DIPARTIMENTO DI MATEMATICA
"Francesco Brioschi"
POLITECNICO DI MILANO

A parallel wavelet-based pricing procedure for Asian options

Corsaro, S.; Marazzina, D.; Marino, Z.

Collezione dei *Quaderni di Dipartimento*, numero **QDD 161**
Inserito negli *Archivi Digitali di Dipartimento* in data 23-10-2013

Piazza Leonardo da Vinci, 32 - 20133 Milano (Italy)

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

‡Politecnico di Milano, Dipartimento di Matematica, Milano, Italy

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. 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 options 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 date, 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 average. 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 Černý and Kyriakou (2011) and Fusai *et al.* (2011), dealing with both 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 localised both in space and frequency domain. Wavelets bases are thus suitable for adaptive approximation, since the coefficients of the representation contain local information. In particular, for certain operators (Beylkin (1992), Corsaro *et al.* (1999)), if we

* Email: daniele.marazzina@polimi.it

neglect coefficients under a fixed threshold we obtain a sparse, accurate representation of the operator itself; this means that accuracy is preserved with a significative gain in efficiency.
Wavelets have been recognized as effective tools in financial and economic time series analysis (see Gençay *et al.* (2002), Ramsey (2002) among the others and references therein). As an example, in Mariani *et al.* (2008) authors use wavelet expansions to approximate the kernel of an integral equation arising in the context of Heston stochastic volatility model calibration. In Carton De Wiart *et al.* (2011) a PDE method, that makes use of wavelets, for evaluating financial derivatives is introduced.

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 matrix, obtaining a fast algorithm preserving the accuracy of the original method. Numerical results reported in this paper confirm the effectiveness of our approach.

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 the pricing 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; finally, in Section 5. we discuss the parallel implementation.

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

$$\left(I_M - cS_M\right)^+,$$

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

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

for Floating Strike calls; if S_0 is included in the average, we set $\gamma = 1$, otherwise $\gamma = 0$; α is the option coefficient of partiality.

The following recursion holds for the option price:

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

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

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

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

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

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 analitically known for $x \geq 0$ and so the integral equation becomes

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

with

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

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, 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. A wavelet $\psi(t)$ is defined as a function belonging to $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ such that

$$\int_{\mathbb{R}} \psi(t)dt = 0.$$

In probabilistic terms, most of a wavelets mass is concentrated in a compact subset of $\mathbb{R}$, that is, one of the main features of wavelets is "localization". Each wavelet basis is derived by a *mother wavelet* by means of dilation and translation; in particular, the dilation factor corresponds to a scale within the *Multiresolution Analysis* (MRA). Projecting a function onto a space of a MRA allows one to obtain information about it, at a level of detail which depends on the *resolution* of the space. The mapping that leads from the l-th (fine) level resolution to the $(l-1)$-th (coarse) one retaining the information that is lost in this process is the Discrete Wavelet Transform (DWT). More precisely, the DWT of a vector provides both scaling and wavelet coefficients. The former give the coarse representation, the latter the detail information loss when passing from the fine level to the coarse one. If we neglect coefficients under a fixed threshold (*Hard Threshold* technique (HT) Mallat (2008)), accuracy can be preserved with a significative gain in efficiency. Wavelet coefficients allow one to reconstruct the original data, that is, the DWT is an invertible operator.

