Algorithm 929: A Suite on Wavelet Differentiation Algorithms

MANI MEHRA and KAVITA GOYAL, Indian Institute of Technology Delhi

A collection of the Matlab routines that compute the values of the scaling and wavelet functions ($\phi(x)$ and $\psi(x)$ respectively) and the derivative of an arbitrary function (periodic or non periodic) using wavelet bases is presented. Initially, the case of Daubechies wavelets is taken and the procedure is explained for both collocation and Galerkin approaches. For each case a Matlab routine is provided to compute the differentiation matrix and the derivative of the function $f^{(d)} = \mathcal{D}^{(d)} f$. Moreover, the convergence of the derivative is shown graphically as a function of different parameters (the wavelet genus, D and the scale, J) for two test functions. We then consider the use of spline wavelets.

Categories and Subject Descriptors: G.1.2 [**Numerical Analysis**]: Approximation—*Wavelets and fractals*; G.1.8 [**Numerical Analysis**]: Partial Differential Equations—*Multigrid and multilevel methods*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Differentiation matrices, wavelets, wavelet methods (Galerkin and collocation), splines.

ACM Reference Format:
Mehra, M. and Goyal, K. 2013. Algorithm 929: A suite on wavelet differentiation algorithms ACM Trans. Math. Softw. 39, 4, Article 27 (July 2013), 28 pages.
DOI: http://dx.doi.org/10.1145/2491491.2491497

27

1. INTRODUCTION

The term *differentiation matrix* $\mathcal{D}^{(d)}$ denotes the transformation between grid point values of a function and its d-th order derivative. In the case of the Galerkin approach [Canuto et al. 1988] this matrix is the product of three matrices, that is, $\mathcal{D}^{(d)} = C^{-1}D^{(d)}C$. The matrix C, called the *quadrature matrix*, constructs an approximation, $P_N f$, of a given function $f(x)$ on the interval $[a, b]$ from a vector of point values of $f(x)$, that is, from $\{f(x_j) : 0 \le j \le N-1\}$. The approximation $P_N f$ belongs to a finite-dimensional space. The second matrix, $D^{(d)}$ called the *differentiation projection matrix* results from differentiating $P_N f$ and projecting it back to a finite-dimensional space. Hence $D^{(d)}$ is defined by the linear transformation between $P_N f$ and $P_N(d^d/dx^d)P_N f$, (See Jameson [1993] for details). In the case of the collocation approach [Canuto et al. 1988] the elements of the matrix $\mathcal{D}^{(d)}$ are values of the derivatives of the basis functions (translates and dilates of the scaling function) at the grid points.

Differentiation matrices may be used to solve differential equations arising in the real world. Many attractive mathematical properties of wavelets, for instance, efficient

M. Mehra acknowledges the support provided by Department of Science and Technology, India, under grant number RP02417. K. Goyal would like to thank the Council of Scientific and Industrial Research for providing a PhD scholarship.
Author's addressess: M. Mehra, Indian Institute of Technology Delhi, Hauz Khas, New Delhi-110 016; email: mmehra@maths.iitd.ac.in; K. Goyal, Indian Institute of Technology Delhi, Hauz Khas, New Delhi-110 016; email: goyalkavita9@gmail.com.

DOI: http://dx.doi.org/10.1145/2491491.2491497

multiscale decompositions, compact support, vanishing moments, the existence of fast wavelet transform and spectral type of convergence of numerical approximation of differential operators etc. together with the techniques for preconditioning and compression of operators and matrices, motivated their use for numerical solution of PDEs. Wavelet methods have been developed for most kinds of PDEs such as Laplace/Poisson equations [Dahmen 1997; Vasilyev and Kevlahan 2005] and advection diffusion problems [Mehra and Kumar 2005], Burgers equation [Vasilyev and Bowman 2000], reaction-diffusion equations [Schneider et al. 1997], Stokes equation [Dahmen et al. 2002] and for PDEs on manifolds [Mehra and Kevlahan 2008]. In this article wavelet differentiation matrix for the functions on an interval will be examined. A collection of the Matlab routines that compute the values of the scaling and wavelet functions ($\phi(x)$ and $\psi(x)$ respectively) and the derivative of an arbitrary function using Daubechies and spline wavelet bases is presented. Moreover, the convergence of the derivative is shown graphically as a function of different parameters, that is, the wavelet genus (D), and the scale J.

The article consists of three sections. Section 1 is this introduction. Section 2 describes the Galerkin and the collocation approach for construction of differentiation matrices using Daubechies wavelets while Section 3 deals with spline wavelets. Our software consists of 26 Matlab functions, which are described in an electronic appendix.

2. DAUBECHIES WAVELET

2.1. Real Line

Multiresolution analysis (MRA) [Daubechies 1992; Mallat 1989] is the theory that was used by I. Daubechies to show that for any nonnegative integer n there exists an orthogonal wavelet with compact support such that all the derivatives up to order n exist; this is characterized by the following axioms:

(1) $\mathcal{V}^j \subset \mathcal{V}^{j+1}$ (subspaces are nested),
(2) $f(x) \in \mathcal{V}^j$ iff $f(2x) \in \mathcal{V}^{j+1}$ for all $j \in \mathbb{Z}$ (invariance to dilation),
(3) $\overline{\bigcup_{j\mathcal{V}^j} \mathcal{V}^j} = L^2(\mathbb{R})$,
(4) $\{\phi(x-k) | k \in \mathbb{Z}\}$ is an orthonormal basis for $\mathcal{V}^0$ (invariance to translation) for a function $\phi(x) \in \mathcal{V}^0$; this is called the *scaling function*.

$\mathcal{W}^j$ is defined as the orthogonal complement of $\mathcal{V}^j$ in $\mathcal{V}^{j+1}$, that is, $\mathcal{V}^j \perp \mathcal{W}^j$ and

$$\mathcal{V}^{j+1} = \mathcal{V}^j \oplus \mathcal{W}^j.$$

Since the set $\{\phi(x-k) | k \in \mathbb{Z}\}$ is an orthonormal basis for $\mathcal{V}^0$ by axiom (4) of MRA, it follows by repeated application of axiom (2) that $\{\phi_k^j(x) = 2^{j/2}\phi(2^j x - k) | k \in \mathbb{Z}\}$ is an orthonormal basis for $\mathcal{V}^j$ and similarly, there exists a function $\psi(x) \in \mathcal{W}^0$ (which is called the *mother wavelet*) such that $\{\psi_k^j(x) = 2^{j/2}\psi(2^j x - k) | k \in \mathbb{Z}\}$ is an orthonormal basis for $\mathcal{W}^j$.

Since $\phi_0^0(x) = \phi(x) \in \mathcal{V}^0 \subset \mathcal{V}^1$, we have

$$\phi(x) = \sum_{k=-\infty}^{\infty} h_k \phi_k^1(x),$$

where

$$h_k = \int_{-\infty}^{\infty} \phi(x)\phi_k^1(x) dx.$$

For Daubechies, compactly supported scaling functions, only finitely many $h_k, k = 0, 1, \ldots D-1$ will be nonzero, where D is an even positive integer called the *wavelet genus*. Hence we have

$$\phi(x) = \sqrt{2}\sum_{k=0}^{D-1} h_k \phi(2x-k). \tag{1}$$

Equation (1) is called the *dilation equation* (a two scale relation for the scaling function) and and $h_0, h_1, \ldots, h_{D-1}$ are called *low-pass filter coefficients*. Similarly, Daubechies compactly supported wavelet $\psi(x) \in \mathcal{W}^0 \subset \mathcal{V}^1$ is defined by

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

Equation (2) is called the *wavelet equation* (a two scale relation for the wavelet function) and $g_0, g_1, \ldots, g_{D-1}$ are called *high-pass filter coefficients*. These filter coefficients are connected by the relation $g_k = (-1)^k h_{D-1-k}, k = 0, 1, \ldots, D-1$. We make use of the Matlab wavelet toolbox and especially the function wfilters which computes both the low pass and the high pass filter coefficients. An important consequence of equations (1) and (2) is that $supp(\phi) = supp(\psi) = [0, D-1]$, see [Daubechies 1992]. It follows that

$$supp\left(\phi_k^j\right) = supp\left(\psi_k^j\right) = I_k^j,$$

where

$$I_k^j = \left[\frac{k}{2^j}, \frac{k+D-1}{2^j}\right].$$

2.1.1. An Algorithm for the Numerical Evolution of ϕ and ψ One should notice that there is in general, no closed form analytic (explicit) formulae for either Daubechies scaling functions, $\phi(x)$ or wavelet functions, $\psi(x)$. An exception is the Haar scaling function ($\phi(x) = 1$ if $x \in [0, 1)$, $\phi(x) = 0$ otherwise) and the Haar wavelet function ($\psi(x) = 1$ if $x \in [0, .5)$, $\psi(x) = -1$ if $x \in [.5, 1)$, $\psi(x) = 0$ otherwise). However, values for general scaling and wavelet functions can be computed at dyadic points using the cascade algorithm [Daubechies 1992; Strang and Nguyen 1996] as follows

Computing ϕ at integers. The scaling function ϕ has support on the interval $[0, D-1]$, with $\phi(0) = 0$ and $\phi(D-1) = 0$ for $D \geq 4$ [Daubechies 1992].

By substituting $x = 0, 1, \ldots, D-2$ in (1), we obtain a homogeneous linear system of equations. For $D = 6$ we have

$$\begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = \sqrt{2}\begin{bmatrix} h_0 & & & & \\ h_2 & h_1 & h_0 & & \\ h_4 & h_3 & h_2 & h_1 & h_0 \\ & h_5 & h_4 & h_3 & h_2 \\ & & & h_5 & h_4 \end{bmatrix} \times \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = A_0\Phi(0), \tag{3}$$

where the vector-valued function $\Phi(x)$ is defined as

$$\Phi(x) = [\phi(x), \phi(x+1), \ldots, \phi(x+D-2)]^T.$$

It can be observed that solving (3) is equivalent to solving the eigenvalue problem

$$A_0\Phi(0) = \lambda\Phi(0). \tag{4}$$

The solution of (3) is the eigenvector of A_0 corresponding to the eigenvalue $\lambda = 1$ (note that the eigenvalues of A_0 include $\lambda = 2^{-m},\ m = 0, 1, \ldots, D/2-1$, [Strang and Nguyen

1996]). The requirement that ϕ has unit area gives rise to the following result

$$\sum_{k=0}^{D-1} \phi(k) = 1$$

(see Nielsen [1998] for details). We use this result to fix the multiplicative constant which arises from solving (4).

