Journal of Physical Mathematics
ISSN: 2090-0902

Hajji, J Phys Math 2016, 7:1
DOI: 10.4172/2090-0902.1000157

Research Article Open Access

# On the Exact Values of Daubechies Wavelets

**Hajji MA***

*Department of Mathematical Sciences, United Arab Emirates University, United Arab Emirates*

**Abstract**

In this paper we propose an algorithm for the calculation of the exact values of compactly-supported Daubechies wavelet functions. The algorithm is iterative, performing a single convolution operation at each step. It requires solving, at the first step only, a linear system of a relatively small size. The novelty of the algorithm is that once the values at dyadic points at a certain level *j* are calculated they do not need to be updated at the next step. We find that this algorithm is superior to the well-known cascade algorithm proposed by Ingrid Daubechies. This algorithm can serve well in wavelet based methods for the numerical solutions of differential equations. The algorithm is tested on Daubechies scaling functions as well as Daubechies coiflets. Comparison with the values obtained using the cascade algorithm is made. We found that the cascade algorithm results converge to ours.

**Keywords**: Daubechies wavelets; Daubechies coiflets; Multiresolution analysis; The cascade algorithm; Convolution

## Introduction

The widespread interest in wavelets and their applications started in the 1980s after the breakthrough made by Daubechies [1,2] in constructing the first orthogonal compactly-supported wavelets with arbitrary regularity. Since then many researchers from different fields of science and engineering jumped into the world of wavelets with different intentions. Some were interested in ways to apply wavelets in their fields and others were interested in developing new theories and generalizations.In the engineering side, wavelets have found great success in signal processing such as in the analysis of sound patterns and image processing [3-6]. Wavelets have also found great success in the design and efficient implementation of numerical algorithms for the solution of differential equations [7-16].

Wavelet collocation based methods for solving different equations require knowledge of the values of the wavelet basis elements at collocation points. However, many of the available wavelets, such as the well-known Daubechies compactly-supported wavelets, do not have explicit formulas. Instead, they are defined recursively by refinement equations. In many applications having accurate values of the wavelet bases functions is very important in obtaining accurate solution to the problem. In [2] Daubechies described an algorithm known as the cascade algorithm for computing approximate values of the compactly-supported scaling and wavelet functions with arbitrary high precision. This algorithm works as a refinement scheme. At each step approximately twice as many values are computed, values at odd dyadic points $2^{-j}(2k+1)$ are computed for the first time and values at even dyadic points $2^{-j}(2k)$ are refined from the previous step. In the long run, i.e., as $j\to\infty$, the cascade algorithm produces the exact values. In this work, we propose an algorithm to calculate the *exact* values of Daubechies scaling and wavelet functions. The proposed algorithm avoids the refinement step in the cascade algorithm. Moreover, our algorithm, at each step computes only values at odd dyadic points $2^{-j}(2k+1)$. The values at even dyadic points $2^{-j}(2k)$ have already been computed at the previous step and no need for refinement because the values are exact.

The paper is organized as follows. In section 2, we briefly review compactly-supported scaling and wavelets functions and some related properties. In section 3, we outline the cascade algorithm in [2]. In section 4, we describe the proposed algorithm. In section 5, we apply the proposed algorithm to Daubechies scaling functions and compare our results to those obtained by the cascade algorithm. Finally, we conclude by some remarks in section 6.

## Preliminaries

Wavelet basis is a doubly-index family of $L^2(R)$ functions, $\psi_{j,k}$, $j,k\in Z$, defined by

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), \tag{1}$$

where $\psi(x)$ is the mother wavelet defined in terms of a mother scaling function $\varphi(x)$ via the refinement equation:

$$\psi(x) = \sqrt{2}\sum_k g_k \phi(2x-k). \tag{2}$$

The mother scaling function, $\varphi(x)$, is itself defined recursively by the refinement equation:

$$\phi(x) = \sqrt{2}\sum_k h_k \phi(2x-k). \tag{3}$$

The coefficients $h_k$ and $g_k$ are known, in the language of signal processing, as the low- and high-pass filter coefficients, respectively. For orthogonal wavelets, they are related by $g_k = (-1)^k h_{1-k}$.

