DIPARTIMENTO DI MATEMATICA
"Francesco Brioschi"
POLITECNICO DI MILANO

Wavelet techniques for option pricing on advanced architectures

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

Collezione dei *Quaderni di Dipartimento*, numero **QDD 88**
Inserito negli *Archivi Digitali di Dipartimento* in data 8-3-2011

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

QFinLab Working Paper Series

Working Paper Series of the Quantitative Finance Laboratory *QFinLab*.
The main research areas include: mathematical finance, corporate finance, asset allocation, computational finance, financial engineering, financial time series analysis, corporate governance.

QDD 42 L. V. Ballestra & C. Sgarra, *The Evaluation of American Options in a Stochastic Volatility Model with Jumps: a Finite Element Approach* (2009)

QDD 81 E. Barucci & F. Gazzola, *Prices in the utility function and demand monotonicity* (2010)

QFinLab website: www.mate.polimi.it/qfinlab

Wavelet techniques for option pricing on advanced architectures

Stefania Corsaro[1,2], Daniele Marazzina[3], and Zelda Marino[1]

[1] Dipartimento di Statistica e Matematica per la Ricerca Economica
Università degli Studi di Napoli "Parthenope"
Via Medina 40, I-80133 Napoli, Italy
{stefania.corsaro,zelda.marino}@uniparthenope.it
[2] Istituto per il Calcolo Parallelo e i Supercalcolatori (ICAR-CNR)
Via Pietro Castellino 111, I-80131 Napoli, Italy
[3] Dipartimento di Matematica - Politecnico di Milano
Via Bonardi 9, I-20133 Milano, Italy
daniele.marazzina@polimi.it

Abstract. This work focuses on the development of a parallel pricing algorithm for Asian options based on the Discrete Wavelet Transform. Following the approach proposed in [6], the pricing process requires the solution of a set of independent Fredholm integral equations of the second kind. Within this evaluation framework, our aim is to develop a robust parallel pricing algorithm based on wavelet techniques for the pricing problem of discrete monitoring arithmetic Asian options. In particular, the Discrete Wavelet Transform is applied in order to approximate the kernels of the integral equations. We discuss both the accuracy of the method and its scalability properties.

Keywords: Asian options, Discrete Wavelet Transform, Parallel Computing

1 Introduction

The backward recursion that arises in option pricing can be converted into a set of independent Fredholm integral equations of the second kind by means of the z-transform. This approach is described in [7] for European, Barrier and Lookback options and in [6] for Asian options. Moreover, the development of a grid-enabled pricing algorithm for plain vanilla options is presented in [5].
In this paper we focus on the pricing procedure for Asian options, based on the randomization technique described in [6]; as authors point out, the pricing procedure turns out to be computational demanding. Our purpose in this framework is to develop an accurate

and efficient pricing algorithm based on wavelet techniques. In probabilistic terms, most of a wavelet mass is concentrated in a compact subset of $\mathbb{R}$, that is, one of the main features of wavelets is "localization". This property motivates the use of wavelet bases for data compression: wavelet coefficients contain local information, thus, if we neglect the coefficient under a fixed threshold, accuracy can be preserved with a significative gain in efficiency. Even if most applications of wavelets deal with signal analysis, wavelets have been applied in the numerical solution of partial differential and integral equations, and in the approximation and interpolation of data [1]. Much effort has been indeed devoted to the development of routines that perform the computation of the *Discrete Wavelet Transform* (DWT) both on serial and parallel architectures (see, for example, [2] and [4] and references therein).
We project the linear systems which arise from the discretization of the integral equations onto wavelet spaces in order to obtain a sparse representation of the discrete operators, preserving information so to preserve accuracy. We furthermore discuss the parallelization of the pricing wavelet-based procedure.

In Section 2 we briefly describe the pricing method, addressing to existing literature for details. In Section 3 we introduce the DWT operator. In Section 4 we describe the wavelet-based pricing algorithm we developed, which is tested in Section 5. Finally, Section 6 deals with the parallel implementation and the performance analysis.

