Published in Proceedings of the International Conference "Wavelets and Applications", Toulouse, 1992;
Y. Meyer and S. Roques, edt., Editions Frontieres, 1993

On the fast algorithm for multiplication of functions in the wavelet bases

G. Beylkin
Program in Applied Mathematics
University of Colorado at Boulder
Boulder, CO 80309-0526

I Introduction

The wavelet bases provide a system of coordinates in which wide classes of linear operators are sparse. As a result, the cost of evaluating Calderón-Zygmund or pseudo-differential operators on a function is proportional to the number of significant wavelet coefficients of this function, i.e., the number of wavelet coefficients above a given threshold of accuracy. Consequently, fast algorithms are now available for solving integral equations with operators from these classes [3].

In order to use the wavelet bases for solving partial differential equations, one is led to consider differential operators and operators of multiplication by a function. Numerical issues of representing differential operators has been addressed in [2] and, it turns out, such operators require $O(1)$ coefficients for their description in the wavelet bases (in the non-standard form).

On the other hand, the operator of the multiplication by a function $a(x)$, $x \in \mathbf{R}^d$, seems to require $O(N^d)$ coefficients for its description independently of the properties of the function $a(x)$. Indeed, the operator of the multiplication by $a(x)$ has the generalized kernel $a(x)\delta(x-y)$ with the singularity along the diagonal $x = y$ even if the function $a(x)$ is smooth and non-oscillatory.

Heuristically, if the solution of a partial differential equation is smooth and non-oscillatory on most of its support but is singular or oscillatory at a few locations, then solving in the wavelet bases should lead to fast and adaptive algorithms where the number of operations is proportional to the number of the significant coefficients in the representation of the solution in the wavelet bases. However, if in the process of solving this equation it is necessary to multiply the solution by a smooth function then, due to our previous remark, the algorithm will be insensitive to the smoothness properties of the functions involved. Indeed, this point is most clear for the nonlinear equations. For example, considering Burgers equation we observe that it is necessary to compute the square of the solution at each time step and, therefore, the gains in the sparsity of the representation of the solution are lost.

In this paper we address the problem of pointwise multiplication of functions in the wavelet bases. We will consider computing $F(u) = u^2$ in the wavelet bases since the product of two functions may be written as $uv = \frac{1}{4}[(u+v)^2 - (u-v)^2]$.

It appears that the straightforward algorithm which would require computing the expansion of the products of the basis functions, storing and using them to perform the multiplication is inefficient. Such algorithm requires computing the coefficients

$$c_{k,k',l}^{j,j',m} = \int_{-\infty}^{+\infty} \psi_k^j(x)\, \psi_{k'}^{j'}(x)\, \psi_l^m(x)\, dx,$$

where $\psi_k^j(x) = 2^{-j/2}\psi(2^{-j}x - k)$ are the basis functions. While computing $c_{k,k',l}^{j,j',m}$ does not present a problem, the number of the nonzero of coefficients is large and, what is more important, the number of operations to compute u^2 is proportional to N_s^3, where N_s is the number of significant coefficients in the representation of u.

In a number of applications the functions of interest are the functions that are singular or oscillatory at a few locations. The number of significant wavelet coefficients of such functions is $O(1)$ on each scale so that N_s is proportional to $\log(N)$. Asymptotically the straightforward algorithm will perform better that an $O(N)$ algorithm, but for problems of practical size it is inefficient.

In this paper we develop a novel approach to the pointwise multiplication of functions in the wavelet bases based on uncoupling the interactions between scales. The complexity of this algorithm is automatically adaptable to the complexity of the wavelet representation of u and is proportional to N_s. A preliminary version of this algorithm was presented at INRIA [1]. The algorithm permits a generalization for computing $F(u)$ directly in the wavelet basis, where F is a smooth function and u is represented in a wavelet basis.

II Multiresolution algorithm for evaluating u^2

II.1 Uncoupling the interaction between scales

Let us consider the projections of $u \in \mathbf{L}^2(\mathbf{R})$ on subspaces $\mathbf{V}_j$,

$$u_j = P_j u, \quad u_j \in \mathbf{V}_j, \tag{2.1}$$

where $\{\mathbf{V}_j\}_{j\in\mathbf{Z}}$,is a multiresolution analysis of $\mathbf{L}^2(\mathbf{R})$. In order to uncouple the interaction between scales, we write a "telescopic" series,

$$u_0^2 - u_n^2 = \sum_{j=1}^{j=n} \left[(P_{j-1}u)^2 - (P_j u)^2\right] = \sum_{j=1}^{j=n} (P_{j-1}u + P_j u)(P_{j-1}u - P_j u) \tag{2.2}$$