The scaling function $\varphi$ generates an orthogonal multiresolution analysis (MRA) [17] which is an increasing sequence of subspaces $V_j$ of $L^2(R)$ (approximation spaces) with the following properties

- $\bigcap_j V_j = \{0\}$ and $\overline{\bigcup_j V_j} = L^2(R)$.
- $f(x)\in V_j \Leftarrow f(2x)\in V_{j+1}$.
- $f(x)\in V_j \Leftarrow f(x-2^{-j}k)\in V_j, \forall\, k\in Z$.
- $\{\phi(x-n), n\in Z\}$ is an orthonormal basis of $V_0$, where $\int\phi(x)dx \neq 0$.

A consequence of the above MRA sructure are the following:

***Corresponding author:** Hajji MA , Department of Mathematical Sciences, United Arab Emirates University, United Arab Emirates, Tel: 971 3 7136385; E-mail: mahajji@uaeu.ac.ae

**Received** January 12, 2016; **Accepted** February 01, 2016; **Published** February 05, 2016

**Citation:** Hajji MA (2016) On the Exact Values of Daubechies Wavelets. J Phys Math 7: 157. doi:10.4172/2090-0902.1000157

• The set of functions $\{_{j,k}(x), k\in Z\}$ is an orthonormal basis for $V_j$, where $\varphi_{j,k}(x)=2^{j/2}\varphi(2^j x-k)$.

• Associated with each $V_j$ there is a *wavelet* space $W_j$, its orthogonal complement in $V_{j+1}$, i.e., $V_j \oplus W_j = W_{j+1}$ .

• An orthonormal basis for $W_j$ is the set $\{\psi_{j,k}(x), k\in Z\}$.

• $\oplus_{j\in Z} W_j = L^2(R)$ .

Let $P_j$ and $Q_j$ be the orthogonal projections onto $V_j$ and $W_j$, respectively. Any function $f \in L^2(R)$ can be approximated by a function $f_j \in V_j$ through

$$f \approx f_j = P_j(f) = \sum_k s_k^j \phi_{j,k}, \qquad s_k^j = < f, \phi_{j,k} > = \int_{-\infty}^{\infty} f(x)\phi_{j,k}(x)dx$$

Similarly, the orthogonal projection of $f$ onto $W_j$ gives

$$Q_j(f) = \sum_k d_k^j \psi_{j,k}, \qquad d_k^j = < f, \psi_{j,k} > = \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx$$

By the structure of the MRA ( $V_j = V_{j-1} \oplus W_{j-1}$ ), we have

$$P_j(f) = P_{j-1}(f) + Q_{j-1}(f).$$

The coefficients $s_k^{j-1}$ and $d_k^{j-1}$ at a lower scale are computed from $s_k^j$ via Mallat's algorithm [17]:

$$s_k^{j-1} = \sum_n h_{n-2k} s_n^j, \tag{4}$$

$$d_k^{j-1} = \sum_n g_{n-2k} s_n^j, \tag{5}$$

and conversely $s_k^j$ are reconstructed via

$$s_k^j = \sum_n h_{k-2n} s_n^{j-1} + \sum_n g_{k-2n} d_n^{j-1}. \tag{6}$$

For compactly-supported wavelets, the sums in (2) and (3) are finite:

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2x-k), \tag{7}$$

$$\psi(x) = \sqrt{2}\sum_{k=0}^{k=0} g_k \phi(2x-k), \tag{8}$$

with $g_k = (-1)^k h_{L-k-1}$ so that both and $\psi$ are supported in [0, *L*-1] .The integer $L=2M$ where $M$ is the number of vanishing moments of $\psi$:[2]

$$\int_{-\infty}^{\infty} x^m \psi(x)dx = 0, \qquad 0 \le m \le M-1.$$

## The Cascade Algorithm

The cascade algorithm is an iterative scheme proposed by Daubechies explanied in [1-3] to calculate approximate values of the scaling and wavelet functions, $\varphi$ and $\psi$, at rational dyadic points $x=2^{-m}k$ . In order to compare this algorithm to our proposed algorithm, it is worthwhile to describe it.

The cascade algorithm is based on the key fact that the scaling function $\varphi$ is the unique function satisfying

$$< f(x), \phi(x \quad n) > = \delta_{0,} \ , \tag{9}$$

$$< f(x), \psi_{j,k}(x) > = 0, \qquad \forall\ j \ge 0, k \in Z. \tag{10}$$

It is also based on the fact that $2^j\varphi(2^j x)$ is an approximate $\delta$-function as $j\to\infty$ in the sense of the following proposition [2].

