APPLIED AND COMPUTATIONAL HARMONIC ANALYSIS 2, 363–381 (1995)

On the Fast Fourier Transform of Functions with Singularities

G. BEYLKIN[1]

Program in Applied Mathematics, University of Colorado at Boulder, Boulder, Colorado 80309-0526

Communicated by Leslie F. Greengard

Received July 21, 1994; revised February 22, 1995

We consider a simple approach for the fast evaluation of the Fourier transform of functions with singularities based on projecting such functions on a subspace of Multiresolution Analysis. We obtain an explicit approximation of the Fourier Transform of generalized functions and develop a fast algorithm based on its evaluation. In particular, we construct an algorithm for the Unequally Spaced Fast Fourier Transform and test its performance in one and two dimensions. The number of operations required by algorithms of this paper is $O(N \cdot \log N + N_p \cdot (-\log \epsilon))$ in one dimension and $O(N^2 \cdot \log N + N_p \cdot (-\log \epsilon)^2)$ in two dimensions, where ϵ is the precision of computation, N is the number of computed frequencies and N_p is the number of nodes. We also address the problem of using approximations of generalized functions for solving partial differential equation with singular coefficients or source terms.

I. INTRODUCTION

In this paper we propose a simple approach for fast evaluation of the Fourier transform of functions with singularities based on projecting such functions on a subspace of Multiresolution Analysis (MRA). We obtain an explicit approximation of the Fourier Transform of generalized functions and develop a fast algorithm based on its evaluation. In particular, we construct an algorithm for the Unequally Spaced Fast Fourier Transform and test its performance in one and two dimensions. We also begin to address the problem of using such approximations for solving partial differential equation with singular coefficients or source terms.

The Fast Fourier Transform (FFT) algorithm [7] requires sampling on an equally spaced grid which proves to be a significant limitation in many applications. It is clear that the direct evaluation of trigonometric sums

$$\hat{g}_n = \sum_{l=0}^{N_p-1} g_l e^{-2\pi i x_l \xi_n}, \quad n = 0, \pm 1, \ldots, N, \qquad (1.1)$$

where $g_l \in \mathbb{C}, \xi_n \in [-N, N]$ and $x_l \in [0, 1]$, is costly and requires $O(N \cdot N_p)$ operations. Algorithms for the fast evaluation of (1.1) constitute Unequally Spaced Fast Fourier Transform (USFFT) algorithms which are far reaching generalizations of the FFT and have numerous applications in numerical analysis and signal processing. In [11] Press and Rybicki suggest using Lagrange interpolation to replace the function values at an arbitrary point by several function values on an equally spaced grid and provide an algorithm in the form of a FORTRAN subroutine. Essentially the same idea was proposed by Brandt in [5]. For computing (1.1) one may also use the Taylor expansion to correct for deviations from an equally spaced grid [17]. Although such approaches are significantly better than the direct evaluation of (1.1), they do not lead to very efficient algorithms especially in multidimensional generalizations. The speed of this type of algorithm depends on the number of points of an equally spaced grid which are used to replace the function value and on the oversampling factor. In the papers mentioned above the dependence of accuracy and speed of the algorithms on the choice of an interpolation scheme and on the oversampling factor has not been investigated.

A more careful analysis and a much faster algorithm to evaluate (1.1) has recently appeared in the paper by Dutt and Rokhlin [9] where a specialized interpolation scheme using Gaussian bells has been developed and implemented. Also, in a recent paper Sorets [15] describes an algorithm for fast and accurate computation of the Fourier transform of functions with jump discontinuities. One of the motivations for considering the problem of fast and accurate evaluation of integrals of functions with jump discontinuities is VLSI design. Namely, the goal is to compute the integrals

$$\hat{f}(n, n') = \int_0^1 \int_0^1 f(x, y) e^{-2\pi i n x} e^{-2\pi i n' y} \, dx dy, \qquad (1.2)$$

with a given accuracy ϵ for $-N \leq n \leq N$ and $-N' \leq N'$, where f is a piecewise constant function or, more generally, a piecewise smooth function. If the number of discontinuities of f is large, then the direct computation of (1.2) is

[1] This research was partially supported by ARPA Grant F49620-93-1-0474 and ONR Grant N00014-91-J4037.

1063-5203/95 $12.00

costly. If N_D is the number of subdomains of $[0,1]\times[0,1]$ where the function f is smooth, then the number of operations in the direct algorithm may be estimated as $N_D \cdot N \cdot N'$.

Our approach addresses the problems of evaluating both (1.1) and (1.2). By introducing the generalized function

$$g(x) = \sum_{l=0}^{N_p-1} g_l \delta(x - x_l), \tag{1.3}$$

evaluating (1.1) may be viewed as evaluating

$$\hat{g}(\xi) = \int g(x) e^{-2\pi i x \xi}\, dx, \tag{1.4}$$

for $|\xi_n| \leq N, n = 0, \pm 1, \ldots, \pm N$ with a given accuracy ϵ. We also develop and test a fast algorithm to compute the Fourier transform of the generalized function

$$g(x,y) = \sum_{l=1}^{N_p} g_l \delta(x - x_l)\delta(y - y_l), \tag{1.5}$$

where $(x_l, y_l) \in [0,1]\times[0,1]$ and $\{g_l\}_{l=1}^{N_p}$ is a set of N_p complex numbers. This algorithm is equivalent to the two-dimensional USFFT for computing the trigonometric sum

$$\hat{g}_{n,n'} = \sum_{l=1}^{N_p} g_l e^{-2\pi i n x_l} e^{-2\pi i n' y_l}, \tag{1.6}$$

for $-N \leq n \leq N$ and $-N' \leq n' \leq N'$.

We address the problem of computing (1.2), (1.4) or the Fourier transform of (1.5) by projecting the function on a subspace of a Multiresolution Analysis,[2] effectively replacing it by its bandlimited version. Using the projection, we obtain an explicit approximation of the Fourier transform and develop a simple practical algorithm based on an implementation of this approximation. Our implementation consists of three steps, two of which are similar to the corresponding steps in algorithms based on interpolation. The first step of our algorithm which we view as a projection replaces an interpolation scheme (Lagrange interpolation in [11], Taylor expansion in [17], a specialized interpolation scheme involving Gaussian bells in [9] and the steps in [15] involving the use of the Gauss-Legendre quadratures and Lagrange interpolation). The second step is the same as in all algorithms of this type and involves the FFT of a $(\nu N)\times(\nu N')$ matrix, where ν is an oversampling factor. The third step is a modification (or correction) step which involves multiplying values at each frequency by a precomputed factor. At this step we effectively generate a representation involving the Battle–Lemarié scaling function [10, 1].

[2] The fact that such an approach may be used has also been recognized by Coifman [13].

We develop fast practical algorithms for problems (1.1) and (1.2) which compare well with those in [9] and [15] (see Section VIII). In [9] the authors develop two approaches for computing (1.1) and (1.2): algorithms based on Fast Multipole Method (FMM) and based on an interpolation scheme involving Gaussian bells. The approach of this paper is related to the second kind whereas a wavelet-based counterpart of FMM is currently being developed [4].

In fact, we address a more general problem of computing the Fourier transform of generalized functions with singularities of the type $x_+^\lambda/\Gamma(\lambda+1)$, where Γ is the gamma function and x_+^λ is defined as x^λ for $x > 0$ and zero for $x \leq 0$. For example, $\lambda = 0$ yields the jump discontinuity, whereas $\lambda = -1$ corresponds to the δ-function. We provide tight estimates showing that for any ϵ we may choose the space of splines of appropriate order and scale so that the projection on that subspace contains sufficient information to account for half of the frequencies (the oversampling factor $\nu = 2$) with accuracy ϵ. Such projections of generalized functions may be used for solving partial differential equations with singular coefficients or source terms. Approximations of generalized functions (the so-called discrete approximation to singular functions) appear for example in the context of the Immersed Interface method (see e.g. [8, 12]). We note that our estimates provide an additional argument in favor of using wavelets from the "spline family" in problems of solving partial differential equations and signal processing. Although the analysis of splines is a well-established subject (see e.g. [14]), several families of bases were constructed only recently. The "spline family" of wavelets includes those constructed by Stromberg [16], Battle and Lemarié [10, 1], as well as nonorthogonal families (see [6] and [18]).

We start the paper with preliminary considerations in Section II where we describe a general approach to the problem. We then explain the choice of basis in Section III and describe the one dimensional USFFT algorithm in Section IV. In Section V we consider algorithms in the multidimensional setting. In Section VI we develop an interpolation scheme to evaluate the Fourier transform at unequally spaced points. Then, in Section VII, we discuss using the approximation of functions with singularities for solving partial differential equations. Numerical examples and performance evaluation of the algorithms are presented in Section VIII. Finally, conclusions and generalizations are discussed in Section IX where we also briefly discuss applications of these algorithms in numerical analysis and signal processing.

II. PRELIMINARY CONSIDERATIONS

As an example let us consider the problem of computing

$$\hat{f}(m) = \int_0^1 f(x) e^{-2\pi i m x}\, dx \tag{2.1}$$

for $-M \leq m \leq M$, where f is a complex-valued piecewise smooth function or a generalized function. In what follows we assume (without loss of generality) that f is zero in a narrow strip near the boundary of [0, 1] and we extend f by zero outside [0, 1]. Alternatively (see Appendix D for the details) we may consider a periodic extension of f outside [0, 1] without assuming that f is zero near the boundary. With a slight abuse of notation we use the same symbol f for such extensions.

We note that an equally spaced discretization of (2.1) (which enables one to use the FFT) leads to a numerical error of order $1/N$, where N is the number of samples. Such behavior of the error due to jump discontinuities makes an accurate evaluation of (2.1) using an equally spaced discretization very expensive and not practical, especially in higher dimensions.

We now outline our approach. Let us consider an MRA of $\mathbf{L}^2(\mathbf{R})$,

$$\cdots \subset \mathbf{V}_2 \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \mathbf{V}_{-2} \subset \cdots, \tag{2.2}$$

where $\mathbf{V}_j$, $j < 0$, is a subspace of the MRA which is spanned by translations of a scaling function ϕ,

$$\phi_{kj}(x) = 2^{-j/2}\phi(2^{-j}x - k), \tag{2.3}$$

where $k \in \mathbf{Z}$. We start by projecting the function f on $\mathbf{V}_j$, $j < 0$, i.e., by computing the integrals

$$f_k = \int_{-\infty}^{\infty} f(x)\phi_{kj}(x)\,dx. \tag{2.4}$$

In order to estimate the Fourier transform of the function f,

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x\xi}\,dx, \tag{2.5}$$

we substitute

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)e^{2\pi i x\xi}\,d\xi \tag{2.6}$$

into (2.4) and obtain

$$f_k = 2^{-j/2}\int_{-\infty}^{\infty} \hat{f}(2^{-j}\xi)\bar{\hat{\phi}}(\xi)e^{2\pi i\xi k}\,d\xi, \tag{2.7}$$

where

$$\hat{\phi}(\xi) = \int_{-\infty}^{\infty} \phi(x)e^{-2\pi i x\xi}\,dx, \tag{2.8}$$

and $^-$ denotes the complex conjugate. Replacing the integral over the real line by that over the interval $[-\frac{1}{2}, \frac{1}{2}]$ in (2.7), we have

$$f_k = 2^{-j/2}\int_{-1/2}^{1/2} e^{2\pi i\xi k}\sum_{l\in\mathbf{Z}} \hat{f}(2^{-j}(\xi + l))\bar{\hat{\phi}}(\xi + l)\,d\xi. \tag{2.9}$$

Introducing the Fourier series

$$F(\xi) = \sum_{k\in\mathbf{Z}} f_k e^{-2\pi i\xi k}, \tag{2.10}$$

where F is periodic, $F(\xi) = F(\xi + 1)$, and using (2.9), we arrive at

$$F(\xi) = 2^{-j/2}\sum_{l\in\mathbf{Z}} \hat{f}(2^{-j}(\xi + l))\bar{\hat{\phi}}(\xi + l), \tag{2.11}$$

or

$$2^{j/2}\frac{F(\xi)}{\bar{\hat{\phi}}(\xi)} - \hat{f}(2^{-j}\xi) = \sum_{l=\pm1,\pm2\ldots} \hat{f}(2^{-j}(\xi + l))\frac{\bar{\hat{\phi}}(\xi + l)}{\bar{\hat{\phi}}(\xi)}. \tag{2.12}$$

We then have

$$\left|2^{j/2}\frac{F(\xi)}{\bar{\hat{\phi}}(\xi)} - \hat{f}(2^{-j}\xi)\right| \leq \sum_{l=\pm1,\pm2\ldots} C_{\hat{f}}(l,\alpha)\frac{|\hat{\phi}(\xi + l)|}{|\hat{\phi}(\xi)|}, \tag{2.13}$$

for $|\xi| \leq \alpha$, where

$$C_{\hat{f}}(l,\alpha) = \sup_{|\xi|\leq\alpha} |\hat{f}(2^{-j}(\xi + l))|, \tag{2.14}$$

and $\alpha > 0$ is a parameter. As we will see below the parameter α is related to the oversampling factor ν by $\alpha = 1/(2\nu)$.

