Available online at http://scik.org
Adv. Inequal. Appl. 2018, 2018:12
https://doi.org/10.28919/aia/3682
ISSN: 2050-7461

A QUANTITATIVE METHOD FOR CHOOSING OPTIMAL DAUBECHIES WAVELETS

ERIC STACHURA*, ANDREW HUNTER

Department of Mathematics and Statistics, Haverford College, Haverford, PA 19041, USA

Abstract. We develop a new method for calculating specific values of Daubechies wavelets in one dimension. The novelty of this approach is its ability to calculate exact values of the Daubechies scaling functions and, by extension, wavelets, without calculating values of the scaling function at other unnecessary dyadic rationals. We then provide a new quantitative method for choosing the best wavelet for applications by giving a rigorous error analysis of the wavelet transform for functions of various smoothness.

Keywords: Daubechies wavelets; Cascade algorithm; eigensystem method; branch method; error analysis; Hölder continuity.

2010 AMS Subject Classification: 65T60, 41A65, 41A25.

1. Introduction

Since the introduction of Daubechies wavelets in [4], [5], mathematicians, physicists, and engineers alike have employed them to study various problems in their respective fields. Some applications include numerically solving partial differential equations [3], [17], recognition and

*Corresponding author
E-mail address: estachura@haverford.edu

Received February 15, 2018

visualization of various signals [10], [18], and regularization of inverse problems [6]. These are just a handful of results in the vast wavelet literature.

Frequently in these applications one needs to know the exact value of the wavelet at certain points. This tends to be difficult though since wavelets generally don't have a closed form, but rather satisfy some recursion relation. In the first part of this note we provide a new way to calculate the exact value of Daubechies wavelets that does not require calculating values of the scaling function at unnecessary dyadic rationals. The second part of this note is devoted to analyzing the convergence of the wavelet transform for functions of various smoothness. This analysis gives a quantitative method for choosing the best suited wavelet for the application at hand, see Theorem 5.3 which gives error estimates (10) and (12). See also the estimate (14) for C^m smooth functions.

The layout of this article is as follows. In Section 2 we provide the necessary background on Daubechies wavelets. Then in Section 3 we review current methods on finding numerical values for Daubechies wavelets, including the Cascade Algorithm. In Section 4 we discuss the new method and provide specific examples. Then in Section 5 we discuss the discrete wavelet transform and compute the error for Lipschitz and Hölder functions. Finally we discuss the convergence of the wavelet transform for smoother functions, and obtain the error bound (14).

2. Daubechies Wavelets

Wavelets are essentially functions that can generate an orthonormal basis for $L^2(\mathbb{R})$ by dilating and translating the wavelet function. Thus, wavelets are useful for approximating functions in $L^2(\mathbb{R})$. In order to dilate and translate a function, we define the subscript notation

$$f_{k,j}(x) := 2^{k/2} f(2^k x - j)$$

for an arbitrary $f \in L^2(\mathbb{R})$. As we will see in the forthcoming sections, a firm understanding of Multiresolution Analysis (MRA) is needed for the derivation of wavelets.

Definition 2.1 ([5])**.** A multiresolution of $L^2(\mathbb{R})$ with scaling function $\varphi \in L^2(\mathbb{R})$ is a sequence of subspaces $\{V_k\}_{k\in\mathbb{Z}}$ that satisfy the following properties:

(1) orthogonality: $\{\varphi(x-j) : j \in \mathbb{Z}\}$ is an orthonormal basis for V_0

(2) nesting: $V_k \subset V_{k+1}$ for all $k \in \mathbb{Z}$

(3) scaling: $f(x) \in V_k$ if and only if $f(2x) \in V_{k+1}$

(4) density: $\overline{\bigcup_{k\in\mathbb{Z}} V_k} = L^2(\mathbb{R})$

(5) separation: $\bigcap_{k\in\mathbb{Z}} V_k = \{0\}$

A simple example of a scaling function that generates a multiresolution of $L^2(\mathbb{R})$ is the indicator function of the interval $[0,1)$: $\varphi(x) = \chi_{[0,1)}(x)$. As we will see later, the scaling function φ plays a crucial rule in constructing a wavelet. Hence, φ is sometimes referred to as a *father wavelet*.

Ingrid Daubechies invented an entire family of wavelets [5] which are referred to as *Daubechies wavelets*. Her discovery was a major breakthrough in wavelet analysis because Daubechies wavelets have compact support and some sort of smoothness. This compromise makes them ideal for applications, which usually exploit their ability to efficiently approximate functions. The importance of certain properties depend on the nature of the application and some of these properties cannot coexist.

The naming scheme we use to describe a specific wavelet in the Daubechies wavelet family is Daub$J$, where J is the number of *constant filter coefficients*, denoted as a_j, in the wavelet expansion.

Definition 2.2. For any $N \in \mathbb{N}$, a Daubechies wavelet of class Daub$\,2N$ is a function $\psi \in L^2(\mathbb{R})$ of the form

$$\psi = \sum_{j=0}^{2N-1} (-1)^j a_{2N-1-j}\varphi_{1,j}$$

where $a_j = \langle \varphi, \varphi_{1,j} \rangle$ and its associated scaling function $\varphi \in L^2(\mathbb{R})$ of the form

$$\varphi = \sum_{j=0}^{2N-1} a_j \varphi_{1,j}$$

that generates a multiresolution of $L^2(\mathbb{R})$ which satisfies the following four properties:

V_k is sometimes defined as $V_k = \overline{\text{span}}\{2^{k/2}\varphi(2^k x - j) : j \in \mathbb{Z}\}$.

(1) ψ and φ are compactly supported on the interval $[0,2N-1]$,

(2) ψ has N vanishing moments,

(3) there are only $2N$ non-zero constant filter coefficients, i.e., $a_j \neq 0$ for $j \in [0,2N)$ and $a_j = 0$ for all other j, and

(4) φ satisfies $\int_{\mathbb{R}} \varphi dx = 1$.

Another useful property of the Daubechies Scaling function is known as a *partition of unity*.

Lemma 2.3. *If φ is a scaling function of a* Daub$2N$ *wavelet, then the equality*

$$\sum_{n\in\mathbb{Z}} \varphi(2^{-k}n) = 2^k \sum_{n\in\mathbb{Z}} \varphi(n)$$

is satisfied for all $k \in \mathbb{N}$.

Proof. We will prove by induction. Using φ's scaling relation when $k=1$, the chain of equalities

$$\sum_{n\in\mathbb{Z}} \varphi\left(\frac{n}{2}\right) = \sum_{n\in\mathbb{Z}}\sum_{j\in\mathbb{Z}} a_j\varphi_{1,j}\left(\frac{n}{2}\right) = \sqrt{2}\sum_{j\in\mathbb{Z}} a_j \sum_{n\in\mathbb{Z}} \varphi(n-j) = \sqrt{2}\sum_{j\in\mathbb{Z}} a_j \sum_{n\in\mathbb{Z}} \varphi(n) = 2\sum_{n\in\mathbb{Z}} \varphi(n)$$

hold by (1) below. Now we will prove the equality for k^{th} case assuming that it holds for $(k-1)^{st}$ case:

$$\sum_{n\in\mathbb{Z}} \varphi\left(\frac{n}{2^k}\right) = \sum_{n\in\mathbb{Z}}\sum_{j\in\mathbb{Z}} a_j\varphi_{1,j}\left(\frac{n}{2^k}\right) = \sqrt{2}\sum_{j\in\mathbb{Z}} a_j \sum_{n\in\mathbb{Z}} \varphi\left(\frac{n}{2^{k-1}}-j\right) = 2\sum_{n\in\mathbb{Z}} \varphi\left(\frac{n}{2^{k-1}}\right) = 2^k\sum_{n\in\mathbb{Z}} \varphi(n)$$

as desired.