Each MRA is characterized by two sequences

$$(h_k),\ (g_k),\quad k \in \mathbb{Z},$$

the *low-pass* and the *high-pass filters* respectively. Let $c^l = (c^l_n)_{n\in\mathbb{Z}}$ be the coefficients of the projection of a function $f(t)$ onto the l-th resolution subspace of the MRA; the DWT operator W is defined as follows

$$W:\ c^l \in l^2(\mathbb{Z}) \longrightarrow (c^{l-1}, d^{l-1}) \in l^2(\mathbb{Z}) \times l^2(\mathbb{Z})$$

where $l_2(\mathbb{Z}) = \{(c_k)_{k\in\mathbb{Z}} : c_k \in \mathbb{C}, \sum_k |c_k|^2 < \infty\}$, and

$$\begin{cases} c^{l-1}_n = \sum_{k\in\mathbb{Z}} h_{k-2n} c^l_k, \\ d^{l-1}_n = \sum_{k\in\mathbb{Z}} g_{k-2n} c^l_k. \end{cases} \tag{13}$$

In matrix form, if

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

are the low-pass and the high-pass operator respectively, relations (13) can be written as

$$\begin{pmatrix} c^{l-1} \\ d^{l-1} \end{pmatrix} = \begin{pmatrix} L \\ H \end{pmatrix} \cdot c^l \Longleftrightarrow \begin{cases} c^{l-1} = Lc^l \\ d^{l-1} = Hc^l \end{cases}.$$

The components of the subsequence c^{l-1} are the scaling coefficients, that retain the information about low frequencies, while d^{l-1} contains the wavelet coefficients, that retain the details.
The DWT can be applied recursively as well; at each step, only the scaling coefficients resulting from the previous step are transformed.
Let us recursively define

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

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

$$A^W_s := Q_s A Q_s^\top.$$

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 if the wavelet basis is orthonormal, then all matrices $Q^{(k)}$ are orthogonal, thus $Q_s Q_s^\top = I$. From a computational point of view, it is worth emphasizing that, if m is the length of the two sequences h_k and g_k, then the number of floating-point operations required for the computation of the DWT of a vector of length n is $O(mn)$.

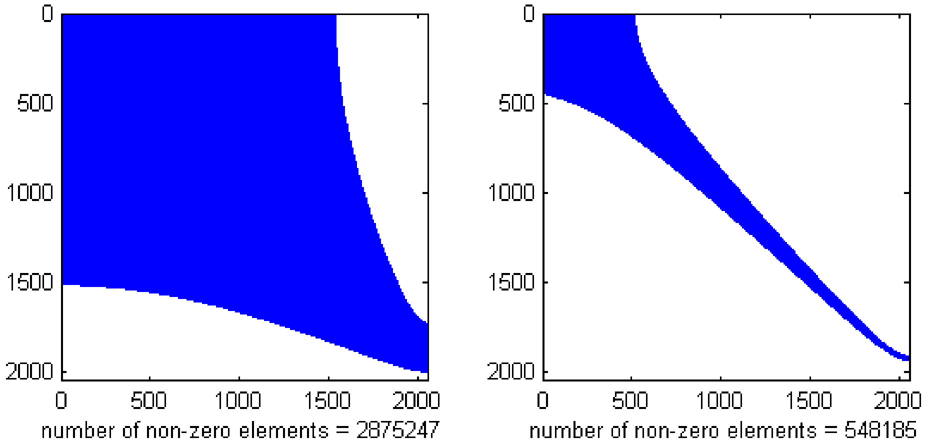

Figure 2. Left: sparsity structure of matrix KD. Right: sparsity structure of the HT of KD with threshold 10^{-8}.

4. The wavelet-based pricing algorithm

In this work, we consider Daubechies wavelets (see Daubechies (1992)), a family of orthonormal compactly supported wavelets. Each family of Daubechies wavelets is characterized by a fixed number of vanishing moments, from which the amplitude of their support depends.
We discretize the integrals in (6) and (7) by means of a quadrature rule on a truncated domain $[l, \lambda]$ for Floating Strike Asian options, $[l, 0]$ for Fixed ones. Different values for 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.
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, decay in magnitude is observed away from this region. This features makes the application of DWT particularly effective. 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 for a threshold level equal to 10^{-8}. We note that the HT operation allows 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