Our main observation is that by an appropriate choice of the scaling function ϕ and the parameter $\alpha > 0$, the ratio

$$\tau_l(\xi) = \frac{|\hat{\phi}(\xi + l)|}{|\hat{\phi}(\xi)|} \tag{2.15}$$

may be made arbitrarily small for $l = \pm1, \pm2\ldots$ and, therefore, the estimate (2.13) leads to an algorithm for computing $\hat{f}(\xi)$ for $|\xi| \leq 2^{-j}\alpha$.

The choice of ϕ, on one hand, should permit an efficient computation of the integrals (2.4). In particular, the function ϕ should have as small support as possible. On the other hand, in order to have a useful estimate in (2.13) we need a tight localization of $\hat{\phi}$ in the frequency domain since we require the ratio τ_l in (2.15) to be small. As an illustration of the contradictory nature of these requirements, let us consider the scaling function for Meyer's basis for which (2.13) is trivial. Meyer's scaling function is compactly supported in the Fourier domain and is given by

$$\hat{\phi}(\xi) = \begin{cases} 1 & \text{for } |\xi| < \frac{1}{3} \\ \cos\left(\frac{\pi}{2} w(3|\xi| - 1)\right) & \text{for } \frac{1}{3} \leq |\xi| \leq \frac{2}{3} \\ 0 & \text{elsewhere,} \end{cases} \quad (2.16)$$

where w is a smooth function, $w(s) = 0$ for $s \leq 0$, $w(s) = 1$ for $s \geq 1$, and $w(s) + w(1 - s) = 1$. For the scaling function in (2.16) the r.h.s. of (2.13) is identically zero for $l \geq 2$ and we have $2^{j/2}F(\xi) = \hat{f}(2^{-j}\xi)$ for $|\xi| \leq \frac{1}{3}$. However, using the scaling function in (2.16) does not lead to an efficient algorithm because this function does not permit effective computation of integrals in (2.4).

It also turns out that using wavelets with compact support does not yield an efficient algorithm since the decay of the Fourier transform of compactly supported wavelets is not sufficiently fast. We demonstrate in the next Section how to satisfy the requirements of localization in both domains by choosing ϕ to be an mth order B-spline.

Let us now show that for symmetric scaling functions computing $F(\xi)/\bar{\hat{\phi}}(\xi)$ in (2.13) may be thought of as essentially an orthogonalization procedure. If the scaling function ϕ is symmetric and, hence, $\bar{\hat{\phi}}(\xi) = \hat{\phi}(\xi)$, then for $|\xi| \leq \alpha$ the function $\hat{\phi}(\xi)$ may be replaced by the periodic function $\sqrt{a(\xi)}$, where

$$a(\xi) = \sum_{l \in \mathbf{Z}} |\hat{\phi}(\xi + l)|^2. \quad (2.17)$$

The function $a(\xi)$ is usually used for orthogonalization of the basis on $\mathbf{V}_0$ since the functions $\{\varphi(x - l)\}_{l \in \mathbf{Z}}$ form an orthonormal basis on $\mathbf{V}_0$, where

$$\hat{\varphi}(\xi) = \frac{\hat{\phi}(\xi)}{\sqrt{a(\xi)}}. \quad (2.18)$$

We have from (2.17) and (2.15) that

$$a(\xi) = |\hat{\phi}(\xi)|^2 \left(1 + \sum_{l = \pm 1, \pm 2 \ldots} \tau_l^2(\xi)\right), \quad (2.19)$$

and thus, for $|\xi| \leq \alpha$,

$$\frac{\sqrt{a(\xi)}}{\hat{\phi}(\xi)} \approx 1 + \frac{1}{2} \sum_{l = \pm 1, \pm 2 \ldots} \tau_l^2(\xi), \quad (2.20)$$

under the same conditions which are sufficient for the validity of (2.13). It follows from (2.11) that

$$\frac{F(\xi)}{\sqrt{a(\xi)}} = 2^{-j/2} \sum_{l \in \mathbf{Z}} \hat{f}(2^{-j}(\xi + l)) \bar{\hat{\varphi}}(\xi + l), \quad (2.21)$$

where $\hat{\varphi}$ is given in (2.18). This implies that by computing the ratio $F(\xi)/\sqrt{a(\xi)}$ we effectively orthogonalize the expansion given by (2.10).

The steps leading up to the approximation of the function $\hat{f}(2^{-j}\xi)$ may be interpreted as follows. Computing the integrals in (2.4), i.e., the projection of the function f on $\mathbf{V}_j$, $j < 0$, may be viewed as a convolution of f with a "blurring" kernel. It plays the same role as the interpolation step in algorithms mentioned in Section I. Evaluating the Fourier series (2.10) brings the computation into the Fourier domain. Finally, calculating the ratio $F(\xi)/\sqrt{a(\xi)}$ may be viewed as a partial deconvolution of the "blurring" introduced at the first step. This deconvolution is performed in the Fourier domain and may be accomplished by the orthogonalization procedure.

In the next section we state and prove our main result.

III. CHOICE OF THE BASIS

In our construction we choose the MRA associated with spaces of polynomial splines. This allows us to use properties of the Battle–Lermarié scaling function while computing integrals only with the B-splines. Such an approach is critical for the efficiency of the algorithm.

We start by computing the integrals in (2.4) with the central B-splines,

$$f_k = \int_{-\infty}^{\infty} f(x) \beta_{kj}^{(m)}(x)\, dx, \quad (3.1)$$

where $\beta_{kj}^{(m)}(x) = 2^{-j/2}\beta^{(m)}(2^{-j}x - k)$ and $\beta^{(m)}$ is the mth order central B-spline. For convenience we only consider splines of odd order. In what follows we use the Fourier transform of $\beta^{(m)}$,

$$\hat{\beta}^{(m)}(\xi) = \left(\frac{\sin \pi\xi}{\pi\xi}\right)^{m+1}, \quad (3.2)$$

the periodic function $a^{(m)}$,

$$a^{(m)}(\xi) = \sum_{l=-\infty}^{l=\infty} |\hat{\beta}^{(m)}(\xi + l)|^2 = \sum_{l=-m}^{l=m} \beta^{(2m+1)}(l) e^{2\pi i l \xi}, \quad (3.3)$$

and the Fourier transform of the Battle–Lemarié scaling function,

$$\hat{\varphi}^{(m)}(\xi) = \frac{\hat{\beta}^{(m)}(\xi)}{\sqrt{a^{(m)}(\xi)}}. \quad (3.4)$$

The values $\beta^{(2m+1)}(l)$ may be computed using the recursion in (3.21) below (see, e.g., [6]) and, therefore, the function $a^{(m)}$ may be easily evaluated.

Let us summarize our results as

THEOREM III.1. *Let E_∞ be the error in approximating the Fourier transform $\hat{f}$ of the generalized function f by a periodic function $2^{j/2}F(\xi)/\sqrt{a^{(m)}(\xi)}$,*

$$E_\infty = \sup_{|\xi|\le\alpha} \left| 2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}} - \hat{f}(2^{-j}\xi) \right| \Big/ \sup_{|\xi|\le\alpha} |\hat{f}(2^{-j}\xi)|, \quad j < 0, \tag{3.5}$$

where $a^{(m)}$ is given by (3.3) *and F is the Fourier series,*

$$F(\xi) = \sum_{k\in\mathbf{Z}} f_k e^{-2\pi i k\xi}, \tag{3.6}$$

with coefficients f_k given in (3.1).

If

$$|\hat{f}(\xi)| \le C(1 + |\xi|)^\sigma, \quad \sigma < m, \tag{3.7}$$

then we have

$$E_\infty \le \frac{1}{2\hat{\varphi}(\alpha) - 1}\left[1 - \hat{\varphi}(\alpha) + \frac{1}{C_{\hat{f}}(0,\alpha)} \times \sum_{l=\pm1,\pm2,\ldots} C_{\hat{f}}(l,\alpha)\left(\frac{\alpha}{|l| - \alpha}\right)^{m+1}\right], \tag{3.8}$$

where

$$C_{\hat{f}}(l,\alpha) = \sup_{|\xi|\le\alpha} |\hat{f}(2^{-j}(\xi + l))|. \tag{3.9}$$

For any $\epsilon > 0$ we may choose m, the order of the central B-spline, and the parameter $\alpha > 0$ so that for $|\xi| \le \alpha$

$$E_\infty \le \epsilon. \tag{3.10}$$

The proof of the theorem may be found in Appendix A.

In order to see that Theorem III.1 provides a close estimate for the error of the approximation, let us consider the r.h.s. of (3.8). One of the reasons for choosing the MRA associated with spaces of polynomial splines is the behavior in the Fourier domain of the Battle-Lemarié scaling function $\hat{\varphi}^{(m)}$ given in (3.4). We show in Appendix B that the function $\hat{\varphi}^m(\xi)$ is strictly monotone for $\xi \in (0,1)$ and $\xi \in (-1,0)$ and we obtain

$$\sup_{|\xi|\le\alpha} |1 - \hat{\varphi}^m(\xi)| = 1 - \hat{\varphi}^{(m)}(\alpha). \tag{3.11}$$

To estimate $1 - \hat{\varphi}^{(m)}(\alpha)$, we may use the expansion of $\hat{\varphi}^{(m)}(\xi)$ around $\xi = 0$ and find[3]

$$\hat{\varphi}^{(m)}(\xi) = 1 + O(\xi^{2m+2}). \tag{3.12}$$

Let us estimate the radius of convergence of (3.12) by examining the denominator in (3.4). As it is shown in [6], the $2m$ roots $\lambda_1, \lambda_2, \ldots, \lambda_{2m}$ of the polynomial

$$E(z) = (2m+1)!z^m \sum_{l=-m}^{l=m} \beta^{(2m+1)}(l)z^l \tag{3.13}$$

are all negative and simple, $0 > \lambda_1 > \lambda_2 > \cdots > \lambda_{2m}$, and form reciprocal pairs, $\lambda_1\lambda_{2m} = \cdots = \lambda_{m-1}\lambda_m = 1$. Therefore, the complex zeros of $a^{(m)}(\xi)$ in (3.3) are of the form

$$\xi_k = \frac{1}{2} \pm i\frac{\log(-\lambda_k)}{2\pi}, \quad k = 1,\ldots,m \tag{3.14}$$

and, thus, the radius of convergence of the expansion in (3.12) is greater than $\frac{1}{2}$. Consequently, we may use (3.12) to estimate

$$1 - \hat{\varphi}^{(m)}(\alpha) \approx C\alpha^{2m+2}, \tag{3.15}$$

where C is a constant. Setting $\alpha = \frac{1}{4}$ (which corresponds to the oversampling factor of 2), we compute $1 - \hat{\varphi}^{(m)}(\frac{1}{4})$ as a function of m and display the results in Table 1.