Computing ϕ at dyadic rationals. Having obtained $\Phi(0)$ from (3) we can again use (1) to obtain ϕ at all the midpoints between the integers in the interval, namely the vector $\Phi(1/2)$. Substituting $x = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots$ into (1) gives

$$\Phi\left(\frac{1}{2}\right) = \begin{bmatrix} \phi(\frac{1}{2}) \\ \phi(\frac{3}{2}) \\ \phi(\frac{5}{2}) \\ \phi(\frac{7}{2}) \\ \phi(\frac{9}{2}) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_1 & h_0 & & & \\ h_3 & h_2 & h_1 & h_0 & \\ h_5 & h_4 & h_3 & h_2 & h_1 \\ & & h_5 & h_4 & h_3 \\ & & & & h_5 \end{bmatrix} \times \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \\ \phi(4) \end{bmatrix} = A_1 \Phi(0). \quad (5)$$

Next for the rationals of the form $k/4$, where k is odd, we obtain

$$\begin{bmatrix} \phi(\frac{1}{4}) \\ \phi(\frac{3}{4}) \\ \phi(\frac{5}{4}) \\ \phi(\frac{7}{4}) \\ \phi(\frac{9}{4}) \\ \phi(\frac{11}{4}) \\ \phi(\frac{13}{4}) \\ \phi(\frac{15}{4}) \\ \phi(\frac{17}{4}) \\ \phi(\frac{19}{4}) \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_0 & & & & \\ h_1 & h_0 & & & \\ h_2 & h_1 & h_0 & & \\ h_3 & h_2 & h_1 & h_0 & \\ h_4 & h_3 & h_2 & h_1 & h_0 \\ h_5 & h_4 & h_3 & h_2 & h_1 \\ & h_5 & h_4 & h_3 & h_2 \\ & & h_5 & h_4 & h_3 \\ & & & h_5 & h_4 \\ & & & & h_5 \end{bmatrix} \times \begin{bmatrix} \phi(0) \\ \phi(\frac{1}{2}) \\ \phi(\frac{3}{2}) \\ \phi(\frac{5}{2}) \\ \phi(\frac{7}{2}) \\ \phi(\frac{9}{2}) \end{bmatrix},$$

which can be written as

$$\Phi\left(\frac{1}{4}\right) = A_0 \Phi\left(\frac{1}{2}\right),$$

$$\Phi\left(\frac{3}{4}\right) = A_1 \Phi\left(\frac{1}{2}\right).$$

Using the same two matrices for all the steps in the algorithm we can continue as follows until a desired resolution 2^q is obtained, for $i = 2, 3, \ldots, q$ and $k = 1, 3, 5, \ldots, 2^{j-1} - 1$,

$$\Phi\left(\frac{k}{2^i}\right) = A_0 \left(\frac{k}{2^{i-1}}\right),$$

$$\Phi\left(\frac{k}{2^i} + \frac{1}{2}\right) = A_1 \left(\frac{k}{2^{i-1}}\right).$$

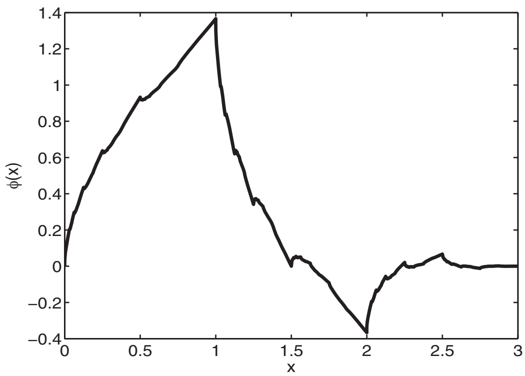

Fig. 1. Daubechies scaling function $\phi(x)$.

Function values of ψ can be computed from the values of ϕ using the wavelet equation (2)

$$\psi(m/2^q) = \sqrt{2}\sum_{k=0}^{D-1} g_k \phi(2m/2^q - k). \tag{6}$$

The computation of ϕ and ψ at dyadic rationals is implemented by the function `cascade.m`; more details may be found in the accompanied User Manual, Section 1. The function $\phi(x)$ is plotted using `cascade.m` in Figure 1 for $D = 4$ and $q = 8$.

2.2. Periodic Domain

So far, our functions have been defined on the entire real line, for instance, $f \in L^2(\mathbb{R})$. In most practical applications such as image processing, data fitting or problems involving differential equations, the space domain is a finite interval, say, for simplicity, the interval is [0, 1] and the function f is periodic, that is, $f(0) = f(1)$. These cases can be dealt with by using periodic scaling and wavelet functions that are defined as follows

Definition 2.1. Let $\phi \in \mathcal{L}^2(\mathbb{R})$ and $\psi \in \mathcal{L}^2(\mathbb{R})$ be the basic scaling function and the basic wavelet from an MRA as defined in Section 2.1. For any $j, k \in \mathbb{Z}$ we define the 1-periodic scaling function

$$\tilde{\phi}_k^j(x) = \sum_{n=-\infty}^{\infty} \phi_k^j(x+n) = 2^{j/2} \sum_{n=-\infty}^{\infty} \phi(2^j(x+n) - k), x \in \mathbb{R},$$

and the 1-periodic wavelet

$$\tilde{\psi}_k^j(x) = \sum_{n=-\infty}^{\infty} \psi_k^j(x+n) = 2^{j/2} \sum_{n=-\infty}^{\infty} \psi(2^j(x+n) - k), x \in \mathbb{R}. \tag{7}$$

The 1-periodicity can be verified as follows

$$\tilde{\phi}_k^j(x+1) = \sum_{n=-\infty}^{\infty} \phi_k^j(x+n+1) = \sum_{m=-\infty}^{\infty} \phi_k^j(x+m) = \tilde{\phi}_k^j(x)$$

and similarly $\tilde{\psi}_k^j(x+1) = \tilde{\psi}_k^j(x)$.
Some of the important results regarding periodic scaling and wavelet functions [Nielsen 1998] are as follows.

(1) $\tilde{\phi}_k^j(x)$ is constant and is equal to $2^{-j/2}$ for $j \leq 0$.
(2) $\tilde{\psi}_k^j(x) = 0$ for $j \leq -1$.
(3) $\tilde{\phi}_k^j(x)$ and $\tilde{\psi}_k^j(x)$ are periodic in the shift parameter k with period 2^j for $j > 0$.

Now suppose

$$\tilde{\mathcal{V}}^j = \overline{\left\langle \left\{ \tilde{\phi}_k^j(x), x \in [0, 1] \right\}_{k=0}^{2^j-1} \right\rangle} \text{ and } \tilde{\mathcal{W}}^j = \overline{\left\langle \left\{ \tilde{\psi}_k^j(x), x \in [0, 1] \right\}_{k=0}^{2^j-1} \right\rangle},$$

it can be observed that the $\tilde{\mathcal{V}}^j$ are nested in a similar way as the $\mathcal{V}^j$ in axiom 1 of MRA (see Section 2.1), that is,

$$\tilde{\mathcal{V}}^0 \subset \tilde{\mathcal{V}}^1 \subset \tilde{\mathcal{V}}^2 \subset \cdots \subset \mathcal{L}^2([0, 1]),$$

and $\overline{\bigcup_{j=0}^{\infty} \tilde{\mathcal{V}}^j} = \mathcal{L}^2([0, 1])$. The orthogonality, which is a property of nonperiodic scaling and wavelet functions carried over to the periodic versions restricted to the interval [0,1] implies that

$$\tilde{\mathcal{V}}^j \oplus \tilde{\mathcal{W}}^j = \tilde{\mathcal{V}}^{j+1}.$$

So the space $\mathcal{L}^2([0, 1])$ has the decomposition

$$\mathcal{L}^2([0, 1]) \approx \tilde{\mathcal{V}}^{J_0} \oplus \left(\bigoplus_{j=J_0}^{\infty} \tilde{\mathcal{W}}^j \right),$$

for some $J_0 > 0$.

2.2.1. Evaluation of Scaling Function Coefficients. Suppose a function $f \in \mathcal{L}^2([0, 1])$, with $f(0) = f(1)$, is given and we wish to compute the projection of this function in the space $\tilde{\mathcal{V}}^j$, that is,

$$P_{\tilde{\mathcal{V}}^j} f(x) = \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^j(x), \quad x \in [0, 1]. \tag{8}$$

Note that there are two natural ways to obtain the scaling function coefficients c_k^j in (8)

(1) *Projection.* Because of the orthogonality of the basis functions, the coefficients c_k^j can be obtained using the relation

$$c_k^j = \int_I f(x) \tilde{\phi}_k^j(x) dx.$$

This method is called *orthogonal projection*. The integral can be approximated by any sufficiently accurate quadrature method.

(2) *Interpolation.* The coefficients c_k^j are chosen such that the projection of f on $\mathcal{V}^j$, and f coincides at the node points at level j, that is,

$$f\left(\frac{l}{2^r}\right) = \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^j \left(\frac{l}{2^r}\right), \quad l = 0, 1, \ldots, 2^r - 1,$$

where $r \in \mathbb{N}$ is called the dyadic resolution of the function.

Using the interpolation technique to find the c_k^j we can obtain from (8)

$$
\begin{aligned}
f\left(\frac{l}{2^r}\right) &= \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^j\left(\frac{l}{2^r}\right) \\
&= \sum_{k=0}^{2^j-1} c_k^j \sum_{n\in\mathbb{Z}} \phi_k^j\left(\frac{l}{2^r}+n\right) \\
&= 2^{j/2} \sum_{k=0}^{2^j-1} c_k^j \sum_{n\in\mathbb{Z}} \phi\left(\frac{m(l,k)+2^{j+q}n}{2^q}\right),
\end{aligned}
$$

where $m(l,k) = l2^{j+q-r} - k2^q$. Now if j is such that $2^j \geq D-1$, then we have

$$
f\left(\frac{l}{2^r}\right) = 2^{j/2} \sum_{k=0}^{2^j-1} c_k^j \phi\left(\frac{\langle m(l,k)\rangle_{2^{j+q}}}{2^q}\right), \quad l = 0, 1, \ldots, 2^r-1 \tag{9}
$$

(see Nielsen [1998] for details). From (9) we see that $m(l,k)$ serves as an index into the vector of precomputed values of ϕ. For this to make sense $m(l,k)$ must be an integer, which leads to the restriction

$$
j + q - r \geq 0.
$$

Suppose $c_j = [c_0^j, c_1^j, \ldots, c_{2^j-1}^j]^T$ and $f_r = [f(0), f(1/2^r), \ldots, f((2^r-1)/2^r)]^T$, then (9) can be written as

$$
f_r = T_{r,j} c_j, \tag{10}
$$

where $T_{r,j}$ is a matrix of size $2^r \times 2^j$. For the case $D = 4, r = 4$ and $j = 3$, we have

$$
\begin{bmatrix} f(0) \\ f(\frac{1}{16}) \\ f(\frac{2}{16}) \\ f(\frac{3}{16}) \\ f(\frac{4}{16}) \\ f(\frac{5}{16}) \\ f(\frac{6}{16}) \\ f(\frac{7}{16}) \\ f(\frac{8}{16}) \\ f(\frac{9}{16}) \\ f(\frac{10}{16}) \\ f(\frac{11}{16}) \\ f(\frac{12}{16}) \\ f(\frac{13}{16}) \\ f(\frac{14}{16}) \\ f(\frac{15}{16}) \end{bmatrix}
= 2^{\frac{3}{2}}
\begin{bmatrix}
\phi(0) & & & & & & \phi(2) & \phi(1) \\
\phi(\frac{1}{2}) & & & & & & \phi(\frac{5}{2}) & \phi(\frac{3}{2}) \\
\phi(1) & \phi(0) & & & & & & \phi(2) \\
\phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & & & & & \phi(\frac{5}{2}) \\
\phi(2) & \phi(1) & \phi(0) & & & & & \\
\phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & & & & \\
& \phi(2) & \phi(1) & \phi(0) & & & & \\
& \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & & & \\
& & \phi(2) & \phi(1) & \phi(0) & & & \\
& & \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & & \\
& & & \phi(2) & \phi(1) & \phi(0) & & \\
& & & \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & & \\
& & & & \phi(2) & \phi(1) & \phi(0) & \\
& & & & \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2}) & \\
& & & & & \phi(2) & \phi(1) & \phi(0) \\
& & & & & \phi(\frac{5}{2}) & \phi(\frac{3}{2}) & \phi(\frac{1}{2})
\end{bmatrix}
\times
\begin{bmatrix} c_3^0 \\ c_3^1 \\ c_3^2 \\ c_3^3 \\ c_3^4 \\ c_3^5 \\ c_3^6 \\ c_3^7 \end{bmatrix}.
$$

Given f_r, calculating c_j using (10) is termed *discrete scaling function transformation* (DST) and given the c_j, calculating f_r using (10) is termed *inverse discrete scaling function transformation* (IDST). We have implemented these transformations as the functions `dst.m` and `idst.m` (see the accompanied User Manual, Section 2 for more details).