We also need the following:

Proposition 2.4. *If φ is a scaling function of a* Daub$2N$ *wavelet, then the following equality holds.*

$$\sum_{n\in\mathbb{Z}} \varphi(n) = 1$$

Proof. We will show that $\int \varphi(x)dx = \sum_n \varphi(n)$; the desired equality follows then as a direct result of Property 4 from Definition 2.2. We can expand the integral of φ as a Riemann sum. We will partition the support of φ with the intervals $[2^{-k}(j-1), 2^{-k}j]$ for $j = 1, 2 \dots, 2^k(2N-1)$. We can then express the integral as limit of Riemann sums with increasing partitions:

$$\int_{\mathbb{R}} \varphi(x)dx = \lim_{k\to\infty} \sum_{j=1}^{2^k(2N-1)} 2^{-k}\varphi(x_{k,j}), \qquad x_{k,j} \in \left[2^{-k}(j-1), 2^{-k}j\right].$$

If we choose $x_{k,j} = 2^{-k}(j-1)$ and utilize Lemma 2.3, we can see that

$$\lim_{k\to\infty} \sum_{j=1}^{2^k(2N-1)} 2^{-k}\varphi(x_{k,j}) = \lim_{k\to\infty} \sum_{j=1}^{2^k(2N-1)} 2^{-k}\varphi(2^{-k}(j-1)) = \lim_{k\to\infty} \sum_{j=1}^{2^k(2N-1)} \varphi(j-1) = \sum_{j\in\mathbb{Z}} \varphi(j)$$

because $2^k(2N-1)$ tends to infinity as k increases and $\varphi(x) = 0$ for all $x < 0$. Therefore the equality

$$1 = \int_{\mathbb{R}} \varphi(x)dx = \sum_{j\in\mathbb{Z}} \varphi(j)$$

holds, as desired.

By combining various properties of Daubechies wavelets, we find a system of equations that the filter coefficients must satisfy:

$$\sqrt{2} = \sum_{j\in\mathbb{Z}} a_j \tag{1}$$

$$\delta_{0,n} = \sum_{j\in\mathbb{Z}} a_{2n+j}a_j \qquad \text{for } n \in \mathbb{Z} \tag{2}$$

$$0 = \sum_{j\in\mathbb{Z}} (-1)^j j^m a_j \qquad \text{for } m = 0, 1, \dots, N-1 \tag{3}$$

$$0 = a_j \qquad \text{for } j \in \mathbb{Z} \setminus [0, 2N) \tag{4}$$

This system of equations does not necessarily give a unique solution, however, the number solutions is finite. One of the solutions does indeed produce a Daubechies wavelet. There is a method for finding the exact values of the constant filter coefficients that utilizes the Fourier transform, see [5] and [13]. The constant filter coefficients of a Daub$2N$ wavelet for $N = 1, 2, 3, 4$ are listed in Table 1.

We only get a unique solution when $N = 1$, forming the Daub2 (Haar) wavelet.

	$N=1$	$N=2$	$N=3$	$N=4$
a_0	$\frac{1}{\sqrt{2}}$	$\frac{1+\sqrt{3}}{4\sqrt{2}}$	$\frac{\sqrt{2}+2\sqrt{5}+\sqrt{10+4\sqrt{10}}}{32}$	$\approx \quad 0.23037781330889673$
a_1	$\frac{1}{\sqrt{2}}$	$\frac{3+\sqrt{3}}{4\sqrt{2}}$	$\frac{5\sqrt{2}+2\sqrt{5}+3\sqrt{10+4\sqrt{10}}}{32}$	$\approx \quad 0.7148465705529157$
a_2		$\frac{3-\sqrt{3}}{4\sqrt{2}}$	$\frac{5\sqrt{2}-2\sqrt{5}+\sqrt{10+4\sqrt{10}}}{16}$	$\approx \quad 0.6308807679298589$
a_3		$\frac{1-\sqrt{3}}{4\sqrt{2}}$	$\frac{5\sqrt{2}-2\sqrt{5}-\sqrt{10+4\sqrt{10}}}{16}$	$\approx -0.027983769416859854$
a_4			$\frac{5\sqrt{2}+2\sqrt{5}-3\sqrt{10+4\sqrt{10}}}{32}$	$\approx -0.18703481171909309$
a_5			$\frac{\sqrt{2}+2\sqrt{5}-\sqrt{10+4\sqrt{10}}}{32}$	$\approx \quad 0.030841381835560764$
a_6				$\approx \quad 0.0328830116668852$
a_7				$\approx -0.010597401785069032$

TABLE 1. Constant filter coefficients (a_j) of Daub $2N$ wavelets.

3. Numerical Values of Wavelets

Wavelet collocation methods [3] are frequently used to solve various integral and differential equations, but these methods often require knowledge of the values of the wavelet basis at specific points.

In order to find numerical values of Daubechies wavelets, we first need to find numerical values of their scaling functions. This task is not trivial due to the recursive definition of Daubechies scaling functions. This section explores methods that give good approximations to scaling function values and, by extension, their respective wavelets.

3.1. **Cascade Algorithm**

The cascade algorithm (developed by Daubechies [5]) is a fixed point method that generates a sequence of functions that converge to the Daub $2N$ wavelet scaling function, Φ.

We begin by discussing the scaling relation Φ will satisfy. If $\{V_k\}$ is a sequence of subspaces that form a multiresolution of $L^2(\mathbb{R})$ with scaling function φ, then it is clear that $\varphi \in V_0 \subset V_1$. Thus, φ has the *scaling relation*

$$\varphi = \sum_{j\in\mathbb{Z}} a_j \varphi_{1,j} \tag{5}$$

where $a_j := \langle \varphi, \varphi_{1,j} \rangle$.

Suppose now that F assigns the expression

$$F[\Lambda](x) = \sqrt{2} \sum_{j\in\mathbb{Z}} a_j \Lambda(2x-j) \tag{6}$$

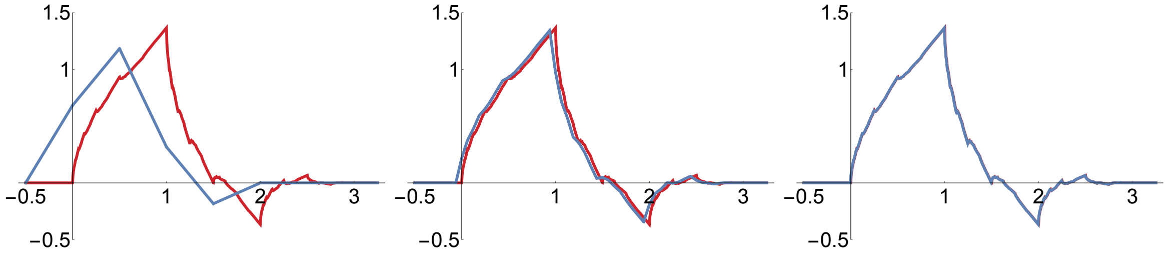
to the function Λ. Existence and uniqueness of integrable solutions to the general fixed point equation (6) was shown in [7]. In particular, using (5), we can see that Φ is a fixed point of F. One can use F to generate a sequence of functions $\{v_i\}$ that converge to Φ.

