that for Floating Strike options function H satisfies the integral equation

$$H(x, q) = q \int_{-\infty}^{\lambda} K(x, y)\, H(y, q)\, dy + (1-q)\phi(x) \tag{6}$$

where

$$K(x, y) = -e^{-r\Delta} f\left(\log\left(\frac{x}{y-\lambda}\right)\right) \frac{x}{(y-\lambda)^2}$$

and

$$\phi(x) = (x-c)^+ .$$

For Fixed Strike options, the value of $v(x,k), k = 1, ..., M$, is analytically known for $x \geq 0$ and so the integral equation becomes

$$H\left(x,q\right) = q \int_{-\infty}^{0} K\left(x,y\right) H\left(y,q\right) dy + (1-q)\widetilde{\phi}\left(x,q\right) \tag{7}$$

with

$$\widetilde{\phi}\left(x,q\right) = \phi\left(x\right) + \frac{q}{1-q} \int_{0}^{\lambda} K\left(x,y\right) H\left(y,q\right) dy \tag{8}$$

where, if $y \geq 0$,

$$H(y,q) = \frac{y(1-q)}{1-qe^{-r\Delta}} + \frac{e^{(r-d)\Delta}(1-q)}{(M+\gamma)(1-e^{(r-d)\Delta})}\left(\frac{1}{1-qe^{-r\Delta}} - \frac{1}{1-qe^{-d\Delta}}\right).$$

In this way, starting from the integral recursion for V defined in (4), an integral equation for H (Equation (6) and (7) for Floating and Fixed Strike Asian options, respectively) is obtained. This equation is the core of the evaluation process.
Applying a quadrature rule, with N nodes x_i and weight w_i, to it, we obtain the linear system

$$(I - qKD)h = b \tag{9}$$

where I is the identity matrix and, for $i,j = 1, \cdots, N$, the vector and matrices elements are given by

$$\begin{aligned} h(i) &= H(x_i, q) \\ K(i,j) &= K(x_i, x_j) \\ b(i) &= (1-q)\Phi(x_i, q) \\ D(i,i) &= w_i, \quad D(i,j) = 0 \text{ if } i \neq j \end{aligned}$$

with $\Phi(x_i, q) = \phi(x_i)$ ($\Phi(x_i, q) = \widetilde{\phi}(x_i, q)$) for Floating (Fixed) Strike Asian options. System (9) is the main computational kernel in the algorithm.
The option price is recovered de-randomizing the option maturity, that is, exploiting the complex inversion integral

$$v(\lambda_0, M) = \frac{1}{2\pi\rho^M} \int_0^{2\pi} \frac{H\left(\lambda_0, \rho e^{is}\right)}{1-\rho e^{is}} e^{-iMs} ds. \tag{10}$$

In particular, we approximate numerically (10) using:

$$v_h(\lambda_0, M) = \frac{\frac{H(\lambda_0,\rho)}{1-\rho} + (-1)^M \frac{H(\lambda_0,-\rho)}{1+\rho} + 2\sum_{j=1}^{M-1}(-1^j) Re\left(\frac{H(\lambda_0,\rho e^{ij\pi/M})}{1-\rho e^{ij\pi/M}}\right)}{2M\rho^M} \tag{11}$$

where $Re(\cdot)$ denotes the real part function and ρ is set to $10^{-4/M}$.
The procedure consists in the following steps:

- solve Equation (6) for q equal to $q_j = \rho e^{ij\pi/M}$, $j = 0, ..., M$;
- approximate $v(\lambda_0, M)$ by $v_h(\lambda_0, M)$ as in (11).

Moreover, Equation (11) can be written as

$$v_h\left(\lambda_0, M\right) = \frac{1}{\rho^M} \sum_{j=0}^{M} (-1)^j a_j H\left(\lambda_0, \rho e^{ij\pi/M}\right)$$

so we can use the Euler summation technique, a convergence-acceleration technique for evaluating alternating series, for its computation. More precisely, fixed two positive integers m_e, n_e, the idea is to approximate the above summation in the following way:

$$\tilde{v}\left(\lambda_0, M\right) \approx \frac{1}{2^{m_e}\rho^M} \sum_{j=0}^{m_e} \binom{m_e}{j} b_{n_e+j}\left(\lambda_0, M\right)$$

where

$$b_k\left(\lambda_0, M\right) = \sum_{j=0}^{k} (-1)^j a_j H\left(\lambda_0, \rho e^{ij\pi/M}\right).$$

The acceleration technique is applied when more than $n_e + m_e$ monitoring dates are considered. When $M > n_e + m_e$ the evaluation of $H\left(\lambda_0, \rho e^{ij\pi/M}\right)$ for $j = 0, ..., n_e + m_e$, is required, thus $n_e + m_e + 1$ systems instead of $M + 1$ as in (11). The number of linear systems which are actually solved is thus

$$N_{sys} = \min(M, n_e + m_e) + 1. \tag{12}$$

In our experiments, we fix these values following Fusai *et al.* (2011).

It is worth emphasizing that, from a computational point of view, Fixed and Floating Strike option valuation slightly differ: in the Floating Strike case all systems have the same right-hand side, while for Fixed Strike options it has to be computed for each one of them, because of its dependence on the parameter q as shown in Equation (7). In Figure 1 a sketch of the pricing algorithm is reported.

```
Procedure Asian_floating_pricing
 1 : compute $K$, $D$ and $b$
 2 : for $j = 0, ..., \min(M, n_e + m_e)$ do
         solve $(I - q_j KD)h_j = b$
     end
 3 : reconstruct $v(\lambda_0, M)$
End Asian_floating_pricing
```

```
Procedure Asian_fixed_pricing
 1 : compute $K$ and $D$
 2 : for $j = 0, ..., \min(M, n_e + m_e)$ do
         compute $b = b(q_j)$
         solve $(I - q_j KD)h_j = b$
     end
 3 : reconstruct $v(\lambda_0, M)$
End Asian_fixed_pricing
```

Figure 1. Asian call randomization pricing algorithm. Up: Floating Strike options; down: Fixed Strike options.

3. The Discrete Wavelet Transform

The solutions of the linear systems in Figure 1 can be computed exploiting wavelets properties. In this section, we briefly report basis on wavelet functions, which are essential to motivate and present our work. We then introduce the DWT. For a comprehensive read on the topic we address to Daubechies (1992).

A function $\psi(x) \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is a wavelet function (*wavelet* in short), if

$$\int_{\mathbb{R}} \psi(x)dx = 0.$$

Therefore, wavelets have either compact or "almost compact" support, meaning that, in probabilistic terms, most of a wavelets mass is concentrated in a compact subset of $\mathbb{R}$. The key property of wavelets is indeed referred to as "localization", which allows one to obtain sparse representation of certain functions and operators in wavelet bases.

The definition of wavelet is a quite general one, thus many wavelet families can be built, which provide bases for $L^2(\mathbb{R})$. They are classified according to certain properties such as orthogonality, the amplitude of the support, smoothness and the number of vanishing moments[1]. Each one of them is most important for specific scopes, thus the choice of the wavelet basis is strongly application dependent. A well-known, brief guide to the choice of a suitable wavelet basis is provided in Torrence *et al.* (1998).

Orthogonality is quite often preferred: in this case the discrete operators are represented by orthogonal matrices, with obvious advantages from the computational point of view. For this reason, in the following we confine ourselves to orthonormal (o.n.) bases.

When the focus is data compression, smoothness and compact, narrow supports are also desiderable: in this case, localization is improved, so that small coefficients are obtained in smooth regions of the approximated function. They can therefore be neglected, preserving information about sub-domains in which the gradient has high values. A key role is played by the number P of vanishing moments. On one hand, high values of P should be preferred since the approximation error by means of wavelet functions decreases with respect to P; on the other, wavelets with P vanishing moments have minimum support width $2P$. Large supports can affect localization, thus a trade-off must be empirically found.