2.2.2. Galerkin Approach.

The differentiation projection matrix.
Let $f \in \mathcal{V}^j$, then

$$f^{(d)}(x) = \sum_{l=-\infty}^{\infty} c_l^j \phi_l^{j(d)}(x), \quad x \in \mathbb{R}, \tag{11}$$

$f^{(d)}$ will in general not belong to $\mathcal{V}^j$ so we project $f^{(d)}$ back onto $\mathcal{V}^j$

$$P_{\mathcal{V}^j} f^{(d)}(x) = \sum_{k=-\infty}^{\infty} c_k^{j(d)} \phi_k^j(x), \quad x \in \mathbb{R},$$

where

$$c_k^{j(d)} = \int_{-\infty}^{\infty} f^{(d)}(x)\phi_k^j(x)dx. \tag{12}$$

Substituting (11) into (12) and doing some manipulations, we get

$$c_k^{j(d)} = \sum_{n=-\infty}^{\infty} c_{n+k}^j 2^{jd} \Gamma_n^d, \quad -\infty < k < \infty, \tag{13}$$

where

$$\Gamma_n^d = \int_{-\infty}^{\infty} \phi(x)\phi_n^{(d)}(x)dx, \quad n \in \mathbb{Z}, \tag{14}$$

are called *connection coefficients*. Now because $\phi(x)$ is compactly supported on $[0, D-1]$, it can be shown that support of $\phi(x)$ and $\phi_n^{(d)}(x)$ overlap only for $-(D-2) \leq n \leq (D-2)$, so there are only $2D-3$ nonzero connection coefficients. Hence (13) reduces to

$$c_k^{j(d)} = \sum_{n=2-D}^{D-2} c_{n+k}^j 2^{jd} \Gamma_n^d, \quad j, k \in \mathbb{Z}. \tag{15}$$

Now if f is 1-periodic function then

$$c_k^j = c_{k+2^j}^j, \quad k \in \mathbb{Z}$$

and

$$c_k^{j(d)} = c_{k+2^j}^{j(d)}, \quad k \in \mathbb{Z}.$$

Hence it is sufficient to consider 2^j coefficients of either type and (15) becomes

$$c_k^{j(d)} = \sum_{n=2-D}^{D-2} c_{\langle n+k \rangle_{2^j}}^j 2^{jd} \Gamma_n^d, \quad k = 0, 1, \ldots, 2^j - 1,$$

which can be written in matrix form as

$$\mathbf{c}^{(d)} = D^{(d)}\mathbf{c}, \tag{16}$$

where $[D^{(d)}]_{k,\langle n+k\rangle_{2^j}} = 2^{jd}\Gamma_n^d, \quad k = 0, 1, \ldots, 2^j - 1;\ n = 2 - D, 3 - D, \ldots, D - 2$ and

$$\mathbf{c}^{(d)} = \left[c_0^{j(d)}, c_1^{j(d)}, \ldots, c_{2^j-1}^{j(d)}\right].$$

The matrix $D^{(d)}$ is the *differentiation projection matrix*. Here, under the assumption of periodicity of $f(x)$, the *quadrature matrix* $C = T^{-1}$ (where T is the matrix $T_{r,j}$ in (10)) and the differentiation projection matrix $D^{(d)}$ are circulant and hence commute. Thus in this case $\mathcal{D}^{(d)} = D^{(d)}$ and

$$\mathbf{f}^{(d)} = \mathcal{D}^{(d)}\mathbf{f}.$$

Note that if f is periodic with period L, then we have

$$\mathbf{f}^{(d)} = \frac{1}{L^d}\mathcal{D}^{(d)}\mathbf{f}.$$

The differentiation matrix has the following structure for $D = 4$ and $j = 3$

$$\mathcal{D}^{(d)} = 2^{3d}\begin{bmatrix} \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 & (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d \\ (-1)^d\Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 & (-1)^d\Gamma_2^d \\ (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 \\ 0 & (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 & 0 \\ 0 & 0 & (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d & 0 \\ 0 & 0 & 0 & (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d & \Gamma_0^d & \Gamma_1^d & \Gamma_2^d \\ \Gamma_2^d & 0 & 0 & 0 & (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d & \Gamma_0^d & \Gamma_1^d \\ \Gamma_1^d & \Gamma_2^d & 0 & 0 & 0 & (-1)^d\Gamma_2^d & (-1)^d\Gamma_1^d & \Gamma_0^d \end{bmatrix}$$

Note that the property $\Gamma_n^d = (-1)^d\Gamma_{-n}^d$ (which can be easily verified) is used while constructing the above matrix. An important case is $d = 1$, and we define

$$\mathcal{D} = \mathcal{D}^{(1)}.$$

We can now see that we need the set $\boldsymbol{\Gamma}^d = \{\Gamma_n^d\}_{n=2-D}^{D-2}$ for the construction of *differentiation matrix*.

Algorithm for computing connection coefficients. Using the dilation equation (1) we obtain

$$\phi_l(x) = \phi(x - l) = \sqrt{2}\sum_{k=0}^{D-1} h_k\phi(2(x - l) - k) = \sqrt{2}\sum_{k=0}^{D-1} h_k\phi_{2l+k}(2x). \tag{17}$$

Differentiating (17) d times gives

$$\phi_l^{(d)}(x) = 2^d\sqrt{2}\sum_{k=0}^{D-1} h_k\phi_{2l+k}^{(d)}(2x). \tag{18}$$

Substituting (1) and (18) into (14), we have

$$\begin{aligned}
\Gamma_n^d &= \int_{-\infty}^{\infty}\left[\sqrt{2}\sum_{r=0}^{D-1} h_r\phi_r(2x)\right]\left[2^d\sqrt{2}\sum_{s=0}^{D-1} h_s\phi_{2n+s}^{(d)}(2x)\right]dx \\
&= 2^{d+1}\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} h_r h_s \int_{-\infty}^{\infty}\phi_r(2x)\phi_{2n+s}^{(d)}(2x)dx, \quad x \leftarrow 2x \\
&= 2^{d}\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} h_r h_s \int_{-\infty}^{\infty}\phi_r(x)\phi_{2n+s}^{(d)}(x)dx, \quad x \leftarrow x-r \\
&= 2^{d}\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} h_r h_s \int_{-\infty}^{\infty}\phi(x)\phi_{2n+s-r}^{(d)}(x)dx, \quad x \leftarrow x-r,
\end{aligned}$$

or

$$\sum_{r=0}^{D-1}\sum_{s=0}^{D-1} h_r h_s \Gamma_{2n+s-r}^d = \frac{1}{2^d}\Gamma_n^d, \quad n \in [2-D, D-2]. \tag{19}$$

Let $m = 2n + s - r$. It has already been explained that Γ_m^d is nonzero only for $m \in [2-D, D-2]$ and that $s = r + m - 2n$, as well as r, must be restricted to $[0, D-1]$. This is fulfilled for $max(0, 2n-m) \leq r \leq min(D-1, D-1+2n-m)$. Let $p = 2n - m$ and define

$$\bar{a}_p = \sum_{r=r_1(p)}^{r_2(p)} h_r h_{r-p},$$

where $r_1(p) = max(0, p)$ and $r_2(p) = min(D-1, D-1+p)$. Hence (19) becomes

$$\sum_{m=2-D}^{D-2} \bar{a}_{2n-m}\Gamma_m^d = \frac{1}{2^d}\Gamma_n^d, \quad n \in [2-D, D-2],$$

which can be written in matrix form as

$$(\mathbf{A} - 2^{-d}\mathbf{I})\Gamma^d = \mathbf{0}, \tag{20}$$

where $\mathbf{A}$ is a matrix of order $(2D-3)$ with elements

$$[\mathbf{A}]_{n,m} = \bar{a}_{2n-m}.$$

Equation (20) has a nontrivial solution if 2^{-d} is an eigenvalue of $\mathbf{A}$. Numerical calculations for $D = 4, 6, \ldots, 30$ indicate that 2^{-d} is an eigenvalue for $d = 0, 1, \ldots, D-1$. The additional condition needed to normalize the solution is obtained by using the property of vanishing moments, which is explained as follows:

Moments of the scaling functions. For a wavelet of genus D, the number of vanishing moments is $D/2$ and this property implies that the scaling function can represent the polynomials up to degree $(D/2-1)$ exactly, that is,

$$x^p = \sum_{l=-\infty}^{\infty} M_l^p \phi(x-l), \quad p = 0, 1, \ldots, D/2-1, \tag{21}$$

where the p^{th} *moment* of $\phi(x-l)$ is defined as

$$M_l^p = \int_{-\infty}^{\infty} x^p \phi(x-l), \quad l, p \in \mathbb{Z}. \tag{22}$$

It is known that the area under the scaling function $\phi(x)$ is 1, hence

$$M_l^0 = 1, \quad l \in \mathbb{Z}.$$

Let $l = 0$. The dilation equation (1) then yields

$$\begin{aligned} M_0^p &= \int_{-\infty}^{\infty} x^p \phi(x) dx \\ &= \sqrt{2} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} x^p \phi(2x-k) dx \\ &= \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} a_k \int_{-\infty}^{\infty} y^p \phi(y-k) dy, \quad y = 2x, \end{aligned}$$

or

$$M_0^p = \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} a_k M_k^p. \tag{23}$$

Using the variable transformation $y = x - l$ in (22)

$$\begin{aligned} M_l^p &= \int_{-\infty}^{\infty} (y+l)^p \phi(y) dy \\ &= \sum_{n=0}^{p} \binom{p}{n} l^{p-n} \int_{-\infty}^{\infty} y^n \phi(y) dy, \end{aligned}$$

or

$$M_l^p = \sum_{n=0}^{p} \binom{p}{n} l^{p-n} M_0^n, \tag{24}$$

Substituting (24) into (23) we obtain

$$\begin{aligned} M_0^p &= \frac{\sqrt{2}}{2^{p+1}} \sum_{k=0}^{D-1} a_k \sum_{n=0}^{p} \binom{p}{n} k^{p-n} M_0^n \\ &= \frac{\sqrt{2}}{2^{p+1}} \sum_{n=0}^{p-1} \binom{p}{n} M_0^n \sum_{k=0}^{D-1} a_k k^{p-n} + \frac{\sqrt{2}}{2^{p+1}} M_0^p \underbrace{\sum_{k=0}^{D-1} a_k}_{\sqrt{2}}, \end{aligned}$$

and solving for M_0^p yields

$$M_0^p = \frac{\sqrt{2}}{2(2^p - 1)} \sum_{n=0}^{p-1} \binom{p}{n} M_0^n \sum_{k=0}^{D-1} a_k k^{p-n}. \tag{25}$$

Equation (25) can be used to determine the p^{th} moment of $\phi(x)$, M_0^p for any $p > 0$. The translated moments M_l^p are then obtained by using (24). We have included in our suite the function `moments.m` which implements (25) and computes the p^{th} moments of $\phi(x)$.

The normalisation condition is derived as follows: Differentiating (21) d times we obtain

$$d! = \sum_{l=-\infty}^{\infty} M_l^d \phi^{(d)}(x-l). \tag{26}$$

Multiplying (26) by $\phi(x)$ and integrating we obtain

$$\begin{aligned} d! \int_{-\infty}^{\infty} \phi(x)dx &= \sum_{l=-\infty}^{\infty} M_l^d \int_{-\infty}^{\infty} \phi(x)\phi^{(d)}(x-l)dx \\ &= \sum_{l=2-D}^{D-2} M_l^d \int_{-\infty}^{\infty} \phi(x)\phi^{(d)}(x-l)dx. \end{aligned}$$

Therefore, the additional condition to normalize the solution becomes

$$\sum_{n=2-D}^{D-2} M_n^d \Gamma_n^d = d!. \tag{27}$$

Hence Γ^d is found as follows.