We now turn to the sum in the estimate (3.8) and consider the first term, $l = \pm1$, for $\alpha = \frac{1}{4}$. We have $(\frac{1}{3})^{m+1}(C_{\hat{f}}(1,\alpha) + C_{\hat{f}}(-1,\alpha))/C_{\hat{f}}(0,\alpha)$ and note that it provides a close approximation to the error observed in the numerical experiments (see Section VIII).

To illustrate why $2^{j/2}F(\xi)/\sqrt{a^{(m)}(\xi)}$ is a good approximation of $\hat{f}(2^{-j}\xi)$ for $|\xi| \le \alpha$, we obtain from (2.21)

$$2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}} = \hat{f}(2^{-j}\xi)\hat{\varphi}^{(m)}(\xi) + \sum_{l=\pm1,\pm2,\ldots} \hat{f}(2^{-j}(\xi + l))\hat{\varphi}^{(m)}(\xi + l), \tag{3.16}$$

TABLE 1
Values $1 - \hat{\varphi}^{(m)}(\frac{1}{4})$ as a Function of the Order of the Battle–Lemarié Scaling Function

Order m	$1 - \hat{\varphi}^{(m)}(\frac{1}{4})$
3	0.77580E-04
5	0.94292E-06
7	0.11619E-07
9	0.14340E-09
11	0.17701E-11
13	0.21878E-13
15	0.81743E-15

[3] For comparison, we have from (3.2)

$$\hat{\beta}^{(m)}(\xi) = 1 - \frac{(1+m)\pi^2}{6}\xi^2 + O(\xi^4).$$

where $\hat{\varphi}^{(m)}$ is the Fourier transform of the Battle–Lemarié scaling function [10, 1]. We plot $\hat{\varphi}^{(m)}(\xi), \hat{\varphi}^{(m)}(\xi + 1)$ and $\hat{\varphi}^{(m)}(\xi - 1)$ for $m = 23$ in Fig. 1. We note that for $\alpha = \frac{1}{4}$ and $|\xi| \leq \alpha$ the values of $\hat{\varphi}^{(m)}(\xi)$ are equal to one (with double precision accuracy), whereas $\hat{\varphi}^{(m)}(\xi + 1)$ and $\hat{\varphi}^{(m)}(\xi - 1)$ are equal to zero with the same precision.

Using Theorem III.1 and setting $\alpha = \frac{1}{4}$, we have (with accuracy ϵ)

$$\hat{f}(l) = \frac{1}{L^{1/2}\sqrt{a^{(m)}(l/L)}} \sum_{k \in \mathbf{Z}} f_k e^{-2\pi i k l/L}, \tag{3.17}$$

for $-L/4 + 1 \leq l \leq L/4$, where $L = 2^{-j}$. Since we have extended f by zero outside $[0, 1]$ and assumed that f is zero in a narrow strip near the boundary of $[0, 1]$, we may always arrange $f_k = 0$ for $k < 0$ and $k \geq L$. Thus, we replace the series in (3.17) by a finite sum and obtain

$$\hat{f}(l) = \frac{1}{L^{1/2}\sqrt{a^{(m)}(l/L)}} \sum_{k=0}^{L-1} f_k e^{-2\pi i k l/L}, \qquad -L/4 \leq l \leq L/4, \tag{3.18}$$

which may be evaluated using the FFT.

As a result we have a simple algorithm based on the approximation in Theorem III.1. The algorithm consists of three steps:

1. computing integrals in (3.1)
2. computing the sum in (3.18) via FFT
3. multiplying by the factor $1/\sqrt{a^{(m)}(l/L)}$ in (3.18).

EXAMPLE III.1. The error of computing via (3.18) is illustrated in Fig. 2 for the characteristic function of an interval ($f(x) = \chi([0.3, 0.55])$), where $j = -9, L = 512$ and $m = 23$. We observe that for $-128 \leq l \leq 128$ the absolute error of computed values $\hat{f}(l)$ is below 10^{-14}.

It is clear that we may use Theorem III.1 in a region $|\xi| \leq \alpha$, where $\alpha < \frac{1}{4}$. Choosing a smaller α permits us to select a lower order B-spline to achieve a given precision and, thus, decreases the number of operations necessary to project f on $\mathbf{V}_j$. On the other hand, choosing a smaller α increases the number of operations at the second step due to the larger oversampling factor. The choice of $\alpha = \frac{1}{4}$ results in the oversampling factor of 2 which we use in all of our experiments in Section VIII.

Let us now consider the projection step of the algorithm, namely computing integrals in (3.1). Given a MRA, we always have the corresponding two-scale difference equation and may use it to evaluate integrals in (3.1). For some functions the resulting algorithm is efficient and reduces the problem to that of solving a small linear system (see e.g. [2]). Specifically, for the central B-splines of odd order the two-scale difference equation

$$\beta^{(m)}(x) = \sum_{k=-(m+1)/2}^{(m+1)/2} \frac{1}{2^m} \times \binom{m+1}{k + (m+1)/2} \beta^{(m)}(2x - k), \tag{3.19}$$

may be used to compute integrals (2.4). However, an algorithm based on the two-scale difference equation tends to be inefficient for general functions f. One of the reasons for choosing the MRA associated with spaces of polynomial splines is the availability of two additional (well-known) methods for evaluating integrals with the B-splines, namely, an approach based on the explicit representation of splines as piecewise polynomials and an approach based on a recursion over the order of the B-splines.

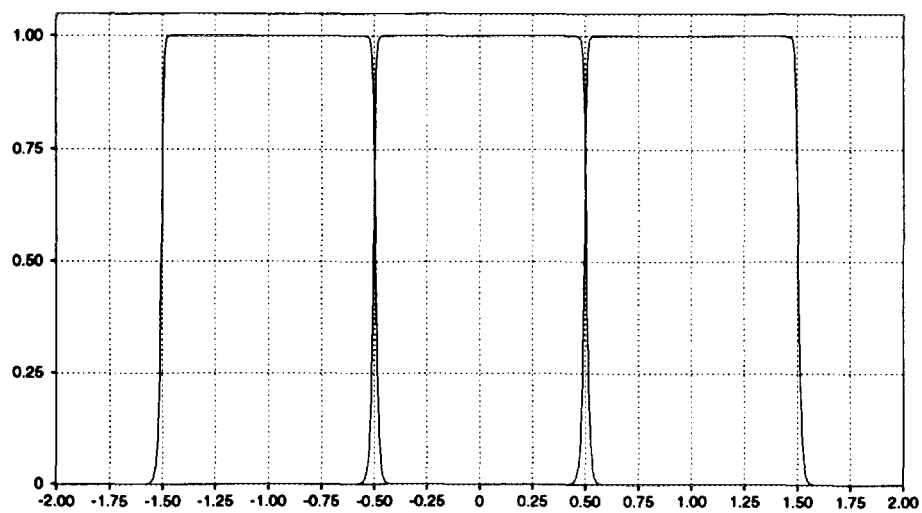

FIG. 1 The Fourier transform of Battle–Lemarié scaling function of order $m = 23$. Shown are functions $\hat{\varphi}^{(m)}(\xi), \hat{\varphi}^{(m)}(\xi + 1)$ and $\hat{\varphi}^{(m)}(\xi - 1)$.

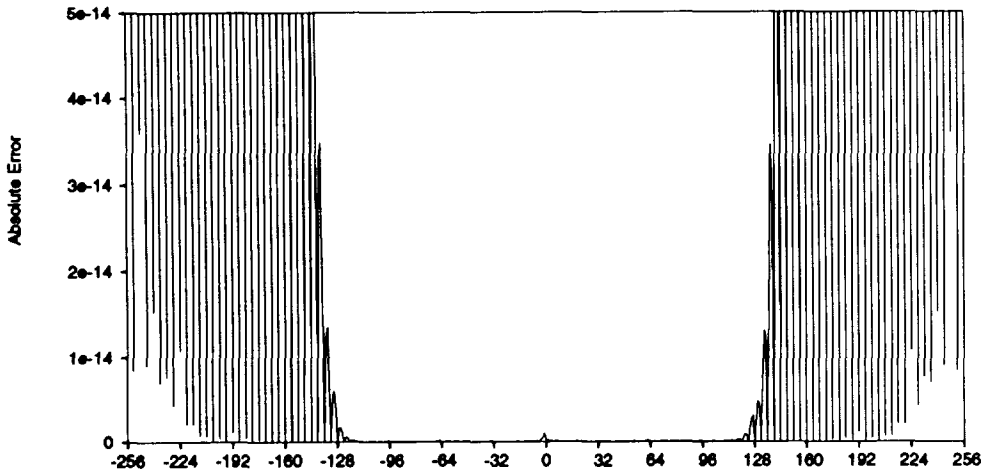

FIG. 2 The absolute error of $\hat{f}(l)$ computed via (3.18) in Example III.1.

As piecewise polynomials, the central B-splines may be written as

$$\beta^{(m)}(x) = \sum_{l=0}^{m+1}(-1)^l \binom{m+1}{l} \frac{(x + (m+1)/2 - l)_+^m}{m!}, \quad (3.20)$$

where x_+^m is defined as x^m for $x > 0$ and zero for $x \leq 0$. Using (3.20) for computing (2.4) reduces the problem to that of evaluating integrals of f with polynomials.

For several important functions (e.g., the characteristic function of a rectangle) the integrals with the B-splines may be computed analytically and expressed in terms of the values of B-splines (of higher order). The values of the B-splines may be obtained using recursion over the spline order,

$$\beta^{(m)}(x) = \frac{(m+1)/2 + x}{m}\beta^{(m-1)}(x + \tfrac{1}{2}) + \frac{(m+1)/2 - x}{m}\beta^{(m-1)}(x - \tfrac{1}{2}), \quad (3.21)$$

where $m = 1, 2, \ldots$ and $\beta^{(0)}(x)$ is the characteristic function of the interval $[-\frac{1}{2}, \frac{1}{2}]$. In our implementations we used a piecewise polynomial representation as well as (3.21).

IV. THE UNEQUALLY SPACED FFT

Theorem III.1 yields a fast algorithm for the evaluation of trigonometric sums

$$\hat{g}_n = \sum_{l=0}^{N_p-1} g_l e^{2\pi i x_l n}, \quad (4.1)$$

where $x_l \in [0, 1]$ and $\{g_l\}_{l=0}^{N_p-1}$ is a set of N_p complex numbers. Indeed, we may replace (4.1) by the integral

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{2\pi i x \xi}\, dx, \quad (4.2)$$

where

$$f(x) = \sum_{l=0}^{N_p-1} g_l \delta(x - x_l), \quad (4.3)$$

is a sum of δ-functions and $\hat{g}_n = \hat{f}(n)$.

Using Theorem III.1 we compute the coefficients of the Fourier series in (3.1),

$$f_k = 2^{-j/2} \sum_{l=0}^{N_p-1} g_l \beta^{(m)}(2^{-j}x_l - k), \quad (4.4)$$

and then evaluate

$$F(\xi) = \sum_{k \in \mathbf{Z}} f_k e^{-2\pi i \xi k}, \quad (4.5)$$

to obtain

$$\sup_{|\xi| \leq \alpha} \left| 2^{j/2} \frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}} - \hat{f}(2^{-j}\xi) \right| \leq \epsilon \sup_{|\xi| \leq \alpha} |\hat{f}(2^{-j}\xi)| \quad j > 0, \quad (4.6)$$

for $|\xi| \leq \alpha$. Thus, the algorithm for evaluating the trigonometric sum in (4.1) is a special case of the algorithm described in Section III. For example, if we choose the oversampling factor $\nu = 2$ then we have

$$\hat{g}_n = \hat{f}(n) = \frac{1}{N^{1/2}\sqrt{a^{(m)}(n/N)}} \sum_{k=0}^{N-1} f_k e^{-2\pi i k n/N}, \quad (4.7)$$

for $-N/4 + 1 \leq n \leq N/4$ and the accuracy is controlled by the choice of the order m of the B-splines.