The wavelets we use in our experiments were introduced in Daubechies (1988). These families of compactly supported wavelets have become very popular in past years, and have been used in many, different contexts, as a richest literature shows. For each integer P, an o.n. wavelet basis with P vanishing moments and (minimum width) support $[0, 2P-1]$ can be obtained.

The framework within which the DWT is introduced is the MultiResolution Analysis (MRA). A MRA is a sequence of closed, nested subspaces $(V_l)_{l \,\in\, \mathbb{Z}}$ of $L^2(\mathbb{R})$ which recovers the whole space:

(i) $V_l \subset V_{l+1}$
(ii) $v(x) \in V_l \Leftrightarrow v(2x) \in V_{l+1}$
(iii) $v(x) \in V_0 \Leftrightarrow v(x+1) \in V_0$
(iv) $\overline{\left(\bigcup_l V_l\right)} = L^2(\mathbb{R})$
(v) $\bigcap_l V_l = \{0\}$
(vi) $\exists\, \varphi(x) \in V_0$ such that the set $\{\varphi(x-k)\},\ k \in \mathbb{Z}$ is an unconditional o.n. basis of $L_2(\mathbb{R})$.

[1] A wavelet ψ is said to have P vanishing moments if

$$\int_{\mathbb{R}} x^p \psi(x)dx = 0, \qquad p = 0, \ldots, P-1.$$

φ is the scaling function of the MRA. It satisfies the normality condition

$$\int_{\mathbb{R}} \varphi(x)dx = 1.$$

The set $\{\varphi_{l,k}(x) = 2^{l/2}\varphi(2^l x - k),\ k \in \mathbb{Z}\}$ is an o.n. basis of V_l.
Let $f(x) \in L^2(\mathbb{R})$ and $\mathcal{P}_l$ be the orthogonal projector on V_l. It holds

$$f = \mathcal{P}_l f + \sum_{n \geq l} g_n$$

where $g_n = \mathcal{P}_{n+1} f - \mathcal{P}_n f$ is the information that is lost moving from resolution $n+1$ to n. Note that $g_n \in V_{n+1} - V_n$.
We then define the detail space of scale l as the set

$$W_l := V_{l+1} - V_l.$$

Detail spaces form a sequence of nested subspaces of $L^2(\mathbb{R})$ which satisfy

$$\bigoplus_{l \in \mathbb{Z}} W_l = L^2(\mathbb{R}).$$

It can be proven that there exists a wavelet function $\psi(x) \in W_0$ such that the set $\{\psi_{l,k}(x) = 2^{l/2}\psi(2^l x - k),\ k \in \mathbb{Z}\}$ is an o.n. basis of W_l.
The resolution index l of each subspace V_l in the MRA must be intended as a scale: the higher the scale, the more accurate the approximation $\mathcal{P}_l f$. In particular, property (ii) suggests us to think of an element of a MRA as a screen with a certain resolution: the successive element in the sequence would then be a screen with a double number of pixels along each dimension. It can be indeed shown that the approximation error $||\mathcal{P}_l f - f||_2$ has order of magnitude $O(2^{-lP})$, where P is the number of vanishing moments of the wavelet of the MRA. This result, moreover, states the significance of P.
The tool for moving between resolutions is the DWT. The key issue for introducing the DWT are the two following relations:

$$\varphi(x) = \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k)$$
$$\psi(x) = \sum_{k \in \mathbb{Z}} g_k \varphi(2x - k)$$

called refinement equation and wavelet equation, respectively. The sequences $\mathbf{h} := \{h_k\}_{k \in \mathbb{Z}}$ and $\mathbf{g} := \{g_k\}_{k \in \mathbb{Z}}$ are usually referred to as *filters* of the MRA. They stay at the core of the DWT operator. With the notations introduced above, if

$$\mathcal{P}_{l+1} f = \sum_{k \in \mathbb{Z}} c_{l+1,k} \varphi_{l+1,k}$$
$$\mathcal{P}_l f = \sum_{k \in \mathbb{Z}} c_{l,k} \varphi_{l,k}$$
$$\mathcal{Q}_l f = \sum_{k \in \mathbb{Z}} d_{l,k} \psi_{l,k}$$

are the projections of f on V_{l+1}, V_l and W_l respectively, then it holds:

$$c_{l,k} = \sum_{n\in\mathbb{Z}} h_{n-2k} c_{l+1,n}, \qquad d_{l,k} = \sum_{n\in\mathbb{Z}} g_{n-2k} c_{l+1,n} \tag{13}$$

and

$$c_{l+1,k} = \sum_{n\in\mathbb{Z}} h_{2k-n} c_{l,n} + \sum_{n\in\mathbb{Z}} g_{2k-n} d_{l,n}. \tag{14}$$

Relations (13), (14) ensure that one needs to know nothing but the filters of the MRA to move between different levels of resolution. The operator defined by (13) is the DWT, applied to $\mathbf{c_{l+1}}$, where we denote with $\mathbf{c_{l+1}}$ the set $\{c_{l+1,k}\}_{k\in\mathbb{Z}}$.
One computes $\mathbf{c_l}$ convolving $\mathbf{h}$ with $\mathbf{c_{l+1}}$: since $\mathbf{c_l}$ contains the coefficients of the projection of the function at a lower resolution (*scaling coefficients*), $\mathbf{h}$ is usually referred to as *low-pass filter*, following signal analysis terminology. The vector $\mathbf{d_l}$ contains the *wavelet coefficients*, which retain the information that is lost when moving from resolution $l+1$ to resolution l; $\mathbf{h}$ is thus called the *high-pass filter*.
The above observations suggest that if we neglect wavelet coefficients under a fixed threshold (*Hard Threshold* technique (HT), Mallat (2008)), accuracy can be preserved with a significative gain in efficiency.
Equation (14) states that the DWT is an invertible operator; its inverse is the Inverse Discrete Wavelet Transform (IDWT).
In matrix form, if

$$L = (\tilde{h}_{i,j} = h_{j-2i}), \quad H = (\tilde{g}_{i,j} = g_{j-2i})$$

relations (13) can be written as

$$\begin{pmatrix} \mathbf{c_{l-1}} \\ \mathbf{d_{l-1}} \end{pmatrix} = \begin{pmatrix} L \\ H \end{pmatrix} \cdot \mathbf{c_l} \iff \begin{cases} \mathbf{c_{l-1}} = L\mathbf{c_l} \\ \mathbf{d_{l-1}} = H\mathbf{c_l}. \end{cases} \tag{15}$$

The DWT can be applied recursively; at each step, only the scaling coefficients resulting from the previous step are transformed.
Relation (15) is a matrix-vector product: one could then argue that the computational time of the DWT is $O(N^2)$ when applied to a vector with N elements. When compactly supported wavelets are considered, the filters have a finite, typically small number R of elements and the DWT is a $O(NR)$ operator. For instance, if Daubechies wavelets with P vanishing moments are considered, then $R = 2P$; it is always, in practical applications, $R \ll N$, the DWT is actually a $O(N)$ operator.