$$(I - qKD_W)h_W = b_W. \tag{14}$$

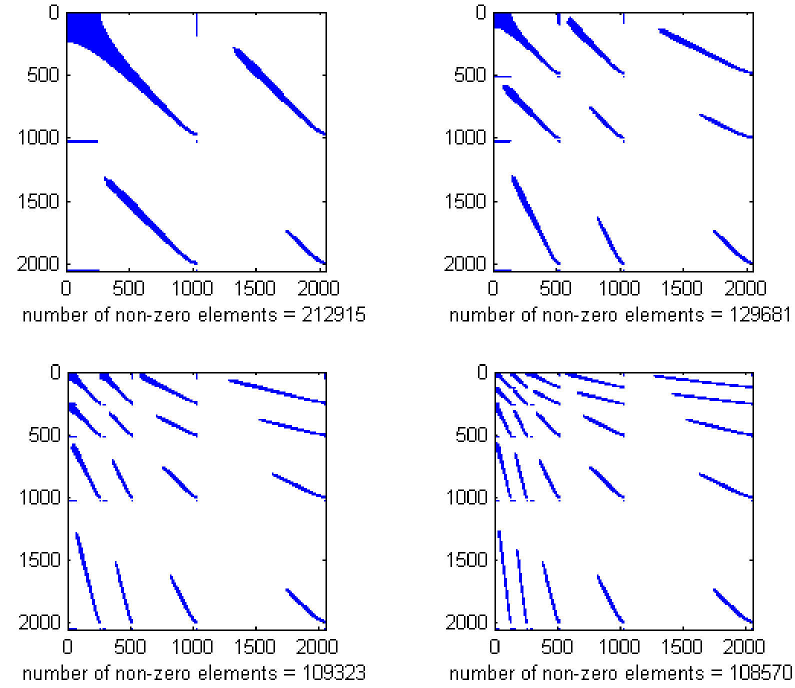

Figure 3. Sparsity structure of the HT of the DWT of matrix KD, with threshold 10^{-8}, for different numbers of transform steps: 1 (up-left), 2 (up-right), 3 (down-left), 4 (down-right).

We then apply a hard threshold to the coefficient matrix of (14), thus we actually solve the linear system

$$(I - q(KD_W)^{\epsilon})y = b_W \tag{15}$$

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 (15), thus an approximation of h, $Q_{\epsilon}^{\top} y$, 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. More precisely, we use the Daubechies wavelets with support $[0, 3]$ (2 vanishing moments), and we show that the integration of the DWT into the pricing procedure allows us to increase its efficiency while preserving accuracy.
In Fusai *et al.* (2011) the linear systems arising from the pricing model were originally solved via GMRes; authors tested a solution method, proposed by Reichel, to speed-up the process. Here we use the GMRes to solve (15). In the following, for sake of brevity, we denote with `GMRes` the original procedure, with `Reichel` the procedure which uses Reichel's method to solve the system and with `DWT`, with an explicit reference to the number of transform steps, the wavelet-based pricing algorithm. We employ the Gauss-Lobatto quadrature rule. 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 our numerical tests, this choice improves accuracy [1].

[1]For our selected values of N, it holds $N + 2500 < 3N$

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	$-\delta\left(\sqrt{\alpha^2-(\beta+i\omega)^2} - \sqrt{\alpha^2-\beta^2}\right)$
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 GMRes) for Fixed Strike Asian option: CGMY process.

		GMRes		Reichel	
N		Error	CPUtime	Error	CPUtime
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
		1 DWT step		2 DWT steps	
N	ϵ	Error	CPUtime	Error	CPUtime
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

In Černý and Kyriakou (2011) benchmarks for Fixed Strike Asian option, with strike $K \in \{90, 100, 110\}$, are given. We use these values to compare accuracy of `GMRes`, `Reichel` and `DWT`. More precisely, 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. We set $T = 1$, $r = 0.04$, $d = 1$, $M = 50$, $\gamma = 1$, $S_0 = 100$, 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). 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 `GMRes` is reported together with the error. These numerical tests refer to a MatLab R2011b implementation [2] on a PC equipped with a Pentium Dual-Core 2.70 GHz and 4 GB RAM.
From these tables, we note that that the `DWT` achieves the same accuracy than `GMRes` 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 execution time of the `DWT`, in percentage, decreases with respect to N for $N \leq 2048$, while it increases for $N = 4096$. 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 investigate the impact of the threshold ϵ and of the number of DWT steps on

[2]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 `GMRes`) for Fixed Strike Asian option: NIG process.