The cost of computing (4.4) is proportional to mN_p, where m is chosen to be proportional to $-\log\epsilon$, the desired number of accurate digits. The cost of computing (4.7) via the FFT requires $O(\nu N \log N)$ operations, where $\nu = 1/(2\alpha)$ is the oversampling factor. The modification step requires $O(N)$ operations. For a given accuracy, ν is proportional to $1/m$ and the total cost may be estimated as

$$C_1 mN_p + C_2 \frac{1}{m} N \log N + C_3 N,$$

where $C_i, i = 1,2,3$ are constants which depend on the implementation. The USFFT algorithm has been implemented and we present the results in Section VIII. The two-dimensional version of the algorithm is described in the next section.

V. ALGORITHMS FOR FUNCTIONS OF SEVERAL VARIABLES

Let us outline an extension of Theorem III.1 of Section III for functions of several variables. We describe this extension by stating the result for functions of two independent variables which is illustrative of the general case.

As an approximation to the function $\hat{f}(2^{-j}\xi, 2^{-j}\eta)$, $j < 0$, $|\xi| \leq \alpha$ and $|\eta| \leq \alpha'$, where

$$\hat{f}(\xi,\eta) = \int_0^1 \int_0^1 f(x,y) e^{-2\pi i\xi x} e^{-2\pi i\eta y}\, dxdy, \tag{5.1}$$

we consider the function

$$\mathcal{F}(\xi,\eta) = \frac{2^j F(\xi,\eta)}{\sqrt{a^{(m)}(\xi)a^{(m')}(\eta)}}, \tag{5.2}$$

where the periodic function $a^{(m)}$ is given in (3.3). The function $F(\xi,\eta)$ is described by its trigonometric series

$$F(\xi,\eta) = \sum_{k,k'\in\mathbf{Z}} f_{kk'} e^{-2\pi ik\xi} e^{-2\pi ik'\eta}, \tag{5.3}$$

with the coefficients

$$f_{kk'} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\beta_{kj}^{(m)}(x)\beta_{k'j}^{(m')}(y)\, dxdy, \tag{5.4}$$

where $\beta^{(m)}$ is the mth order central B-spline. We note that we may use B-splines of different orders m and m'.

Let E_∞ be the error in approximating $\hat{f}(2^{-j}\xi, 2^{-j}\eta)$ ($j < 0$) by the periodic function $\mathcal{F}(\xi,\eta)$,

$$E_\infty = \sup_{|\xi|\leq\alpha,|\eta|\leq\alpha'} |\mathcal{F}(\xi,\eta) - \hat{f}(2^{-j}\xi, 2^{-j}\eta)| / \sup_{|\xi|\leq\alpha,|\eta|\leq\alpha'} |\hat{f}(2^{-j}\xi, 2^{-j}\eta)|. \tag{5.5}$$

By using the same arguments as in the proof of Theorem III.1 in Appendix A, it is easy to derive an estimate for E_∞ similar to that in Theorem III.1. Here the parameters α and α' play the same role as in the case of one independent variable. By evaluating the function $\mathcal{F}$ we obtain an algorithm for computing (5.1) for functions of two independent variables.

As an example, let us provide formulas for computing the Fourier transform of generalized functions of two variables using the oversampling factor $\nu = 2$ ($\alpha = \alpha' = \frac{1}{4}$) in both variables. We have

$$\hat{f}(n,n') = \frac{1}{N\sqrt{a^{(m)}(n/N)a^{(m)}(n'/N)}} \times \sum_{k=0}^{N-1}\sum_{k'=0}^{N-1} f_{kk'} e^{-2\pi ikn/N} e^{-2\pi ik'n'/N}, \tag{5.6}$$

for $-N/4 + 1 \leq n, n' \leq N/4, N = 2^{-j}$, where the coefficients $f_{kk'}$ are given in (5.4). The accuracy ϵ is controlled by an appropriate choice of the order of the B-spline.

The algorithm for evaluating (5.6) is completely analogous to that described in Section III. The first step of the algorithm consists of computing the integrals of the function f with the central B-splines in (5.4). The second step involves the FFT of a $(\nu N) \times (\nu N)$ matrix, where $\nu = 1/(2\alpha) = 2$. The third step of the algorithm consists of multiplying values at each frequency by the pre-computed factor $\sqrt{a^{(m)}(n/N)a^{(m)}(n'/N)}$ to obtain the result. At this step we effectively generate a representation involving the Battle–Lemarié scaling function.

The two-dimensional version of the USFFT for computing

$$\hat{g}_{n,n'} = \sum_{l=1}^{N_p} g_l e^{-2\pi inx_l} e^{-2\pi in'y_l}, \quad -N \leq n, n' \leq N, \tag{5.7}$$

where $\{g_l\}_{l=1}^{N_p}$ is a set of N_p complex numbers, is obtained by computing (5.1) for the generalized function

$$f(x,y) = \sum_{l=1}^{N_p} g_l \delta(x - x_l)\delta(y - y_l). \tag{5.8}$$

We obtain from (5.4)

$$f_{kk'} = \sum_{l=1}^{N_p} g_l \beta_{kj}^{(m)}(x_l)\beta_{k'j}^{(m)}(y_l), \tag{5.9}$$

and then use the FFT to evaluate (5.6).

The cost of generating the matrix $f_{kk'}$ in (5.9) is proportional to m^2N_p where m is the order of the central B-spline (which is usually chosen to be proportional to $-\log\epsilon$). The second step, the FFT of the matrix $f_{kk'}$, requires $O(\nu^2N^2\log N)$ operations, where ν is the oversampling factor. The third step requires $O(N^2)$ operations and its cost is negligible if compared to the first two steps. For a given accuracy, the oversampling factor ν is inversely proportional to m and the total cost may be estimated as

$$C_1m^2N_p + C_2\frac{1}{m^2}N^2\log N + C_3N^2,$$

where $C_i, i = 1,2,3$ are constants which depend on the implementation. The choice of the parameter m may be used to optimize the performance of the algorithm. In our computations we used the oversampling factor $\nu = 2$. We provide the run times for all steps of the algorithm in Section VIII.

VI. EVALUATING THE FOURIER TRANSFORM AT UNEQUALLY SPACED POINTS

According to Theorem III.1, the function $\hat{f}(2^{-j}\xi)$ for $|\xi| \leq \alpha, j < 0$, is well approximated by $2^{j/2}F(\xi)/\sqrt{a^{(m)}(\xi)}$, where F is given in (3.6),

$$F(\xi) = \sum_{k\in\mathbf{Z}} f_k e^{-2\pi ik\xi}. \tag{6.1}$$

In order to evaluate $\hat{f}(2^{-j}\xi)$ at an arbitrary set of points $|\xi_l| \leq \alpha, l = 1,2,\ldots,L$, it is sufficient to construct a fast algorithm for computing $F(\xi)$ at these points. This may be accomplished by interpolation and we now describe a scheme which uses central B-splines.

Let us consider

$$F(\xi) = \sum_{k=0}^{N-1} f_k e^{-2\pi ik\xi}, \tag{6.2}$$

and describe an algorithm to evaluate (6.2) at an arbitrary set of points $|\xi_l| \leq \frac{1}{2}, l = 1,2,\ldots,N_p$. We rewrite (6.2) as

$$F(\xi) = e^{-\pi iN\xi}\sum_{k=-N}^{N-1} w_k e^{-2\pi ik\xi}, \tag{6.3}$$

where

$$w_k = \begin{cases} f_{k+N/2} & \text{for } -N/2 \leq k \leq N/2-1 \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

and look for an approximation of $F(\xi)$ as

$$F(\xi) = e^{-\pi iN\xi}\sum_{l=-N}^{N-1} F_l\beta^{(m)}(N\xi - l), \tag{6.5}$$

where $\beta^{(m)}$ is the mth order central B-spline and F_l are the coefficients to be determined. The estimate for such approximation follows from Theorem III.1 since the function in (6.3) (apart from oscillatory factor) is replaced by its projection on a subspace of B-splines. As elsewhere in the paper, we use the oversampling factor $\nu = 1/(2\alpha) = 2$ in (6.4) and note that for a given accuracy using a larger oversampling factor allows one to use lower order B-splines.

In order to find the coefficients F_l in (6.5) we have the set of linear equations

$$\sum_{k=-N}^{N-1} w_k e^{-2\pi ikn/N} = \sum_{l=-N}^{N-1} F_l\beta^{(m)}(n-l), \tag{6.6}$$

where $n = -N,\ldots,N-1$. Considering

$$\hat{F}_k = \sum_{l=-N}^{N-1} F_l e^{2\pi ilk/(2N)}, \tag{6.7}$$

and

$$\hat{b}_k = \sum_{l=-(m-1)/2}^{(m-1)/2} \beta^{(m)}(l)e^{2\pi ilk/(2N)}, \tag{6.8}$$

for $k = -N,\ldots,N-1$, we have

$$\hat{F}_k = \frac{w_k}{\hat{b}_k}. \tag{6.9}$$

Thus, we obtain an algorithm consisting of the following steps:

1. Modification of the coefficients w_k by the factor $\hat{b}_k$ in (6.9). This step requires $O(N)$ operations.
2. Application of the FFT to compute the coefficients F_l in (6.5) from coefficients $\hat{F}_k$ in (6.9). This step requires $O(N\log N)$ operations.
3. Evaluation of (6.5) at points $|\xi_l| \leq \frac{1}{2}$ to obtain $F(\xi_l)$ $l = 1,2,\ldots,L$. This step requires $O(N)$ operations.

These steps are similar to those of the algorithm in Section III but are taken in the reverse order. Therefore, the computational complexity of the interpolation algorithm is exactly the same as that of algorithms in Section III and Section IV. The accuracy of the scheme is illustrated in Table 2.

The generalization of this interpolation scheme to higher dimensions is straightforward and we outline it below for two independent variables. We seek a fast algorithm to evaluate

TABLE 2
Accuracy of the Interpolation Scheme as the Function of the Order of B-Splines

Order m	L_∞-norm	L_2-norm
3	0.95312E-01	0.16955E-01
5	0.88320E-02	0.16509E-02
7	0.80374E-03	0.16821E-03
9	0.78343E-04	0.17344E-04
11	0.79291E-05	0.18033E-05
13	0.80319E-06	0.18899E-06
15	0.81645E-07	0.19960E-07
17	0.83404E-08	0.21235E-08
19	0.85673E-09	0.22743E-09
21	0.88510E-10	0.24503E-10
23	0.91993E-11	0.26551E-11
25	0.96501E-12	0.29075E-12
27	0.19185E-12	0.35252E-13

Note. The estimate was obtained using pseudo-random coefficients f_k in (6.2). In this test $N = 128$ and $L = 127$.

$$F(\xi,\eta) = \sum_{k,k'=0}^{N-1} f_{kk'} e^{-2\pi i k\xi} e^{-2\pi i k'\eta}, \tag{6.10}$$

at an arbitrary set of points (ξ_l, η_l), where $|\xi_l|, |\eta_l| \leq \frac{1}{2}, l = 1, 2, \ldots, N_p$. We approximate $F(\xi,\eta)$ as

$$F(\xi,\eta) = e^{-\pi i N\xi} e^{-\pi i N\eta} \times \sum_{l,l'=-N}^{N-1} F_{ll'} \beta^{(m)}(N\xi - l)\beta^{(m)}(N\eta - l'), \tag{6.11}$$

where $\beta^{(m)}$ is the mth order central B-spline. The coefficients $F_{ll'}$ are determined using

$$\hat{F}_{kk'} = \sum_{l,l'=-N}^{N-1} F_{ll'} e^{2\pi i l k/(2N)} e^{2\pi i l' k'/(2N)}, \tag{6.12}$$

where

$$\hat{F}_{kk'} = \frac{w_{kk'}}{\hat{b}_k \hat{b}'_k}, \quad -N \leq k, k' \leq N-1. \tag{6.13}$$

In (6.13) factors $\hat{b}_k$ are given in (6.8) and

$$w_{kk'} = \begin{cases} f_{k+N/2,k'+N/2} & \text{for } -N/2 \leq k, k' \leq N/2 - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.14}$$

The steps of the interpolation algorithm in the case of two independent variables are in the reverse order as compared to the steps of the algorithm in Section V and, therefore, the interpolation algorithm has the same computational complexity as USFFT algorithm in Section V.