To begin generating the sequence of functions, we need to choose a starting function v_0 that is our "initial guess" for the fixed point of F. It is known that Φ is supported on $[0, 2N-1]$ and obtains a maximum on this interval. So a reasonable first guess is a function that is zero at all integers except $v_0(0) = 1$ (see [1]). If we connect the integer values of v_0 with lines we get

$$v_0(x) = \begin{cases} 1+x & \text{if } -1 \leq x < 0 \\ 1-x & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} .$$

We can generate the first iteration of the cascade algorithm using $v_1(x) = F(v_0)(x)$. Once we have v_1, we can generate the second iteration using $v_2(x) = F(v_1)(x)$. This process continues until the desired preciseness of the approximation is obtained. Figure 1 demonstrates the fast convergence of $v_i \to {}_2\Phi$ as $i \to \infty$. In particular, the following rate of convergence is obtained

Another reasonable first guess is $v_0(x) = \chi_{[0,1)}(x)$, see [2].



(A) 1^{st} iteration: v_1 (B) 4^{th} iteration: v_4 (C) 8^{th} iteration: v_8

FIGURE 1. Selected iterations of the cascade algorithm (blue) approximating the Daub 4 scaling function (red)

in [5, Chapter 7, Equation (7.2.17)]:

$$||\Phi_2 - v_j||_{L^\infty} \leq C \cdot 2^{-\alpha j}$$

where $\alpha = |\log((1+\sqrt{3})/4)|/\log(2) = 0.550$.

3.2. **Eigensystem Method**

The eigensystem method is an alternative approach to finding numerical values of a Daub $2N$ scaling function for $N > 1$ and is inspired by [5] and [19].

To begin, we will find all integers values that fall within the support of the Daub $2N$ scaling function, i.e., $\varphi(0), \varphi(1), \ldots, \varphi(2N-1)$. Recall that φ has the scaling relation in (5), so the integer values of φ must satisfy

$$\frac{1}{\sqrt{2}}\varphi(i) = \sum_{j\in\mathbb{Z}} a_j \varphi(2i-j) = \sum_{j=0}^{2N-1} a_{2i-j}\varphi(j), \qquad \text{for } i = 0, 1, \ldots, 2N-1.$$

We can express this system of equations in matrix from as

$$\frac{1}{\sqrt{2}} v = Av \tag{7}$$

where A is $2N \times 2N$ and the i, j^{th} entry is $A_{i,j} = a_{2i-j}$ for $i, j = 0, 1, \ldots, 2N-1$ and 0 else, and $v = [\varphi(0)\ \ \varphi(1)\ \cdots\ \varphi(2N-1)]^T$ is an eigenvector of A with eigenvalue $\frac{1}{\sqrt{2}}$. Once v is found, it must be scaled so that the sum of its entries satisfies $\sum_i \varphi(i) = 1$, agreeing with Proposition 2.4. Solving for these eigenvectors can be done by hand or using an algorithm (e.g. the QR algorithm [20]; note also that the matrix A is sparse). However, it is usually more efficient for the integer values to be calculated beforehand and stored. Solving for eigenvectors is very computationally expensive when compared to the recursive style branching and should be avoided when possible.

Now that we have the values of $\varphi(i)$ for $i = 0, 1, \ldots, 2N-1$, we can use them to find the values of $\varphi\left(\frac{i}{2}\right)$ for $i = 0, 1, \ldots, 4N-2$ with

$$\varphi\left(\tfrac{i}{2}\right) = \sqrt{2} \sum_{j=0}^{2N-1} a_j \varphi(i-j)$$

because $(i-j)$ is clearly an integer. We can extend this to solve for $\varphi\left(\frac{i}{2^k}\right)$ for any $i, k \in \mathbb{Z}$ using

$$\varphi\left(\tfrac{i}{2^k}\right) = \sqrt{2} \sum_{j=0}^{2N-1} a_j \varphi\left(\frac{i - j2^{k-1}}{2^{k-1}}\right) \tag{8}$$

given that the value of $\varphi\left(\frac{i}{2^{k-1}}\right)$ is known for $i = 0, 1, \ldots, 2^{k-1}(2N-1)$.

Remark 3.1. *This method does not work for the* Daub 2 *(Haar) wavelet because matrix* $\sqrt{2}A$ *becomes the identity matrix so that we cannot solve for the eigenvector* v *in (7). However, it is not hard to see that* $\varphi(x) = {}_1\Phi(x) = \chi_{[0,1)}(x)$ *satisfies the scaling relation* $\varphi(x) = \varphi(2x) + \varphi(2x-1)$ *and that the wavelet formed,* ${}_1\Psi(x) = \chi_{\left[0,\frac{1}{2}\right)}(x) - \chi_{\left[\frac{1}{2},1\right)}(x)$*, satisfies all the properties of a* Daub 2 *wavelet.*

We end this section by discussing a recent algorithm [12] which has been developed for calculating exact values of Daubechies wavelets, which can be viewed as an improvement of the Cascade algorithm discussed in the previous section. This algorithm is based on calculating exact integer values of the scaling function, as in the previous section. Once these values are found then Ma uses a convolution operation to compute the value of the scaling function at odd-dyadics. The convolution operation is equivalent to (8) and yields the vector

$$\left[\varphi\left(\frac{1}{2^k}\right) \quad \varphi\left(\frac{3}{2^k}\right) \quad \cdots \quad \varphi\left(\frac{2^k(2N-1)-1}{2^k}\right)\right]$$

when give the value of $\varphi\left(\frac{i}{2^{k-1}}\right)$ at $i = 0, 1, \ldots, 2^{k-1}(2N-1)$. In the next section we introduce the new method for calculating values of the scaling function φ.

4. The Branch Method

Sometimes the value of the Daub $2N$ scaling function φ is needed only at a single point. Both the cascade algorithm and the eigensystem method require values of φ at many points to find the value of φ at a specific point. The branch method finds $\varphi\left(\frac{i}{2^k}\right)$ for any $i, k \in \mathbb{Z}$ using recursion and can be thought of as the eigensystem method in reverse. All integer values of φ must be known to use the branch method. The values of $\varphi(n)$ for $n = 0, 1, \dots, 2N-1$ can be calculated using the method discussed in Section 3.2.

Recall the modified scaling relation in (8). In order to find the value of $\varphi\left(\frac{i}{2^k}\right)$, we must find the values of $\varphi\left(\frac{i-j2^{k-1}}{2^{k-1}}\right)$ for $j = 0, 1, \dots, 2N-1$. If we apply the scaling relation to each $\varphi\left(\frac{i-j2^{k-1}}{2^{k-1}}\right)$, then we would need to find the values of $\varphi\left(\frac{n_u}{2^{k-2}}\right)$ for some numerators $n_0, n_1, \dots, n_{2N-1} \in \mathbb{Z}$. Applying the scaling relation can be thought of creating at most $2N$ branches that need to be summed together in order to calculate the parent node as shown in Figure 2 where $h_j := \sqrt{2}a_j$. Note that exactly $2N$ branches are not always created because of the cases when the argument of $\varphi(x)$ falls outside the interval $[0, 2N-1]$. In these cases, $\varphi(x) = 0$ due to the compact support of φ and that branch need not be considered.