**Proposition 1** *If f is a continuous function. Then for any* $x\in R$,

$$\lim_{j\to\infty} \int_{-\infty}^{\infty} f(x+y)2^j \phi(2^j y)dy = f(x). \tag{11}$$

Moreover if $f$ is uniformly continuous, then the convergence in (11) is uniform as well, and If $f$ is H ö lder continuous with exponent α, i.e., $|f(x)-f(y)| \le C|x-y|^{\alpha}, \forall\ x, y \in R$ , then the convergence in (11) is exponentially fast in *j*, i.e.,

$$|f(x) - \int_{-\infty}^{\infty} f(x+y)2^j \phi(2^j y)dy| \le C2^{-j\alpha}. \tag{12}$$

A consequence of the above proposition is the following.

**Lemma 2** *For any dyadic rational* $x=2^{-m}k$,

$$\phi(2^{-m}k) = \lim_{j\to\infty} 2^{j/2} < \phi, \phi_{j,2^{j-m}k} > . \tag{13}$$

Moreover, if $j$ is sufficiently large, say $j > j_0$ , we have the estimate

$$|\phi(2^{-m}k) - 2^{j/2} < \phi, \phi_{j,2^{j-m}k} > | \le C2^{-j\alpha}, \tag{14}$$

Where $C$ and $j_0$ depend on $m$ and $k$.

Proof By Proposition 1, we have

$$\phi(k2^{-m}) = \lim_{j\to\infty} \int_{-\infty}^{\infty} \phi(2^{-m}k + y)2^j \phi(2^j y)dy$$

$$= \lim_{j\to\infty} 2^{j/2} \int_{-\infty}^{\infty} \phi(u)2^{j/2} \phi(2^j u - 2^{j-m}k)du$$

$$= \lim_{j\to\infty} 2^{j/2} < \phi, \phi_{j,2^{j-m}k} >$$

which proves (13). Estimate (14) follows from (12).

Lemma 2 suggests that at any $j$ -level, $\varphi(2^j x)$ can be approximated by

$$\phi(2^{-j}k) \approx 2^{j/2} < \phi, \phi_{j,k} >, \tag{15}$$

with the error $|\phi(2^{-j}k) - 2^{j/2} < \phi, \phi_{j,k} > | \le C2^{-j\alpha}$ (see estimate (14)).

For $j,k\in Z$ , define the coefficients $c_k^j = < \phi, \phi_{j,k} >$ . Then by (15), we have

$$\phi(2^{-j}k) \approx 2^{j/2} c_k^j. \tag{16}$$

The coefficients $c_k^j$ can be reconstructed recursively using Mallat's algorithm (6) starting at the scale $j$=0. At scale $j$=0, by (9), we have

$$c_k^0 = \delta_{0,k} = \begin{cases} 1, & if \quad k = 0 \\ 0, & otherwise. \end{cases} \tag{17}$$

Since $\phi \perp \psi_{j,k}$ for all $j,k \in Z$ , $d_k^j = < \phi, \psi_{j,k} > = 0$ . It follows from (6) that $c_n^j$ are given by

$$c_k^j = \sum_n h_{k-2n} c_n^{j-1}. \tag{18}$$

The cascade algorithm summarizes as follows:

• Start with the sequence $c_n^0 = \delta_{n,0}$ which can be viewed as a first approximation of at the integers.

• For $j \ge 1$ compute $c_k^j$ recursively via (18). The sequence $c_k^j$ gives the approximation of $\varphi$ at the $j$ -level dyadics $x = k/2^j$: $\phi(k/2^j) \approx 2^{j/2} c_k^j$.

We note that if the length of the filter $h$ is $L$ then the length of the sequence $c_k^j$ is $2^j(L-1)-(L-2)$ . At every step $j$ , the algorithm computes for the first time approximations of $\varphi$ at the odd dyadics $x = (2k+1)/2^j$ and refines the approximations at the even dyadics $x = (2k)/2^j = k/2^{j-1}$ which were obtained at the previous $j$-1 step. To be more precise at step $j$, the algorithm computes a total of $2^j(L-1)-(L-2)$ values (the length $c_k^j$ ) of which $2^{j-1}(L-1)$ values comprise the new approximations (at the odd dyadics) and the rest ( $2^{j-1}(L-1)-(L-2)$) values constitute refinements of the old approximations obtained at the previous $j$-1 step.