The construction of the MRA, and, therefore, the DWT, can be extended to the bi-dimensional case considering the tensor products between the sub-spaces, as described in Mallat (2008).
In order to introduce the bi-dimensional DWT, which we use to expand the kernel of the integral equations in wavelet bases, let us recursively define the following matrices:

$$Q^{(1)} := \begin{pmatrix} L \\ H \end{pmatrix}, \qquad Q^{(k)} := \begin{pmatrix} Q^{(k-1)} & I \\ I & I \end{pmatrix}, \qquad k \geq 2$$

and let

$$Q_s := \prod_{k=1}^{s} Q^{(i)}$$

then, the bi-dimensional DWT in s steps of a discrete operator A is defined as

$$A_s^W := Q_s A Q_s^\top. \tag{16}$$

In practice, it is performed in two stages: computing the product $Q_s A$ actually requires to transform s times the columns of A; then, multiplying by $Q_s^\top$, the rows are transformed. Note that, since the wavelet basis is o.n., then all matrices $Q^{(k)}$ are orthogonal, that is, $Q_s^{-1} = Q_s^\top$, so $Q_s Q_s^\top = I$. Therefore one can then easily invert (16). It holds:

$$Q_s^\top Q_s A Q_s^\top Q_s = Q_s^\top A_s^W Q_s \Leftrightarrow A = Q_s^\top A_s^W Q_s.$$

4. The wavelet-based pricing algorithm

We discretize the integrals in (6) and (7) by means of a quadrature rule on a truncated domain $[\mathcal{L}, \lambda]$ for Floating Strike Asian options, $[\mathcal{L}, 0]$ for Fixed ones. Different values for $\mathcal{L}$, chosen according to the optimality criterion discussed in Fusai *et al.* (2011), have been tested. Numerical results shown in the following refer to the value which provides the best accuracy.

In order to present our approach, we here report some results obtained in Beylkin *et al.* (1991), which is one of the pioneering works on the application of wavelet techniques to obtain sparse representation of integral operators.

Let $K(x, y)$ be the kernel of an integral operator and P the number of wavelet vanishing moments. Let us suppose that a certain level in the MRA has been fixed; we denote with $\phi_J, \psi_{J'}$ respectively the scaling and the wavelet function at the fixed level, having supports J, J'. The coefficients of the expansion of $K(x, y)$ are given by:

$$\begin{aligned} \alpha_{JJ'} &= \int_{\mathbb{R}} \int_{\mathbb{R}} K(x,y)\psi_J(x)\psi_{J'}(y)dxdy \\ \beta_{JJ'} &= \int_{\mathbb{R}} \int_{\mathbb{R}} K(x,y)\psi_J(x)\phi_{J'}(y)dxdy \\ \gamma_{JJ'} &= \int_{\mathbb{R}} \int_{\mathbb{R}} K(x,y)\phi_J(x)\psi_{J'}(y)dxdy. \end{aligned}$$

If K is "sufficiently" smooth on the square $J \times J'$, that is, if partial derivatives up to order P exist here, than it can be proven that:

$$|\alpha_{JJ'}| + |\beta_{JJ'}| + |\gamma_{JJ'}| \le C|J|^{P+1} \sup_{(x,y)\in J\times J'} \sum_j \Big| \frac{\partial^P}{\partial x^j \partial y^{P-j}} K(x,y) \Big|.$$

Thus, to have small wavelet coefficients, one should have a small right-hand side; this is small whenever either $|J|$ or the derivatives are small. The first issue ($|J|$ small) points out the importance of having wavelet bases with narrow supports, while the second suggests the classes of integral operators for which a wavelet-based representation can be effective.

Inspired from this, in Corsaro *et al.* (1999) authors identify a class of matrices (*compressible* matrices) for which one can obtain accurate, sparse representation in wavelet bases. They essentially have most significant elements about the main diagonal. The transition probability density of the underlying log-price obviously decreases close to the bounds of the integration domain. For this reason, the most significant elements of the matrix KD are about its diagonal and decay in magnitude is observed away from this region. This feature suggested us to integrate the DWT into the pricing procedure.

In Figure 2 the sparsity structure of matrix KD, for 2048 quadrature nodes and a Gauss-Lobatto quadrature rule, is represented, together with the sparsity structure of the HT of KD

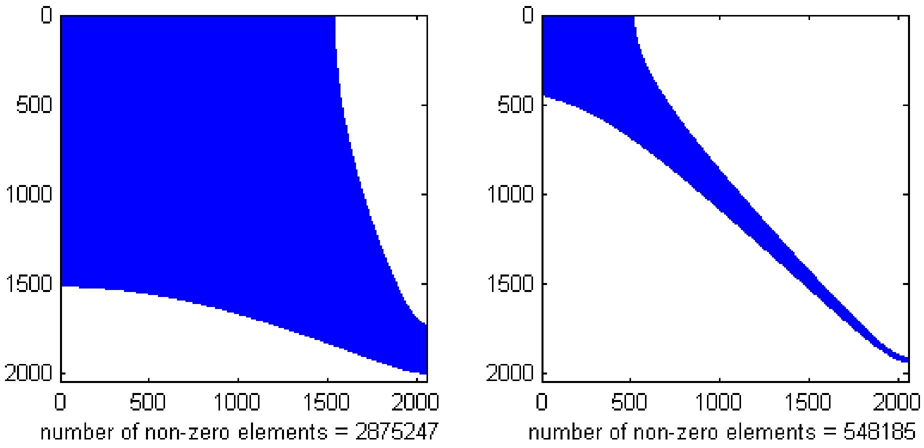

Figure 2. Left: sparsity structure of matrix KD. Right: sparsity structure of the HT of KD with threshold 10^{-8}.

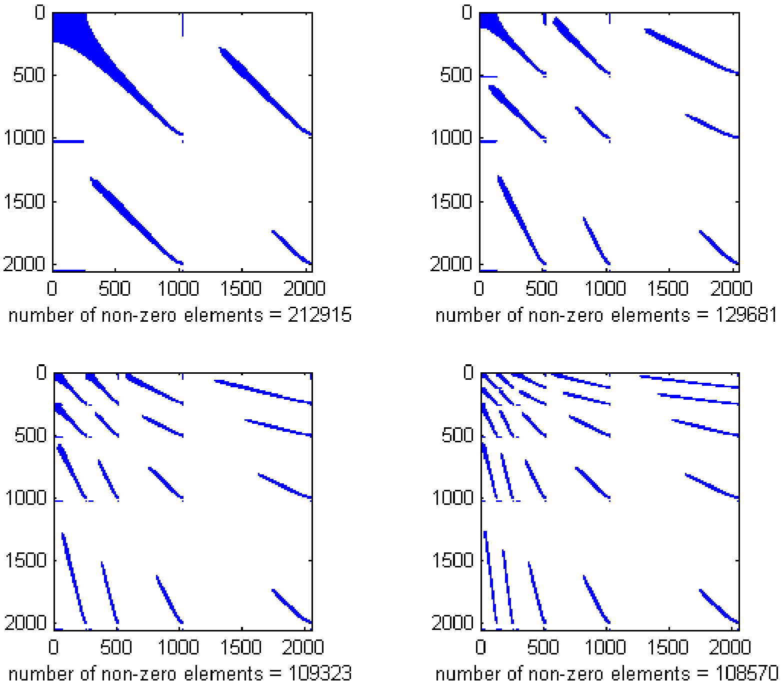

Figure 3. Sparsity structure of the HT of the DWT of matrix KD, with threshold 10^{-8}, for different numbers of transform steps: 1 (up-left), 2 (up-right), 3 (down-left), 4 (down-right).

for a threshold level equal to 10^{-8}. We note that the HT operation allows one to neglect about the 80% of the elements.

In Figure 3 we represent the sparsity structure of the HT of the transformed matrix, for different transform steps and the same threshold. We note that the percentage of neglected elements is greater than 90% even for a single DWT step; it raises up to 96% when 4 steps are performed. Therefore, the impact of the HT increases when it is applied in the wavelet space, thus efficiency is improved. Moreover, accuracy is preserved, as results shown in the following reveal.