2 The Randomization pricing algorithm

In this section we briefly recall the Asian fixed call randomization pricing algorithm presented in [6]. Authors show that, under the assumption that the underlying asset evolves according to a generic Lévy process, the price of a call option with fixed strike K, N equidistant monitoring dates (Δ being the time interval between them) and maturity T is equal to

$$e^{-rT} \int_{-\infty}^{+\infty} \left(\frac{S_0}{N+1} \left(1 + e^x\right) - K \right)^+ f_{B_1}\left(x\right) dx \qquad (1)$$

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

$$u(x,k) = \int_{\mathbb{R}} K(x,y)\, u(y,k-1)\, dy,\ \ k = 1,\ldots,N-1 \tag{2}$$

with initial condition $u(x,0) = f(x)$, where $u(x,k) = f_{B_{N-k}}(x)$ 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.
The randomization technique consists in making the expiry date T to be random according to a geometric distribution of the parameter q and then computing the value of $U(x,q) := (1-q)\sum_{k=0}^{+\infty} q^k u(x,k)$. With some manipulations on (2), we get that the function $U(x,q)$ satisfies the integral equation:

$$U(x,q) = q\int_{\mathbb{R}} K(x,y)\, U(y,q)\, dy + (1-q) f(x) \tag{3}$$

Therefore a recursive integral equation for $u(x,k)$ is transformed into an integral equation for $U(x,q)$. If we approximate the integral equation (3) with a quadrature rule with nodes x_i, $i = 1,\cdots,m$, we obtain the linear system

$$\mathbf{u} - q\mathbf{K}\mathbf{D}\mathbf{u} = \mathbf{f} \tag{4}$$

with $(\mathbf{u})_i = U(x_i)$, $(\mathbf{K})_{ij} = K(x_i,x_j)$, $(\mathbf{f})_i = (1-q)f(x_i)$ and $\mathbf{D}$ being the diagonal matrix of the quadrature weights. The system (4) is the main computational kernel in the procedure. The unknown function $u(x,N-1)$, i.e., $f_{B_1}(x)$, can be then obtained by de-randomizing the option maturity exploiting the complex inversion integral

$$u(x,N-1) = \frac{1}{2\pi\rho^{N-1}} \int_0^{2\pi} \frac{U(x,\rho e^{is})}{1-\rho e^{is}} e^{-i(N-1)s} ds \tag{5}$$

Approximating (5) with a trapezoidal formula, and applying the Euler summation, a convergence-acceleration technique well suited for evaluating alternating series, we obtain

$$f_{B_1}(x) = u(x,N-1) \approx \frac{1}{2^{m_e}\rho^{N-1}} \sum_{j=0}^{m_e} \binom{m_e}{j} b_{n_e+j}(x)$$

where $b_k(x) = \sum_{j=0}^{k}(-1)^j a_j U(x, q_j)$ with $q_j = \rho e^{ij\pi/(N-1)}$, $a_0 = (2(N-1)(1-q_0))^{-1}$, $a_j = ((N-1)(1-q_j))^{-1}$, $j \geq 1$, assuming $N > n_e + m_e$. A sketch of the pricing algorithm is reported in Fig. 1.

Procedure
- `compute` $\mathbf{K}$, $\mathbf{D}$ `and` $\mathbf{f}$
- `for` $j = 0, \cdots, n_e + m_e$, `solve the integral equations` $(\mathbf{I} - q_j\mathbf{KD})\mathbf{u} = \mathbf{f}$
- `reconstruct` $f_{B_1}(x)$ `by means of the solutions of the integral equation`
- `compute the integral` (1)

End Procedure

Fig. 1. Asian fixed call randomization pricing algorithm.

3 The Discrete Wavelet Transform

A wavelet $\psi(t)$ is defined as a function belonging to $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ such that $\int_{\mathbb{R}} \psi(t)dt = 0$. Wavelets have either compact support or the most of information contained in them is concentrated in a compact subset of $\mathbb{R}$ [3]. 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, depending on the *resolution* of the space. The mapping that leads from the l-th level resolution to the $(l-1)$-th level, retaining the information that is lost in this process, is the Discrete Wavelet Transform. The aforementioned properties justify the use of DWT for data compression [9]: wavelet coefficients contain the detail information, thus, if we neglect the coefficients under a fixed threshold, accuracy can be preserved with a significative gain in efficiency.