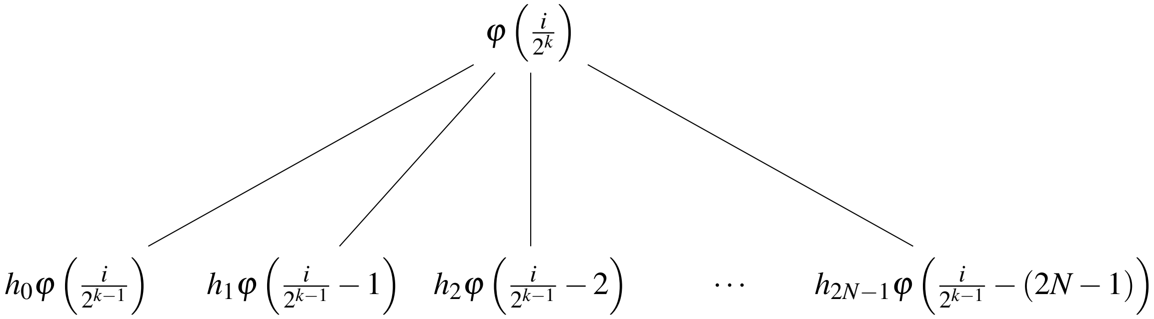

$\varphi\left(\frac{i}{2^k}\right)$
$h_0\varphi\left(\frac{i}{2^{k-1}}\right)$
$h_1\varphi\left(\frac{i}{2^{k-1}}-1\right)$
$h_2\varphi\left(\frac{i}{2^{k-1}}-2\right)$
$\cdots$
$h_{2N-1}\varphi\left(\frac{i}{2^{k-1}}-(2N-1)\right)$

FIGURE 2. Visualization of the branch method

The modified scaling relation in (8) can be applied k times so that all the lowest branches are of the form $\varphi(n)$ for some $n \in \mathbb{Z}$. Then we only need to add together the many integer values of φ with their respective filter coefficients to find $\varphi\left(\frac{i}{2^k}\right)$; see Algorithm 1 below.

Algorithm 1 Branch Method

```
// Assume the values of constant filter coefficients $a_0, a_1, \ldots, a_{2N-1}$ and the integer values of
// the scaling function $\varphi(0), \varphi(1) \ldots, \varphi(2N-1)$ for a Daub $2N$ wavelet are known
function BM($i,k,N$)
    // Returns value of Daub $2N$ scaling function $\varphi(x)$ at $x = i \cdot 2^{-k}$
    $x \leftarrow i \cdot 2^{-k}$
    if $0 \leq x \leq 2N-1$ then
        if $x \in \mathbb{Z}$ then
            return $\varphi(x)$
        else
            $y \leftarrow 0$
            for $j = 0, 1, \ldots, 2N-1$ do
                $y \leftarrow y + a_j \cdot \mathrm{BM}(i - j \cdot 2^{k-1}, k-1, N)$
            return $\sqrt{2} \cdot y$
    else
        return 0
```

The plot in Figure 3 shows calculations made by the Branch Method of exact values of the Daub 4 scaling function at dyadic rationals. The dyadic rationals $i \cdot 2^{-k}$ are categorized by their k value to make the dyadic rational 'levels' more visible.

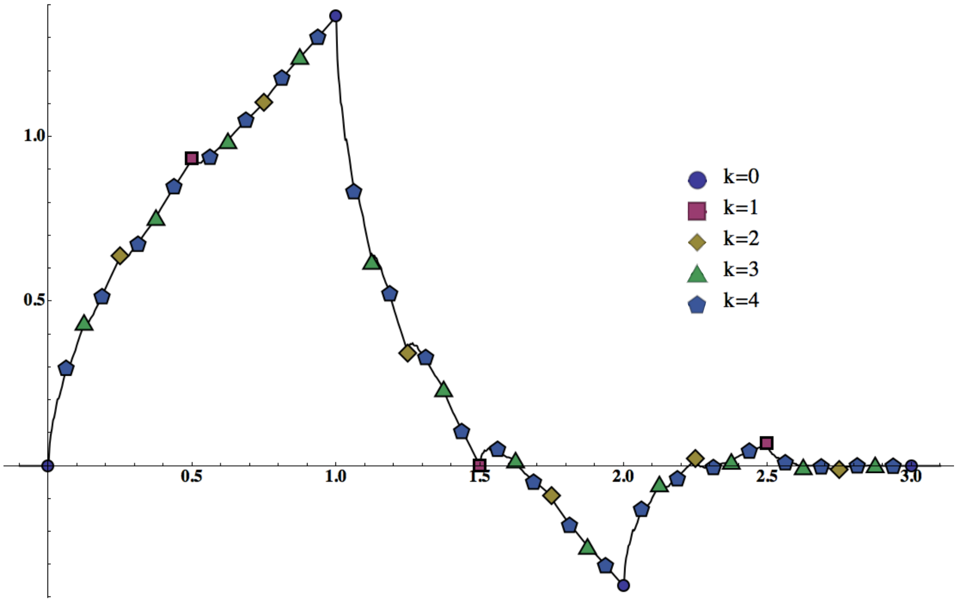


FIGURE 3. Daub 4 scaling function approximations using branch method at $x = i \cdot 2^{-k}$ at $k = 0, 1, 2, 3, 4$ and all relevant i

Remark 4.1. *If it is desired to find the value of $\varphi(x)$ for some $x \neq \frac{i}{2^k}$ for all $i,k \in \mathbb{Z}$, one can approximate x with $x \approx \frac{i'}{2^{k'}}$ for some $i',k' \in \mathbb{Z}$ (dyadic rationals are dense in $\mathbb{R}$). Since φ is continuous, $\varphi(x) \approx \varphi\left(\frac{i'}{2^{k'}}\right)$. Note that the* Daub 2 *scaling function is not continuous but it has a closed form expression. Thus, this method is not necessary to find numerical values of the* Daub 2 *scaling function.*

Example 1. *We will find the value of the* Daub 4 *scaling function $\varphi\left(\frac{13}{8}\right)$ using the branch method. Using the filter coefficients from Table 1, if we solve for the eigenvector in*

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\varphi(0)\\ \varphi(1)\\ \varphi(2)\\ \varphi(3)\end{bmatrix} = \begin{bmatrix}a_0 & 0 & 0 & 0\\ a_2 & a_1 & a_0 & 0\\ 0 & a_3 & a_2 & a_1\\ 0 & 0 & 0 & a_3\end{bmatrix}\begin{bmatrix}\varphi(0)\\ \varphi(1)\\ \varphi(2)\\ \varphi(3)\end{bmatrix}$$

and scale the eigenvector such that $\sum_n \varphi(n) = 1$, we obtain:

$$\varphi(0) = 0 \qquad \varphi(1) = \frac{1+\sqrt{3}}{2} \qquad \varphi(2) = \frac{1-\sqrt{3}}{2} \qquad \varphi(3) = 0.$$