Using $P_{j-1} = P_j + Q_j$, we obtain

$$u_0^2 - u_n^2 = \sum_{j=1}^{j=n} (2P_j u + Q_j u)(Q_j u), \tag{2.3}$$

or

$$u_0^2 = 2\sum_{j=1}^{j=n} (P_j u)(Q_j u) + \sum_{j=1}^{j=n} (Q_j u)(Q_j u) + u_n^2. \tag{2.4}$$

In (2.4) there is no interaction between different scales j and j', $j \neq j'$. Let consider each term of (2.4) as a bilinear mapping

$$M_{VW}^j : \mathbf{V}_j \times \mathbf{W}_j \to \mathbf{L}^2(\mathbf{R}) = \mathbf{V}_j \bigoplus_{j'\leq j} \mathbf{W}_{j'}, \tag{2.5}$$

and

$$M_{WW}^j : \mathbf{W}_j \times \mathbf{W}_j \to \mathbf{L}^2(\mathbf{R}) = \mathbf{V}_j \bigoplus_{j'\leq j} \mathbf{W}_{j'}, \tag{2.6}$$

and note that in (2.5) and (2.6) we select the representation of $\mathbf{L}^2(\mathbf{R})$ depending on the scale j.

Remark 1. For the numerical purposes we need formulas (2.3) or (2.4) with a finite number of scales, though it is clear that by taking limits $j \to \infty$ and $j \to -\infty$ we have

$$u^2 = \sum_{j\in\mathbf{Z}} (2P_j u + Q_j u)(Q_j u), \tag{2.7}$$

which is essentially the para-product of J.M. Bony [5], [6], [4], [7].

II.2 Computing u^2 in the Haar basis

Let us start by considering an example of (2.4) in the Haar basis. We have the following explicit relations,

$$\begin{aligned} (\chi_k^j(x))^2 &= 2^{-j/2}\chi_k^j(x), \\ (h_k^j(x))^2 &= 2^{-j/2}\chi_k^j(x), \\ \chi_k^j(x)h_k^j(x) &= 2^{-j/2}h_k^j(x), \end{aligned} \tag{2.8}$$

where $\chi_k^j(x) = 2^{-j/2}\chi(2^{-j}x-k)$, $h_k^j(x) = 2^{-j/2}h(2^{-j}x-k)$, χ is the characteristic function of the interval $(0,1)$ and h is the Haar function, $h(x) = \chi(2x) - \chi(2x-1)$.

Expanding u_0 into the Haar basis,

$$u_0(x) = \sum_{j=1}^{j=n}\sum_{k\in\mathbf{Z}} d_k^j h_k^j(x) + \sum_{k\in\mathbf{Z}} s_k^n \chi_k^n(x), \tag{2.9}$$

and using (2.8), we obtain from (2.4)

$$u_0^2(x) = 2\sum_{j=1}^{j=n} 2^{-j/2}\sum_{k\in\mathbf{Z}} d_k^j s_k^j\, h_k^j(x) + \sum_{j=1}^{j=n} 2^{-j/2}\sum_{k\in\mathbf{Z}} (d_k^j)^2\, \chi_k^j(x) + 2^{-n/2}\sum_{k\in\mathbf{Z}} (s_k^n)^2\, \chi_k^n(x). \tag{2.10}$$

On denoting

$$\begin{aligned} \hat{d}_k^j &= 2^{-j/2+1} d_k^j s_k^j, \\ \hat{s}_k^j &= 2^{-j/2}(d_k^j)^2, \\ \hat{\hat{s}}_k^n &= 2^{-n/2}(s_k^n)^2, \end{aligned} \tag{2.11}$$

we rewrite (2.10) as

$$u_0^2(x) = \sum_{j=1}^{j=n}\sum_{k\in\mathbf{Z}} \hat{d}_k^j\, h_k^j(x) + \sum_{j=1}^{j=n}\sum_{k\in\mathbf{Z}} \hat{s}_k^j\, \chi_k^j(x) + \sum_{k\in\mathbf{Z}} \hat{\hat{s}}_k^n\, \chi_k^n(x). \tag{2.12}$$

Remark 2. We note that *if the coefficient d_k^j is zero then there is no need to keep the corresponding average s_k^j*. In other words, we need to keep averages only near the singularities, i.e., where the wavelet coefficients d_k^j (or products $s_k^j d_k^j$) are significant for a given accuracy.