Given a MRA, two sequences $(h_k)_{k\in\mathbb{Z}}$ and $(g_k)_{k\in\mathbb{Z}}$, the *low-pass* and the *high-pass filters* of the MRA, respectively, define a change of level within the MRA. More precisely, let $\mathbf{c}^l = (c_n^l)_{n\in\mathbb{Z}}$ be the vector of 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:

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

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

$$\begin{cases} c_n^{l-1} = \sum_{k\in\mathbb{Z}} h_{k-2n} c_k^l \\ d_n^{l-1} = \sum_{k\in\mathbb{Z}} g_{k-2n} c_k^l. \end{cases}$$

In matrix form, if $\mathbf{L} = (\tilde{h}_{i,j} = h_{j-2i})$ is the low-pass operator and $\mathbf{H} = (\tilde{g}_{i,j} = g_{j-2i})$ is the high-pass operator, the above relation can be written in the following way:

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

The vector $\mathbf{c}^{l-1}$ retains the information about the low frequencies, while the filters g_k "detect" the high frequencies: so the vector $\mathbf{d}^{l-1}$ contains the *details*, that is, the information that is lost passing from the resolution l to the resolution $l-1$. From a computational point of view, it is worth emphasizing that, if s 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 m is $O(sm)$.
If $\mathbf{Q} := (\mathbf{L},\ \mathbf{H})^\top$, then the DWT of a matrix $\mathbf{A}$ is defined as $\mathbf{QAQ}^\top$. In practice, the bidimensional DWT is computed in two stages: the product $\mathbf{QA}$ actually requires to transform the columns of the matrix; then, the DWT is applied to the rows of the intermediate matrix $\mathbf{QA}$. Note that if the wavelet basis is orthonormal, then the matrix $\mathbf{Q}$ is orthogonal, thus $\mathbf{QQ}^\top = \mathbf{I}$.

4 The wavelet-based pricing algorithm

In this work, we consider the Daubechies Wavelets [3], 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 the support depends. Our idea is to increase the sparsity of the coefficient matrices of the linear systems to be solved, so to improve efficiency, by means of a *hard threshold*

[9] applied to the projection of the discrete operators onto wavelet spaces, which allows one to better preserve information for the sake of accuracy.
Let us discretize the (3) by means of a quadrature rule on a truncated integration domain $[l, u]$. The bounds are chosen in such a way that the tails of the density outside of it are less than 10^{-8} [6]. Moreover, since f is the transition probability density of the log-price of the underlying asset, it is reasonable to expect a decay of the values of f moving towards l or u. For this reason, we expect the most significant elements of the matrix $\mathbf{KD}$ to be localized in a region near its diagonal, and smoothness away from this region. Let us refer to the pricing algorithm reported in Fig. 1: in the solution of the linear systems in step 2, we apply to both sides the DWT operator $\mathbf{Q}$, so for each value of q we obtain the linear system, equivalent to (4), $(\mathbf{I}-q\mathbf{Q}(\mathbf{KD})\mathbf{Q}^\top)\mathbf{Qu} = \mathbf{Qf}$. Therefore, if we denote by $\mathbf{KD}_W, \mathbf{u}_W, \mathbf{f}_W$ the DWT of $\mathbf{KD}, \mathbf{u}, \mathbf{f}$ respectively, we have:

$$(\mathbf{I} - q\mathbf{KD}_W)\mathbf{u}_W = \mathbf{f}_W \tag{6}$$

We then apply a *hard threshold* to the coefficient matrix of (6), thus we actually solve the linear system:

$$(\mathbf{I} - q\mathbf{KD}^\epsilon_W)\mathbf{y} = \mathbf{f}_W \tag{7}$$

where $\mathbf{KD}^\epsilon_W$ is the hard threshold of $\mathbf{KD}$ with threshold ϵ. Finally, the inverse DWT is applied to the solution $\mathbf{y}$ of (7), thus an approximation of $\mathbf{u}$, $\mathbf{Q}^\top\mathbf{y}$, is obtained.

5 Numerical results

In this section we price an Asian fixed option with 100 monitoring dates, maturity $T = 1$ and strike $K = 100$. The Market data are $S_0 = 100$, $r = 3.67\%$ and the underlying asset is assumed to follow a Jump Diffusion Merton Lévy process with parameters $\sigma = 0.126349$, $\alpha = -0.390078$, $\lambda = 0.174814$ and $\delta = 0.338796$. All the computations have been performed in Matlab using an Intel Personal Computer equipped with 6GB of RAM and Intel Core i7 Q720-1600MHz processor.
We consider a Gauss-Legendre quadrature rule with $m = 2048$ nodes

and three methods to solve the integral equations: the “standard” quadrature method with Gauss-Legendre nodes, the wavelet transform method, and the Reichel algorithm, which is a fast solution method for integral equations based on a low-rank representation of the kernel of the equation. The quadrature rule considered for the wavelet transform method is the Gauss-Legendre one. We fix $n_e = 12$ and $m_e = 10$.