		GMRes		Reichel	
N		Error	CPUtime	Error	CPUtime
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
		1 DWT step		2 DWT steps	
N	ϵ	Error	CPUtime	Error	CPUtime
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

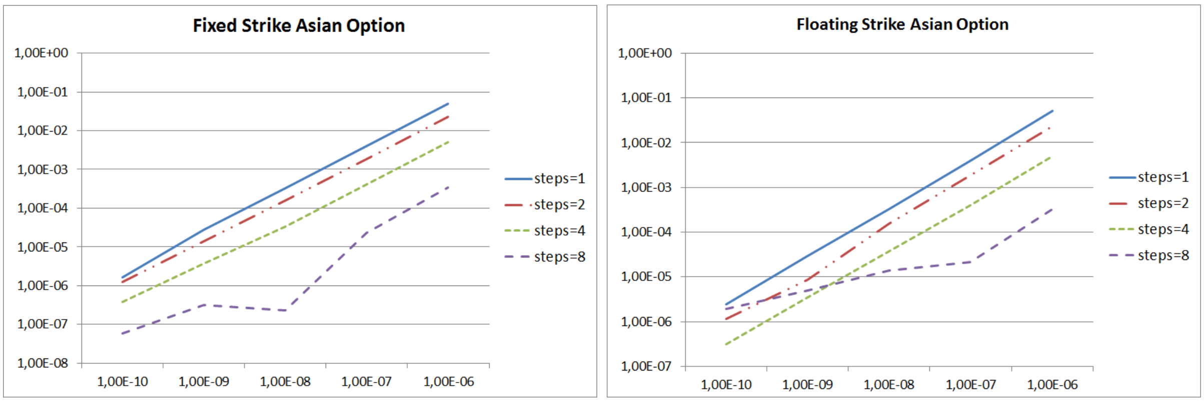

Figure 4. Absolute error: Effects of the threshold tolerance ϵ (x-axis) and of the number of DWT steps.

accuracy, 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. The implemented software 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)).

More precisely, in Figure 4 we present two test cases taken from Fusai *et al.* (2011): a Fixed and a Floating Strike Asian option, assuming a Jump Diffusion (JD) distribution with parameters $r = 0.0367$, $d = 0$, $T = 1$, $S_0 = 100$, $M = 50$, $K = 100$, $\gamma = 1$, $\sigma = 0.126349$, $\lambda = 0.174814$, $\alpha = -0.390078$, 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 does not exceed 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 `GMRes`, 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 $DWTsteps = 8$. For further details, we refer to Section 5.1, where we analyze the performances of the parallel algorithm.

5. Parallel implementation

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

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

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: this slightly increases parallel efficiency.

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 ϵ;
- 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 $T = 1$, $r = 0.04$, $d = 1$ 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

$$S_{np} = \frac{T_1}{T_{np}} \tag{16}$$

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 faster the parallel program is. Typically S_{np} is a number between 1 and np. The ideal speed-up is $S_{np} = np$. An important theoretical limit to the maximum improvement obtainable by a parallel algorithm is Amdahl's Law. It states that

$$S_{np} \leq \frac{1}{\alpha + \frac{1-\alpha}{np}} \tag{17}$$

where α, with $0 < \alpha < 1$, is the fraction of the algorithm which is strictly serial.

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.