Finally, to compute the coefficients of the wavelet expansion of the function u_0^2, we need to expand the second sum in (2.12) into the wavelet basis. Starting from the

scale $j = 1$, we compute the differences and averages $\bar{d}_k^{j+1}$ and $\bar{s}_k^{j+1}$. We then add $\bar{s}_k^{j+1}$ to $\hat{s}_k^{j+1}$ before expanding it further according to the following pyramid scheme

$$\begin{array}{ccccccccc} \{\hat{s}_k^1\} & \longrightarrow & \{\bar{s}_k^2\} & \longrightarrow \{\bar{s}_k^2\} + \{\hat{s}_k^2\} & \longrightarrow & \{\bar{s}_k^3\} & \longrightarrow \{\bar{s}_k^3\} + \{\hat{s}_k^3\} & \cdots & \\ & \searrow & & & \searrow & & & \searrow & \\ & & \{\bar{d}_k^2\} & \longrightarrow \{\bar{d}_k^2\} + \{\hat{d}_k^2\} & & \{\bar{d}_k^3\} & \longrightarrow \{\bar{d}_k^3\} + \{\hat{d}_k^3\} & \cdots & \end{array} \tag{2.13}$$

(The formulas for evaluating the differences and averages $\bar{d}_k^{j+1}$ and $\bar{s}_k^{j+1}$ may be found in [3]). As a result, we compute $\bar{d}_k^j$, $j = 2, \ldots, n$, (we set $\bar{d}_k^1 = 0$) and $\bar{s}_k^n$ and obtain

$$u_0^2(x) = \sum_{j=1}^{j=n} \sum_{k \in \mathbf{Z}} (\hat{d}_k^j + \bar{d}_k^j)\, h_k^j(x) + \sum_{k \in \mathbf{Z}} (\bar{s}_k^n + \hat{s}_k^n + \hat{\hat{s}}_k^n)\, \chi_k^n(x). \tag{2.14}$$

It is clear, that the number of operations for computing the Haar expansion of u_0^2 is proportional to the number of significant coefficients d_k^j in the wavelet expansion of u_0. In the worst case, if the original function is represented by a vector of the length N, then the number of operations is proportional to N. If the original function is represented by $O(\log_2 N)$ significant Haar coefficients, then the number of operations to compute its square is proportional to $\log_2 N$. The algorithm in the Haar basis easily generalizes to the multidimensional case.

II.3 Computing u^2 in the wavelet bases

We now return to the general case of wavelets and derive an algorithm to expand (2.4) into the wavelet bases. Unlike in the case of the Haar basis, the product on a given scale "spills over" into the finer scales and we develop an efficient approach to handle this problem. We use compactly supported wavelets though our considerations are not restricted to such wavelets. We denote the scaling function by ϕ and the wavelet by ψ. The wavelet basis is then given by $\psi_k^j(x) = 2^{-j/2}\psi(2^{-j}x - k)$, $k, j \in \mathbf{Z}$ (see [8]). We consider the multiresolution analysis associated with such basis.

In order to expand each term in (2.4) into the wavelet basis we are led to consider the integrals of the products of the basis functions, for example

$$M_{WWW}^{j,j'}(k, k', l) = \int_{-\infty}^{+\infty} \psi_k^j(x)\, \psi_{k'}^j(x)\, \psi_l^{j'}(x)\, dx, \tag{2.15}$$

where $j' \le j$. It is clear, that the coefficients $M_{WWW}^{j,j'}(k, k', l)$ are identically zero for $|k - k'| > k_0$, where k_0 depends on the overlap of the supports of the basis functions. The number of necessary coefficients may be reduced further by observing that

$$M_{WWW}^{j,j'}(k, k', l) = 2^{-j'/2} \int_{-\infty}^{+\infty} \psi_0^{j-j'}(x)\, \psi_{k-k'}^{j-j'}(x)\, \psi_{2^{j-j'}k-l}^0(x)\, dx, \tag{2.16}$$

so that

$$M^{j,j'}_{WWW}(k,k',l) = 2^{-j'/2}\tilde{M}^{j-j'}_{WWW}(k-k',2^{j-j'}k-l). \tag{2.17}$$

We also observe that the coefficients in (2.17) decay as the distance $r = j - j'$ between the scales increases. Rewriting (2.17) as

$$\tilde{M}^{r}_{WWW}(k-k',2^{r}k-l) = 2^{-r}\int_{-\infty}^{+\infty}\psi(2^{-r}x)\,\psi(2^{-r}x-k+k')\,\psi(x-2^{r}k+l)\,dx, \tag{2.18}$$

and recalling that the regularity of the product $\psi(2^{-r}x)\,\psi(2^{-r}x-k+k')$ increases linearly with the number of vanishing moments of the function ψ, we obtain

$$|\tilde{M}^{r}_{WWW}(k-k',2^{r}k-l)| \le C2^{-r\lambda M} \tag{2.19}$$

with some λ (see [8], [9]).