Let us refer to the pricing procedure reported in Figure 1: in the solution of the linear systems (step **2**), we apply to both sides the DWT operator Q_s, for a fixed number of DWT steps s; for each value of q we thus obtain the linear system

$$\begin{aligned} Q_s(I - qKD)h = Q_s b &\Leftrightarrow (Q_s - qQ_s(KD)Q_s^\top Q_s)h = Q_s b \\ &\Leftrightarrow (I - qQ_s(KD)Q_s^\top)Q_s h = Q_s b \end{aligned}$$

for the orthogonality of the operator Q_s. Therefore, if we denote by KD^W, h^W, b^W the DWT of KD, h, b respectively, we have

$$(I - qKD^W)h^W = b^W. \tag{17}$$

We then apply a hard threshold to the coefficient matrix of (17), thus we actually solve the linear system

$$(I - q(KD^W)_\epsilon)y = b^W \tag{18}$$

where $(KD^W)_\epsilon$ is the hard threshold of KD^W with threshold ϵ. Finally, the inverse DWT is applied to the solution y of (18), thus an approximation of h, $Q_s^\top y$, is obtained.

4.1 *Numerical Results*

The aim of this section is to discuss the accuracy and the computational efficiency of the proposed wavelet-based pricing algorithm. In particular, we show that the integration of the DWT into the pricing procedure allows us to increase its efficiency while preserving accuracy.

In the experiments shown in the rest of the paper Daubechies wavelets with support $[0, 3]$ (2 vanishing moments) have been used. As discussed in Section 3., a tradeoff between the number of vanishing moments and amplitude of the support must be empirically found. In our experiments the value $P = 2$ exhibited the best performance.

We recall that our starting point is the pricing procedure presented in Fusai *et al.* (2011); in that paper the linear systems arising from the pricing model were originally solved via the GMRes (Generalized Minimal Residual) method. It is an iterative method, belonging to the class of Krylov subspace ones, that is, projection methods onto Krylov subspaces. It is a generalization to non-symmetric linear systems of the conjugate gradient method, one of the most effective ones for solving large sparse symmetric positive definite systems. The method was introduced in Saad *et al.* (1986). In order to speed up the pricing process, authors acted on step **2** in the procedures sketched in Figure 1, testing also another solver for the systems solution, proposed in Reichel (1989). In the following, we denote with `Base` the original procedure, that is, the one which employs the GMRes algorithm and with `Reichel` the procedure which uses Reichel's one to solve the system. The two procedures are summarized in Appendix A. We finally denote with `DWT`, with an explicit reference to the number of transform steps, the wavelet-based pricing algorithm, presented in this work, e.g., `DWT2` is the 2 DWT steps procedure.

In the wavelet-based procedure, we use the GMRes algorithm for solving the systems. In order to fully exploit the potential of the GMRes method, we couple it with a preconditioner that strongly increases the convergence rate and preserves the sparsity pattern of the matrix in (18). More precisely, we precondition the systems by means of the incomplete factorization, known as ILU(0). The underlying idea is very simple: if LU is the factorization provided by Gaussian elimination, the ILU(0) entails a decomposition of the form $\tilde{L}\tilde{U} - E$ where $\tilde{L}$ and $\tilde{U}$ have the same nonzero structure as the lower and upper parts of the matrix respectively, and E is the error of the factorization. This is rather inexpensive to compute and turns out to be very effective, as the results shown in the following reveal.

Table 1. Characteristic exponents of some parametric Lévy processes

Model	$\psi(\omega)$
CGMY	$C\Gamma(-Y)\left((M-i\omega)^Y - M^Y + (G+i\omega)^Y - G^Y\right)$
NIG	$-\delta\left(\sqrt{\alpha^2-(\beta+i\omega)^2}-\sqrt{\alpha^2-\beta^2}\right)$
DE	$-\frac{1}{2}\sigma^2\omega^2+\lambda\left(\frac{(1-p)\eta_2}{\eta_2+i\omega}+\frac{p\eta_1}{\eta_1+i\omega}-1\right)$
JD	$-\frac{1}{2}\sigma^2\omega^2+\lambda\left(e^{i\omega\alpha-\frac{1}{2}\omega^2\delta^2}-1\right)$

Table 2. Mean absolute error and CPU time (in percentage with respect to Base) for Fixed Strike Asian option: CGMY process.

		Base		Reichel	
N		Error	%vsBASE	Error	%vsBASE
1024		$1.64 * 10^{-4}$	1	$1.63 * 10^{-4}$	0.52
2048		$5.40 * 10^{-5}$	1	$5.31 * 10^{-5}$	0.53
4096		$1.53 * 10^{-5}$	1	$1.16 * 10^{-5}$	0.64

		DWT1		DWT2	
N	ϵ	Error	%vsBASE	Error	%vsBASE
1024	10^{-10}	$1.63 * 10^{-4}$	0.48	$1.64 * 10^{-4}$	0.45
1024	10^{-9}	$1.56 * 10^{-4}$	0.46	$1.60 * 10^{-4}$	0.43
1024	10^{-8}	$2.41 * 10^{-4}$	0.44	$1.22 * 10^{-4}$	0.42
1024	10^{-7}	$6.18 * 10^{-4}$	0.43	$2.29 * 10^{-4}$	0.40
2048	10^{-10}	$5.24 * 10^{-5}$	0.39	$5.32 * 10^{-5}$	0.34
2048	10^{-9}	$3.79 * 10^{-5}$	0.37	$4.59 * 10^{-5}$	0.32
2048	10^{-8}	$1.04 * 10^{-4}$	0.35	$4.14 * 10^{-4}$	0.31
2048	10^{-7}	$1.47 * 10^{-3}$	0.34	$7.19 * 10^{-4}$	0.30
4096	10^{-10}	$1.65 * 10^{-5}$	0.54	$1.59 * 10^{-5}$	0.45
4096	10^{-9}	$2.68 * 10^{-5}$	0.54	$2.12 * 10^{-5}$	0.44
4096	10^{-8}	$3.07 * 10^{-4}$	0.50	$1.52 * 10^{-4}$	0.43
4096	10^{-7}	$3.00 * 10^{-3}$	0.48	$1.54 * 10^{-3}$	0.42

The discretization parameters are defined as in Fusai *et al.* (2011), with the exception of the number of quadrature nodes for the computation of the integral in (8): we use $3N$ quadrature nodes (instead of $N + 2500$), since, according to numerical tests, this choice improves accuracy. Moreover, we employ the Gauss-Lobatto quadrature rule (Quarteroni *et al.* 2007, Section 10.2). First of all, we compare the wavelet-based pricing procedure with the Base, and Reichel, from both accuracy and efficiency point of view. In Tables 2-3 we report the mean absolute error over the strike values, obtained with the three procedures, simulating the CGMY and the NIG distribution respectively. The error has been evaluated using as benchmark the ones proposed in Černý and Kyriakou (2011) for Fixed Strike Asian option, setting $K \in \{0.9, 1, 1.1\}$, $T = 1$, $r = 0.04$, $d = 0$, $M = 50$, $\gamma = 1$, $S_0 = 1$, and $C = 0.9795$, $G = 3.512$, $M = 10.96$, $Y = 0.8$ for the CGMY process, $\alpha = 7.4046$, $\beta = -3.5302$, $\delta = 1.2573$ for the NIG process (see Table 1). In the following, we denote with ϵ the threshold in the HT. We compare the procedures in terms of execution time as well: in the tables the percentage of time with respect to the CPU time required by Base (%vsBASE) is reported together with the error. For example, in Table 2 the execution times for $N = 4096$ is 143 for the Base algorithm (percentage 1 in Table 2), 92 for the Reichel one (percentage 0.64) and 63 for the DWT2 with a 10^{-9} threshold tolerance (percentage 0.44). Therefore the improvement of the DWT algorithm with respect to the Base and Reichel ones is significant. These numerical tests refer to a MatLab R2011b implementation [1] on a PC equipped with a Pentium Dual-Core 2.70 GHz and 4 GB RAM.

From these tables, we note that the DWT achieves the same accuracy than Base and it is always more efficient than Reichel. As expected, for smaller values of ϵ the error decreases and CPU time increases, since the linear system matrix becomes less sparse. We note that the computational gain of the DWT increases with respect to N for $N \leq 2048$, while it decreases for $N = 4096$.

[1] The MatLab implementation was realized for comparing the different methods employed in the pricing procedure. We developed our parallel software in C; details are given in next section.