—Let $\mathbf{v}^d$ be an eigenvector corresponding to the eigenvalue 2^{-d} of matrix $\mathbf{A}$.
—Then $\Gamma^d = k\mathbf{v}$ for some constant k.
—The constant k is fixed by using (27).

Remark 2.2. There is an exception to the statement that 2^{-d} is an eigenvalue of $\mathbf{A}$ for $d = 0, 1, \ldots, D-1$. For $D = 4$ eigenvalues of $\mathbf{A}$ are

$$\frac{1}{8}, \frac{1}{4} + 6.4765 \times 10^{-9}i, \frac{1}{4} - 6.4765 \times 10^{-9}i, \frac{1}{2}, 1.$$

Consequently $\frac{1}{4}$ is not an eigenvalue of $\mathbf{A}$ and the connection coefficients for the combination $D = 4$, $d = 2$ are not well defined.

The functions `conn.m` and `gal_difmatrix_periodic.m` compute the connection coefficients and differential matrix respectively (see the accompanied User Manual, Section 4 for details).

Convergence results. If $f(x)$ is a 1-periodic function and the error is defined as

$$E^{(d)}(f, j) = \max_{k=0,1,\ldots,2^j-1} \left| \left[f_n^{(d)} \right]_k - f^{(d)}\left(\frac{k}{2^j}\right) \right|,$$

where $f_n^{(d)}$ and $f^{(d)}$ denotes the numerical and analytic value of the d-order derivative of f, then the following convergence result holds

$$f \in C^D(\mathbb{R}) \quad \Rightarrow \quad E^{(1)}(f, j) \leq C2^{-jD}, \tag{28}$$

where $C^D(\mathbb{R})$ denotes the space of functions having continuous derivatives of order $\leq D$ and C is a constant. Assuming that a similar result holds for higher order of differentiation, that is,

$$E^{(d)}(f, j) \leq C2^{-jR}, \tag{29}$$

where R probably depends on d and N. It follows that

$$R = log_2(E^{(d)}(f, j)) - log_2(E^{(d)}(f, j+1)),$$

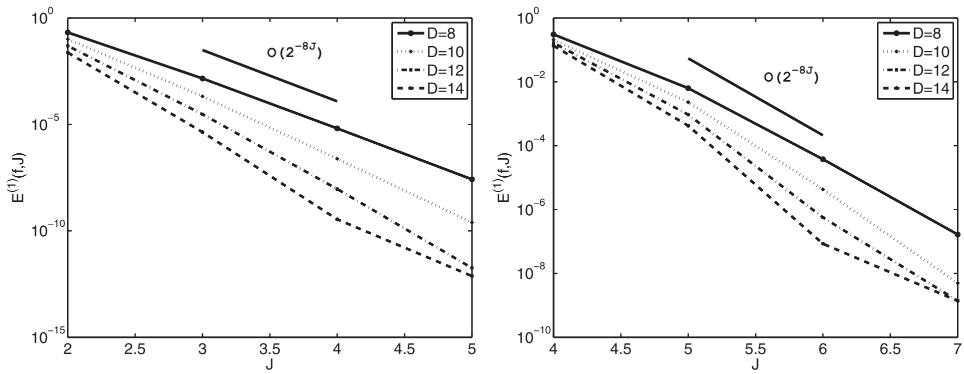

Fig. 2. $E^{(1)}(f, J)$ as a function of scale number J: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

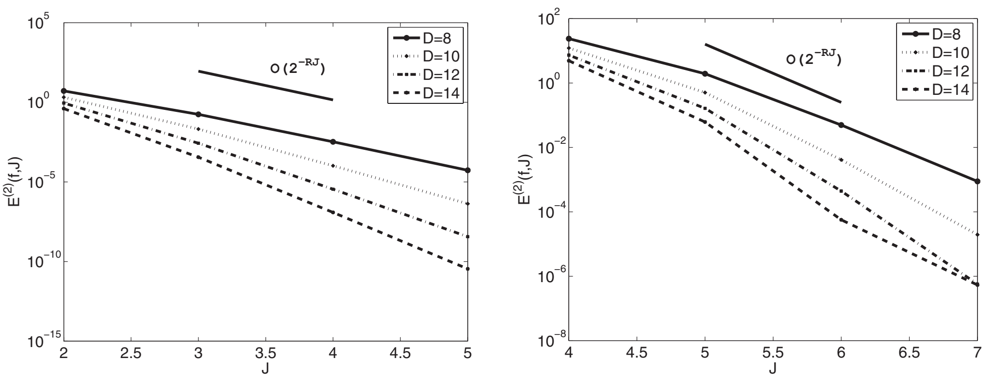

Fig. 3. $E^{(2)}(f, J)$ as a function of scale number J: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

and numerically

$$R = D - 2\lfloor d/2 \rfloor, \quad d = 0, 1, \ldots, D-1.$$

Remark 2.3. The convergence rate $R = D$ for $d = 1$ can also be achieved for higher orders by redefining the differentiation process for $d > 1$ as

$$\bar{f}^{(d)} = (\mathcal{D} * \cdots d \text{ times } \cdots * \mathcal{D})f = \mathcal{D}^d f,$$

that is, the d-order derivative of f is approximated by repeated application of the first order differentiation matrix. Define

$$\bar{E}^{(d)}(f, j) = \max_{k=0,1,\ldots,2^j-1} \left| \left[\bar{f}_n^{(d)} \right]_k - f^{(d)}\left(\frac{k}{2^j}\right) \right|,$$

then

$$\bar{E}^{(d)}(f, j) \leq C 2^{-j\bar{R}}, \quad \bar{R} \in \mathbb{R},$$

with $\bar{R} = D$. See Nielsen [1998] for details.

Figure 2 shows the convergence of first order derivatives with respect to the scale J for $D = 8$ and verifies the relation given in (28). Figure 3 shows the convergence of second order derivatives with respect to the scale J for $D = 8$ and verifies the relation (29) (note that $R = 8 - 2\lfloor 2/2 \rfloor = 6$). Figure 4 shows the convergence of first order derivatives

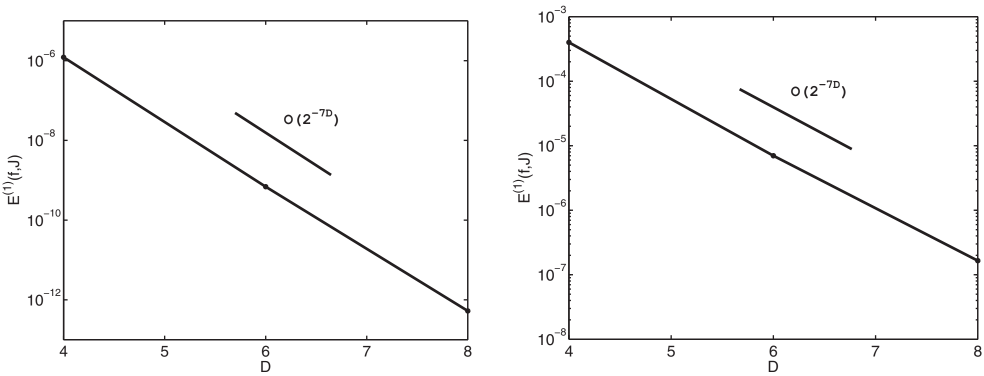

Fig. 4. $E^{(1)}(f, J)$ as a function of wavelet genus D for $J = 7$: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

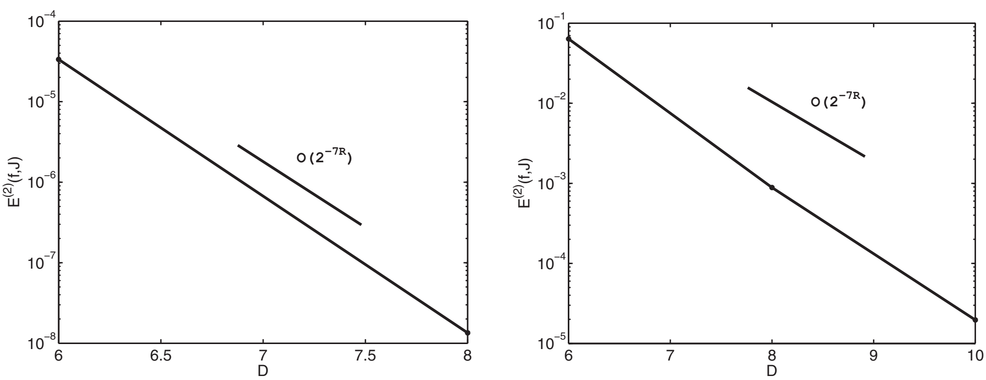

Fig. 5. $E^{(2)}(f, J)$ as a function of wavelet genus D for $J = 7$: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

with respect to the wavelet genus D and verifies the relation in (28). Figure 5 shows the convergence of second order derivatives with respect to D and verifies (29). These figures were generated using `gal_diff_periodic.m`.

2.2.3. Collocation approach. The function $f(x)$ is approximated in the space $\tilde{\mathcal{V}}^j$ by

$$P_{\tilde{\mathcal{V}}^j} f(x) = \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^j(x), \tag{30}$$

where c_k^j are scaling function coefficients. Differentiating d (nonnegative integer) times (30) gives

$$f^{(d)}(x) = 2^{jd} \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^{j(d)}(x). \tag{31}$$

In the collocation method the approximated function will coincide with the actual function at the nodal points in the domain at level j (collocation points), therefore (31) becomes

$$f^{(d)}(l/2^j) = 2^{jd} \sum_{k=0}^{2^j-1} c_k^j \tilde{\phi}_k^{j(d)}(l/2^j), \text{ where } l = 0, \ldots, 2^j - 1. \tag{32}$$

Therefore, calculating the scaling function coefficients, c_k^j, reduces to solving a matrix equation

$$f^{j(d)} = \mathcal{D}^{(d)} c^j, \tag{33}$$

where $f^{j(d)} = (f^{(d)}(0), \ldots, f^{(d)}(\frac{2^j-1}{2^j}))$, $c^j = (c_0^j, \ldots, c^j_{\frac{2^j-1}{2^j}})$ and matrix $\mathcal{D}^{(d)}$ is given by

$$\mathcal{D}^{(d)} = 2^{jd+j/2} \begin{bmatrix} 0 & 0 & \cdots & 0 & \phi_{D-2}^{(d)} & \cdots & \phi_2^{(d)} & \phi_1^{(d)} \\ \phi_1^{(d)} & 0 & \cdots & \vdots & 0 & \vdots & \vdots & \phi_2^{(d)} \\ \phi_2^{(d)} & \phi_1^{(d)} & \cdots & \vdots & \vdots & \vdots & \phi_{D-1}^{(d)} & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & 0 & \phi_{D-2}^{(d)} \\ \phi_{D-2}^{(d)} & \phi_{D-3}^{(d)} & \cdots & \vdots & \vdots & \cdots & \vdots & 0 \\ 0 & \phi_{D-2}^{(d)} & \cdots & 0 & \vdots & \cdots & \vdots & \vdots \\ \vdots & 0 & \cdots & \phi_1^{(d)} & 0 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \phi_1^{(d)} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \phi_{D-3}^{(d)} & \vdots & \cdots & 0 & \vdots \\ 0 & \cdots & \cdots & \phi_{D-2}^{(d)} & \phi_{D-3}^{(d)} & \cdots & \phi_1^{(d)} & 0 \end{bmatrix}.$$

To solve for c_k^j using (33), we need to construct $\mathcal{D}^{(d)}$, which requires the values of $\phi^{(d)}$ at the dyadic rationals.