Let us define j_0 as the distance between the scales such that for a given ϵ all the coefficients in (2.19) with labels $r = j - j'$, $r > j_0$, have absolute values less than ϵ. For the purpose of computing with accuracy ϵ, we replace the mappings in (2.5) and (2.6) by

$$M^{j}_{VW} : \mathbf{V}_j \times \mathbf{W}_j \to \mathbf{V}_j \bigoplus_{j_0 \le j' \le j} \mathbf{W}_{j'}, \tag{2.20}$$

and

$$M^{j}_{WW} : \mathbf{W}_j \times \mathbf{W}_j \to \mathbf{V}_j \bigoplus_{j_0 \le j' \le j} \mathbf{W}_{j'}. \tag{2.21}$$

Since

$$\mathbf{V}_j \bigoplus_{j_0 \le j' \le j} \mathbf{W}_{j'} = \mathbf{V}_{j_0-1}, \tag{2.22}$$

and

$$\mathbf{V}_j \subset \mathbf{V}_{j_0-1}, \quad \mathbf{W}_j \subset \mathbf{V}_{j_0-1}, \tag{2.23}$$

we may consider the bilinear mappings (2.20) and (2.21) on $\mathbf{V}_{j_0-1} \times \mathbf{V}_{j_0-1}$. For the evaluation of (2.20) and (2.21) as mappings

$$\mathbf{V}_{j_0-1} \times \mathbf{V}_{j_0-1} \to \mathbf{V}_{j_0-1}, \tag{2.24}$$

we need significantly fewer coefficients than for the mappings (2.20) and (2.21). Indeed, it is sufficient to consider only the coefficients

$$M(k,k',l) = 2^{-j/2}\int_{-\infty}^{+\infty}\phi(x-k)\,\phi(x-k')\,\phi(x-l)\,dx, \tag{2.25}$$

and it easy to see that $M(k,k',l) = 2^{-j/2}M_0(k-l,k'-l)$, where

$$M_0(p,q) = \int_{-\infty}^{+\infty}\phi(x-p)\,\phi(x-q)\,\phi(x)\,dx. \tag{2.26}$$

Though it is a simple matter to derive and solve a system of linear equations to find $M_0(p,q)$, we advocate a different approach to evaluate (2.24) in the next subsection.

Let us now explain the reasons for considering (2.20) and (2.21) as mappings (2.24). On a given scale j the procedure of "lifting" the projections $P_j u$, $Q_j u$ into a "finer" subspace is accomplished by the pyramid reconstruction algorithm (see e.g. [3]). Let us assume that only a small number of the coefficients of $Q_j u$ are above the threshold of accuracy. We note (see Remark 2 for the Haar basis) that only those coefficients of $P_j u$ that contribute to the product $(P_j u)(Q_j u)$ (above the threshold ϵ) need to be kept. In fact, one may consider the function $Q_j u$ as a "cutoff function" for $P_j u$.

Computing $Q_j u$ and the corresponding part of $P_j u$ in the "finer" subspaces $\mathbf{V}_{j-1}, \mathbf{V}_{j-2}, \ldots, \mathbf{V}_{j_0-1}$ via the pyramid reconstruction algorithm roughly doubles the number of the coefficients on each scale. The procedure is similar to that of "oversampling" by the factor of two (from scale to scale) and, as we reach $\mathbf{V}_{j_0-1}$, we have increased the number of coefficients by the factor of approximately 2^{j_0}. This factor is a dominant constant in the complexity estimate of the algorithm. As an example, to maintain the single precision accuracy ($\epsilon \approx 10^{-6}$), we have $j_0 = 6$ and the "oversampling" factor $2^{j_0} = 64$ (for wavelets with six vanishing moments).

Examples of functions for which it is indeed necessary to use $j_0 = 6$ are the basis functions themselves. It is clear that the smoother are the basis functions, the smaller is the distance j_0. The compactly supported wavelets, however, are not very smooth since there is a trade-off between the number of vanishing moments and the smoothness of the function (see [8]).