Before closing this section, we would like to mention that at any scale $j$ the cascade algorithm does not cover all $j$ -level dyadic points in the interval $[0, L-1]$ ; it only covers dyadics $_{k=0,1,\ldots,2^j(L-1)-(L-2)}$ for $k = 0,1,\ldots,2^j(L-1)-(L-2)$ . Since there are $(2^j(L-1)+1)$ $j$-level dyadics in $[0, L-1]$ , the $(L-3)$ dyadic points $x = L-1-k/2^j$ for $k = 0,1,\ldots,L-4$ , are not covered; some of them, but not all, will be covered at next step $j+1$ .

## The Proposed Algorithm

The algorithm we propose in this paper covers all dyadics at any given scale *j*. It also yields the exact values of the scaling and wavelet functions $\varphi$ and $\psi$ at dyadic points $x = 2^{-j}m,\ j,m \in Z$. Since $\psi$ is given in terms of $\varphi$, it is suffices to describe the algorithm for . The algorithm is based on finding the exact values of $\varphi$ at the integers $n = 1,2,\ldots,L-2$ (the $0$ -level dyadics). These are found by solving an eigenvalue problem. Once the values of $\varphi$ at the integers are obtained, at each subsequent step, the algorithm performs a single convolution operation to calculate the values of $\varphi$ at only odd-dyadics. The algorithm is explained in the following.

Since $\varphi$ is supported in $[0, L-1]$, we have, by continuity, $\varphi(x) = 0$ for $x \le 0$ and $x \ge L-1$. Let

$$\Phi^{(0)} = [\phi(1)\ \ \phi(1)\ \ \phi(2)\ \ \ldots\ \ \phi(L-2)]^T \tag{19}$$

be the column vector containing the values of $\varphi$ at the integers. Then, according to the refinement equation,

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2x-k), \tag{20}$$

$\Phi^{(0)}$ satisfies the linear system

$$\Phi^{(0)} = A\Phi^{(0)} \tag{21}$$

where $A$ is an $(L\text{-}2)\times(L\text{-}2)$ matrix with entries $a_{ij}$, $1 \le i, j \le L-2$, given by

$$a_{ij} = \begin{cases} \sqrt{2}\, h_{2i-j}, & \text{if } 0 \le (2i-j) \le L-1, \\ 0. & \text{otherwise.} \end{cases} \tag{22}$$

Equation (21) suggests that $\Phi^{(0)}$ is an eigenvector of $A$ corresponding to the eigenvalue $=1$. The existence of the eigenvalue $\lambda=1$ is justified by the following proposition.

**Proposition 3** *Suppose that there exists a continuous solution $\varphi$ to (20). Then $\lambda=1$ is an eigenvalue of the matrix $A$ in (22) and $\Phi^{(0)}$ is an associated eigenvector.*

Proof Let $\Phi^{(0)}$ be as in (19). Then by (20) $\Phi^{(0)}$ satisfies (21). If $\Phi^{(0)}=0$ (the zero vector), then $\varphi(n)=0$ for $k \in \mathbb{N}, l \in \mathbb{Z}$. This implies, using (20), that $\varphi\ (2^{-k}l)=0$ for all $k \in \mathbb{N}, l \in \mathbb{Z}$. Then by continuity of $\varphi$, this would imply that $\varphi=0$ which is a contradiction. It follows that $\Phi^{(0)} \neq 0$ and consequently $\Phi^{(0)}$ is an eigenvector of $A$ associated to the eigenvalue $\lambda=1$.

Since $\varphi$ satisfies $\sum_l \phi(x-l) = 1$ for any $x$, $\sum_{n=1}^{L-2} \phi(n) = 1$. Then the vector $\Phi^{(0)}$ is equal to the normalized eigenvector of $A$ corresponding to $\lambda=1$. Precisely, $\Phi^{(0)}$ is the unique solution to the $(L\text{-}1)\times(L\text{-}2)$ nonhomogeneous system

$$Bx = b \tag{23}$$

where

$$B = \begin{bmatrix} A - I_{(L-2)} \\ \hline 11\ldots1 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

and $I_{(L-2)}$ is the identity matrix of order $(L\text{-}2)$.