Table 3. Mean absolute error and CPU time (in percentage with respect to `Base`) for Fixed Strike Asian option: NIG process.

		`Base`		`Reichel`	
N		Error	%vs`BASE`	Error	%vs`BASE`
1024		$4.50 * 10^{-4}$	1	$4.49 * 10^{-4}$	0.54
2048		$8.97 * 10^{-5}$	1	$8.90 * 10^{-5}$	0.54
4096		$2.54 * 10^{-5}$	1	$2.20 * 10^{-5}$	0.64

		`DWT1`		`DWT2`	
N	ϵ	Error	%vs`BASE`	Error	%vs`BASE`
1024	10^{-10}	$4.49 * 10^{-4}$	0.47	$4.49 * 10^{-4}$	0.45
1024	10^{-9}	$4.42 * 10^{-4}$	0.45	$4.45 * 10^{-4}$	0.44
1024	10^{-8}	$3.67 * 10^{-4}$	0.44	$4.08. * 10^{-4}$	0.43
1024	10^{-7}	$3.58 * 10^{-4}$	0.43	$1.52 * 10^{-4}$	0.40
2048	10^{-10}	$8.82 * 10^{-5}$	0.39	$8.90 * 10^{-5}$	0.34
2048	10^{-9}	$7.34 * 10^{-5}$	0.37	$8.14 * 10^{-5}$	0.32
2048	10^{-8}	$7.34 * 10^{-4}$	0.35	$3.90 * 10^{-5}$	0.31
2048	10^{-7}	$1.49 * 10^{-3}$	0.33	$7.13 * 10^{-4}$	0.30
4096	10^{-10}	$2.44 * 10^{-5}$	0.54	$2.49 * 10^{-5}$	0.46
4096	10^{-9}	$2.47 * 10^{-5}$	0.52	$2.01 * 10^{-5}$	0.44
4096	10^{-8}	$3.04 * 10^{-4}$	0.50	$1.43 * 10^{-4}$	0.43
4096	10^{-7}	$3.11 * 10^{-3}$	0.48	$1.57 * 10^{-3}$	0.42

For instance, dealing with Table 2, when $N = 1024$ and $\epsilon = 10^{-10}$, `DWT1` requires 48% CPU time of `Base` - the gain is 52% -, for the same parameters and $N = 2048$ the computational time falls to 39%, providing a gain of 61%. Finally, for $N = 4096$ the gain decreases to 46%. To explain this behavior we monitored the different stages of the pricing procedure: we observed that when 1024 quadrature nodes are considered, the CPU time required to compute the price is small, thus the overhead of the DWT affects performance, and we do not have great advantages. On the opposite, if we consider 4096 grid points, the computational cost of the construction of matrices and vectors involved in the algorithm is high: this does not depend on the specific method used to solve the systems, thus performance is affected also if the application of the `DWT` speeds up the linear systems solution process. Tables 2-3 seem to suggest that efficiency increases with respect to the number of DWT steps.

In order to validate our approach, we now compare the performance of our wavelet-based pricing procedure with some algorithms chosen among the ones most employed for our financial problem, for a fixed combination of DWT steps and threshold (4 DWT steps and a 10^{-8} threshold tolerance). We note that the specific choice of the test case does not affect results, as our experiments revealed: the same behavior was observed in all our numerical tests.

As it is shown below, results suggest that the wavelet-based pricing procedure presented here is particularly competitive when the number of monitoring dates is large.

We consider the Double Exponential (DE) distribution (see Table 1), with the parameters considered in Fusai and Meucci (2008), i.e., $r = 0.0367$, $\sigma = 0.120381$, $\lambda = 0.330966$, $p = 0.2071$, $\eta_1 = 9.65997$, and $\eta_2 = 3.13868$. Moreover we set $d = 0$, $\gamma = 1$, $S_0 = 1$. We focus on a Fixed Strike Asian option with strike $K = 1$, different maturities T and daily monitoring, i.e., $M = 252T$. More precisely, in Table 4 we compare the option price computed with the method presented in Fusai and Meucci (2008), here denoted with `FM` and described in Appendix A, the Monte Carlo (`MC`) algorithm and `DWT4` with a 10^{-8} threshold tolerance. For both `FM` and `DWT4` $N = 4096$ grid points have been used, while for `MC` 250000 simulations with antithetic variable variance reduction technique are considered. The prices obtained with both our procedure and the `FM` one are very close and fall into the 95% confidence interval (C.I.) computed with the `MC` algorithm. We point out that the execution time of our procedure is the only one that scales with respect to the maturity, that is, the CPU time is almost constant with respect to T and the number of monitoring dates M. This is only partially due to the use of Euler summation technique, which bounds the number of linear systems to be solved, so to avoid severe increase of the execution with respect to the number of monitoring dates. The fundamental issue that makes our algorithm scalable is the sparsity of the involved matrices, which is obtained applying the DWT. This, together with the effectiveness of the selected preconditioning technique, allows

Table 4. Option price and CPU time (in seconds) for Fixed Strike Asian option: DE process. The threshold tolerance for the `DWT4` algorithm is set equal to 10^{-8}.

		FM		DWT4		MC	
T	M	Price	CPUtime	Price	CPUtime	Price (95% C.I.)	CPUtime
1	252	0.05069	14	0.05070	31	(0.05054, 0.05082)	89
2	504	0.08080	22	0.08080	32	(0.08057, 0.08127)	180
3	756	0.10535	29	0.10536	31	(0.10493, 0.10554)	267
4	1008	0.12636	40	0.12635	32	(0.12581, 0.12659)	366
5	1260	0.14479	46	0.14475	32	(0.14440, 0.14532)	454

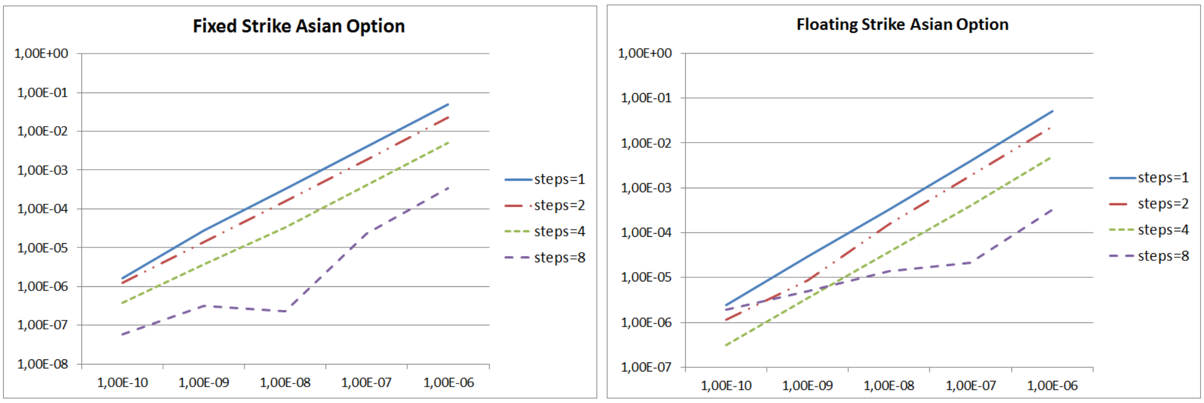

Figure 4. Absolute error: Effects of the threshold tolerance ϵ (x-axis) and of the number of DWT steps.

us to solve each linear system with 8 GMRes iterations independently of the values of T and M.