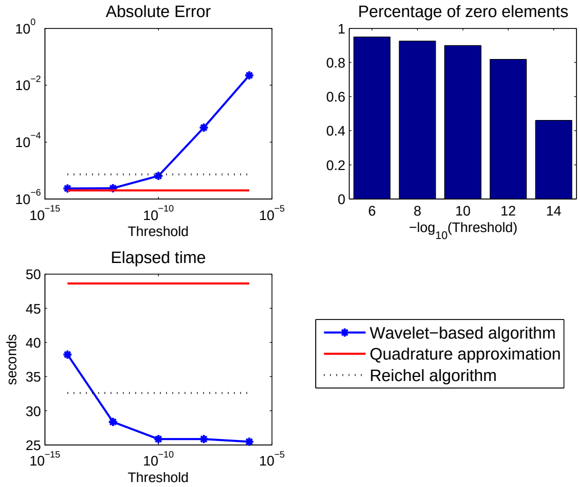

Fig. 2. Level 2 Daubechies wavelets. Top-left: approximation error; top-right: percentage of neglected elements in the hard threshold following the DWT; bottom-left: execution time.

In Fig. 2 (3) we report the results concerning a simulation in which two (four) steps of DWT, based on Daubechies wavelets with four vanishing moments, have been performed, for different threshold values ranging between 10^{-14} and 10^{-6}. In the top-left picture the absolute error is represented for different threshold levels, considering the results in [6] as exact solution, the corresponding execution time being represented in the bottom-left graphic. In both Figures 2-3, we see that the DWT-based approach is almost always the most efficient and the approximation error has the same order of magnitude up to 10^{-10} threshold level. On the other hand, when the threshold is in the range $10^{-10} - 10^{-6}$ more than the 80% of the elements are set to zero, as it can be seen in the top-right graphic,

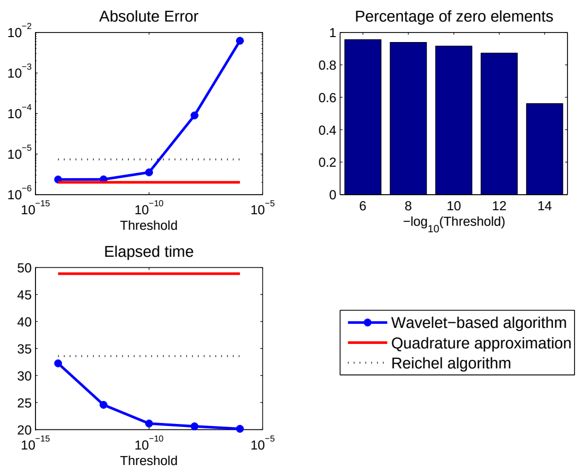

Fig. 3. Level 4 Daubechies wavelets. Top-left: approximation error; top-right: percentage of neglected elements in the hard threshold following the DWT; bottom-left: execution time.

where the percentage of elements which are neglected in the threshold procedure are reported, for different threshold values. Finally, going from level 2 (Fig.2) to level 4 (Fig.3) we notice an increase of the computational efficiency of the wavelet-based approach.

6 Parallel implementation

The performances of our method can be improved using High Performance Computing methodologies. Parallelism has been introduced both in the linear systems solution process and in the DWT computation. In this section we describe the parallel algorithm and we present numerical results from the implementation of the developed software.

As already pointed out, the computation of the bidimensional DWT is performed in two stages; we distribute the matrix **KD** in a row-block fashion. In the first stage, processors concurrently compute the DWT of rows; then, communication is required for globally transposing the matrix, so, processors can concurrently transform the columns of the intermediate matrix. Finally, the matrix is globally transposed again.

While to apply the DWT the matrix **KD** has to be distribute among processor, to solve the linear system each processor can build independently from the others the coefficients matrix. We have then $N_{sys} = n_e + m_e$ linear systems to be solved. We distribute them among processors, so that each one solves $\lfloor N_{sys}/nprocs \rfloor$ systems; in this phase, processors work concurrently. In Fig. 4 a sketch of the algorithm is reported.

Procedure
- `apply the DWT operator to` $\mathbf{KD}$ `and to` $\mathbf{f}$ `in parallel`
  - `distribute the matrix` $\mathbf{KD}$ `in row-block fashion`
  - `each processor computes the DWT of a row block`
  - `global transposition`
  - `each processor computes the DWT of a column block`
  - `global transposition`
- `collect the matrix so that each processor stores the whole matrix`
- `neglect the elements of` $\mathbf{KD}_W$ `below the fixed threshold` ϵ;
- `for` $j = 0, \cdots, N_{sys}$, `solve in parallel` $(\mathbf{I} - q_j \mathbf{KD}^{\epsilon}_W)\mathbf{y} = \mathbf{f}_W$
  - `distribute the values of` q `among processors;`
  - `each processor solves the linear systems distributed to it;`
  - `each processor applies the inverse transform to` $\mathbf{y}$.
- `processor` 0 `collects local solutions`

End Procedure

Fig. 4. Sketch of the parallel pricing algorithm for Asian options based on the DWT.