VII. APPROXIMATIONS OF SINGULAR FUNCTIONS AND THE IMMERSED INTERFACE METHOD

In this section we consider properties of projections of functions with singularities on spaces of polynomial splines. We demonstrate that such projections may be used to obtain the so-called discrete approximation of functions with singularities, in particular, the δ-function. Discrete approximations of singular functions appears as a tool in the Immersed Interface (Boundary) method (see [8, 12] and references therein). The idea of this method is to discretize a partial differential equation on a uniform rectangular grid and then distribute the singularities of the coefficients or of the source term to the neighboring grid points so that the overall accuracy of the scheme is maintained. Such a procedure permits the development of efficient and accurate algorithms for problems with complicated or moving boundaries [8, 12]. The central question in this approach is how to approximate functions with singularities.

Let f be a generalized function and let us consider its projection on the subspace $\mathbf{V}_j$, $j < 0$,

$$p(x) = \sum_{k\in\mathbf{Z}} f_k \gamma_{kj}^{(m)}(x), \tag{7.1}$$

where f_k are given in (3.1),

$$f_k = \int_{-\infty}^{\infty} f(x)\beta_{kj}^{(m)}(x)\,dx, \tag{7.2}$$

and where $\gamma_{kj}^{(m)}(x) = 2^{-j/2}\gamma^{(m)}(2^{-j}x - k)$. The function $\gamma^{(m)}$ is the dual of the central B-spline $\beta^{(m)}$ (see e.g., [6]), and is defined so that

$$\int_{-\infty}^{\infty} \beta^{(m)}(x-k)\gamma^{(m)}(x-l)\,dx = \delta_{kl}, \tag{7.3}$$

and

$$\hat{\gamma}^{(m)}(\xi) = \frac{\hat{\beta}^{(m)}(\xi)}{a^{(m)}(\xi)}, \tag{7.4}$$

where $\hat{\beta}^{(m)}$ and $a^{(m)}$ are given in (3.2) and (3.3).

We also consider an alternative expression for the projection (7.1) using the Battle–Lemarié scaling function $\varphi^{(m)}$,

$$p_o(x) = \sum_{k\in\mathbf{Z}} f_k^o \varphi_{kj}^{(m)}(x), \tag{7.5}$$

where $\varphi_{kj}^{(m)}(x) = 2^{-j/2}\varphi^{(m)}(2^{-j}x - k)$ and

$$f_k^o = \int_{-\infty}^{\infty} f(x)\varphi_{kj}^{(m)}(x)\,dx. \tag{7.6}$$

Let us verify that the projections are indeed the same, $p_o(x) = p(x)$, by computing their Fourier transform. Using (7.1) we have

$$\hat{p}(\xi) = \int_{-\infty}^{\infty} p(x)e^{-2\pi ix\xi}\,dx = 2^{j/2}\hat{\gamma}^{(m)}(2^j\xi)F(2^j\xi) = \frac{2^{j/2}\hat{\beta}^{(m)}(2^j\xi)F(2^j\xi)}{a^{(m)}(2^j\xi)}, \tag{7.7}$$

where F is defined in (2.10). Alternatively, using (7.5) we obtain

$$\hat{p}_o(\xi) = \int_{-\infty}^{\infty} p_0(x)e^{-2\pi ix\xi}\,dx = 2^{j/2}\hat{\varphi}^{(m)}(2^j\xi)\sum_{k\in\mathbf{Z}} f_k^o e^{-2\pi ik2^j\xi}, \tag{7.8}$$

or

$$\hat{p}_o(\xi) = 2^{j/2}\hat{\varphi}^{(m)}(2^j\xi)\frac{F(2^j\xi)}{\sqrt{a^{(m)}(2^j\xi)}}, \tag{7.9}$$

so that $\hat{p}_o(\xi) = \hat{p}(\xi)$.

LEMMA VII.1. *For any $\epsilon > 0$ we may choose the order m of the central* B*-spline and $\alpha > 0$ so that for $|\xi| \leq 2^{-j}\alpha$*

$$\sup_{|\xi|\leq 2^{-j}\alpha} |\hat{p}(\xi) - \hat{f}(\xi)| \leq \epsilon \sup_{|\xi|\leq 2^{-j}\alpha} |\hat{f}(\xi)|. \tag{7.10}$$

Proof. Using (7.9) we have

$$|\hat{p}(\xi) - \hat{f}(\xi)| \leq \left|\hat{\varphi}^{(m)}(2^j\xi)\left(2^{j/2}\frac{F(2^j\xi)}{\sqrt{a^{(m)}(2^j\xi)}} - \hat{f}(\xi)\right)\right| + |(\hat{\varphi}^{(m)}(2^j\xi) - 1)\hat{f}(\xi)|, \tag{7.11}$$

or, since $\sup_{|\xi|\leq 2^{-j}\alpha} \hat{\varphi}^{(m)}(2^j\xi) = \hat{\varphi}^{(m)}(\alpha) < 1$ (see Appendix B),

$$\sup_{|\xi|\leq 2^{-j}\alpha} |\hat{p}(\xi) - \hat{f}(\xi)| \leq \sup_{|\xi|\leq 2^{-j}\alpha} \left|2^{j/2}\frac{F(2^j\xi)}{\sqrt{a^{(m)}(2^j\xi)}} - \hat{f}(\xi)\right| + (1 - \hat{\varphi}^{(m)}(\alpha)) \sup_{|\xi|\leq 2^{-j}\alpha} |\hat{f}(\xi)|. \tag{7.12}$$

Applying Theorem of Section III, we obtain the result.

Generalized functions are defined as functionals by their action on a space of smooth tests functions. Let us show that in this sense projections p and p_o ($p_o = p$) provide a good approximation of generalized functions.

LEMMA VII.2. *For all test functions from $\mathbf{V}_j$ the generalized function f and its projection p_o are identical as functionals.*

Proof. We obtain using (7.5) and (7.6)

$$\int_{-\infty}^{\infty} p_o(x)w(x)\,dx = \sum_{k\in\mathbf{Z}} f_k^o \int_{-\infty}^{\infty} \varphi_{kj}^{(m)}(x)w(x)\,dx = \int_{-\infty}^{\infty} f(x)w_j(x)\,dx, \tag{7.13}$$

where

$$w_j(x) = \sum_{k\in\mathbf{Z}} \varphi_{kj}^{(m)}(x)\int_{-\infty}^{\infty} \varphi_{kj}^{(m)}(x)w(x)\,dx \tag{7.14}$$

is the projection of the test function w on the subspace $\mathbf{V}_j$, $j < 0$. Thus, we have

$$\int_{-\infty}^{\infty} f(x)w(x)\,dx - \int_{-\infty}^{\infty} p_o(x)w(x)\,dx = \int_{-\infty}^{\infty} f(x)(w(x) - w_j(x))\,dx. \tag{7.15}$$

If the test function $w \in \mathbf{V}_j$, then $w = w_j$ and we obtain Lemma VII.2.

VII.1. Approximation of the δ-Function

Let $f(x) = \delta(x - x_0)$ in (7.6), then we have from (7.5)

$$p_o(x) = \sum_{k\in\mathbf{Z}} \varphi_{kj}^{(m)}(x_0)\varphi_{kj}^{(m)}(x). \tag{7.16}$$

LEMMA VII.3. *Let w be a test function, $w \in C^\infty$. Then*

$$w(x_0) - \int_{-\infty}^{\infty} p_o(x)w(x)\,dx = O(h^{m+1}), \tag{7.17}$$

where $h = 2^j$, $j < 0$.

Proof. Since the Battle–Lemarié scaling function $\varphi^{(m)}$ has $2m + 1$ vanishing moments, expanding w into the Taylor series yields

$$\int_{-\infty}^{\infty} \varphi_{kj}^{(m)}(x)w(x)\,dx = 2^{j/2}w(2^jk) + O(h^{2m+2+1/2}), \tag{7.18}$$

where $h = 2^j$, $j < 0$. Thus, we have

$$w(x_0) - \int_{-\infty}^{\infty} p_o(x)w(x)\,dx = w(x_0) - \sum_{k\in\mathbf{Z}} 2^{j/2}w(2^jk)\varphi_{kj}^{(m)}(x_0) + O(h^{2m+2+1/2}). \tag{7.19}$$

We then expand w into Taylor series at the point x_0,

TABLE 3
Timing and Accuracy for Example VIII.1 (One Rectangle)

N	T_p	T_{FFT}	T_m	T_{tot}	Error E_∞	T_{dir}
64	0.04	0.50	0.01	0.55	4.9e-15	0.20
128	0.15	2.55	0.05	2.75	1.6e-15	1.89
256	0.60	11.45	0.20	12.24	1.2e-15	3.91
512	2.15	52.22	0.82	55.18	1.4e-15	16.86

$$w(2^j k) = w(x_0) + \sum_{l=1}^{\infty} \frac{h^l}{l!} w^{(l)}(x_0)(k - 2^{-j}x_0)^l, \tag{7.20}$$

substitute $w(2^j k)$ into (7.19) and use Lemma XII.1 of Appendix C to obtain (7.17).

Remark. Since $p(x) = p_o(x)$, we have

$$w(x_0) = \int_{-\infty}^{\infty} p(x)w(x)\,dx + O(h^{m+1}), \tag{7.21}$$

where

$$p(x) = \sum_{k\in\mathbf{Z}} \beta_{kj}^{(m)}(x_0)\gamma_{kj}^{(m)}(x). \tag{7.22}$$

The dual of the B-spline $\gamma^{(m)}$ and, therefore, the projection p do not have compact support. Since in the Immersed Interface Method only points in the neighborhood of the singularity are modified, it may appear that there is no connection between projections described in this section and approximations considered in [8, 12]. Yet, the connection with the Immersed Interface Method may be established if we use (7.22).

As an example let us consider the problem

$$u'' = -C\delta(x - x_0), \quad u(0) = u(1) = 0, \tag{7.23}$$

with the solution $u(x) = Cx(1 - x_0)$ for $x \leqslant x_0$ and $u(x) = Cx_0(1 - x)$ for $x \geqslant x_0$. The problem (7.23) is one of the example considered in [12]. We choose the order of splines $m = 1$ so that $\beta^{(1)}$ is the so-called "hat" function and look for the solution of (7.23) of the form

TABLE 5
Timing for Example VIII.3 (40,000 rectangles)

N	T_p	T_{FFT}	T_m	T_{tot} (sec)	T_{dir} in hours (est.)
64	8.94	0.50	0.01	9.45	3 h
128	9.73	2.55	0.05	12.33	11 h
256	11.79	11.45	0.21	23.45	44 h
512	16.02	52.22	0.82	69.06	176 h

$$u(x) = \sum_{k\in\mathbf{Z}} u_k \beta_{kj}^{(1)}(x). \tag{7.24}$$

Let us replace the δ-function in (7.23) by its approximation $p(x)$ in (7.22). We have

$$\sum_{k\in\mathbf{Z}} u_k \frac{d^2}{dx^2}\beta_{kj}^{(1)}(x) = -C\sum_{k\in\mathbf{Z}} \beta_{kj}^{(1)}(x_0)\gamma_{kj}^{(1)}(x). \tag{7.25}$$

Multiplying both sides of (7.25) by $\beta_{kj}^{(1)}(x)$, integrating over x and using (7.3), we obtain

$$\frac{1}{h^2}(u_{l-1} - 2u_l + u_{l+1}) = -C\beta_{lj}^{(1)}(x_0), \tag{7.26}$$

which is the same approximation as in [12]. Since $u(k) = u_k$ in (7.24), we may consider (7.26) as a finite-difference scheme. The r.h.s. of (7.26) is the discrete δ-function used in [8].