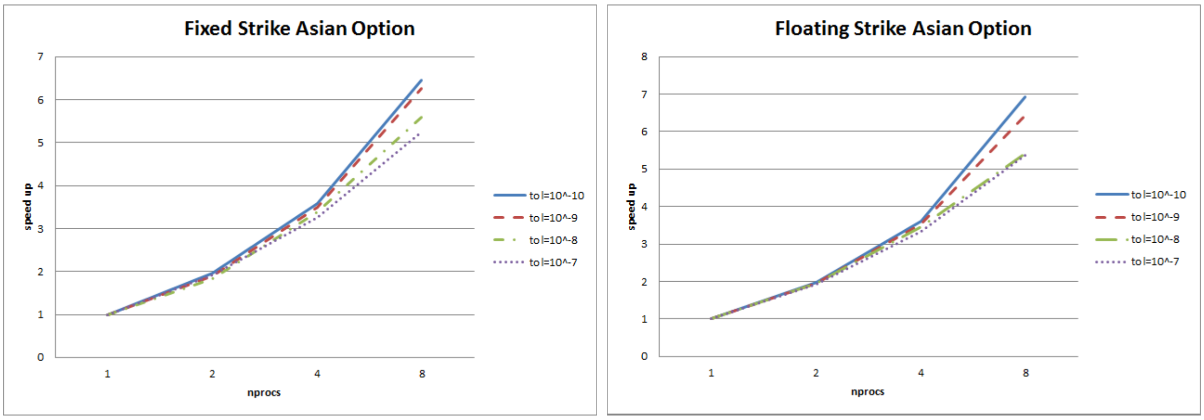

Figure 6. Speed-up: Effects of the threshold tolerance ϵ for $N = 2^{11}$ and 8 DWT steps. Left: Fixed Strike Asian Option; right: Floating Strike Asian Option.

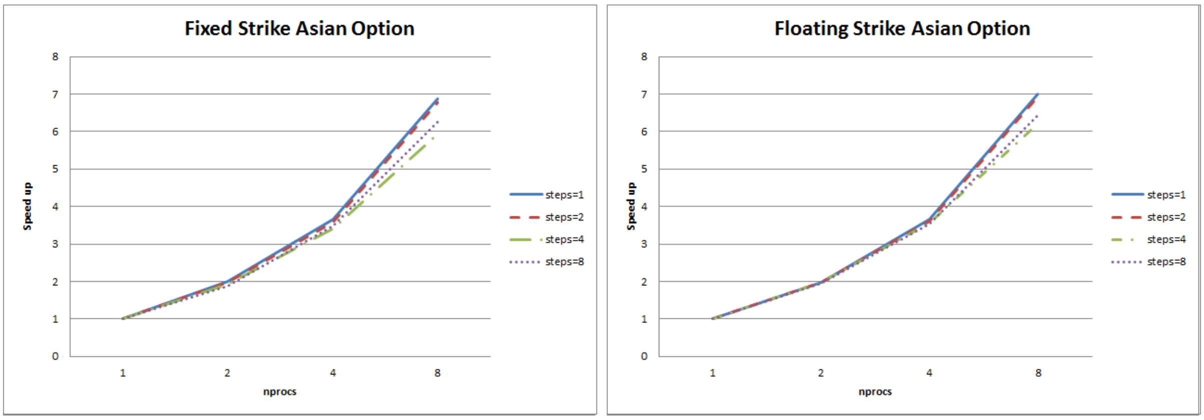

Figure 7. Speed-up: Effects of the number of DWT steps, with $N = 2^{11}$ and ϵ set at 10^{-9}. Left: Fixed Strike Asian Option; right: Floating Strike Asian Option.

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 systems arising from the involved integral equations. Numerical results, shown in this work, confirm the effectiveness of our approach.
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. Our wavelet-based method can instead be used to speed-up the solution of the arising linear systems, like in this work. This is at present work in progress.

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.

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

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.

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).

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.

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

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.

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

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