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. The implemented software is written in C language, using the Message Passing Interface (MPI) communication system. We use the freely available GSL Library [8] to perform the wavelet transform, while for the global matrix transposition we use the routine `pdtrans` of the PUMMA library [10].
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 [12].

To evaluate the parallel performance of the algorithm, in Fig. 5 we report the speed-up for $m = 2^{10}$, $m = 2^{11}$ and $m = 2^{12}$, considering the same pricing problem presented in Section 5. We use the

Daubechies wavelets of length 4 and with 4 level of resolution. The graph reveals a decrease in terms of performance with four processors. This is due to the communication overhead of the global transposition. Better results could be obtained if the transposed matrices were built, so to avoid one transposition, as we plan to do in the next future.

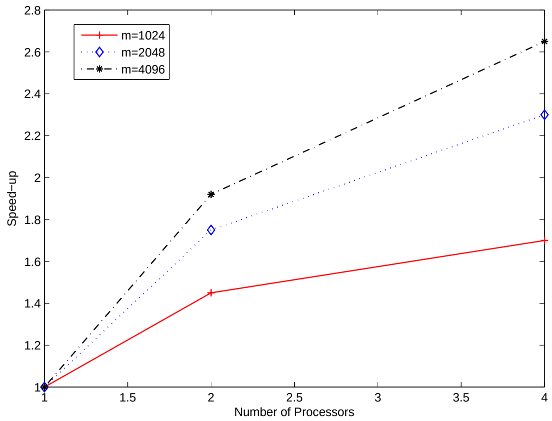

Fig. 5. Speed-up

7 Conclusion

In this paper we focus on the use of wavelet techniques in a pricing procedure for Asian options based on randomization. Preliminary experiments reveal that wavelet bases allow one to improve efficiency without loss in accuracy. Moreover, we discuss the parallelization of the proposed algorithm; parallelism is introduced at two levels, both in the wavelet transform and in the solution of the linear systems arising from the discretization of the involved integral operators. Parallel performance results reveal that the parallel algorithm needs some revisions which we are planning to implement in the next future.

References

1. Cohen, A.: Numerical analysis of wavelet methods. Elsevier, Amsterdam (2003)
2. Corsaro, S., D'Amore, L., Murli, A.: On the Parallel Implementation of the Fast Wavelet Packet Transform on MIMD Distributed Memory Environments. In: Zinterhof, P., Vajteršic, M., Uhl, A. (eds.) ParNum'99. LNCS, vol. 1557, 357–366. Springer, Heidelberg (1999)
3. Daubechies, I.: Ten lectures on wavelets. Society for Industrial and Applied Mathematics, Philadelphia (1992)
4. Franco, J., Bernabé, G., Fernández, J., Acacio, M.E.: A Parallel Implementation of the 2D Wavelet Transform Using CUDA. In: PDP '09: Proceedings of the 2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing, pp. 111-118. IEEE Computer Society, Washington (2009)
5. Fusai, G., Marazzina, D., Marena, M.: Option Pricing, Maturity Randomization and Distributed Computing, Parallel Comput., 36-7, 403-414 (2010)
6. Fusai, G., Marazzina, D., Marena, M.: Pricing Discretely Monitored Asian Options by Maturity Randomization, SEMeQ Working Paper, University of Piemonte Orientale(2009). To appear in *SIAM Journal on Financial Mathematics*
7. Fusai, G., Marazzina, D., Marena, M., Ng, M.: Z-Transform and Preconditioning Techniques for Option Pricing, SEMeQ Working Paper, University of Piemonte Orientale (2009). To appear in *Quantitative Finance*
8. Galassi, M., et al: GNU Scientific Library Reference Manual (2009)
9. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press, San Diego (2008)
10. Choi, J., Dongarra, J.J., Petitet, A., Walker, D.W.: Pumma Reference Manual, http://www.netlib.org/scalapack (1993)
11. Reichel, L.: Fast Solution Methods for Fredholm Integral Equations of the Second Kind, Numer. Math., 57-1, 719-736 (1989)
12. 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)