From all the numerical results presented above, it emerges that the gain in terms of efficiency obtained with the `DWT` increases with respect to the number of transform steps and decreases with respect to the threshold: one could then argue that the best combination is a great number of transform steps and a small value of threshold. In order to investigate on this in Figure 4 we report the absolute error of the computed price of the Asian call option versus the threshold employed in the HT procedure, with 4096 quadrature points. More precisely, in Figure 4 we price a Fixed and a Floating Strike Asian option, assuming a Jump Diffusion (JD) distribution with parameters $r = 0.0367$, $d = 0$, $T = 1$, $S_0 = 1$, $M = 50$, $K = 1$, $\gamma = 1$, $\sigma = 0.126349$, $\lambda = 0.174814$, $\alpha = -0.390078$, and $\delta = 0.338796$ (see Table 1). In each graph the four lines refer to different number of DWT steps. We tested all the possible values of DWT steps, that are the naturals less or equal to $\log_2 N$. We observed that the execution time decreases, with respect to the number of transform steps, as far as this number is lower than 8; so the four lines refer to a number of DWT steps equal to $1, 2, 4$ and 8. The absolute error is computed with respect to a reference price obtained with `Base`, the same number of grid points, with a very small tolerance in the stopping criterion of the linear systems solver, without applying the wavelet transform: Figure 4 shows that errors corresponding to different values of DWT steps are comparable, even if increasing the number of DWT steps mostly results in a slightly increase of the accuracy. Moreover, in all cases, the most significant digits are obtained as far as threshold does not exceed 10^{-8}. On the other hand, the percentage of zeros in the matrix is about 30%; after the HT applied to the transformed matrix, more than 90% of the elements are neglected even for the threshold set to 10^{-6}. Also in these numerical tests, an increase in the DWT steps does not always result in an increase of efficiency. So we can use the threshold to control the accuracy and the number of DWT steps to control the efficiency. In particular reducing the threshold we improve accuracy and increasing the number of DWT steps we reduce the execution time. The best combination of threshold value and number of DWT steps has to be determined; performance obviously depends on computational environment. In our case, the best combination is $\epsilon = 10^{-9}$ and 8 DWT steps. For further details, we refer to Section 5.1, where we analyze the performances of the parallel algorithm.

5. Parallel implementation

In this section we describe the parallel software that we developed. It is written in C language and uses the freely available GSL Library (see Galassi *et al.* (2009)) to perform the wavelet transform. As pointed out, the matrix arising from the threshold applied to the projection of the discrete operators onto wavelet spaces is strongly sparse. We solve the sparse linear systems by means of the GMRES solver, with Incomplete Factorization ILU(0) preconditioner, implemented in the SPARSKIT library (see Saad (1990)).

Parallelism has been introduced in the two most computationally intensive stages of the pricing procedure, that are the DWT of the KD matrix and the solution of the N_{sys} linear systems. The computation of the bi-dimensional DWT is performed in two independent stages. We distribute the matrix KD in a column-block fashion, that is, blocks of contiguous columns are assigned to each process. In the first stage, processes concurrently compute the DWT of columns; then, communication is required for globally transposing the matrix, so, processes can concurrently transform the rows of the intermediate matrix.
Moreover, the algorithm requires the solution of N_{sys} linear systems, with N_{sys} defined in (12). The systems are independent one of each other, thus, they can be solved concurrently; they are distributed among processes, that is, processes build and solve them without communication.
If np processes are involved in the computation, let

$$N_{sys}^{loc} := \lfloor N_{sys}/np \rfloor, \quad mod := \quad \mathrm{mod}\ (N_{sys}, np),$$

where we denote with $\lfloor \cdot \rfloor$ the integer part, then, mod processes solve $N_{sys}^{loc} + 1$ systems, the others N_{sys}^{loc}.

In Figure 5 a sketch of the parallel algorithm for the Floating Strike Asian option is reported. For Floating Strike Asian option the right hand side of the systems has to be computed once, while for Fixed Strike Asian Option it has to be recomputed for each one of them, for its dependence on the parameter q (Figure 1). In the second case each process has to build the right hand side of the system, compute its DWT and then solve the linear system, so some more calculations, which are done in parallel, are required.

```
Procedure
• compute $KD_W$ and $h_W$ in parallel
    - distribute $KD$ in column-block fashion
    - each process computes the DWT of a column block
    - global transposition
    - each process computes the DWT of a row block
• collect the matrix so that each process stores the whole matrix
• neglect the elements of $\mathbf{KD}_W$ below the fixed threshold $\epsilon$;
• solve in parallel $(I - q_j KD_W^{\epsilon})y_j = f_W, \quad j = 0, N_{sys}$
    - distribute the values $(q_j)_{j=0}^{N_{sys}}$ to processes;
    - each process solves the linear systems distributed to it;
    - each process applies the inverse DWT to local solutions $y_j$.
• process $0$ collects local solutions
End Procedure
```

Figure 5. Sketch of the parallel pricing algorithm for Floating Strike Asian option pricing based on the DWT.

The parallel software uses the Message Passing Interface (MPI) communication system. We use the routine `pdtrans` of the PUMMA library (see Choi *et al.* (1993)) for the global matrix transposition. We carried out our experiments on an IBM Bladecenter installed at University of Naples Parthenope. It consists of 6 Blade LS 21, each one of which is equipped with 2 AMD Opteron 2210 and with 4 GB of RAM.

5.1 *Numerical results*

To test the parallel software we simulate the CGMY distribution with parameters $C = 0.9795$, $G = 3.512$, $M = 10.96$ and $Y = 0.8$. We set $T = 1$, $r = 0.04$, $d = 0$ and $M = 50$. To evaluate the parallel performance of the algorithm, in the following we show the speed-up. Speed-up (see Foster (1995)) is the usual metrics to evaluate a parallel software and is defined as

$$\mathcal{S}_{np} = \frac{T_1}{T_{np}} \tag{19}$$

where np is the number of parallel processes and T_{np} is the execution time of the algorithm with np parallel processes. The speed-up denotes how much the parallel program is faster than the sequential one. Typically $\mathcal{S}_{np}$ is a number between 1 and np. The ideal speed-up is $\mathcal{S}_{np} = np$. An important theoretical limit to the maximum improvement obtainable by a parallel algorithm is Amdahl's Law. It states that

$$\mathcal{S}_{np} \leq \frac{1}{\alpha + \frac{1-\alpha}{np}} \tag{20}$$

where α, with $0 < \alpha < 1$, is the fraction of the algorithm which is strictly serial, that is, the part of the algorithm that cannot be parallelized.

Both the threshold of the HT procedure and the number of DWT steps affect performance, so, in Figure 6 and 7 we report the speed-up values, versus the number of processes, for $N = 2048$ and different values of the threshold and the DWT steps.

Since we find that the optimal value for the number of DWT steps is 8 (see Section 4.1), in Figure 6 we fix the number of DWT steps to 8 and we plot the speed-up for 4 different threshold values $\epsilon = 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}$. The two graphics refer to the Fixed Strike (left) and Floating Strike (right) Asian Option. For higher values of threshold the sparsity of the KD matrix substantially increases: as a consequence, the computational time is lower and the communication overhead is higher, affecting parallel efficiency. Indeed the speed-up values are very close for all the values of the threshold up to four processes, while when eight processes are involved the speed-up decay is more evident. Similar behaviors are observed for both Floating Strike and Fixed Strike Asian Option.
In Figure 7 the threshold is fixed at 10^{-9} and the speed-up is represented for $1, 2, 4$ and 8 DWT steps. Again, the two graphics in figure refer to the Fixed Strike (left) and Floating Strike (right) Asian Option. The figure reveals a decrease in terms of performance with eight processes. In most cases, the higher the number of DWT steps, the lower the speed-up. This because the time required for systems solution, that is the execution time of the parallel part of the algorithm (the parallel time) is strongly reduced, thus the communication overhead and the execution time of the part of the algorithm that cannot be parallelized (the serial time) affect speed-up according to Amdahl's law. Also in this case we observe slight difference between Floating Strike and Fixed Strike Asian Option.