Next, once the values of $\varphi$ at the integers are known, we apply the refinement equation to find the values of $\phi$ at the *odd* half integers $x = \frac{2n-1}{2}, n = 1,2\ldots,L-1$,

$$\phi((2n-1)/2) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2n-k-1), \qquad n = 1,2,\ldots,L-1. \tag{24}$$

Note that we do need to calculate $\varphi$ at the *even* half integers $x = \frac{2n}{2}$ as they were computed in $\Phi^{(0)}$. Let $\Phi^{(1)}$ be the column vector containing the values of $\varphi$ at the odd $1$ -level dyadics $x = \frac{2n-1}{2}, n = 1,2\ldots,L-1$. Then (24) can be viewed as a convolution operation followed by downsampling by 2,

$$\Phi^{(1)} = conv(\sqrt{2}\,h,\ \Phi^{(0)}) \downarrow 2, \tag{25}$$

where "conv" denotes convolution and $\downarrow$ denotes downsampling by 2.

Let $\Phi^{(j)}$ $j \ge 1$, be the vector of length $2^{j-1}$ $(L\text{-}1)$ containing the values of $\varphi$ at the odd *j*-level dyadics, i.e.,

$$\Phi^{(j)} = [\phi(1/2^j)\ \ \phi(3/2^j)\ \ \ldots\ \ \phi(L-1-1/2^j)]. \tag{26}$$

A careful examination of the refinement equation (7) gives

$$\Phi^{(j)} = conv(\tilde{h}^j,\ \Phi^{(j-1)}), \qquad for\ \ j \ge 2, \tag{27}$$

where $\tilde{h}^j$ is an "upsampled" version of the vector $\sqrt{2}\,h$, obtained by inserting $(2^{j-2}\text{-}1)$ zeros between every two successive entries in $\sqrt{2}\,h$. Explicitly,

$$\tilde{h}_k^j = \begin{cases} \sqrt{2}\, h_m, & \text{if } \ k = m2^{j-2}, 0 \le m \le L-1, \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

The length of the vector $\tilde{h}^j$ is equal to $2^{j-2}(L\text{-}1)+1$. Note that the length of $^{(j)}$ given by the convolution formula (27) is the sum of the lengths of $\tilde{h}^j$ and $\Phi^{(j-1)}$ less one, i.e., $(2^{j-2}(L-1)+1)+(2^{j-2}(L-1))-1 = 2^{j-1}(L-1)$, and it agrees with the length of $\Phi^{(j)}$ given by formula (26).

Our proposed algorithm summarizes in the following steps:

1. Compute the values of $\varphi$ at the integers($x$=1, 2,…,$L$-2)given by the normalized eigenvector of $A$ corresponding to the eigenvalue 1 and collect them in a vector $\Phi^{(0)}$.

2. Convolve $\Phi^{(0)}$ with the vector $\sqrt{2}\,h$ and downsample by 2 to get $\Phi^{(1)}$. This gives the values of $\varphi$ at the odd 1-level dyadics $(x = \frac{2n-1}{2}, n = 1,2,\ldots,L-1)$.

3. For $j \ge 2$, compute the values of $\varphi$ at the odd *j*-level dyadics by convolving of the vector $\tilde{h}^j$ with the vector $\Phi^{(j-1)}$ where, again, $\Phi^{(j-1)}$ is the vector containing only the values of $\varphi$ at the odd $(j\text{-}1)$-evel dyadics.

The values of the wavelet function $\psi(x)$ at the any *j*-level dyadics can be computed using the relation

$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k \phi(2x-k) \tag{29}$$

All of the steps in the proposed algorithm are computationally trivial except perhaps for the first one, where an eigenvector of $A$ needs to be found. The matrix $A$ being sparse, however, this is not very difficult. Using the normalization of $\varphi$, $\sum_{n=1}^{L-2} \phi(n) = 1$, the sought eigenvector $\Phi^{(0)}$ can be obtained as the solution of (23).

As a comparison between the two algorithms we note the following:

- Our proposed algorithm is similar to the well-known cascade algorithm in that the computations of both algorithms involve convolutions, except for the first step in our proposed algorithm, where an relatively small size linear system has to be solved.

- The first step in our algorithm is the most expensive but it is the

crucial one because it yields the exact values at the integers from which everything else is derived.