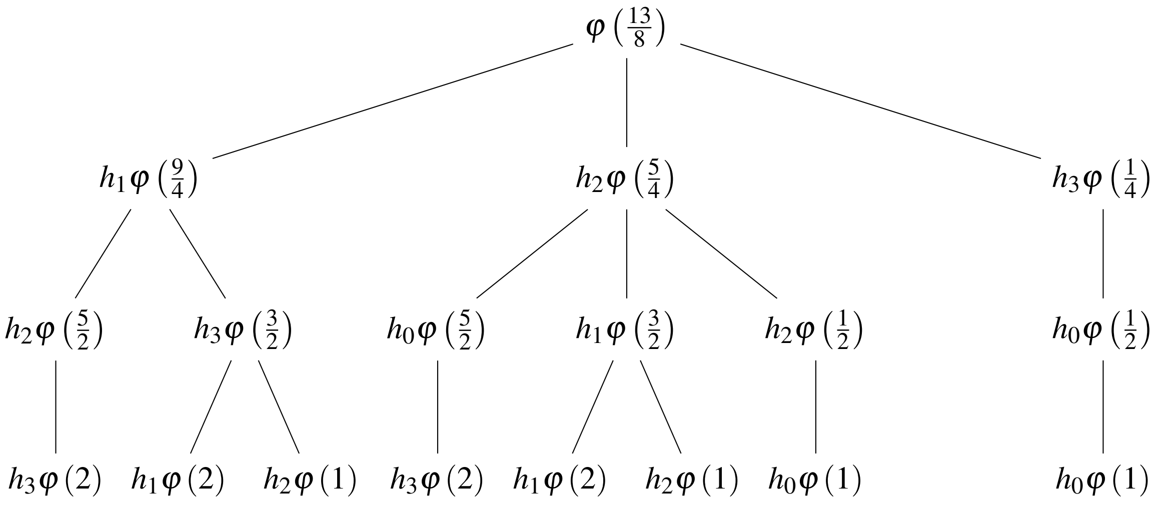

$\varphi\left(\frac{13}{8}\right)$
$h_1\varphi\left(\frac{9}{4}\right)$
$h_2\varphi\left(\frac{5}{4}\right)$
$h_3\varphi\left(\frac{1}{4}\right)$
$h_2\varphi\left(\frac{5}{2}\right)$
$h_3\varphi\left(\frac{3}{2}\right)$
$h_0\varphi\left(\frac{5}{2}\right)$
$h_1\varphi\left(\frac{3}{2}\right)$
$h_2\varphi\left(\frac{1}{2}\right)$
$h_0\varphi\left(\frac{1}{2}\right)$
$h_3\varphi(2)$
$h_1\varphi(2)$
$h_2\varphi(1)$
$h_3\varphi(2)$
$h_1\varphi(2)$
$h_2\varphi(1)$
$h_0\varphi(1)$
$h_0\varphi(1)$

FIGURE 4. Non-zero branches for Daub 4 scaling function $\varphi\left(\frac{13}{8}\right)$

The modified scaling relation in (8) is applied three times to obtain integer arguments, as shown in Figure 4. We get an expression for $\varphi\left(\frac{13}{8}\right)$ from the tree:

$$\begin{aligned}\varphi\left(\tfrac{13}{8}\right) &= h_1\big(h_2h_3\varphi(2)+h_3(h_1\varphi(2)+h_2\varphi(1))\big)\\ &\quad+h_2\big(h_0h_3\varphi(2)+h_1(h_1\varphi(2)+h_2\varphi(1))+h_2h_0\varphi(1)\big)\\ &\quad+h_3h_0h_0\varphi(1)\\ &=\frac{2-\sqrt{3}}{16}\approx 0.0167468.\end{aligned}$$

Remark 4.2. *When implementing the Branch Method, all calculated values of the scaling function, i.e., $\varphi\left(\frac{j}{2^k}\right)$, at each branch node should be stored. The stored values can be used when calculating future branches. This will dramatically increase efficiency as scaling fuction values at each dyadic rational will only be calculated once.*

We end this section by discussing the computational issues arising in the Branch method. The most computationally expensive part of the process is the eigenvector calculation for the $2N\times 2N$ matrix A. Typically in direct dense methods [9], $\mathscr{O}((2N)^3)$ time is required and $\mathscr{O}((2N)^2)$ space is required to compute the eigenvectors of A. However, in our case we have three advantages: namely, the matrix is sparse, the associated eigenvalue is known, and we do not need to calculate *all* eigenvectors of A. Thus a more efficient calculation (via e.g. inverse iteration) can be done.

5. Daubechies Wavelet Transform

Many of the practical applications of wavelets revolve around the wavelet transform. The transform is a method for approximating functions in $L^2(\mathbb{R})$ with a finite number of wavelets in a smart and efficient manner. We start by recalling the definition of the wavelet transform:

Definition 5.1. Suppose $\psi = {}_N\Psi$ is a Daub$2N$ wavelet and $\varphi = {}_N\Phi$ is its associated scaling function. The *Daubechies wavelet transform* of a function $f\in L^2(\mathbb{R})$ is defined as

$$D_nf(x) := \sum_{j\in\mathbb{Z}}\langle f,\varphi_{0,j}\rangle\varphi_{0,j}(x)+\sum_{k=0}^{n-1}\sum_{j\in\mathbb{Z}}\langle f,\psi_{k,j}\rangle\psi_{k,j}(x).$$

In practice, the wavelet transform is used to approximate functions over a finite domain. The function to be approximated can be considered to equal zero outside of this finite domain so that only a finite number of the inner products $\langle f, \psi_{k,j}\rangle$, referred to as *wavelet coefficients*, need to be computed in $D_n f$. We will find that these wavelet coefficients play a large role in the maximum error of $D_n f$ and the overall accuracy of the wavelet approximation.

Lemma 5.2. *If $f \in L^2(\mathbb{R})$ is globally Lipschitz continuous with Lipschitz constant C, then*

$$|\langle f, \psi_{k,j}\rangle| \leq C\left(\frac{2N-1}{2^k}\right)^{3/2}.$$

Proof. The function $\psi_{k,j}$ is supported on the set

$$I_{k,j} := \left[\frac{j}{2^k}, \frac{2N-1+j}{2^k}\right]$$

due to the compact support of ψ over $[0, 2N-1]$. Let $M_{k,j} = \inf\{f(x) : x \in I_{k,j}\}$. Recall from Definition 2.2 that the zeroth moment of ψ vanishes meaning that the equality

$$0 = \int_{\mathbb{R}} \psi(x)dx = 2^{k/2}\int_{I_{k,j}} \psi(2^k x - j)dx = \int_{I_{k,j}} \psi_{k,j}(x)dx$$

holds. We then obtain:

$$\begin{aligned} |\langle f, \psi_{k,j}\rangle| &= \left|\int_{I_{k,j}} f(x)\psi_{k,j}(x)dx - M_{k,j}\int_{I_{k,j}} \psi_{k,j}(x)dx\right| \\ &= \left|\int_{I_{k,j}} \left(f(x) - M_{k,j}\right)\psi_{k,j}(x)dx\right| \\ &\leq \int_{I_{k,j}} \left|f(x) - M_{k,j}\right| \left|\psi_{k,j}(x)\right| dx. \end{aligned}$$

Applying Hölder's inequality yields

$$|\langle f, \psi_{k,j}\rangle| \leq \left(\int_{I_{k,j}} \left|f(x) - M_{k,j}\right|^2 dx\right)^{\frac{1}{2}} \left(\int_{I_{k,j}} \left|\psi_{k,j}(x)\right|^2 dx\right)^{\frac{1}{2}} = \left(\int_{I_{k,j}} \left(f(x) - M_{k,j}\right)^2 dx\right)^{\frac{1}{2}}$$

because $\psi_{k,j}$ is part of an orthonormal basis, hence $\|\psi_{k,j}\|_2 = 1$. This follows from the fact that one can decompose $L^2(\mathbb{R})$ as

$$L^2(\mathbb{R}) = \bigoplus_{k\in\mathbb{Z}} W_k$$

where $W_k = \{f \in V_{k+1} : f \perp V_k\}$, see e.g. [5].