6. Conclusion

In this work we presented a parallel wavelet-based pricing procedure for both Fixed and Floating Strike Asian options, within a discrete monitoring modeling framework. We integrated the DWT into a pricing procedure previously developed by one of the authors and parallelized the main computational kernels. More precisely, the DWT has been applied in order to obtain a sparse, accurate representation of the coefficient matrices - typically large matrices - of the linear sys-

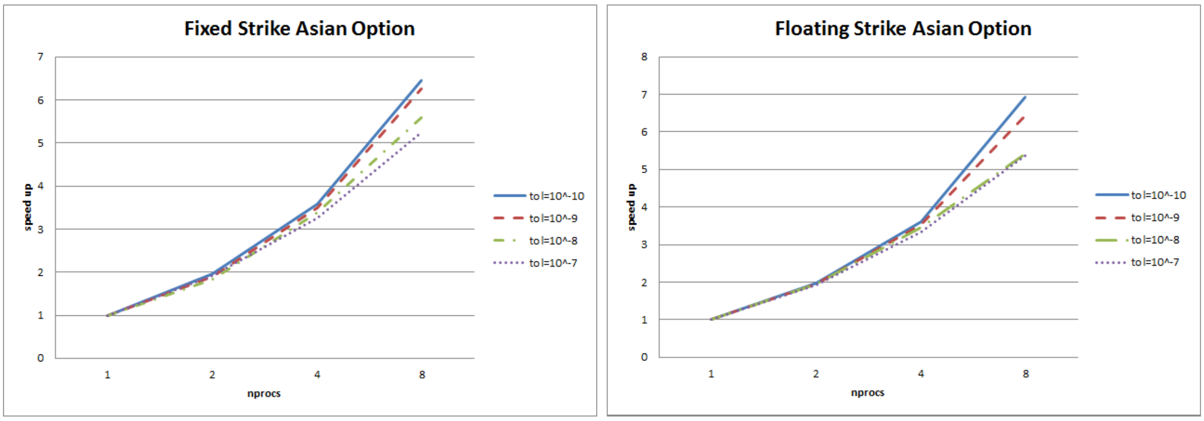

Figure 6. Speed-up: Effects of the threshold tolerance ϵ for $N = 2^{11}$ and 8 DWT steps. Left: Fixed Strike Asian Option; right: Floating Strike Asian Option.

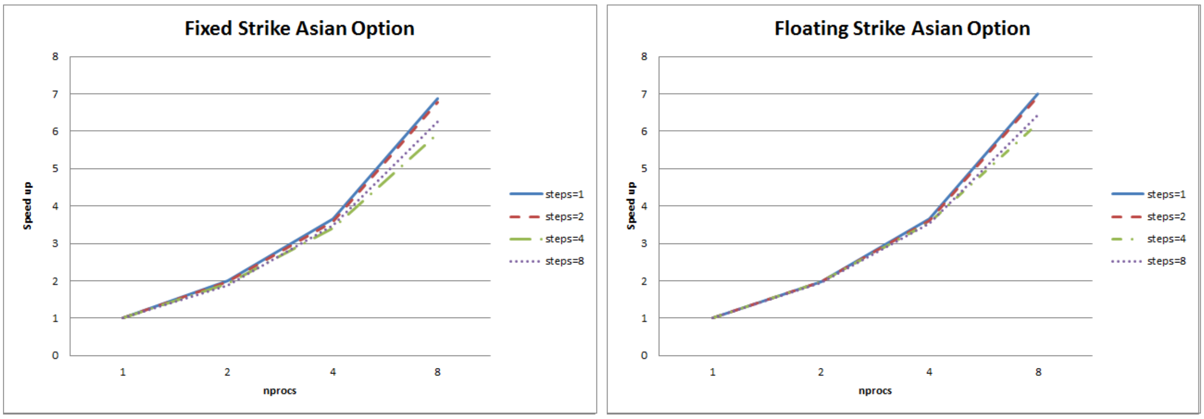

Figure 7. Speed-up: Effects of the number of DWT steps, with $N = 2^{11}$ and ϵ set at 10^{-9}. Left: Fixed Strike Asian Option; right: Floating Strike Asian Option.

tems arising from the involved integral equations. Numerical results confirm the effectiveness of our approach. We have shown, on one hand, that the integration of the DWT into the pricing procedure presented in Fusai *et al.* (2011) turned actually in its improvement, since the computational cost decreases while a comparable accuracy is obtained; on the other, that the execution time of our procedure scales with respect to the maturity, that is, the CPU time is almost constant with respect to T and the number of monitoring dates M. Therefore the proposed pricing method is competitive when the number of monitoring dates is large. Moreover, we have also proved that parallel architectures and methodologies strongly improve efficiency.

We observe that the proposed algorithm can be applied also to other kind of contracts, like barrier and lookback options, starting from the randomization procedure presented in Fusai *et al.* (2012). In that paper a method based on the Fast Fourier Transform (FFT) is proposed; the assumption that the underlying asset evolves as an exponential Lévy process is made. The proposed randomization technique can be applied also to other processes, like the Costant Elasticity of Variance (CEV) one. In this case the FFT-based algorithm cannot be used, since the CEV process is not characterized by independent and identically distributed log-increments. As another example, in Sesana *et al.* (2013) authors propose a pricing approach for valuating different exotic derivatives, such as Asian, barrier, Bermudan, lookback and step options when the underlying asset evolves according to a general process. More precisely, the proposed algorithm exploits the structure of matrices arising from the numerical quadrature of pricing backward formulas, which are close to Equation (4), to devise a convenient factorization that greatly helps in the speed-up of the recursion. Starting from the similarity between the matrix structure described in Sesana *et al.* (2013) and the one presented in Figure 2, if we apply the randomization procedure to the recursive pricing formulas in Sesana *et al.* (2013), our wavelet-based method can instead be used to speed-up the solution of the arising linear systems, like in this work.

References

Albanese, C., D'Ippoliti, F., and Pietronero, G., Margin lending and securization: regulations, modelling and technology. *Wilmott Magazine*, 2011.

Beylkin, G., On the representation of operators in bases of compactly supported wavelets. *SIAM J. Numer. Anal.*, 1992, **6-6**, 1716–1740.

Beylkin, G., Coifman, R., Rokhlin, V., Fast wavelet transforms and numerical algorithms I. *Comm. Pure Appl. Math.*, 1991, **44-2**, 141–183.

Bronstein, A.L., Pages, G., and Wilbertz, B., How to speed up the quantization tree algorithm with an application to swing options. *Quant. Finance*, 2010, **10-9**, 995–1007.

Carton De Wiart, B., Dempster, M.A.H., Wavelet optimized valuation of financial derivatives. *Int. J. of Theoretical and Applied Finance*, 2011, **14-7**, 1113–1137.

Černý, A., and Kyriakou, I., An improved convolution algorithm for discretely sampled Asian options. *Quant. Finance*, 2010, **11-3**, 381–389.

Chang, C., and Tsao, C., Efficient and accurate quadratic approximation methods for pricing Asian strike options. *Quant. Finance*, 2011, **11-5**, 729–748.

Choi, J., Dongarra, J.J., Petitet, A., and Walker, D.W., *Pumma reference manual.* 1993. Available online at: http://www.netlib.org/scalapack

Corsaro, S., D'Amore, L., and Murli, A., On the parallel implementation of the fast Wavelet packet transform on MIMD distributed memory environments. In *Parallel Computation*, LNCS vol. 1557, pp. 357-366, 1999.

Corsaro, S., Marazzina, D., and Marino, Z., Wavelet techniques for option pricing on advanced architectures. In *Euro-Par 2010 Parallel Processing Workshops*, LNCS vol. 6586, pp. 447-454, 2011.

Daubechies, I., *Orthonormal bases of compactly supported wavelets. Commun. Pure Appl. Math.*, 1988, **41**, 909–996.

Daubechies, I., *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics, 1992 (Philadelpia).

Foster, I., *Designing and building parallel programs: concepts and tools for parallel software engineering.* Addison-Wesley, 1995.