To calculate the values of the $\phi^{(d)}$ at the dyadic rationals, we differentiate (1) d times to give

$$\phi^{(d)}(x) = 2^d \sqrt{2} \sum_{k=0}^{D-1} h_k \phi^{(d)}(2x - k). \tag{34}$$

Then putting $x = 0, 1, \ldots, D-1$ in (34) we obtain the system

$$2^{-d} \Phi^{(d)}(0) = A_0 \Phi^{(d)}(0), \tag{35}$$

where matrix A_0 is defined in (3). From (35) it is clear that $\Phi^d(0)$ is nothing but the eigenvector of the matrix A_0 corresponding to the eigenvalue 2^{-d} and the normalization condition is obtained as follows:

Differentiate (21) d times to get

$$d! = \sum_{l=-\infty}^{\infty} M_k^d \phi^{(d)}(x - l), \tag{36}$$

and put $x = 0$ in (36) to give

$$d! = \sum_{l=-\infty}^{\infty} M_k^d \phi^{(d)}(-l),$$

or

$$d! = \sum_{l=-\infty}^{\infty} (-1)^d M_k^d \phi^{(d)}(l),$$

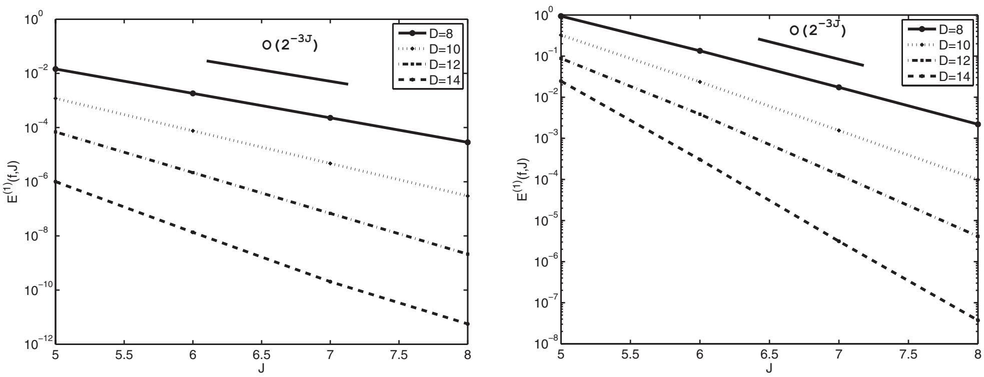

Fig. 6. $E^{(1)}(f, J)$ as a function of scale number J: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

or

$$d! = \sum_{l=0}^{D-1} (-1)^d M_k^d \phi^{(d)}(l).$$

Now substituting $x = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots$ into (34), leads to a matrix equation of the form

$$\Phi^{(d)}\left(\frac{1}{2}\right) = 2^d A_1 \Phi^{(d)}\left(\frac{1}{2}\right),$$

where A_1 is given in (5). Continuing in a similar manner, we may obtain the values of $\Phi^{(d)}$ to a desired resolution as we did with the cascade algorithm described in Section 2.1.1. Differentiating (2) d times gives

$$\psi^{(d)}(x) = 2^d \sqrt{2} \sum_{k=0}^{D-1} g_k \phi^{(d)}(2x - k),$$

which can be used to calculate the values of $\psi^{(d)}(x)$ from the values of $\phi^{(d)}(x)$. The functions `cascade_der.m` and `collo_difmatrix_periodic.m` implement the algorithms (see the accompanied user manual, Section 5, for more details).

Convergence results. The error is defined as

$$E^{(d)}(f, j) = \left\| \left[f_n^{(d)} \right]_k - f^{(d)}\left(\frac{k}{2^j}\right) \right\|_2,$$

where $f_n^{(d)}$ and $f^{(d)}$ denote the numerical and analytic values of the d-order derivative of f. Figure 6 and Figure 7 show the error as a function of the scale number J for $d = 1$ and $d = 2$ respectively. Figure 8 and Figure 9 show the error as a function of the wavelet genus D for $d = 1$ and $d = 2$ respectively. From these graphs it can be inferred that for first and second order derivatives, the method converges like $K^{-\alpha(D)}$, where K is the number of collocation points, that is, $K = 2^J$ and $\alpha(D)$ grows with D. More precisely $\alpha(D) \approx D - 5$ for the first derivative and $\alpha(D) = D - 6$ for the second derivative. Also we can see that convergence is spectral with respect to the wavelet genus D; see Garba [1996] for details. These figures were generated using the function `collo_diff_periodic.m`.

2.3. Nonperiodic Domain

We next consider the case of an interval without the requirement of periodicity of the function f. Again, for simplicity, we assume that the interval is [0, 1]. We explain

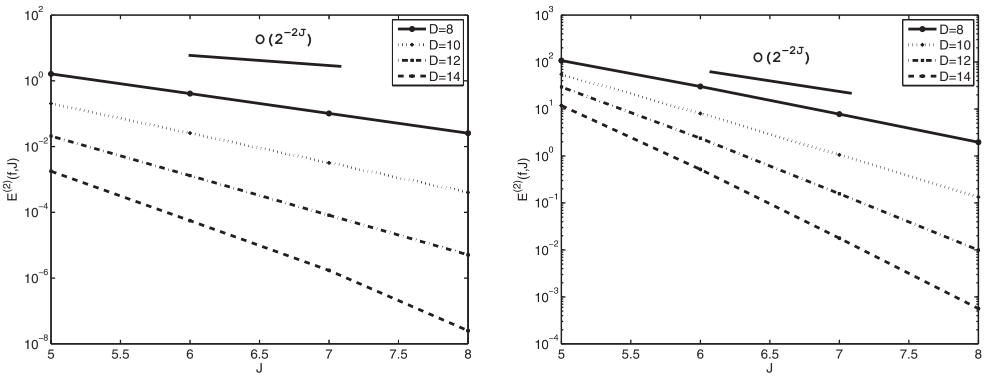

Fig. 7. $E^{(2)}(f, J)$ as a function of scale number J: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

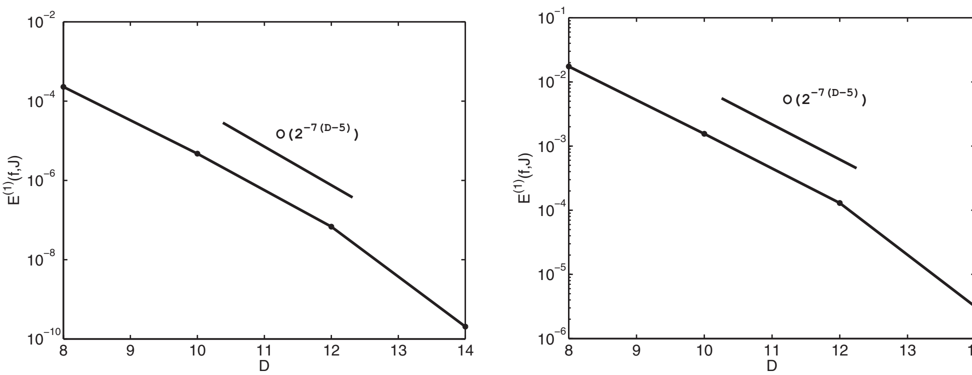

Fig. 8. $E^{(1)}(f, J)$ as a function of wavelet genus D for $J = 7$: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

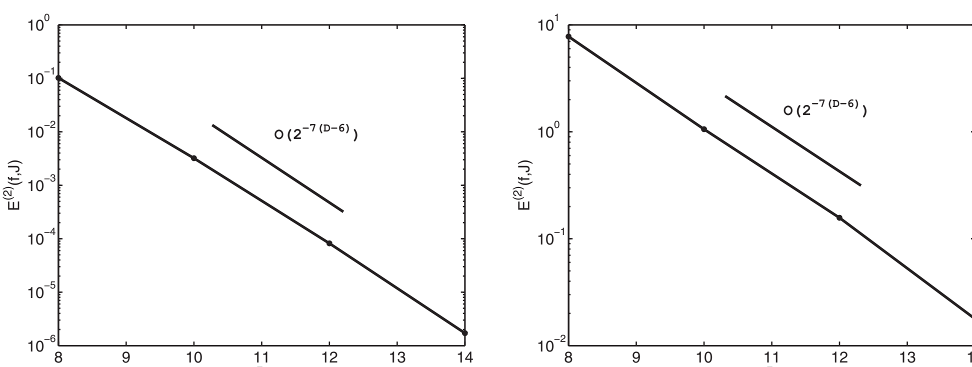

Fig. 9. $E^{(2)}(f, J)$ as a function of wavelet genus D for $J = 7$: (a) $f(x) = 1 + cos(2\pi x)$; (b) $f(x) = e^{-100(x-\frac{1}{2})^2}$.

the construction of the wavelet basis on an interval where the scaling functions and wavelets away from the boundary are the usual Daubechies scaling functions and wavelets. At the boundaries, boundary scaling functions are constructed such that the polynomials of degree up to the number of vanishing moments of the wavelet can be reproduced exactly across the entire interval. The boundary function construction

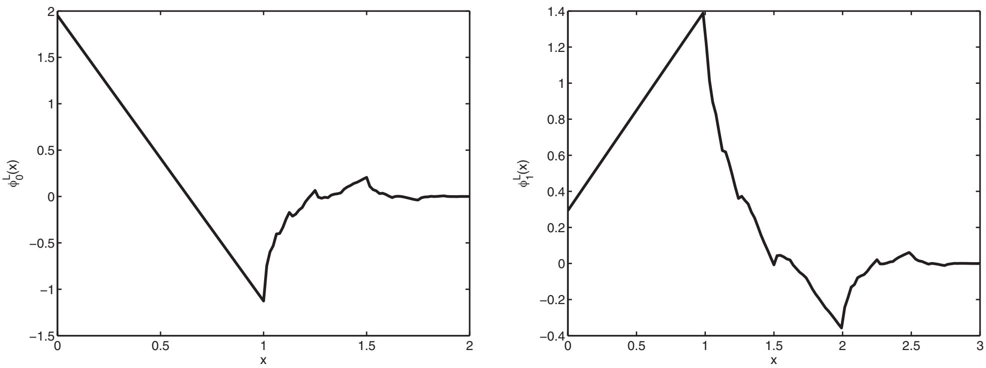

Fig. 10. Left boundary scaling functions $\phi_k^L(x)$ for $D = 4$.

begins by building independent, but not orthogonal, functions

$$\tilde{\phi}^k(x) = \sum_{n=k}^{2M-2} \binom{n}{k} \phi(x + n - M + 1),$$

where $\phi(x)$ is the usual Daubechies scaling functions and $M(= D/2)$ is the number of vanishing moments of the associated wavelet. These functions are compactly supported and their staggered supports are given by

$$supp(\tilde{\phi}^k) = [0, 2M - 1 - k].$$

The staggered support yields independence, and the boundary functions are defined by simply orthonormalizing these functions using the Gram-Schmidt method.

The left and right boundary scaling functions are defined recursively as follows

$$\phi_k^L(x) = \sqrt{2} \sum_{l=0}^{M-1} h_{k,l}^L \phi_l^L(2x) + \sqrt{2} \sum_{m=M}^{M+2k} h_{k,m}^L \phi(2x - m), \tag{37}$$

$$\phi_k^R(x) = \sqrt{2} \sum_{l=0}^{M-1} h_{k,l}^R \phi_l^R(2x) + \sqrt{2} \sum_{m=M}^{M+2k} h_{k,m}^R \phi(2x + m + 1). \tag{38}$$

Note that $\phi_k^L(x) = \phi_k^{L,0}(x)$ and $\phi_k^R(x) = \phi_k^{R,0}(x)$. If we work on the interval [0, 1] and start with a scale fine enough so that the two edges do not interact, that is, $2^j \geq 2M$, then there are the following.

—$2^j - 2M$ total interior scaling functions $\phi_k^j(x), k = M, \ldots, 2^j - M - 1$.
—M total left boundary scaling functions $\phi_k^{L,j}(x), k = 0, 1, \ldots, M - 1$.
—M total right boundary scaling functions $\phi_k^{R,j}(x), k = 2^j - M, \ldots, 2^j - 1$.