• The cascade algorithm begins by making an initial guess for $\varphi(x)$ (a Dirac delta function), whereas our algorithm begins by computing $\varphi(x)$ at the integers (0-level dyadics).

• A clear advantage of our algorithm is that (i) it provides the exact values and (ii) once the initial step has been performed, at every subsequent step, only the values of $\varphi(x)$ at the odd dyadics need to be computed, the even dyadics at step $j$ being the odd dyadics at step $j$-1. In contrast, the cascade algorithm requires refinement of $\varphi(x)$ at the even dyadics.

## Numerical Results

We have implemented the proposed algorithm in Matlab and tested it to produce the values of Daubechies' scaling functions as well as Daubechies' coiflets. Plots of Daubechies scaling functions db4 and db6 ($M$=4 and $M$=6 ) as well as Daubechies' coiflets coif2 and coif4 ($M$=2 and $M$=3), obtained by our algorithm, are displayed in Figures 1 and 2.

We compared our numerical values with the ones obtained using the cascade algorithm which is implemented in Matlab under the function "wavefun". Samples of the numerical values obtained are displayed in Tables 1 and 2. Table 1 displays the values of db4 at the integers obtained by the cascade algorithm for $j$=5,10,15, 20 and the ones obtained by our algorithm using only $j$=0, i.e., the solution of the system (23). Table 2 displays selected values of db6 at the $j$=5 level dyadics obtained by the cascade algorithm for $j$=5,10,15, 20 and the ones obtained by our algorithm using $j$=5. The results clearly show that the cascade algorithm results converge to ours as $j$ tends to infinity.We remark that one has to iterate the cascade algorithm for larger value of $j$ to obtain accurate results at a lower $k$ level dyadics. For instance, from Table 2, we needed to iterate the cascade algorithm until $j$=20 to obtain as closer results to the ones obtained by our algorithm with only $j$=5.