Higher order extensions are obtained if we simply use higher order B-splines in (7.24), i.e.,

$$u(x) = \sum_{k\in\mathbf{Z}} u_k \beta_{kj}^{(m)}(x). \tag{7.27}$$

Substituting (7.27) into the partial differential equation and using dual of the central B-splines to project the singular functions, we obtain a finite system of equations, as in our example. After the coefficients u_k are determined, the solution may be evaluated via (7.27) by taking advantage of the compact support of B-splines.

To summarize, our approach to approximating singular

TABLE 4
Timing and Accuracy for Example VIII.2 (1225 Rectangles)

N	T_p	T_{FFT}	T_m	T_{tot}	Error E_∞	FFT	T_{dir}
64	0.40	0.50	0.01	0.91	5.4e-15	0.09	296
128	0.62	2.55	0.05	3.21	1.4e-15	0.44	1,095
256	1.26	11.45	0.20	12.91	1.0e-15	2.55	4,756
512	3.60	52.22	0.81	56.63	8.3e-16	11.45	20,605

TABLE 6
Timing for Example VIII.4 (160,000 Rectangles)

N	T_p	T_{FFT}	T_m	T_{tot} (sec)	T_{dir} in hours (est.)
64	33.20	0.50	0.01	33.71	11 h
128	34.13	2.55	0.05	36.73	44 h
256	36.93	11.45	0.21	48.59	176 h
512	44.39	52.22	0.82	93.43	704 h

functions by constructing their projections on a subspace of MRA appears to be applicable to the Immersed Interface Method where it provides higher order extensions. Although our considerations in this section address functions of one variable, the extension to multiple dimensions is straightforward. Further work in this direction is required and results will be reported elsewhere.

VIII. NUMERICAL EXPERIMENTS

The algorithms developed in Sections IV, V, and VI have been implemented and numerical experiments have been carried out for several examples. In the examples below we run programs written in FORTRAN 77 on a SPARC-10 workstation using the Pro-Fortran v3.0 compiler with −O3 optimization option under the Solaris 2.3 operating system.

In the first series of examples we evaluate the integral (1.2) for $-N + 1 \leq n, n' \leq N$, where the function f is a piecewise constant function constructed as a linear combination of characteristic functions of rectangles randomly positioned within the unit square. In all examples the oversampling factor $\nu = 2$. We denote by T_p the run time for the projection step of the algorithm where we compute the integrals in (3.1). T_{FFT} denotes the run time for the FFT step, and T_m for the modification step of the algorithm (multiplying by the factor $(La^{(m)}(l/L))^{-1/2}$ in (3.18)). The error E_∞ is the maximal absolute error among all computed frequencies obtained by comparing the output of the algorithm with the output of the direct evaluation. T_{tot} denotes the total run time, $T_{\text{tot}} = T_p + T_{\text{FFT}} + T_m$.

The direct algorithm for rectangular subdomains consists of evaluating $\hat{f}(n, n') = \sum_j \hat{f}_j(n, n')$ as a sum of contributions from each rectangle $[a_j, b_j] \times [c_j, d_j]$, where

TABLE 8
Timing for the One-Dimensional Double Precision USFFT Algorithm with the Time of Initialization Estimated Separately

N_p	Alg. init.	Alg. eval.	T_{tot}	FFT
2048	0.05	0.08	0.13	0.03
4096	0.10	0.17	0.27	0.05
8192	0.19	0.34	0.53	0.11
16384	0.38	0.76	1.16	0.26
32768	0.75	1.82	2.57	0.62

$$\hat{f}_j(n, n') = \left(\frac{e^{-2\pi i n b_j} - e^{-2\pi i n a_j}}{-2\pi i n} \right) \times \left(\frac{e^{-2\pi i n' d_j} - e^{-2\pi i n' c_j}}{-2\pi i n'} \right), \quad (8.1)$$

for $j = 1, \ldots, N_D$ and $-N + 1 \leq n, n' \leq N$. T_{dir} denotes the run time required for the direct computation.

As a point of reference in some tables we display the run time of the ordinary FFT for same number of evaluated frequences.

EXAMPLE VIII.1. We compute the Fourier transform of the function f which is a complex constant and has a rectangle of area approximately 0.64 as its support. We compare the run time of our algorithm with that of the direct algorithm for $N = 2^n, n = 6, 7, 8, 9$ and report the results in Table 3. We observe that the direct algorithm for a single rectangle is faster only by a factor of approximately 3.

EXAMPLE VIII.2. In this example we consider f to be a linear combination of characteristic functions of a pseudo-random combination of 1225 rectangles of total area of approximately 0.64 and perimeter of approximately 112. The number of rectangles in this example is chosen to provide a comparison with Example 2 of [15]. Each rectangle was projected separately. The results are shown in Table 4. We compare the run time and accuracy with that of the direct algorithm. We note that in this example our algorithm is dominated by the FFT step.

EXAMPLE VIII.3. In this example f is a linear combination of characteristic functions of a pseudo-random combi-

TABLE 7
Timing and Accuracy for the One-Dimensional Double Precision USFFT Algorithm

N_p	T_p	T_{FFT}	T_m	T_{tot}	Error E_∞	Error E_2	FFT
2048	0.05	0.05	0.01	0.12	7.0e-14	1.2e-13	0.03
4096	0.10	0.11	0.02	0.24	1.1e-13	2.4e-13	0.05
8192	0.19	0.26	0.04	0.49	1.5e-13	5.0e-13	0.11
16384	0.38	0.62	0.09	1.09	2.7e-13	1.0e-12	0.26
32768	0.77	1.44	0.19	2.40	4.2e-13	2.0e-12	0.62

TABLE 9
Timing and Accuracy for the One-Dimensional Single Precision USFFT Algorithm

N_p	T_p	T_{FFT}	T_m	T_{tot}	E_∞	E_2	FFT	E_∞^{dir}	E_2^{dir}
2048	0.010	0.023	0.006	0.039	3.1e-5	4.4e-5	0.011	4.3e-5	8.2e-5
4096	0.020	0.049	0.012	0.081	3.6e-5	8.7e-5	0.023	7.4e-5	1.6e-4
8192	0.040	0.102	0.024	0.166	5.4e-5	1.8e-4	0.049	1.2e-4	3.5e-4
16384	0.082	0.274	0.048	0.404	9.8e-5	3.5e-4	0.102	1.7e-4	6.9e-4
32768	0.162	0.640	0.096	0.898	1.2e-4	7.0e-4	0.274	4.7e-4	1.3e-3

nation of 40,000 rectangles of total area of approximately 0.64 and perimeter of approximately 640. The results are shown in Table 5.

EXAMPLE VIII.4. In this example f is a linear combination of characteristic functions of a pseudo-random combination of 160,000 rectangles of total area of approximately 0.64 and perimeter of approximately 1280. The results are shown in Table 6. As individual rectangles become smaller, we observe that the first step of the algorithm does not significantly depend on the number of frequencies.

The next series of examples illustrates the performance of the USFFT algorithms in dimensions one and two. In these examples the number of frequencies calculated within the accuracy ϵ is fixed $N = N_p$, where N_p is the number of points. The order of the B-splines was chosen so that the accuracy ϵ corresponds to the double and single precision.

EXAMPLE VIII.5. We evaluate the performance and the accuracy of the double and single precision one-dimensional USFFT. The accuracy of the algorithm is compared with that of the direct evaluation and the errors are computed as

$$E_\infty = \max_{-N/2+1\le n\le N/2} |\hat{f}(n) - \hat{f}_{\mathrm{dir}}(n)| / \max_{-N/2+1\le n\le N/2} |\hat{f}_{\mathrm{dir}}(n)|,$$

and

$$E_2 = \sqrt{\sum_{-N/2+1\le n\le N/2} |\hat{f}(n) - \hat{f}_{\mathrm{dir}}(n)|^2 / \sum_{-N/2+1\le n\le N/2} |\hat{f}_{\mathrm{dir}}(n)|^2},$$

where $\hat{f}(n)$ is output of the algorithm and $\hat{f}_{\mathrm{dir}}$ is the result of the direct computation. In Tables 7–10 N_p denotes the number of points randomly distributed in [0, 1]. The steps of the algorithm are timed separately and the results are shown in Table 7. For the double precision algorithm the order of the B-splines is $m = 27$ and for the single precision algorithm $m = 17$. In order to provide a point of reference we also display run times of the FFT for the same number of frequences. A typical error curve is shown in Fig. 3.

If it is necessary to apply the USFFT algorithm several times but the location of points does not change, then the algorithm may be split into the initialization and the evaluation steps as in [9]. Successive evaluations require only a single initialization. We display timing results for the initialization and the evaluation steps separately in Table 8. In Table 9 we report the accuracy and the run time for the single precision USFFT. The steps of the algorithm are timed separately and the accuracy is compared with that of the double precision computation. For comparison we provide the run time of the FFT algorithm and the accuracy of the direct single precision evaluation. In Table 10 we display the run time for the initialization and evaluation steps separately.

EXAMPLE VIII.6. We evaluate the performance and the accuracy of the double and single precision two-dimensional USFFT. The accuracy of the algorithm is compared with that of the direct evaluation in Tables 11 and 12. The points are chosen at pseudo-random locations within the square $[0, 1] \times [0, 1]$. The errors are computed as

$$E_\infty = \max_{-N/2\pm 1\le n,n'\le N/2} |\hat{f}(n,n') - \hat{f}_{\mathrm{dir}}(n,n')| / \max_{-N/2+1\le n,n'\le N/2} |\hat{f}_{\mathrm{dir}}(n,n')|,$$

and

$$E_2 = \Big(\sum_{-N/2+1\le n,n'\le N/2} |\hat{f}(n,n') - \hat{f}_{\mathrm{dir}}(n,n')|^2 / \sum_{-N/2+1\le n,n'\le N/2} |\hat{f}_{\mathrm{dir}}(n,n')|^2 \Big)^{1/2},$$

TABLE 10
Timing for the One-Dimensional Single Precision USFFT Algorithm with the Time of Initialization Estimated Separately

N_p	Alg. init.	Alg. eval.	T_{tot}	FFT
2048	0.009	0.030	0.039	0.011
4096	0.019	0.062	0.081	0.023
8192	0.038	0.142	0.180	0.049
16384	0.076	0.304	0.380	0.102
32768	0.144	0.743	0.887	0.274

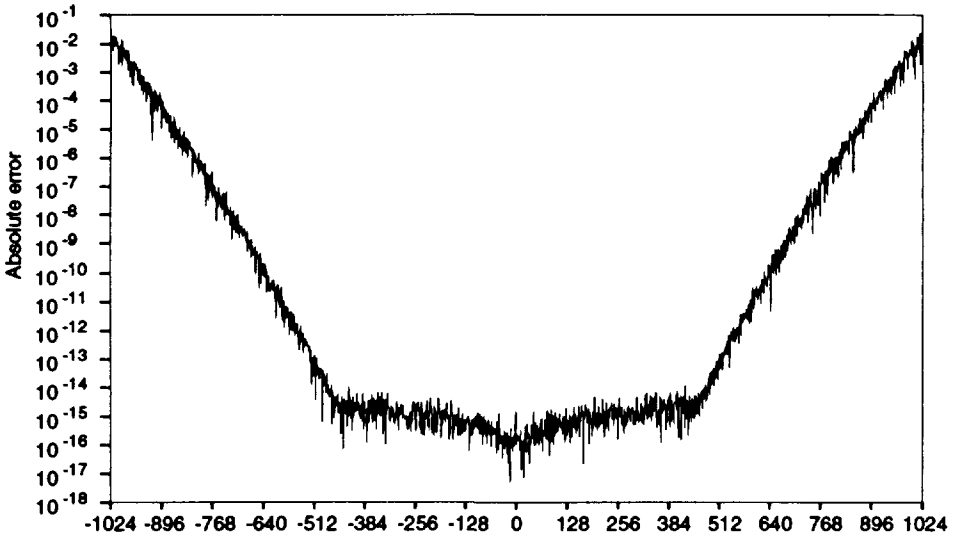

FIG. 3. Behavior of E_∞ error of computing the USFFT in the one-dimensional case. Here the oversampling factor is $\nu = 2$.

where $\hat{f}(n,n')$ is output of the algorithm and $\hat{f}_{\text{dir}}$ is the result of the direct computation. We display run times of the double precision algorithm in Table 13 and of the single precision algorithm in Table 14.