Moreover, $h^L = \{h_{k,l}^L, k = M, \ldots, 2^j - M - 1, l = 0, \ldots, M + 2k\}$ and $h^R = \{h_{k,l}^R, k = 2^j - M, \ldots, 2^j - 1, l = 0, \ldots, M + 2k\}$ are the left and right low pass filter coefficients respectively [Cohen et al. 1993]. Routines `L_daubfilt.m` and `R_daubfilt.m` are available to compute the coefficients h^L and h^R respectively. Figures 10 and 11 show the left and right boundary scaling functions for $D = 4$.

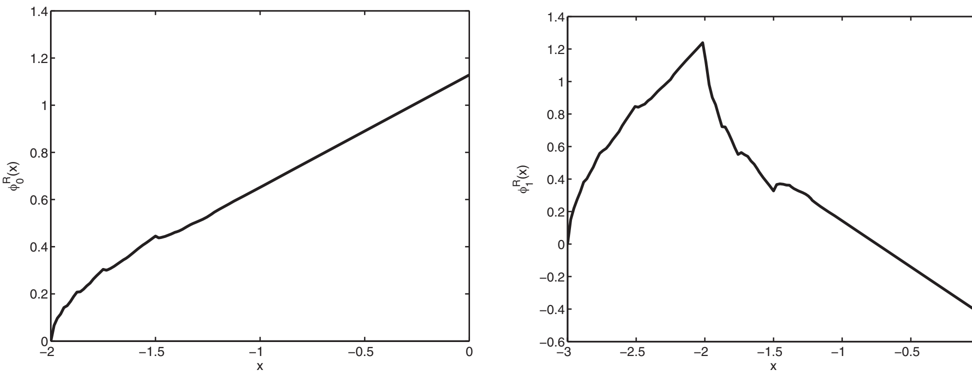

Fig. 11. Right boundary scaling functions $\phi_k^R(x)$ for $D = 4$.

2.3.1. Evaluation of scaling function coefficients. Let $f(x) \in \mathcal{L}^2([0, 1])$, then

$$P_{\mathcal{V}^j} f(x) = \sum_{k=0}^{N-1} c_k^j b_k^j(x), \tag{39}$$

where $\{b_k^j(x), k = 0, \ldots, N-1\}$ denotes the basis functions of the space $\mathcal{V}^j$, $N = 2^j$ and $c_k^j = \langle f, b_k^j \rangle$ (because of the orthonormality of b_k^j) and

$$b_k^j(x) = \begin{cases} \phi_k^{L,j}(x), & supp\left(\phi_k^{L,j}\right) = [0, M+k], & k = 0, \ldots, M-1 \\ \phi_k^j(x), & supp(\phi_k j) = [-M+1, M], & k = M, \ldots, N-M-1 \\ \phi_k^{R,j}(x), & supp\left(\phi_k^{R,j}\right) = [k-M+1, N], & k = N-M, \ldots, 2^j - 1 \end{cases}$$

We will use the technique of orthogonal projection explained in Section 2.2.1 to obtain the c_k^j of (39) [Jameson 1996]

$$c_k^j = \left\langle f, b_k^j \right\rangle \quad k = 0, 1, \ldots, N-1.$$

Using the interval $[0, N]$ and $j = 0$ we have

$$\begin{aligned} c_k^L = c_k^{L,0} &= \int_0^N \phi_k^L(x) f(x) dx, \quad k = 0, 1, \ldots, M-1 \\ c_k = c_k^0 &= \int_0^N \phi(x-k) f(x) dx, \quad k = M, \ldots, N-M-1 \\ c_k^R = c_k^{R,0} &= \int_0^N \phi_k^R(x) f(x) dx, \quad k = N-M, \ldots, N-1. \end{aligned} \tag{40}$$

A quadrature method is required to approximate the integrals in (40). For simplicity assume that $D = 4$ and $N = 8$ and let the grid chosen be

$$\vec{x} = \begin{bmatrix} .5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 \end{bmatrix}^T$$

then

$$c_0^L = \int_0^8 \phi_0^L(x) f(x) dx. \tag{41}$$

To calculate the integral on the right-hand side of (41) we use the quadrature formula

$$\int_0^8 \phi_0^L f(x)dx = s_{1,1} f(0.5) + s_{1,2} f(1.5), \tag{42}$$

so that

$$c_0^L = s_{1,1} f(0.5) + s_{1,2} f(1.5).$$

If $f(x) = 1$, we get from (42)

$$\int_0^8 \phi_0^L(x)dx = s_{1,1}1 + s_{1,2}1 \tag{43}$$

and if $f(x) = x$, (42) implies

$$\int_0^8 \phi_0^L(x)xdx = s_{1,1}0.5 + s_{1,2}1.5. \tag{44}$$

Equations (43) and (44) can be written in matrix form as

$$\begin{bmatrix} \int_0^8 \phi_0^L(x)dx \\ \int_0^8 \phi_0^L(x)xdx \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ .5 & 1.5 \end{bmatrix} \begin{bmatrix} s_{1,1} \\ s_{1,2} \end{bmatrix}. \tag{45}$$

The left-hand side of (45) is just the vector of moments of the left hand side boundary scaling functions. Similarly, while calculating the c_k^R and c_k, we need moments of both the right-hand side boundary scaling functions $\phi_k^R(x)$ and the usual Daubechies scaling function $\phi(x)$. The moments of $\phi(x)$ have already been explained in Section 2.2.2.

Moments of the boundary scaling functions. The p^{th} moment of ϕ_k^L is defined as

$$m_k^{L,p} = \int_0^\infty \phi_k^L(x)x^p dx.$$

We begin by calculating the 0^{th} moment of ϕ_k^L

$$m_k^{L,0} = \int_0^\infty \phi_k^L(x)x^0 dx = \int_0^\infty \phi_k^L(x)dx.$$

Integrating (37) with respect to x from 0 to ∞,

$$\int_0^\infty \phi_k^L(x)dx = \sqrt{2}\sum_{l=0}^{M-1} h_{k,l}^L \int_0^\infty \phi_l^L(2x)dx + \sqrt{2}\sum_{m=M}^{M+2k} h_{k,m}^L \int_0^\infty \phi(2x-m)dx$$

then substituting $y = 2x$ gives, after some manipulation,

$$\sqrt{2}\int_0^\infty \phi_k^L(x)dx = \sum_{l=0}^{M-1} h_{k,l}^L \int_0^\infty \phi_l^L(x)dx + \sum_{m=M}^{M+2k} h_{k,m}^L \int_0^\infty \phi(x-m)dx,$$

that is,

$$\sqrt{2}m_k^{L,0} = \sum_{l=0}^{M-1} h_{k,l}^L m_l^0 + \sum_{m=M}^{M+2k} h_{k,m}^L M_m^0, \quad k = 0, \ldots, M-1. \tag{46}$$

Hence we obtain a system of M equations which can be solved for $m_k^{L,0}$, $k = 0, \ldots, M-1$. For $D = 4$ the system can be written in the form

$$\begin{bmatrix} h_{0,0}^L - \sqrt{2} & h_{0,1}^L \\ h_{1,0}^L & h_{1,1}^L - \sqrt{2} \end{bmatrix} \times \begin{bmatrix} m_0^{L,0} \\ m_1^{L,0} \end{bmatrix} = \begin{bmatrix} -h_{0,2}^L \\ -h_{1,2}^L - h_{1,3}^L - h_{1,4}^L \end{bmatrix}. \tag{47}$$

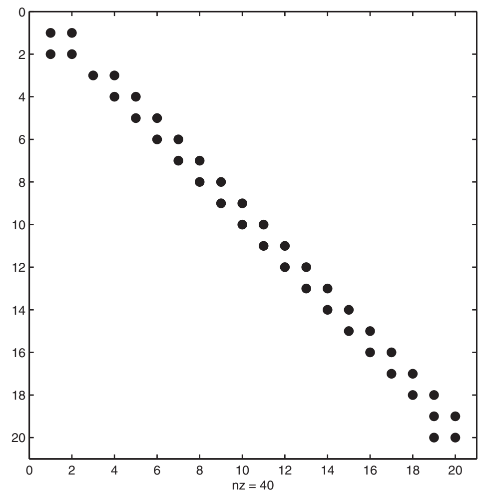

Fig. 12. Quadrature matrix C.

Once the moments are calculated, we can use (42) and (45) to obtain the matrix *quadrature matrix* C such that

$$\mathbf{c} = C\mathbf{f}.$$

The structure of the matrix C is given in Figure 12.

The functions `L_moments.m`, `R_moments.m` and `dstmat_nonper.m` are available to compute the left and right side boundary scaling functions and the quadrature matrix respectively. More details of their implementations and use may be found in the accompanied User Manual, Sections 7 and 8.

2.3.2. Galerkin Approach.

The differentiation projection matrix. Differentiating (39) with respect to x we get

$$\frac{d}{dx} P_{\mathcal{V}^j} f(x) = \sum_{k=0}^{N-1} c_k^j b_k^{(1)j}(x), \tag{48}$$

where $b_k^{(1)j}(x) = \frac{d}{dx} b_k^j(x)$. The derivative takes $P_{\mathcal{V}^j} f(x)$ out of $\mathcal{V}^j$. Projecting back into $\mathcal{V}^j$ we obtain

$$P_{\mathcal{V}^j} \frac{d}{dx} P_{\mathcal{V}^j} f(x) = \sum_{l=0}^{N-1} \left\langle \frac{d}{dx} P_{\mathcal{V}^j} f, b_l^j \right\rangle b_l^j(x),$$

which after using (48) becomes

$$P_{\mathcal{V}^j} \frac{d}{dx} P_{\mathcal{V}^j} f(x) = \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} s_k \left\langle b_k^{(1)j}, b_l^j \right\rangle b_l^j(x).$$

The elements $\langle b_k^{(1)j}, b_l^j \rangle$ comprise the differentiation projection matrix $D^{(1)}$. To find the elements of $D^{(1)}$ we need to find the interaction of the derivative of each basis function with every other basis function. Note that the left- and right-hand side boundary functions do not interact with each other. For simplicity, assume that $j = 0$. We define the following terms for the left-hand side boundary scaling functions

$$\begin{aligned}
\rho^L_{k,p} &= \int_I \phi^{(1)L}_k(x)\phi^L_p(x)dx,\\
\alpha^L_{m,i} &= \int_I \phi^{(1)}(x-m)\phi^L_i(x)dx,\\
\beta^L_{l,q} &= \int_I \phi^{(1)L}_l(x)\phi(x-q)dx,\\
r_{m,q} &= r_{q-m} = \int_I \phi^{(1)}(x-m)\phi(x-q)dx,
\end{aligned} \tag{49}$$

with similar definitions for the right-hand side. With this notation, the matrix $D^{(1)}$ for $D=4$ and $j=3$ will take the form

$$D^{(1)} = \begin{bmatrix}
\rho^L_{0,0} & \rho^L_{1,0} & \alpha^L_{2,0} & 0 & 0 & 0 & 0 & 0\\
\rho^L_{0,1} & \rho^L_{1,1} & \alpha^L_{2,1} & \alpha^L_{3,1} & 0 & 0 & 0 & 0\\
\beta^L_{0,2} & \beta^L_{1,2} & r_0 & r_1 & r_2 & 0 & 0 & 0\\
0 & \beta^L_{1,3} & r_{-1} & r_0 & r_1 & r_2 & 0 & 0\\
0 & 0 & r_{-2} & r_{-1} & r_0 & r_1 & \beta^R_{1,4} & 0\\
0 & 0 & 0 & r_{-2} & r_{-1} & r_0 & \beta^R_{1,5} & \beta^R_{0,5}\\
0 & 0 & 0 & 0 & \alpha^R_{4,1} & \alpha^R_{5,1} & \rho^R_{1,1} & \rho^R_{0,1}\\
0 & 0 & 0 & 0 & 0 & \alpha^R_{5,0} & \rho^R_{1,0} & \rho^R_{0,0}
\end{bmatrix}.$$