Fortunately, in many applications the projections $P_j u$ and $Q_j u$ may be smoother than the basis functions. In this case the scale distance $\tilde{j}_0$ may be chosen according to the smoothness of $P_j u$ and $Q_j u$ and, thus, will be less that j_0. If we use the coefficients in (2.15) for computing the product, then we cannot make use of this observation in an adaptive manner. On the other hand, by computing $Q_j u$ and the corresponding part of $P_j u$ in the "finer" subspaces we may decide to terminate the process adaptively.

We note that the number of operations for this step of the algorithm is proportional to the number of the significant wavelet coefficients of the function.

II.4 Relations between values of functions and their wavelet coefficients.

The result of "oversampling" $P_j u$ and $Q_j u$ is that their product (for a given accuracy ϵ) is in the same subspace as the multiplicands (see (2.24)) . To compute the product, one may use the coefficients (2.26) but such approach does not lead to an efficient algorithm in the multidimensional case.

Instead of (2.24), it is sufficient to consider the mapping

$$\mathbf{V}_0 \times \mathbf{V}_0 \to \mathbf{V}_0. \tag{2.27}$$

It is easy to see that for $f \in \mathbf{V}_0$,

$$f(x) = \sum_k f_k \phi(x-k), \tag{2.28}$$

the values of f at integer points may be written as

$$f(m) = \sum_k \phi_{mk} f_k, \tag{2.29}$$

where $\phi_{mk} = \phi(m-k)$. Computing ϕ^{-1}_{mk} allows us to evaluate the coefficients f_l from the values of the function at integer points,

$$f_l = \sum_m \phi^{-1}_{lm} f(m). \tag{2.30}$$

Formula (2.30) is the quadrature formula for $f \in \mathbf{V}_0$. For most wavelets the entries of the matrix $\phi^{-1}_{mk} = \phi^{-1}(m-k)$ decay fast away from the diagonal and for a given accuracy ϵ the sum in (2.30) has very few terms.

The algorithm to evaluate the product via mapping in (2.27) is now clear. First, we compute the values of the multiplicands at the integer points using (2.29). Then we compute the product via ordinary multiplication at these points. Finally, we use (2.30) to obtain the coefficients of the product.

The virtue of this approach is that both operations (2.29) and (2.30) are convolutions. In the multidimensional case, the convolutions are separable and, therefore, lead to a faster algorithm than the algorithm using coefficients in (2.26).

Remark 3. Daubechies' wavelet with five vanishing moments is the only wavelet for which the author found that the decay of entries of the matrix ϕ^{-1}_{mk} away from the diagonal is relatively slow. Wavelet with five vanishing moments and with "almost linear phase", however, yields ϕ^{-1}_{mk} with a fast decay.

Remark 4. The use of (2.30) requires an *a priori* knowledge that the product is in the same subspace as the multiplicands.

Acknowledgments

The research was partially supported by ONR grant N00014-91-J4037 and a grant from Chevron Oil Field Research Company.

References

[1] G. Beylkin. Wavelets, Multiresolution analysis and Fast Numerical Algorithms. *A draft of INRIA Lecture Notes*, 1991.

[2] G. Beylkin. On the representation of operators in bases of compactly supported wavelets. *SIAM J. Numer. Anal.*, 1992. to appear.

[3] G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, 44:141–183, 1991. Yale University Technical Report YALEU/DCS/RR-696, August 1989.

[4] J. M. Bony. Interaction des singularités pour les équations aux dérivées partielles non-linéaires. Sém. e.d.p., 1979/80, 22, 1981/82, 2 et 1983/84, 10, Centre de Mathématique, Ecole Polytechnique, 91128-Palaisau, France.

[5] J. M. Bony. Calcul symbolique et propagation des singularités pour les équations aux dérivées partielles non-linéaires. *Ann. Scient. E.N.S.*, 14:209–246, 1981.

[6] J. M. Bony. Propagation et interaction des singularités pour les solutions des équations aux dérivées partielles non-linéaires. In *Proceedings of the International Congress of Mathematicians, Warszawa*, pages 1133–1147, 1983.

[7] R. R. Coifman and Y. Meyer. Au delà des opérateurs pseudo-différentiels. In *Astérisque, 57, (seconde édition revue et augmentée)*. Société Mathématique de France.

[8] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure and Appl. Math.*, 41:909–996, 1988.

[9] I. Daubechies and J. Lagarius. Two-scale difference equations, I. Global regularity of solutions & II. Local regularity, infinite products of matrices and fractals. *SIAM J. Math. Anal.*, 1991.