Next, define the coefficient

$$\omega_{k,j} = \sup\left\{|f(x_1) - f(x_2)| : x_1, x_2 \in I_{k,j}\right\},$$

We see that the inequality

$$0 \le f(x) - M_{k,j} \le \omega_{k,j}$$

holds for all $x \in I_{k,j}$. This inequality assures us that the integral in the previous calculations is less than or equal to the area of a rectangle with dimensions of $\omega_{k,j}^2$ and the width of the domain $I_{k,j}$, so that

$$|\langle f, \psi_{k,j}\rangle| \le \left(\omega_{k,j}^2\left(\frac{2N-1+j}{2^k} - \frac{j}{2^k}\right)\right)^{\frac{1}{2}} = \frac{\omega_{k,j}\sqrt{2N-1}}{2^{k/2}}.$$

Since f is globally Lipschitz continuous, we can see that the inequality $\omega_{k,j} \le C(2N-1)2^{-k}$ is satisfied. By using this upper bound of $\omega_{k,j}$, we are left with the inequality

$$|\langle f, \psi_{k,j}\rangle| \le \frac{\left(C(2N-1)2^{-k}\right)\sqrt{2N-1}}{2^{k/2}} = C\left(\frac{2N-1}{2^k}\right)^{3/2}$$

as desired.

Wavelets are well suited to approximate piecewise smooth signals [21]. In these types of cases where the signal function f is globally Lipschitz continuous, we can establish uniform convergence of the Daubechies wavelet transform.

Theorem 5.3. *If $f \in L^2(\mathbb{R})$ is globally Lipschitz continuous with Lipschitz constant C, then $D_n f$ converges uniformly to f.*

Proof. Since the set $\{\phi_{0,j}, \psi_{k,j} : j,k \in \mathbb{Z},\ k \ge 0\}$ is an orthonormal basis of $L^2(\mathbb{R})$, we have that f may be expressed as

$$f = \sum_{j\in\mathbb{Z}} \langle f, \varphi_{0,j}\rangle \varphi_{0,j} + \sum_{k=0}^{\infty}\sum_{j\in\mathbb{Z}} \langle f, \psi_{k,j}\rangle \psi_{k,j}$$

and the difference between the original function and the transform is

$$f - D_n f = \sum_{k=n}^{\infty}\sum_{j\in\mathbb{Z}} \langle f, \psi_{k,j}\rangle \psi_{k,j}.$$

$\omega_{k,j}$ has a similar role to the *modulus of continuity* of f.

To prove uniform convergence, it suffices to show that $\|f - D_n f\|_\infty \to 0$ as $n \to \infty$. The following calculations use the results of Lemma 5.2 further simplify the difference:

$$
\begin{aligned}
|f(x) - D_n f(x)| &\le \sum_{k=n}^{\infty} \sum_{j\in\mathbb{Z}} |\langle f, \psi_{k,j}\rangle| |\psi_{k,j}(x)| \\
&\le \sum_{k=n}^{\infty} \sum_{j\in\mathbb{Z}} 2^{k/2} C \left(\frac{2N-1}{2^k}\right)^{3/2} |\psi(2^k x - j)| \qquad (9)\\
&= C(2N-1)^{3/2} \sum_{k=n}^{\infty} 2^{-k} \sum_{j\in\mathbb{Z}} |\psi(2^k x - j)|.
\end{aligned}
$$

Due to the compact support of ψ there are at most $2N-1$ different integer values of j where $|\psi(2^k x - j)| \neq 0$ for any given x. Thus, the inequality

$$\sum_{j\in\mathbb{Z}} |\psi(2^k x - j)| \le (2N-1)\|\psi\|_\infty$$

holds. This inequality allows to continue our calculations in (9) yielding:

$$
\begin{aligned}
|f(x) - D_n f(x)| &\le C(2N-1)^{3/2} \sum_{k=n}^{\infty} 2^{-k}(2N-1)\|\psi\|_\infty \\
&\le C(2N-1)^{5/2}\|\psi\|_\infty \sum_{k=n}^{\infty} 2^{-k} \\
&\le C(2N-1)^{5/2}\|\psi\|_\infty 2^{1-n}.
\end{aligned}
$$

The result of these calculations allow us to conclude that the inequality

$$\|f - D_n f\|_\infty \le C(2N-1)^{5/2}\|\psi\|_\infty 2^{1-n}$$

also holds. Therefore $\lim_{n\to\infty} \|f - D_n f\|_\infty = 0$.

Remark 5.4. *Uniform convergence of $D_n f$ to f is still obtained even if f is only uniformly continuous on $\mathbb{R}$ as well as convergence in L^2 for any $f \in L^2(\mathbb{R})$* [8].

5.1. **Error Analysis**

If E_n is the error of $D_n f$ using wavelets of the class Daub$2N$ where $D_n f(x) = f(x) + E_n(x)$, the result of the proof of Theorem 5.3 assure us that the error of the transform is at most

$$\|E_n\|_\infty \le C(2N-1)^{5/2}\|{}_N\Psi\|_\infty 2^{1-n} \qquad (10)$$

N	1	2	3	4	5	6	7	8	9	10
$\|{}_N\Psi\|_\infty$	1	1.73205	1.70112	1.35907	1.19308	1.12634	1.12161	1.10919	1.06982	1.02596

TABLE 2. Approximate values of $\|{}_N\Psi\|_\infty$ for different values of N

if f is globally Lipschitz continuous with Lipschitz constant C.

We also mention that the L^2 error was first analyzed rigorously in [14], which resulted in a characterization of Sobolev spaces in terms of error decay rates of wavelet expansions.

5.1.1. Hölder Continuous Functions

In the case that f is Hölder continuous with the additional constant $0 < \alpha \leq 1$, by performing the same steps as we did in the proof of Theorem 5.3, we find that

$$\omega_{k,j} \leq C\left(\frac{2N-1}{2^k}\right)^{\alpha}$$

which implies that

$$|\langle f, \psi_{k,j}\rangle| \leq \frac{\omega_{k,j}(2N-1)^{1/2}}{2^{k/2}} \leq C(2N-1)^{\alpha+\frac{1}{2}} \cdot 2^{-k(\alpha+\frac{1}{2})}. \tag{11}$$

Notice that this estimate agrees with [16, Equation 5.4], but we have explicitly written the constant depending on N. In particular, (11) is equivalent to $f \in C^{0,\alpha}(\mathbb{R})$. This stricter restriction on $\omega_{k,j}$ leads us to the following estimate:

$$\|E_n\|_\infty \leq \|f - D_n f\|_\infty \leq C(2N-1)^{\alpha+\frac{3}{2}} \|{}_N\Psi\|_\infty \frac{2^{\alpha(1-n)}}{2^\alpha - 1} \tag{12}$$

The Daub 2 wavelet has the smallest error bound for all α, making it the ideal choice of wavelet to use to approximate a Hölder continuous function (assuming that the function is not twice differentiable).

5.1.2. Differentiable Functions

How will the transform error be affected if the function f is more smooth and has multiple derivatives? To answer this question we will need to find a more strict condition on $\langle f, \psi_{k,j}\rangle$ for cases when f has m derivatives.

Proposition 5.5. *If f is m times continuously differentiable where the i^{th} derivative of f is bounded by B_i over $I_{k,j}$ and ψ is a* Daub$2N$ *wavelet, then the inequality*

$$\left|\langle f,\psi_{k,j}\rangle\right| \leq 2^{\left(\frac{1-k}{2}\right)} \sum_{i=\tau}^{m} \frac{B_i\left(N-\frac{1}{2}\right)^{i+\frac{1}{2}}}{2^{ki} i! \sqrt{2i+1}} \tag{13}$$

is satisfied where $\tau = \min\{m,N\}$.