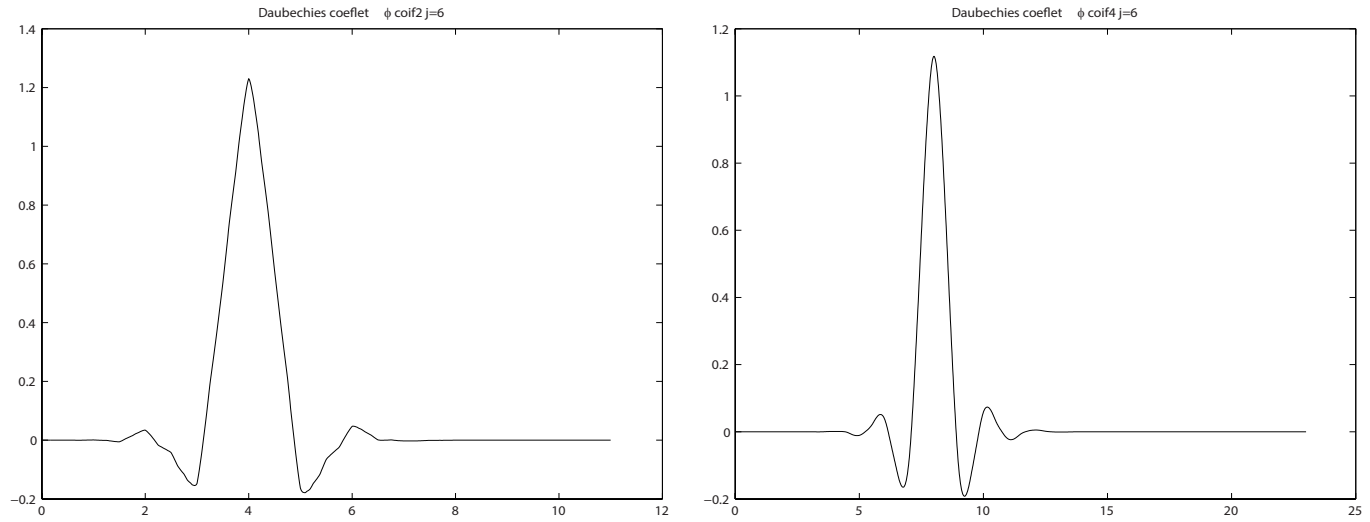


**Figure 1:** A plot of Daubechies db4 and db6.

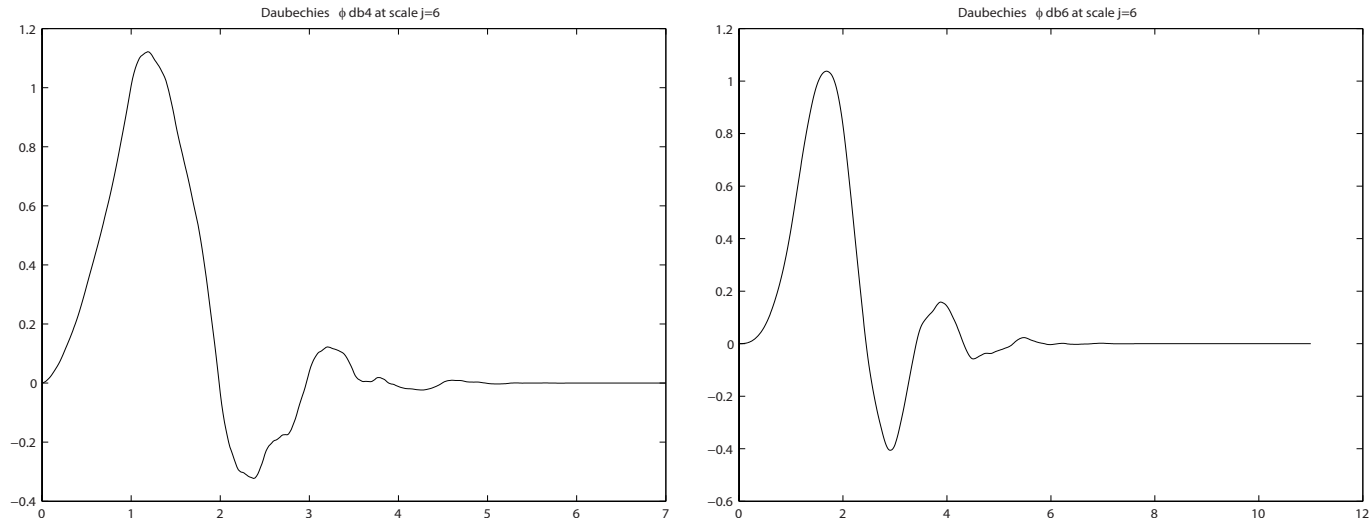


**Figure 2:** A plot of Daubechies coiflets coif2 and coif4.

| Values of Daubechies' scaling function db4 | | | | | |
|---|---|---|---|---|---|
| $*x$ | The cascade algorithm | | | | Our algorithm |
| | $j=5$ | $j=10$ | $j=15$ | $j=20$ | $j=0$ |
| | 1.00747364 | 1.00717932 | 1.00717027 | 1.00716998 | 1.00716997 |
| | -0.03432410 | -0.03385159 | -0.03383741 | -0.03383696 | -0.03383695 |
| | 0.03983843 | 0.03961669 | 0.03961065 | 0.03961046 | 0.03961046 |
| | -0.01180456 | -0.01176501 | -0.01176437 | -0.01176435 | -0.01176435 |
| | -0.00120268 | -0.00119824 | -0.00119796 | -0.00119795 | -0.00119795 |
| | 0.00001926 | 0.00001883 | 0.00001882 | 0.00001882 | 0.00001882 |

**Table 1:** Values of db4 at the integers.

| Values of Daubechies' scaling function db6 | | | | | |
|---|---|---|---|---|---|
| $*x$ | The cascade algorithm | | | | Our algorithm |
| | $j=5$ | $j=10$ | $j=15$ | $j=20$ | $j=5$ |
| $\frac{2}{2^5}$ | 0.0004331 | 0.0002749 | 0.0002707 | 0.0002705 | 0.0002705 |
| $\frac{22}{2^5}$ | 0.1696324 | 0.1623220 | 0.1620967 | 0.1620896 | 0.1620894 |
| $\frac{42}{2^5}$ | 0.8333298 | 0.8208984 | 0.8205067 | 0.8204944 | 0.8204941 |
| $\frac{62}{2^5}$ | 0.9002129 | 0.9135170 | 0.9139223 | 0.9139350 | 0.9139354 |
| $\frac{82}{2^5}$ | -0.1721720 | -0.1591060 | -0.1586933 | -0.1586804 | -0.1586800 |
| $\frac{119}{2^5}$ | 0.1241885 | 0.1214656 | 0.1213838 | 0.1213812 | 0.1213811 |
| $\frac{159}{2^5}$ | -0.0268479 | -0.0277469 | -0.0277748 | -0.0277757 | -0.0277757 |
| $\frac{199}{2^5}$ | 0.0014295 | 0.0014226 | 0.0014213 | 0.0014212 | 0.0014212 |
| $\frac{209}{2^5}$ | -0.0022161 | -0.0023020 | -0.0023046 | -0.0023047 | -0.0023047 |
| $\frac{259}{2^5}$ | 0.0000263 | 0.0000270 | 0.0000270 | 0.0000270 | 0.0000270 |