IX. GENERALIZATIONS AND CONCLUSIONS

Our algorithm may be viewed as a procedure for bandlimiting a function by projecting it on spaces of piecewise polynomial splines. As we have demonstrated, such a projection contains enough information to evaluate frequencies within a certain range. The range of frequencies depends on the required accuracy and the order of the spline. Although for this paper we have implemented the algorithm only for functions which are linear combinations of characteristic functions of rectangular domains or δ-functions, the extension of the algorithm to include a greater variety of functions with singularities is straightforward.

The extension of the algorithm to functions with more than two independent variables is also straightforward. The number of operations required for the projection step of the algorithm in the case of d independent variables is proportional to $m^d N_p$, where m is the order of the spline and N_p is the number of elements being projected.

In the following series of remarks we discuss several variants of the algorithm and its applications.

1. *Multiresolution computation of the Fourier Transform.* Using the projection on subspaces of MRA as a procedure for bandlimiting functions opens an opportunity to compute integrals in (1.2) or (2.1) for frequency intervals not only around zero but also for frequency bands separated from zero, e.g., $N \leq n \leq 3N$. Let us briefly outline such algorithm. The first step is a projection of the function on an appropriate subspace $\mathbf{W}_j$ defined as a complement of $\mathbf{V}_j$ in $\mathbf{V}_{j-1}$. Since the scale $j < 0$ must be chosen so that the projection of functions in $\mathbf{W}_j$ are localized within the frequency band of interest, this may imply a very large vector of coefficients (approximately $\approx 2^{-j}$ coefficients) for high frequencies. The important observation here is that such a vector is sparse. The coefficients of the projection are then shifted in the frequency domain towards the zero frequency by multiplying the coefficients in the original domain by an appropriate exponential factor. The next step consists of resampling the coefficients to a coarser "grid" which may be accomplished by fast interpolation as described in [9] or

TABLE 11
Accuracy Test for the Double Precision Two-Dimensional USFFT

N_{freq}	Error E_∞	Error E_2
$2^7 \times 2^7$	5.0×10^{-14}	1.7×10^{-14}
$2^8 \times 2^8$	7.7×10^{-14}	2.7×10^{-14}
$2^9 \times 2^9$	1.1×10^{-13}	5.0×10^{-14}
$2^{10} \times 2^{10}$	2.1×10^{-13}	9.7×10^{-14}

TABLE 12
Accuracy Test for the Single Precision Two-Dimensional USFFT

N_{freq}	Error E_∞	Error E_2
$2^7 \times 2^7$	1.1×10^{-5}	6.2×10^{-6}
$2^8 \times 2^8$	2.0×10^{-5}	1.2×10^{-5}
$2^9 \times 2^9$	3.9×10^{-5}	2.5×10^{-5}
$2^{10} \times 2^{10}$	7.9×10^{-5}	4.9×10^{-5}

via an algorithm currently being developed [4]. Finally, the FFT step followed by a modification step are applied.

2. *Applications to solving PDEs.* In this paper the projection step was mostly used as a part of the algorithm to compute the Fourier transform. However, once such projection is generated, there are several applications where the projection may be useful by itself. For example, let us consider a variant of the algorithm which is based on the expansion of the function into the wavelet basis, i.e. projection on subspaces $\mathbf{W}_j$ rather than $\mathbf{V}_j$. In this case the Fourier transform may be computed within each subspace (subband) separately. In fact, once such a decomposition is obtained, a variety of operators represented in the wavelet basis may be applied efficiently. The accuracy of such a computation is assured by Theorem III.1 of this paper.

For example, let us briefly consider Poisson's equation,

$$\Delta u = f \tag{9.1}$$

on a unit square with zero boundary conditions. Let us assume that function f is discontinuous and satisfies the same assumptions as in (1.2). Also, let us for simplicity assume that f is zero in a strip near the boundary. Since our algorithm generates $\hat{f}(m,n)$ for $-N/4 \leq m,n \leq N/4$ with accuracy ϵ, it is easy to compute the solution u for which $\hat{u}(m,n)$ is accurate to within the same tolerance. In fact (since we divide by m^2+n^2) the band of frequencies where $\hat{u}(m,n)$ is ϵ-accurate is wider than that for the function f. The observation that we would like to make is that in order to solve (9.1), we may proceed by representing the Green's function in the wavelet system of coordinates in the nonstandard form [3] and apply the operator directly to the wavelet representation of f. This will result in an adaptive algorithm with the number of operations proportional to the complexity of the wavelet representation of f.

3. *Extension of functions.* Let us consider an application of our approach to the extension of a function f from a complicated domain to a larger rectangular domain where we require that the extended function is zero in a strip near the boundary of the rectangle and coincides with f inside the original domain. Let us outline the algorithm for such extension. First we extend the function f into a δ-strip around the original boundary, where the distance δ may be arbitrarily small. Since δ dictates the choice of the scale $j<0$ for further projection into the MRA, we would like δ to be as large as our extension method into a δ-strip permits. One such extension method is a local expansion of the function. Given δ we choose the scale $j<0$ so that the linear dimension of the support of B-splines is less than δ. In this case the edge of the locally extended support of the function does not affect the original support of f as we compute the projection of f on a subspace of the MRA. The projection of the δ-extension of the function f on $\mathbf{V}_j$ is the extension that we are seeking.

4. *Problems of nondestructive evaluation.* The USFFT algorithm may be used as an effective tool in problems of non-destructive evaluation. The problems of X-ray tomography, ultrasound imaging, seismics, and synthetic aperture radar among others, reduce to the evaluation of trigonometric sums as in (5.7). In these problems the discretization in the Fourier domain is dictated by the experimental configuration and usually is not equally spaced. Clearly, the FFT algorithm is not directly applicable and a variety of methods have been developed to circumvent the problem. We note that by developing quadrature formulas to evaluate the inverse Fourier transform on unequally spaced grids, we obtain new algorithms for problems of nondestructive evaluation. We will consider problems of nondestructive evaluation in a separate paper.

TABLE 13
Timing for the Double Precision Two-Dimensional USFFT Algorithm

N_p	T_p	T_{FFT}	T_m	T_{tot}	FFT
$N_p = 2^{14}$	3.62	0.44	0.01	4.07	0.09
$N_p = 2^{16}$	13.11	2.55	0.05	15.71	0.44
$N_p = 2^{18}$	53.37	11.45	0.21	65.03	2.55
$N_p = 2^{20}$	218.84	52.22	0.83	271.89	11.45

Note. Splines of order $m = 27$ were used to achieve accuracy $\approx 10^{-13}$ for $N_{\text{freq}} = N_p$.

TABLE 14
Timing for the Single Precision Two-Dimensional USFFT Algorithm

N_p	T_p	T_{FFT}	T_m	T_{tot}	FFT
$N_p = 2^{14}$	0.60	0.18	0.005	0.79	0.05
$N_p = 2^{16}$	2.41	1.14	0.025	3.58	0.18
$N_p = 2^{18}$	9.99	5.11	0.096	15.20	1.14
$N_p = 2^{20}$	38.10	22.22	0.400	60.72	5.11

Note. Splines of order $m = 17$ were used to achieve accuracy $\approx 10^{-6}$ for $N_{\text{freq}} = N_p$.

X. APPENDIX A

Proof of Theorem III.1. We have

$$\left| 2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}} - \hat{f}(2^{-j}\xi) \right| \leq \left| 2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}} - 2^{j/2}\frac{F(\xi)}{\hat{\beta}^{(m)}(\xi)} \right| + \left| 2^{j/2}\frac{F(\xi)}{\hat{\beta}^{(m)}(\xi)} - \hat{f}(2^{-j}\xi) \right|, \tag{10.1}$$

where

$$\left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}-2^{j/2}\frac{F(\xi)}{\hat{\beta}^{(m)}(\xi)}\right| \leqslant \left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}\right| \times \left|1-\frac{\sqrt{a^{(m)}(\xi)}}{\hat{\beta}^{(m)}(\xi)}\right| = \left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}\right| \frac{1-\hat{\varphi}^{(m)}(\xi)}{\hat{\varphi}^{(m)}(\xi)} \quad (10.2)$$

since

$$\hat{\varphi}^{(m)}(\xi)=\frac{\hat{\beta}^{(m)}(\xi)}{\sqrt{a^{(m)}(\xi)}} \quad (10.3)$$

and $\hat{\varphi}^{(m)}(\xi) \leqslant 1$. Also, we have

$$\left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}\right| \leqslant \left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}-\hat{f}(2^{-j}\xi)\right| + |\hat{f}(2^{-j}\xi)|. \quad (10.4)$$

Combining (10.1), (10.2) and (10.4), we arrive at

$$\left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}-\hat{f}(2^{-j}\xi)\right| \leqslant \frac{1-\hat{\varphi}^{(m)}(\xi)}{2\hat{\varphi}^{(m)}(\xi)-1}|\hat{f}(2^{-j}\xi)| + \frac{\hat{\varphi}^{(m)}(\xi)}{2\hat{\varphi}^{(m)}(\xi)-1}\left|2^{j/2}\frac{F(\xi)}{\hat{\beta}^{(m)}(\xi)}-\hat{f}(2^{-j}\xi)\right|. \quad (10.5)$$

Introducing

$$C_{\hat{f}}(l,\alpha)=\sup_{|\xi|\leqslant\alpha}|\hat{f}(2^{-j}(\xi+l))|, \quad l=0,\pm1,\pm2\ldots \quad (10.6)$$

we have

$$\left|2^{j/2}\frac{F(\xi)}{\hat{\beta}^{(m)}(\xi)}-\hat{f}(2^{-j}\xi)\right| \leqslant \sum_{l=\pm1,\pm2\ldots} C_{\hat{f}}(l,\alpha)\tau_l(\xi), \quad (10.7)$$

where

$$\tau_l(\xi)=\left(\frac{\xi}{\xi+l}\right)^{m+1}. \quad (10.8)$$

Using Lemma XI.1 of Appendix B,

$$\hat{\varphi}^{(m)}(\alpha)=\inf_{|\xi|\leqslant\alpha}|\hat{\varphi}^{(m)}(\xi)|, \quad (10.9)$$

then we obtain from (10.5)

$$\left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}-\hat{f}(2^{-j}\xi)\right| \leqslant \frac{1-\hat{\varphi}^{(m)}(\alpha)}{2\hat{\varphi}^{(m)}(\alpha)-1}|\hat{f}(2^{-j}\xi)| + \frac{1}{2\hat{\varphi}^{(m)}(\alpha)-1}\sum_{l=\pm1,\pm2\ldots} C_{\hat{f}}(l,\alpha)\tau_l(\xi). \quad (10.10)$$

Using explicit expression for τ_l (10.8), we have

$$\sup_{|\xi|\leqslant\alpha}\tau_l(\xi)=\left(\frac{\alpha}{|l|-\alpha}\right)^{m+1}, \quad (10.11)$$

and estimate

$$E_\infty=\sup_{|\xi|\leqslant\alpha}\left|2^{j/2}\frac{F(\xi)}{\sqrt{a^{(m)}(\xi)}}-\hat{f}(2^{-j}\xi)\right| \Big/ \sup_{|\xi|\leqslant\alpha}|\hat{f}(2^{-j}\xi)| \quad (10.12)$$

to obtain

$$E_\infty \leqslant \frac{1}{2\hat{\varphi}^{(m)}(\alpha)-1}\left[1-\hat{\varphi}^{(m)}(\alpha)+\frac{1}{C_{\hat{f}}(0,\alpha)} \times \sum_{l=\pm1,\pm2\ldots} C_{\hat{f}}(l,\alpha)\left(\frac{\alpha}{|l|-\alpha}\right)^{m+1}\right]. \quad (10.13)$$

In order to verify that the sum in (10.13) is finite, we use condition (3.7) of Theorem III.1. Since by choosing m sufficiently large the sum in (10.13) may be made arbitrarily small, we obtain the estimate (3.10).