Proof. We first expand the inner product $\langle f,\psi_{k,j}\rangle$. Starting with its integral form

$$\langle f,\psi_{k,j}\rangle = \int_{\mathbb{R}} f(x)\, 2^{k/2}\psi(2^k x - j)\, dx,$$

substitute $t = 2^k x - j$, yielding

$$\langle f,\psi_{k,j}\rangle = 2^{-k/2}\int_{I_{0,0}} f\left(\frac{t+j}{2^k}\right)\psi(t)dt$$

since the support of ψ is $I_{0,0}$. The Taylor polynomial of f on the interval $\eta_b = I_{k,j}$ centered at the midpoint $b = \frac{2N-1+2j}{2^{k+1}}$ can be substituted for the function f since it is a C^m function:

$$f(x) = p_b(x) + \varepsilon_b(x) = \sum_{i=0}^{m-1} \frac{f^{(i)}(b)}{i!}(x-b)^i + \varepsilon_b(x)$$

where the error $\varepsilon_b(x)$ satisfies

$$|\varepsilon_b(x)| \leq \frac{|x-b|^m}{m!} \sup_{u\in\eta_b} |f^{(m)}(u)|$$

The following simplifies the integral form of $\langle f,\psi_{k,j}\rangle$ by utilizing the Taylor polynomial:

$$\begin{aligned}\langle f,\psi_{k,j}\rangle &= 2^{-k/2}\int_{I_{0,0}}\left(\sum_{i=0}^{m-1}\frac{f^{(i)}(b)}{i!}\left(\frac{t+j}{2^k}-b\right)^i + \varepsilon_b\left(\frac{t+j}{2^k}\right)\right)\psi(t)dt \\ &= 2^{-k/2}\left(\sum_{i=0}^{m-1}\frac{f^{(i)}(b)}{2^{ki}i!}\int_{I_{0,0}}\left(t-N+\tfrac{1}{2}\right)^i\psi(t)dt + \int_{I_{0,0}}\varepsilon_b\left(\frac{t+j}{2^k}\right)\psi(t)dt\right)\end{aligned}$$

For $i<N$, the expansion of $\left(t-N+\frac{1}{2}\right)^i$ is a polynomial of degree less than N. Since ψ has N vanishing moments, ψ is orthogonal to any polynomial of degree less than N. Therefore the equality

$$\int_{I_{0,0}}\left(t-N+\tfrac{1}{2}\right)^i\psi(t)dt = 0$$

holds for all $i = 0, 1, \ldots, N-1$. We are left with the following expansion:

$$\langle f, \psi_{k,j} \rangle = 2^{-k/2} \left(\sum_{i=N}^{m-1} \frac{f^{(i)}(b)}{2^{ki} i!} \int_{I_{0,0}} \left(t - N + \tfrac{1}{2}\right)^i \psi(t) dt + \int_{I_{0,0}} \varepsilon_b \left(\frac{t+j}{2^k} \right) \psi(t) dt \right)$$

We then obtain

$$\begin{aligned}
\left|\langle f, \psi_{k,j} \rangle\right| &\leq 2^{-\frac{k}{2}} \left(\sum_{i=N}^{m-1} \frac{|f^{(i)}(b)|}{2^{ki} i!} \int_{I_{0,0}} \left| \left(t - N + \tfrac{1}{2}\right)^i \right| |\psi(t)| dt + \int_{I_{0,0}} \left| \varepsilon_b \left(\frac{t+j}{2^k} \right) \right| |\psi(t)| dt \right) \\
&\leq 2^{-\frac{k}{2}} \left(\sum_{i=N}^{m-1} \frac{B_i}{2^{ki} i!} \int_{I_{0,0}} \left| t - N + \tfrac{1}{2} \right|^i |\psi(t)| dt + \int_{I_{0,0}} \frac{\left| \frac{t+j}{2^k} - b \right|^m}{m!} \sup_{u \in \eta_b} \left| f^{(m)}(u) \right| |\psi(t)| dt \right) \\
&\leq 2^{-\frac{k}{2}} \left(\sum_{i=N}^{m-1} \frac{B_i}{2^{ki} i!} \int_{I_{0,0}} \left| t - N + \tfrac{1}{2} \right|^i |\psi(t)| dt + \int_{I_{0,0}} \frac{\left| t - N + \frac{1}{2} \right|^m}{2^{km} m!} B_m |\psi(t)| dt \right) \\
&\leq 2^{-\frac{k}{2}} \sum_{i=\tau}^{m} \frac{B_i}{2^{ki} i!} \int_{I_{0,0}} \left| t - N + \tfrac{1}{2} \right|^i |\psi(t)| dt
\end{aligned}$$

because $|f^{(i)}(x)| \leq B_i$ for every x and the result of Taylor's Theorem. The index of the sum starts at τ instead of N because of the possibility that $N > m$. Next, apply Hölder's inequality to the integral. This leaves us with the desired inequality:

$$\begin{aligned}
\left|\langle f, \psi_{k,j} \rangle\right| &\leq 2^{-k/2} \sum_{i=\tau}^{m} \frac{B_i}{2^{ki} i!} \left(\int_{I_{0,0}} \left| t - N + \tfrac{1}{2} \right|^{2i} dt \cdot \int_{I_{0,0}} |\psi(t)|^2 dt \right)^{\frac{1}{2}} \\
&= 2^{-k/2} \sum_{i=\tau}^{m} \frac{B_i}{2^{ki} i!} \left(\int_{I_{0,0}} \left(t - N + \tfrac{1}{2}\right)^{2i} dt \right)^{\frac{1}{2}} \\
&= 2^{-k/2} \sum_{i=\tau}^{m} \frac{B_i}{2^{ki} i!} \left(\frac{2\left(N - \frac{1}{2}\right)^{2i+1}}{2i+1} \right)^{\frac{1}{2}} \\
&= 2^{\left(\frac{1-k}{2}\right)} \sum_{i=\tau}^{m} \frac{B_i \left(N - \frac{1}{2}\right)^{i+\frac{1}{2}}}{2^{ki} i! \sqrt{2i+1}}
\end{aligned}$$

Remark 5.6. *In Proposition 5.5, we only need to consider the non-zero derivatives of f. This is because if $f^{(i)} = 0$, then $B_i = 0$ and the i^{th} term of the sum does not affect the value of the upper bound of $|\langle f, \psi_{k,j} \rangle|$. If f has infinitely many non-zero derivatives, then the value $m = N$ should be chosen to minimize the upper bound of $|\langle f, \psi_{k,j} \rangle|$.*

Using this new inequality from Proposition 5.5 in the calculations in (9) from the proof of Theorem 5.3 yields an error that obeys the inequality

$$\|E_n\|_\infty \leq \|_N\Psi\|_\infty \sum_{i=\tau}^{m} \frac{B_i\sqrt{2i+1}(2N-1)^{i+\frac{3}{2}}}{2^{ni}(4^i-1)\,i!} \leq B\|_N\Psi\|_\infty \sum_{i=\tau}^{m} \frac{\sqrt{2i+1}(2N-1)^{i+\frac{3}{2}}}{2^{ni}(4^i-1)\,i!} \tag{14}$$

where $B = \max\{B_i : i = \tau, \dots, m\}$. Figure 5 displays a log plot of the right side of the above inequality for different values of N using $m = 4$ and $B = 1$. We can see that the error bound is smallest when $N = m$.

A similar analysis can be done on the derivatives of f. Namely, if $f \in C^m(\mathbb{R})$ then we can estimate the difference between the p-th derivative of f, for $0 < p < m$, and its approximation $D_n f$ in the L^∞ norm to get

$$\left\| \frac{d^p f}{dx^p} - \frac{d^p(D_n f)}{dx^p} \right\|_\infty \lesssim 2^{-n(m-p)} \tag{15}$$

where the implicit constant again depends only on B_i, N and $||_N\Psi||_\infty$; compare with [11, Theorem 2] where a similar result is shown for Coiflets.