Fusai, G., Marazzina D., and Marena, M., Option pricing, maturity randomization and distributed Computing. *Parallel Comput.*, 2010, **36-7**, 403–414.

Fusai, G., Marazzina D., and Marena, M., Pricing discretely monitored Asian options by maturity randomization. SIAM J. of Financ. Math., 2011, **2**, 383–403.

Fusai, G., Marazzina D., Marena, M., and Ng, M., Z-transform and preconditioning techniques for option pricing. *Quant. Finance*, 2012, **12-9**, 1381–1394.

Fusai, G., and Meucci, A., Pricing discretely monitored Asian options under Lévy processes, *J. Bank. Financ.*, 2008, **32-10**, 2076–2088.

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M., and Rossi, F., *GNU scientific library reference manual.* Network Theory Ltd, 2009

Gençay, R., Selçuk, F., and Whitcher, B.J., *An introduction to wavelets and other filtering methods in finance and economics.* Academic Press, 2002 (San Diego, London and Tokyo).

In, F. and Kim, S., *An introduction to wavelet theory in finance: a wavelet multiscale approach.* World Scientific, 2012 (Singapore).

Kim, B., and Wee, I., Pricing of geometric Asian options under Heston's stochastic volatility model. *Quant. Finance*, iFirst, 2011.

Leentvaar, C. C. W., and Oosterlee, C. W., Multi-asset option pricing using a parallel Fourier-based technique. *Comp. Finance*, 2008, **12-1**, 1–26.

Mallat, S., *A Wavelet tour of signal processing.* Academic Press, 2008.

Mariani, F., Pacelli, G., and Zirilli, F., Maximum likelihood estimation of the Heston stochastic volatility model using asset and option prices: an application of nonlinear filtering theory. *Optim. Letters*, 2008, **2**, 177–222.

Quarteroni, A., Sacco, R., and Saleri, F., *Numerical mathematics*, Springer Verlag, 2nd edition, 2007.

Ramsey, J.B., Wavelets in economics and finance: past and future. *Stud. Nonlinear Dynam. Economet.*, 2002, **6-3**, 1–27.

Reichel, L., Fast Solution Methods for Fredholm Integral Equations of the Second Kind, *Numer. Math.*, 1989, **57–1**, 719–736.

Saad, Y., SPARSKIT: a basic tool kit for sparse matrix computations. Technical Report 90-20, Research Institute for Advanced Computer Science, NASA Ames Research Center, 1990.

Saad, Y., Schultz H.M., GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 1986, **7-3**, 856–869.

Sak, H., Özekici, S., and Bodurogĭu, I., Parallel computing in Asian option pricing. *Parallel Comput.*, 2007, **33-2**, 92–108.

Sesana, D., Marazzina, D., and Fusai, G., Pricing exotic derivatives exploiting structure, *Eur. J. Oper. Res.*, 2013, *in press.* http://dx.doi.org/10.1016/j.ejor.2013.12.009.

Surkov, V., Parallel option pricing with Fourier space time-stepping method on graphics processing units. *Parallel Comput.*, 2010, **36-7**, 372–380.

Torrence, C., Compo, G.P., A practical guide to wavelet analysis. *Bull. Amer. Meteor. Soc.*, 1998, **79**, 61-78.

Appendix A: Competitive Methods

In this section, we details the competitive methods considered in Section 4.1: the `Base`, `Reichel` and `FM` algorithms.

A.1 *The `Base` and `Reichel` algorithms*

These two algorithms are proposed in Fusai *et al.* (2011): they differ on the method used to solve the linear systems $(I - q_j KD)h_j = b$ in Figure 1. More precisely, while `Base` refers to the case where the linear systems are solved using the GMRes, in Reichel (1989) the author proposes a fast solution method for the one-dimensional Fredholm integral equation

$$H(x,q) = q \int_{\Omega} K(x,y)\, H(y,q)\, dy + (1-q)\phi(x),$$

i.e., Equation (6). Setting $\Omega = (a,b)$, for any $q \in \mathbb{C}$, the idea consists in discretizing the integral equation using a Nystrom quadrature rule based on the N nodes

$$x_j = \frac{1}{2}\left((b+a) + (b-a)\cos\left(\frac{\pi(j-1)}{N-1}\right)\right),\ j = 1, \ldots, N,$$

and the corresponding weights

$$w_j = \frac{\pi}{N-1}\sin\left(\frac{\pi(j-1)}{N-1}\right),\ j = 1, \ldots, N.$$

Therefore, we obtain again a linear system of the form $(I - q_j KD)h_j = b$. The main idea proved in Reichel (1989) is that the matrix K can be well approximated by a matrix K' of rank much smaller than N, returning the new linear system

$$\left(I - qK'D\right) h' = b, \tag{A1}$$

and h' is an approximation to h.

In addition, Reichel proposes an iterative algorithm for the solution of (A1), exploiting a suitable preconditioner (Reichel 1989, Section 4). This algorithm is based on the solution of a $l+1 \times l+1$ linear system rather than $N \times N$, provided that the conditions

$$c_1 N^{\delta} \leq l \leq c_2 N^{2/3}, \quad N > m_1, \tag{A2}$$

are satisfied, for some constants m_1, $0 < c_1 < c_2 < +\infty$, and $0 < \delta < \frac{2}{3}$. In Fusai *et al.* (2011) authors suggest to set $l = N^{2/3}$ and thus conditions (A2) hold. We refer to the original article (Reichel (1989)) for implementation details.
We recall that, as authors shown in (Fusai *et al.* 2011, Tables 7-9), even if the `Reichel` procedure exploits itself the Euler summation, and therefore the number of linear systems to be solved is fixed, increasing the number of monitoring dates the computational cost of the method slightly increases (Fusai *et al.* 2011, Tables 7-8) due to the fact that the number of iterations of the procedure proposed in Reichel (1989) increases itself (Fusai *et al.* 2011, Table 9).

A.2 *The `FM` algorithm*

The pricing method presented in Fusai and Meucci (2008), here denoted with `FM`, is an algorithm based on recursive integrations for pricing Fixed Strike Asian contracts. More precisely, the price of a call option with fixed strike K, $\gamma = 1$, M equidistant monitoring dates and maturity T is equal to

$$e^{-rT} \int_{-\infty}^{+\infty} \left(\frac{S_0}{M+1} \left(1 + e^{x}\right) - K \right)^{+} f_{B_1}(x)\, dx. \tag{A3}$$

The density f_{B_1} is the key variable: it can be computed exploiting the recursion

$$f_{B_{M-k}}(x) = \int_{\mathbb{R}} K(x, y)\, f_{B_{M-k+1}}(x)\, dy, \quad k = 1, \ldots, M-1 \tag{A4}$$

with initial condition $f_{B_M}(x) = f(x)$, and $K(x, y) = f(x - \log(e^{y} + 1))$, being f the transition probability density function from time t to time $t + \Delta$, of the considered Lévy process.
Applying a quadrature rule, with N nodes x_i and weight w_i, to the recursive integral equation (A4), we obtain the recursive formula

$$F_k = (KD) F_{k-1}, \quad k = 1, \ldots, M-1,$$

where the vector and matrices elements are given by

$$F_k(i) = f_{B_{M-k}}(x_i), \quad K(i,j) = K(x_i, x_j), \quad D(i,i) = w_i \text{ and } D(i,j) = 0 \text{ if } i \neq j,$$

for $i, j = 1, \cdots, N$. Therefore the pricing procedure consists in a recursive matrix-vector multiplication procedure, to obtain an approximation of f_{B_1}, and the numerical evaluation of the integral (A3) to obtain the option price. We would like to stress that this pricing procedure, as well as all the procedures based on a recursive computation, e.g., the one presented in Černý and Kyriakou (2011), provides a computational cost which is linear with respect to the number of monitoring dates.