Next we differentiate (37) with respect to x to get

$$\frac{d}{dx}\phi^L_k(x) = 2\sqrt{2}\sum_{l=0}^{M-1} h^L_{k,l}\phi^{(1)L}_l(2x) + 2\sqrt{2}\sum_{m=M}^{M+2k} h^L_{k,m}\phi^{(1)}(2x-m)$$

then multiply by $\phi^L_p(x)$ and integrate to get

$$\begin{aligned}
\int_I \phi^{(1)L}_k(x)\phi^L_p(x)dx = {}& 4\sum_{l=0}^{M-1}\sum_{i=0}^{M-1} h^L_{k,l}h^L_{p,i}\int_I \phi^{(1)L}_l(2x)\phi^L_i(2x)dx\\
&+ 4\sum_{m=M}^{M+2k}\sum_{i=0}^{M-1} h^L_{p,i}h^L_{k,m}\int_I \phi^L_i(2x)\phi^{(1)}(2x-m)dx\\
&+ 4\sum_{l=0}^{M-1}\sum_{q=M}^{M+2p} h^L_{k,l}h^L_{p,q}\int_I \phi^{(1)L}_l(2x)\phi(2x-q)dx\\
&+ 4\sum_{m=M}^{M+2k}\sum_{q=M}^{M+2p} h^L_{k,m}h^L_{p,q}\int_I \phi^{(1)}(2x-m)\phi(2x-q)dx.
\end{aligned} \tag{50}$$

Now using the symbols defined in (49), we may write (50) as

$$\begin{aligned}
\frac{1}{2}\rho^L_{k,p} = {}& \sum_{l=0}^{M-1}\sum_{i=0}^{M-1} h^L_{k,l}h^L_{p,i}\rho^L_{l,i} + \sum_{m=M}^{M+2k}\sum_{i=0}^{M-1} h^L_{p,i}h^L_{k,m}\alpha^L_{m,i}\\
&+ \sum_{l=0}^{M-1}\sum_{q=M}^{M+2p} h^L_{k,l}h^L_{p,q}\beta^L_{l,q} + \sum_{m=M}^{M+2k}\sum_{q=N}^{M+2p} h^L_{k,m}h^L_{p,q}r_{m,q}.
\end{aligned} \tag{51}$$

A similar relation for $\rho^R_{k,p}$ may be obtained using (38). Note that the coefficient $r_{m,q}$ is just the connection coefficient defined in Section 2.2.2 and the Matlab function `conn.m` will calculate these coefficients. Also, it is easy to observe that

$$\beta^L_{i,m} = -\alpha^L_{m,i} \text{ and } \beta^R_{i,m} = -\alpha^R_{m,i},$$

and hence we only need to calculate $\alpha^L_{m,i}$ and $\alpha^R_{m,i}$. The algorithm for the calculation of $\alpha^L_{m,i}$ is as follows; we have

$$\alpha^L_{m,i} = \int_I \phi^{(1)}(x-m)\phi^L_i(x)dx, \quad i = 0,\ldots,M-1, \tag{52}$$

and from (1)

$$\phi^{(1)}(x-m) = 2\sqrt{2}\sum_{k=-M+1}^{M} h_k\phi^{(1)}(2x-(2m+k)). \tag{53}$$

Now using (37) and (53) in (52) and after some manipulation we obtain

$$\begin{aligned}\alpha^L_{m,i} &= 2\sum_{k=-M+1}^{M}\sum_{l=0}^{M-1} h_k h^L_{i,l}\alpha^L_{2m+k,l} \\ &\quad + 2\sum_{k=-M+1}^{M}\sum_{p=M}^{M+2k} h_K h^L_{i,p} r_{2m+k,p}\end{aligned} \tag{54}$$

Calculation of $\{\rho^L_{k,p}\}$: First consider the case with $k = p$, that is, $\{\rho^L_{k,p} | k = p\}$, where

$$\rho^L_{k,k} = \int_0^\infty \phi^{(1)L}_k(x)\phi^L_k(x)dx.$$

Applying integration by parts, we obtain

$$\rho^L_{k,k} = \left(\phi^L_k(x)\right)^2 |_0^\infty - \int_0^\infty \phi^L_k(x)\phi^{(1)L}_k(x)dx.$$

or

$$2\rho^L_{k,k} = \left(\phi^L_k(x)\right)^2 |_0^\infty$$

Since $\phi^L_k(x) = 0$ at $x = \infty$, we have

$$\rho^L_{k,k} = -\frac{\left(\phi^L_k(0)\right)^2}{2}. \tag{55}$$

We can now use (37) to calculate $\phi^L_k(0)$. A number of Matlab functions are available to perform these computations, see the accompanied User manual, Sections 9, 10, 11.

3. SPLINE-BASED WAVELET

Since the Daubechies scaling and wavelet functions do not have explicit forms, they cannot be used efficiently in some situations. In the case of biorthogonal spline wavelets, the basic scaling and wavelet functions are splines, and hence have explicit expressions. To the best of our knowledge, B-splines are the only known scaling functions with an explicit representation. The only price we have to pay is loss of orthogonality.

The $\beta^0(x)$ (B-spline of order 0) is the characteristic function in the interval $[-\frac{1}{2}, \frac{1}{2})$. All B-splines of higher order ($\beta^n(x)$) are generated from $\beta^0(x)$ using the recurrence relation

$\beta^n(x) = \beta^0(x) * \beta^{n-1}(x)$ (where '*' denotes the convolution operator); these have a number of attractive properties. The most important property is their compact support which allows them to be used as scaling function. In fact, the support of these functions is minimal among all polynomial splines of order n.

B-splines are symmetric, bell-shaped functions and have compact support given by $[-\frac{n+1}{2}, \frac{n+1}{2}]$, [Unser and Aldroubi 1992]. An explicit formula for $\beta^n(x)$ is:

$$\beta^n(x) = \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \binom{n+1}{j} \left[x + \frac{n+1}{2} - j \right]_+^n, \tag{56}$$

where $[x]_+^n = max\{0, x\}^n$ is the one-sided power function of degree n.

For each positive integer n, let S^n be the space of polynomial splines of order n with knot sequence $\mathbb{Z}$. A fundamental theorem states that every polynomial spline function ($s^n(x)$) of a given degree can be uniquely represented as a linear combination of shifted B-splines of the same degree (the term B-spline is short for basis-spline), that is,

$$s^n(x) = \sum_{k=-\infty}^{\infty} p_k \beta^n(x-k). \tag{57}$$

The set $\{\phi_k^0(x) = \phi(x-k) = s^n(x-k) : k \in \mathbb{Z}\}$ is a basis for $\mathcal{V}^0$ provided the sequence $\{p_k\}$ is an invertible convolution operator from l^2 to itself (note that there are several ways to select the scaling function when dealing with spline wavelets.). Now, $\{\phi_k^j(x) = 2^{j/2}\phi(2^j x - k), k \in \mathbb{Z}\}$ is a basis for $\mathcal{V}^j$ when n is odd, hence n is assumed to be odd. The dilation relation

$$\phi(x) = \sum_{k=-\infty}^{\infty} h_k \phi(2x-k), \tag{58}$$

is satisfied by the $\phi(x)$ and the wavelet function $\psi(x)$ is defined such that

$$\psi(x) = \sum_{k=-\infty}^{\infty} g_k \phi(2x-k).$$

The dual scaling function $\tilde{\phi}(x)$ and dual wavelet function $\tilde{\psi}(x)$ are not splines and no closed form formula exists for them [Urban 2009], but these functions satisfy the dilation relation $\tilde{\phi}(x) = \sum_{k=-\infty}^{\infty} \tilde{h}_k \tilde{\phi}(2x-k)$ and the wavelet relation $\tilde{\psi}(x) = \sum_{k=-\infty}^{\infty} \tilde{g}_k \tilde{\phi}(2x-k)$; hence the cascade algorithm may be used to calculate the values of $\tilde{\phi}(x)$ and $\tilde{\psi}(x)$.

For linear splines ($n = 1$) and $p_0 = 1; p_k = 0$ for $k \neq 0$, (57) gives us $\phi(x) = \beta^n(x)$. Figure 13 shows the functions $\phi(x)$ and $\psi(x)$ while $\tilde{\phi}(x)$ and $\tilde{\psi}(x)$ are shown in Figure 14.

The two sequences $\{\mathcal{V}^j : j \in \mathbb{Z}\}$ and $\{\tilde{\mathcal{V}}^j : j \in \mathbb{Z}\}$ satisfy the first three axioms of MRA of Section 2.1 but not the fourth. The following relation does hold:

$$\mathcal{V}^{j+1} = \mathcal{V}^j + \mathcal{W}^j, \quad \tilde{\mathcal{V}}^{j+1} = \tilde{\mathcal{V}}^j + \tilde{\mathcal{W}}^j.$$

The biorthogonality implies that $\mathcal{W}^j$ is not necessarily orthogonal to $\mathcal{V}^j$ but it is to $\tilde{\mathcal{V}}^j$, whereas $\tilde{\mathcal{W}}^j$ is not necessarily orthogonal to $\tilde{\mathcal{V}}^j$ but is to $\mathcal{V}^j$.

For any $f(x) \in \mathcal{L}^2(\mathbb{R})$ we have

$$P_{\mathcal{V}^0} f(x) = \sum_k \langle f, \tilde{\phi}_k \rangle \phi_k(x) = \sum_k c_k^0 \phi_k(x), \tag{59}$$

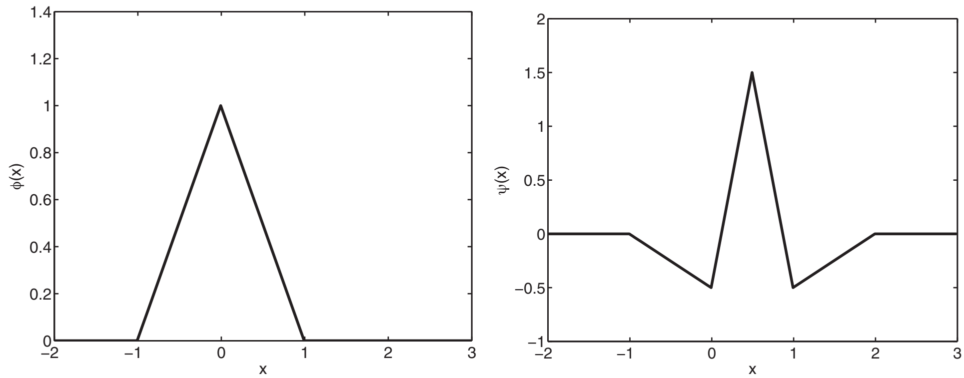

Fig. 13. (a) Linear spline scaling function $\phi(x)$. (b) Linear spline wavelet function $\psi(x)$.

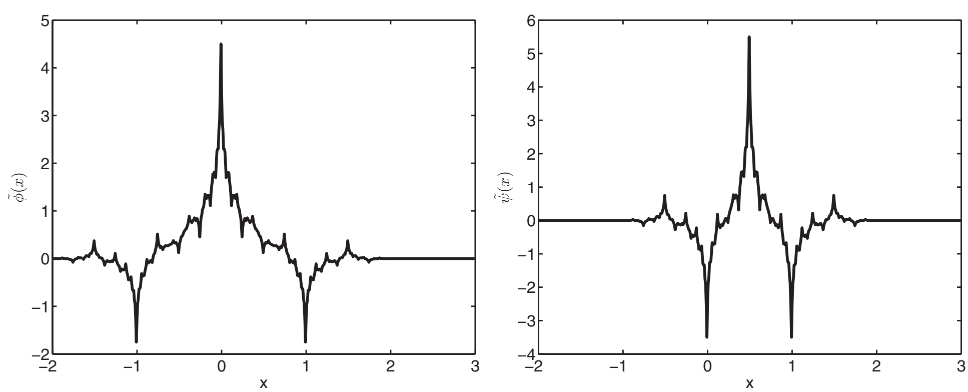

Fig. 14. (a) Linear spline dual scaling function $\tilde{\phi}(x)$. (b) Linear spline dual wavelet function $\tilde{\psi}(x)$.