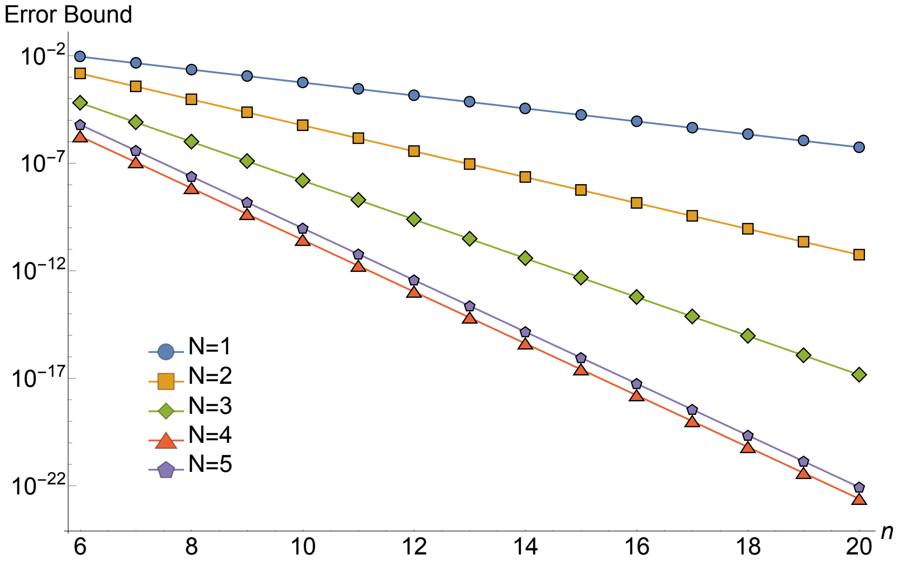


FIGURE 5. Log plot upper of $\|E_n\|_\infty$ using Daub$2N$ wavelets to approximate a C^4 function.

Remark 5.7. *There is also the Zygmund class* [5] *of "differentiable" functions*

$$\Lambda_*^s = \left\{ f \in L^\infty(\mathbb{R}) \cap C^{s-1}(\mathbb{R}) : \frac{d^{s-1}}{dx^{s-1}} f \in \Lambda_* \right\}$$

where

$$\Lambda_* = \left\{ f \in L^\infty(\mathbb{R}) \cap C^0(\mathbb{R}) : \sup_{x,h} \frac{|f(x+h)+f(x-h)-2f(x)|}{h} < \infty \right\}$$

In this case a characterization of the space Λ_*^s for integer s (the analog of C^s functions) was given in [15]*:*

$$f \in \Lambda_*^s \iff |\langle f, \psi_{-k,j}\rangle| \leq C \cdot 2^{-k\left(s+\frac{1}{2}\right)}, \qquad k \geq 0, j \in \mathbb{Z}$$

5.1.3. Minimizing the Error Bound

The results of sections 5.1.1 and 5.1.2 show a correlation between the error bound of the transform $D_n f$ and the support of the wavelet ${}_N\Psi$. Indeed, the error bound is smallest for larger N when $N \geq 2$. The special case $N = 1$, i.e. the Haar wavelet, deserves special attention. Based on Table 2, it would appear that the Haar wavelet is the optimal choice for approximating a globally Lipschitz function (as well as for a Hölder function as discussed in Section 5.1.1). Intuitively this makes sense since Haar wavelets are not differentiable. In particular, since the Haar wavelet has the smallest possible support and only one vanishing moment, it is not ideal for approximating smooth functions, as can be seen in our previous analysis.

The error bound calculations performed in this section bring forward an alternative metric for choosing the best suited Daub$2N$ wavelet. Notice that the maximum value of E_n hinges on the values of the wavelet coefficients, $\langle f, \psi_{k,j}\rangle$. Thus, an alternative method for choosing the optimal wavelet is to *choose a wavelet that produces the maximum number of wavelet coefficients $\langle f, \psi_{k,j}\rangle$ that are close to zero* [13]. This not only minimizes the error of $D_n f$ but it can also drastically speed up the process of calculating the values of $D_n f$ when the wavelet coefficients can be approximated by zero, $\langle f, \psi_{k,j}\rangle \approx 0$.

6. Conclusion

We have developed a new algorithm for finding exact values of Daubechies wavelets. The only computationally challenging aspect of this method is the calculation of an eigenvector, but as the matrix is sparse, this is not an issue. The new method does not require the calculation of values of the scaling function at dyadic rationals. We have also provided a new quantitative method for deciding which wavelet is best for applications. We expect these results to be useful for numerical methods for various partial differential equations, which will be done in future work.

Conflict of Interests

The authors declare that there is no conflict of interests.

REFERENCES

[1] E. Aboufadel, R. Meyers, Wavelets, Introduction, Encycl. Phys. Sci. Technol. (2002), 773-788.

[2] A. E. Age, A survey of the development of Daubechies scaling functions, Master's Thesis, University of South Florida, 2010.

[3] S. Bertoluzza and G. Naldi, A wavelet collocation method for the numerical solution of partial differential equations, Appl. Comput. Harmon. Anal., 3 (1) (1996), 1-9.

[4] I. Daubechies, Orthonormal bases of compactly supported wavelets, Commun. Pure Appl. Math. 41 (7) (1988), 909-996.

[5] I. Daubechies, Ten lectures on Wavelets, SIAM, 1992.

[6] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math. 57 (11) (2004), 1413-1457.

[7] I. Daubechies, J. Lagarias, Two-scale difference equations I: Existence and global regularity of solutions, SIAM J. Math. Anal. 22 (5) (1991), 1388-1410.

[8] K. R. Davidson, A. P. Donsig, Real Analysis with real applications, Prentice Hall, 2002.

[9] I. S. Duff, A. M. Erisman, J. K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, 1986.

[10] R. Kronland-Martinet, J. Morlet, A. Grossmann, Analysis of Sound Patterns through wavelet transforms,Int. J. Pattern Recognit. Artif. Intell. 1 (02) (1987), 273-302.

[11] X. Liu, Y. Zhou, L. Zhang, and J. Wang, Wavelet solutions of Burger's equation with high Reynolds number, Sci. China Tech. Sci. 57 (7) (2014), 1285-1292.

[12] H. Ma, On the Exact values of Daubechies Wavelets, J. Phys. Math. 7(1) (2016).

[13] S. G. Mallat, A wavelet tour of signal processing, Academic Press, 1999.

[14] S. G. Mallat, Multiresolution approximations and wavelet orthonormal bases of L^2, Trans. Am. Math. Soc. 315 (1) (1989), 69-87.

[15] Y. Meyer, Ondelettes et operateurs, Paris, France 1990.

[16] M. J. Mohlenkamp, M. C. Pereyra, Wavelets, their friends, and what they can do for you, European Mathematical Society, 2008.

[17] S. Qian, J. Weiss, Wavelets and the numerical solution of partial differential equations, J. Comput. Phys. 106 (1) (1993), 155-175.

[18] O. Rioul, M. Vetterli, Wavelets and signal processing, IEEE Signal Process Mag. 8 (4) (1991), 14-38.

[19] P. Schniter, Computing the scaling function: The Cascade Algorithm, OpenStax CNX, June 1, 2009, `http://cnx.org/contents/d279ef3c-661f-4e14-bba8-70a4eb5c0bcf@7`.

[20] L. N. Trefethen, D. Bau III, Numerical Linear Algebra, Vol. 50, SIAM, 1997.

[21] M. Vetterli, Wavelets, approximation, and compression, IEEE Signal Process Mag. 18 (5) (2001), 59-73.