**Table 2:** Values of db6 at selected level dyadic.

## Conclusion

In this paper, we have presented an efficient algorithm for the computation of the exact values of refinable functions, in particular Daubechies' scaling and wavelet functions. Our motivation for this work stems from the need for more accurate point values of the widely used Daubechies wavelets. This certainly will be useful in the numerical solutions of differential equations where wavelets are being used. Our proposed algorithm produces the exact values whereas the well-known cascade algorithm produces approximate values. What is good about the proposed algorithm is that, at each step, it computes values only at odd dyadics. The cascade algorithm, however, at each step calculates new values and refines old ones. The only expensive step in our algorithm is the first one where we need to solve a relatively small size linear system. This first step is the crucial one in that it provides us with the exact values (to machine precision) at the integers from which the rest is derived.We believe that having exactly values of wavelet functions will give better results in wavelet based numerical schemes for the solution of differential equations.

### References

1. Daubechies I (1988) Orthonormal bases of compactly supported wavelets. Comm Pure and Appli Math 41: 909-996.
2. Daubechies I (1992) Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics. Philadelphia PA
3. Daubechies I, Lagarias JC (1991) Two-scale difference equations. I. Existence and global regularity of solutions. SIAM J Math Anal 22: 1388-1410.
4. Martinet RK, Morlet J, Grossmann A (1987) Analysis of sound patterns through wavelet transforms. Internat J Patterm Recognition and Artificial Intelligence 1: 273-301
5. Antoni, M Barlaud M Mathieu P, Daubechies I (1992) Image coding using wavelet transform. IEEE Trans Image Proc 1: 205-220.

6. Devore RA, Jawerth B, Lucier BJ (1992) Image compression through wavelet transform coding. IEEE Trans Inf Th 38: 719-746.

7. Xu JC, Shann WC (1992) Galerkin-wavelet methods for two-poiny boundary value problems. Numer Math 63: 123-144.

8. Karami A, Karimi HR, Moshiri B, Maralani PJ (2008) Investigation of Interpolating Wavelets Theory for the Resolution of PDEs. Int J Contemp Math Sci 3: 1017-1029.

9. Cai W , Wang J (1996) Adaptive multiresolution collocation methods for initial boundary value problems of Nonlinear PDEs. SIAM J Numer Anal 33: 937-970.

10. Vasilyev OV, Paolucci A (1996) Dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. J Computational Physics 125: 498-512.

11. Vasilyev OV, Paolucci S (1997) Aast adaptive wavelet collocation algorithm for multidimensional PDEs. J Computational Physics 138: 16-56.

12. Bertoluzza S, Naldi G(1996) A wavelet collocation method for the numerical solution of partial differential equations. Applied and Computational Harmonic Analysis 3: 1-9.

13. Alam JM, Kevlahan NKR, Vasilyev OV (2006) Simultaneous space–time adaptive wavelet solution of nonlinear parabolic differential equations. J Computational Physics 214: 829-857.

14. Vasilyev OV, Kevlahan NKR (2005) An adaptive multilevel wavelet collocation method for elliptic problems. J Computational Physics 206: 412-431.

15. Vasilyev OV, Kevlahan NKR (2005) An adaptive wavelet collocation method for fluid-structure interaction at high Reynolds numbers. SIAM J Sci Comput 26: 1894-1915.

16. MA Hajji, Melkonian S, Vaillancourt R (2004) Two-dimensional wavelet bases for partial differential operators and applications. In Advances in Pseudo-Differential Operators, Birkhäuser Basel, USA.

17. Mallat S (1989) Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$ .Trans Amer Math Soc 315: 69–88.