XI. APPENDIX B

LEMMA XI.1. *The Battle–Lemarié scaling function* $\hat{\varphi}^{(m)}(\xi)$ *is strictly monotone for* $\xi \in (0,1)$ *and* $\xi \in (-1,0)$ *and*

$$\sup_{|\xi|\leqslant\alpha}|1-\hat{\varphi}^{(m)}(\xi)|=1-\hat{\varphi}^{(m)}(\alpha), \quad (11.1)$$

for $\alpha < 1$.

Proof. Let us compute the derivative of $(d/d\xi)\hat{\varphi}^{(m)}(\xi)$ for $\xi \in (-1,1)$. We have from (3.4)

$$\log\hat{\varphi}^{(m)}(\xi)=\log\left(\frac{\sin\pi\xi}{\pi\xi}\right)^{m+1}-\frac{1}{2}\log a^{(m)}(\xi). \quad (11.2)$$

It is sufficient to consider $\xi \in (0,1)$ since $\hat{\varphi}^{(m)}(\xi)$ is symmetric around $\xi = 0$. We obtain from (3.3)

$$\log a^{(m)}(\xi) = \log\left(\frac{\sin \pi\xi}{\pi}\right)^{2m+2} + \log \sum_{l=-\infty}^{l=\infty} \frac{1}{(\xi + l)^{2m+2}}, \quad (11.3)$$

and, thus, arrive at

$$\log \hat{\varphi}^{(m)}(\xi) = -\frac{1}{2} \log \tau(\xi), \quad (11.4)$$

where

$$\tau(\xi) = \sum_{l=\pm 1, \pm 2, \ldots} \frac{1}{(\xi + l)^{2m+2}}. \quad (11.5)$$

We have

$$\frac{d}{d\xi}\hat{\varphi}^{(m)}(\xi) = -\frac{\hat{\varphi}^{(m)}(\xi)}{2\tau(\xi)} \frac{d}{d\xi}\tau(\xi), \quad (11.6)$$

and, therefore, the sign of $(d/d\xi)\hat{\varphi}^{(m)}(\xi)$ is opposite to that of $(d/d\xi)\tau(\xi)$ since $\hat{\varphi}^{(m)}, \tau > 0$ for $\xi \in (0, 1)$.

From (11.5) we obtain

$$-\frac{1}{2m+2}\frac{d}{d\xi}\tau(\xi) = \sum_{l=\pm 1, \pm 2, \ldots} \frac{1}{(\xi + l)^{2m+3}}, \quad (11.7)$$

or

$$-\frac{1}{2m+2}\frac{d}{d\xi}\tau(\xi) = \sum_{l=1}^{l=\infty} \frac{(\xi + l)^{2m+3} + (\xi - l)^{2m+3}}{(\xi^2 - l^2)^{2m+3}}. \quad (11.8)$$

Since

$$(\xi + l)^{2m+3} + (\xi - l)^{2m+3} = 2\xi \sum_{k=0}^{k=m+1} \binom{2m+3}{2k} \times l^{2k} \xi^{2(m-k)+2} \quad (11.9)$$

the numerator in (11.8) is positive for $\xi \in (0, 1)$. The denominator of each term of the series in (11.8) is negative for $\xi \in (0, 1)$ and, thus, we conclude that the derivative $(d/d\xi)\hat{\varphi}^{(m)}(\xi)$ is negative for $\xi \in (0, 1)$.

XII. APPENDIX C

LEMMA XII.1. *The Battle–Lemarié scaling function $\varphi^{(m)}$ satisfies the identity*

$$\sum_{k\in\mathbf{Z}} (x - k)^l \varphi^{(m)}(x - k) = \delta_{0,l} \quad 0 \leq l \leq m. \quad (12.1)$$

Proof. We have

$$\int_{-\infty}^{\infty} \sum_{k\in\mathbf{Z}} (x - k)^l \varphi^{(m)}(x - k) e^{-2\pi i x\xi}\, dx = \left(\sum_{k\in\mathbf{Z}} e^{-2\pi i k\xi}\right) \frac{1}{(-2\pi i)^l} \frac{d^l}{d\xi^l}\hat{\varphi}^{(m)}(\xi). \quad (12.2)$$

Using Poisson's summation formula $\sum_{k\in\mathbf{Z}} \delta(\xi - k) = \sum_{k\in\mathbf{Z}} e^{-2\pi i k\xi}$, we obtain Lemma XII.1 from (12.2) and (3.4).

XIII. APPENDIX D

Let us repeat the considerations of Section II using the periodic extension $f(x + 1) = f(x)$ of the function f defined on $[0, 1]$. Let $N = 2^n$ and consider coefficients

$$f_k = \sqrt{N} \int_{-\infty}^{\infty} f(x)\phi(Nx - k)\, dx, \quad (13.1)$$

for $0 \leq k \leq N - 1$. Changing the domain of integration in (13.1) to $[0, 1]$, we obtain

$$f_k = \sqrt{N} \int_0^1 f(x) \sum_{l\in\mathbf{Z}} \phi(Nx - k - Nl)\, dx, \quad (13.2)$$

Using Poisson's summation formula, we have

$$\sum_{l\in\mathbf{Z}} \phi(Nx - k - Nl) = \frac{1}{N} \sum_{l\in\mathbf{Z}} e^{2\pi i(k - Nx)l/N} \bar{\phi}(l/N), \quad (13.3)$$

and, therefore,

$$f_k = \frac{1}{\sqrt{N}} \sum_{l\in\mathbf{Z}} \bar{\phi}(l/N)\hat{f}(l) e^{2\pi i k l/N} \quad (13.4)$$

where

$$\hat{f}(l) = \int_0^1 f(x) e^{-2\pi i l x}\, dx. \quad (13.5)$$

In order to find an approximation to $\hat{f}(l)$, we rewrite (13.4) by splitting the sum into segments of length N,

$$f_k = \frac{1}{\sqrt{N}} \times \sum_{l=0}^{N-1} \left(\sum_{m\in\mathbf{Z}} \bar{\phi}(l/N + m)\hat{f}(l + Nm)\right) e^{2\pi i k l/N}. \quad (13.6)$$

Using discrete Fourier transform, we obtain

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} f_k e^{-2\pi ikl/N} = \bar{\hat{\phi}}(l/N)\hat{f}(l) + \sum_{m=\pm 1,\pm 2,\ldots} \bar{\hat{\phi}}(l/N+m)\hat{f}(l+Nm) \quad (13.7)$$

and seek approximation

$$\left| \frac{1}{\sqrt{N}\bar{\hat{\phi}}(l/N)} \sum_{k=0}^{N-1} f_k e^{-2\pi ikl/N} - \hat{f}(l) \right| \leqslant \sum_{m=\pm 1,\pm 2,\ldots} \frac{|\hat{\phi}(l/N+m)|}{|\hat{\phi}(l/N)|} |\hat{f}(l+Nm)|. \quad (13.8)$$

From this point on we may proceed by repeating the considerations in Section II and Section III. By choosing ϕ to be the mth order B-spline, we obtain an analog of Theorem III.1 of Section III. Although the estimates are the same, the difference between using the 1-periodic extension of the function f and the extension of f to zero shows in the definition of the coefficients f_k via (13.1) or (2.4). In (13.1) we do not have to introduce an additional assumption (for the discrete algorithm) that the function f is zero in a narrow strip near the boundary. For this reason we prefer to use (13.1) in our implementations.

Using orthogonalization procedure, let us rewrite (13.7) for the central B-splines of order m. We have

$$\frac{1}{\sqrt{N}\sqrt{a^{(m)}(l/N)}} \sum_{k=0}^{N-1} f_k e^{-2\pi ikl/N} = \hat{\varphi}^{(m)}(l/N)\hat{f}(l) + \sum_{k=\pm 1,\pm 2,\ldots} \hat{\varphi}^{(m)}(l/N+k)\hat{f}(l+Nk), \quad (13.9)$$

where $\hat{\varphi}^{(m)}$ is the Fourier transform of the Battle–Lemarié scaling function. Formula (13.9) should be compared with (3.16). Using (13.9) we arrive at the same conclusions as those stated in Theorem III.1. The only difference is in computing coefficients f_k via (13.1) where f is extended periodically.

REFERENCES

1. G. Battle, A block spin construction of ondelettes, Part I, Lemarié functions. *Comm. Math. Phys.* **110** (1987), 601–615.
2. G. Beylkin, On the representation of operators in bases of compactly supported wavelets, *SIAM J. Numer. Anal.* **29**, No. 6 (1992), 1716–1740.
3. G. Beylkin, R. R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I, *Comm. Pure Appl. Math.* **44** (1991) 141–183. Yale University Technical Report YALEU/DCS/RR-696, August 1989.
4. G. Beylkin and R. Cramer, manuscript in preparation.
5. A. Brandt, Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Comput. Phys. Comm.* **65** (1991), 24–38.
6. C. Chui, "*An Introduction to Wavelets*," Academic Press, San Diego, 1992.
7. J. W. Cooley and J. W. Tukey, An algorithm for the machine computation of complex Fourier series, *Math. Comp.* **19** (1965), 297–301.
8. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25** (1977), 220–252.
9. A. Dutt and V. Rokhlin, Fast Fourier Transform for nonequispaced data, *SIAM J. Sci. Stat. Comp.* (1993).
10. P. G. Lemarié, Ondelettes à localisation exponentielles, *J. Math. Pures Appl.* **67** (1988), 227–236.
11. W. H. Press and G. B. Rybicki, Fast Algorithm for spectral analysis of unevenly sampled data, *Astrophys. J.* **338** (1989), 227–280.
12. R. P. Beyer and R. J. LeVeque, Analysis of a one-dimensional model for the immersed boundary method, *SIAM J. Num. Anal.* **29** (1992), 332–364.
13. R. R. Coifman, *personal communication.*
14. I. J. Schoenberg, "*Cardinal Spline Interpolation,*" SIAM, 1973. CBMS NSF Series in Applied Math. Vol. 12.
15. E. Sorets, "Fast Fourier Transform of Piecewise Constant Functions," 1993. Yale University Research Report, YALEU/DCS/RR-986.
16. J. O. Stromberg, A Modified Franklin system and Higher-Order Spline Systems on $\mathbf{R}^n$ as Unconditional Bases for Hardy Spaces, *in* "Conference in Harmonic Analysis in Honor of Antoni Zygmund," Wadworth Math. Series, pp. 475–493, 1983.
17. T. D. Sullivan, A technique for convolving unequally spaced samples using Fast Fourier Transforms, 1990, Sandia Report.
18. M. Unser and A. Aldroubi, Polynomial splines and wavelets—a signal processing perspective, *in* "Wavelets: A Tutorial in Theory and Applications" (C. Chui, Ed.), pp. 91–122. Academic Press, San Diego, 1992.