Using (57) in (59) we obtain

$$P_{\mathcal{V}^0} f(x) = \sum_k c_k^0 \sum_m p_{k,m} \beta^n(x-m) = \sum_k \sum_m c_k^0 p_{k,m} \beta_m^n(x).$$

Using the interpolation technique explained in Section 2.2.1 with the set of integers as node points we have

$$f(l) = \sum_k \sum_m c_k^0 p_{k,m} \beta_m^n(l), \quad l \in \mathbb{Z},$$

or

$$f(l) = \sum_k \left[\sum_m p_{k,m} \beta_m^n(l) \right] c_k^0, \quad l \in \mathbb{Z},$$

which can be written as

$$\mathbf{f} = (Pb)^T \mathbf{c} = T\mathbf{c}.$$

Hence the required *quadrature matrix* is

$$C = T^{-1} = ((Pb)^T)^{-1}. \tag{60}$$

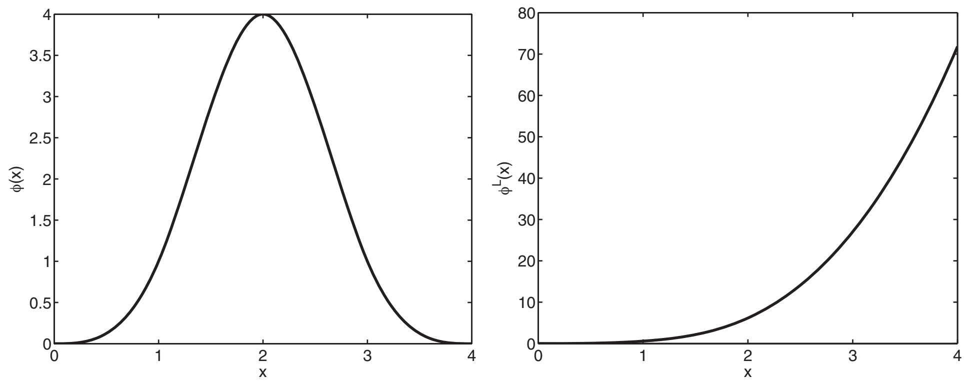

Fig. 15. (a) Scaling function $\phi(x)$. (b) Boundary scaling function $\phi^L(x)$.

Let $A_{l,m} = \langle \beta^{(1)n}(m), \beta^n(l)\rangle$, $B_{m,l} = \langle \beta^n(m), \beta^n(l)\rangle$ and $b(l,m) = \beta^n(l-m)$, then the differentiation matrix $\mathcal{D}^{(1)}$ [Jameson 1995] is given by

$$\mathcal{D}^{(1)} = b^T B^{-1} A (b^T)^{-1}. \tag{61}$$

For the periodic domain, the three matrices b, A and B are circulant and hence commute yielding $\mathcal{D}^{(1)} = B^{-1}A$.

For a nonperiodic domain, if the boundary basis functions are constructed using truncated B-splines, then the matrix b of (61) is an ill-conditioned matrix for $n > 1$. Hence the inversion of matrix b will introduce a huge error. An alternative way to deal with the nonperiodic domain is explained as follows.

Suppose the interval is $I = [0, N]$. We can translate the B-spline, $\beta^n(x)$, given by (56) so that its support becomes $[0, n+1]$. In this case $\beta^n(x)$ is given by

$$\beta^n(x) = \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \begin{pmatrix} n+1 \\ j \end{pmatrix} [x-j]_+^n .$$

To construct an MRA for the Sobolov space $\mathcal{H}_0^2(I)$, two scaling functions are considered, an interior scaling function $\phi(x)$ and left hand side boundary scaling function $\phi^L(x)$ given as follows

$$\phi(x) = \beta^3(x) = \frac{1}{6}\sum_{j=0}^{4}(-1)^j \begin{pmatrix} 4 \\ j \end{pmatrix} [x-j]_+^3 ,$$

and

$$\phi^L(x) = \phi_0^L(x) = \frac{3}{2}x_+^2 - \frac{11}{12}x_+^3 + \frac{3}{2}(x-1)_+^3 - \frac{3}{4}(x-2)_+^3.$$

Figure 15 shows $\phi(x)$ and $\phi^L(x)$.

Let $\mathcal{V}^j = span\{\phi^{L,j}(x); \phi_k^j(x) : 1 \leq 2^j \leq N-3; \phi^{R,j}(x) = \phi^{L,j}(N-x)\}$, then $\{\mathcal{V}^j\}_{j\in\mathbb{Z}^+}$ forms an MRA of $\mathcal{H}_0^2(I)$. If $\{x_i : i = 0, \ldots, N\}$ is the grid used and $h_i = x_i - x_{i-1}$, then the second-order *differentiation matrix* is given by

$$\mathcal{D}^{(2)} = T_1^{(-1)}(T_2 + \Gamma),$$

where

$$T_1 = \begin{pmatrix} \frac{h_1}{3} & \frac{h_1}{6} & 0 & & & & \\ \frac{h_1}{h_1+h_2} & 2 & \frac{h_2}{h_1+h_2} & & & & \\ \ddots & \ddots & \ddots & & & & \\ & & \frac{h_i}{h_i+h_{i+1}} & 2 & \frac{h_{i+1}}{h_i+h_{i+1}} & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \frac{h_{N-1}}{h_{N-1}+h_N} & 2 & \frac{h_N}{h_{N-1}+h_N} \\ & & & & 0 & \frac{h_N}{6} & \frac{h_N}{3} \end{pmatrix}.$$

and

$$T_2 = \begin{pmatrix} -\frac{1}{h_1} & \frac{1}{h_1} & & & \\ \frac{6}{h_1(h_1+h_2)} & -\frac{6}{h_1+h_2}\left(\frac{1}{h_1}+\frac{1}{h_2}\right) & \frac{6}{h_2(h_1+h_2)} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{6}{h_{N-1}(h_{N-1}+h_N)} & -\frac{6}{h_{N-1}+h_N}\left(\frac{1}{h_{N-1}}+\frac{1}{h_N}\right) & \frac{6}{h_N(h_{N-1}+h_N)} \\ & & & \frac{1}{h_N} & -\frac{1}{h_N} \end{pmatrix}.$$

Γ is defined such that

$$\begin{pmatrix} f'(0) \\ 0 \\ \vdots \\ 0 \\ f'(N) \end{pmatrix} = \Gamma \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \\ f(N) \end{pmatrix}.$$

The first order differentiation matrix is given by

$$\mathcal{D}^{(1)} = H_1 \mathcal{D}^{(2)} + H_2,$$

where

$$\begin{aligned} (H_1)_{11} &= 2(H_1)_{12} = -\frac{h_1}{3} \\ (H_1)_{ii} &= 2(H_1)_{ii-1} = \frac{h_{i-1}}{3} \quad i = 2, \ldots, N \\ (H_1)_{N+1N+1} &= 2(H_1)_{N+1N} = \frac{h_N}{3}, \end{aligned}$$

and

$$\begin{aligned} (H_2)_{11} &= -(H_2)_{12} = -h_1^{-1} \\ (H_2)_{ii} &= -(H_2)_{ii-1} = h_{i-1}^{-1} \quad i = 2, \ldots, N \\ (H_2)_{N+1N+1} &= -(H_2)_{N+1N} = -h_N^{-1}. \end{aligned}$$

See Cai and Wang [1996] and Kumar and Mehra [2007] for details. Note that the differentiation matrix obtained above is the same as the differentiation matrix obtained by the collocation method using splines [Gerald and Wheatley 2004].

ELECTRONIC APPENDIX

The electronic appendix to this article is available in the ACM Digital Library.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments. The authors would also like to thank Professor Leland Jameson for his helpful discussion on spline wavelet.

REFERENCES

CAI, W. AND WANG, J. 1996. Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear PDEs. *SIAM J. Numer. Anal. 33*, 937–970.

CANUTO, C., HUSSAINI, M. Y., QUARTERONI, A., AND ZANG, T. A. 1988. *Spectral Methods in Fluid Dynamics*. Springer series in computational physics. Springer.

COHEN, A., DAUBECHIES, I., AND VIAL, P. 1993. Wavelets on the interval and fast wavelet transform. *Appl. Comput. Harmonic Anal. 1*, 54–81.

DAHMEN, W. 1997. Wavelet and multiscale methods for operator equations. *Acta Numerica 6*, 55–228.

DAHMEN, W., URBAN, K., AND VORLOEPER, J. 2002. Adaptive wavelet methods: basic concepts and applications to the Stokes problem. In *Wavelet Analysis: Twenty Years' Developments*, D.-X. Zhou, Ed. Series in Analysis, vol. 1. World Scientific Publishing, Singapore, 39–80.

DAUBECHIES, I. 1992. *Ten Lectures on Wavelets*. Number 61 in CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, PA.

GARBA, A. 1996. A wavelet collocation method for numerical solution of Burgers' equation. Tech. rep., International Center for Theorytical Physics.

GERALD, C. AND WHEATLEY, P. 2004. *Applied Numerical Analysis*, 7th ed. Pearson Education, Inc.

JAMESON, L. 1993. On the Daubechies-based wavelet differentiation matrix. *J. Sci. Comput. 8*, 3, 267–305.

JAMESON, L. 1995. On the spline based wavelet differentiation matrix. *Appl. Numer. Math. 17*, 33–45.

JAMESON, L. 1996. The differentiation matrix for Daubechies based wavelets on an interval. *SIAM J. Sci. Comput. 17*, 498–516.

KUMAR, V. AND MEHRA, M. 2007. Cubic spline adaptive wavelet scheme to solve singularly perturbed reaction diffusion problems. *Int. J. Wavelets Multiresolution Inf. Process. 5*, 317–331.

MALLAT, S. G. 1989. Multiresolution approximations and wavelet orthonormal bases of L2(R). *Trans. Amer. Math. Soc. 315*, 69–87.

MEHRA, M. AND KEVLAHAN, N. K.-R. 2008. An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *J. Comput. Phys. 227*, 5610–5632.

MEHRA, M. AND KUMAR, B. V. R. 2005. Time-accurate solution of advection-diffusion problems by wavelet-Taylor-Galerkin method. *Commun. Numer. Methods Eng. 21*, 313–326.

NIELSEN, O. M. 1998. Wavelets in scientific computing. Ph.D. thesis, Technical University of Denmark, Lyngby.

SCHNEIDER, K. AND FRÖHLICH, J. 1997. An adaptive wavelet-vaguelette algorithm for the solution of PDEs. *J. Comput. Phys. 130*, 90–174.

STRANG, G. AND NGUYEN, T. 1996. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA.

UNSER, M. AND ALDROUBI, A. 1992. Polynomial splines and wavelets: A signal processing perspective. In *Wavelets: A Tutorial in Theory and Applications*, C. K. Chui, Ed. Academic Press Professional, Inc., San Diego, CA, 91–122.

URBAN, K. 2009. *Wavelet Methods for Elliptic Partial Differential Equations*. Oxford University Press.

VASILYEV, O. AND KEVLAHAN, N.-R. 2005. An adaptive multilevel wavelet collocation method for elliptic problems. *J. Comput. Phys. 206*, 412–431.

VASILYEV, O. V. AND BOWMAN, C. 2000. Second generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys. 165*, 660–693.

Received October 2011; revised June 2012; accepted October 2012