ADVANCES IN APPLIED MATHEMATICS 15, 202–250 (1994)

Computing Fourier Transforms and Convolutions on the 2-Sphere*,†

JAMES R. DRISCOLL AND DENNIS M. HEALY, JR.

Department of Mathematics and Computer Science, Dartmouth College, Hanover, New Hampshire 03755

This paper considers the problem of efficient computation of the spherical harmonic expansion, or Fourier transform, of functions defined on the two dimensional sphere, S^2. The resulting algorithms are applied to the efficient computation of convolutions of functions on the sphere. We begin by proving convolution theorems generalizing well known and useful results from the abelian case. These convolution theorems are then used to develop a sampling theorem on the sphere, which reduces the calculation of Fourier transforms and convolutions of band-limited functions to discrete computations. We show how to perform these efficiently, starting with an $O(n(\log n)^2)$ time algorithm for computing the Legendre transform of a function defined on the interval $[-1, 1]$ sampled at n points there. Theoretical and experimental results on the effects of finite precision arithmetic are presented. The Legendre transform algorithm is then generalized to obtain an algorithm for the Fourier transform, requiring $O(n(\log n)^2)$ time, and an algorithm for its inverse in $O(n^{1.5})$ time, where n is the number of points on the sphere at which the function is sampled. This improves the naive $O(n^2)$ bound, which is the best previously known. These transforms give an $O(n^{1.5})$ algorithm for convolving two functions on the sphere.

1. INTRODUCTION

The Fourier expansion of a function f on the sphere,

$$f = \sum_{l \in \mathbb{N}} \sum_{|m| \le l} \hat{f}(l, m) Y_l^m,$$

*An extended abstract of this work appears in "Proceedings of the 30th Annual Symposium on the Foundations of Computer Science."

†This research was supported in part by the NSF under Grant CCR-8809573 and by DARPA as administered by the AFOSR under Contract AFOSR-90-0292.

0196-8858/94 $9.00

expresses that function as a linear combination of special functions $\{Y_m^l\}$, known as spherical harmonics. This is an orthogonal decomposition in the Hilbert space $L^2(S^2)$ of functions square integrable with respect to the standard rotation invariant measure $d\omega$ on S^2. In particular, the Fourier coefficients appearing in the expansion are obtained as inner products:

$$\hat{f}(l, m) = \int_{S^2} f \overline{Y}_l^m \, d\omega.$$

This paper presents an algorithm for efficient numerical computation of these Fourier coefficients. This algorithm is exact (in exact arithmetic) for *band-limited* functions; that is, for those functions in $L^2(S^2)$ with only a finite number of nonzero Fourier coefficients. This leads to algorithms for the efficient computation of the convolution of two such functions on S^2. These results represent some of the first asymptotically fast algorithms for nonabelian harmonic analysis.

In the setting of abelian harmonic analysis, it would be difficult to overstate the impact of an algorithm, popularized by Cooley and Tukey [11] in 1965, for the computation of Fourier coefficients of band-limited functions on the circle. This algorithm computes in $O(n \log n)$ time the n Fourier coefficients of such a function from its values (samples) at n distinct points. Despite the fact that the fast Fourier transform has been extensively studied in both the theoretical and applied literature [2, 3, 5, 13, 31, 32, 38, 51], until recently there has been very little done about the computation of Fourier transforms associated with nonabelian groups. The important new work of Diaconis, Rockmore, and others has attacked this issue in the case of finite nonabelian groups, and sketched some very interesting applications [15, 41, 10].

It seems clear that algorithms for the efficient calculation of the Fourier transform for certain continuous nonabelian groups and their homogeneous spaces may also be of interest. The example of this paper fits into this category, as it concerns the action of the rotation group $SO(3)$ on its quotient S^2. In fact, the need for a fast Fourier transform algorithm in this very case has recently been identified for its applications in computational vision [42].

In addition to computer vision, the Fourier transform on S^2 is applied in a variety of fields, including statistics, tomography, geophysics, seismology, atmospheric science, signal processing, and crystallography. See, for example, [14, 16–18, 25, 29, 44]. Much of the applicability of the transform stems from its nice property of simplifying convolutions and correlations. For example, the 2D convolution on the plane, which is efficiently computed by the two-dimensional abelian Fourier transform, is useful in pattern matching. By convolving an image with a pattern, it is possible to

locate translated copies of the pattern in the image. Efficient spherical convolution, which would follow from a fast transform and inverse, should prove useful in a similar way for pattern matching with directional data.

To our knowledge, however, prior to this work there did not exist an algorithm for exact computation of spherical Fourier transforms asymptotically better than the naive $O(n^2)$ algorithm, where n is the number of sampled values of the function. Many approximate methods for harmonic analysis on the sphere already exist; for example, one method approximates the sphere as a torus by identifying the two poles and proceeds as though the problem were framed on the torus. A refinement of this approach then transforms this back to the sphere to get an approximation of the spherical harmonic coefficients [16]. A much more promising development is the recent work of Alpert and Rokhlin [1], which offers fast approximate methods for evaluating Legendre transforms. Orzag [36] gives a fast method for computing Legendre polynomial transforms to a given accuracy by using a WKB asymptotic expansion. In contrast, we develop *exact* and efficient methods for computing the spherical harmonic transforms of band-limited signals by first finding an appropriate sampling of the functions for which we can then efficiently compute a discrete transform. Of course, in order to justify the claim of exactness we must take as our model of computation the RAM with exact real arithmetic. This model in this context is at least historically justified, since a similar model is used for the method of Gauss quadrature that exactly integrates degree bounded polynomials.

There are two immediate obstacles to the efficient computation of spherical harmonics. First, the abelian fast Fourier transform relies on the fact that the sample points at the roots of unity comprise a discrete subgroup on the circle. The discrete subgroups of $SO(3)$, whose orbits on the sphere might provide plausible sampling sets, consist of cyclic and dihedral groups of rotations about one axis and the symmetries of the platonic solids [12]. Unfortunately, the orbits do not fill up the sphere densely enough to get even a reasonable approximation [23]. The second obstacle is that spherical harmonics are products of exponentials in one coordinate and of associated Legendre functions in the other. We know of no prior exact algorithms for these transforms requiring less than $O(n^2)$ time.

This paper presents an $O(n(\log n)^2)$ algorithm that, given a data structure of size $O(N \log N)$, computes the spherical harmonic expansion of a discrete function on the sphere defined on an equiangular grid of $n = 2^k \leq N$ points. This improves the naive $O(n^2)$ bound, which is the best previously known. Additionally, we describe methods to invert this transform in $O(n^{1.5})$ time, and using these transforms, an $O(n^{1.5})$ time algorithm to convolve band-limited functions on the sphere.

The paper is organized as follows. Section 2 reviews the Fourier transform on S^2. Section 3 develops convolution theorems on the sphere, and these are used in Section 4 to prove a sampling theorem for band-limited signals. It is this sampling theorem that allows the exact computation of the Fourier transform of band-limited functions as sums of sampled data. We next turn to the construction of an *efficient* scheme for evaluating the sums. The first step is given in Section 5, which describes an $O(n(\log n)^2)$ time algorithm to compute the Legendre polynomial transform of a function sampled at the n points $\cos(2\pi i/n)$, for $0 \le i < n$ and $n = 2^k \le N$, given a data structure of size $O(N \log N)$. This improves the best previously known bound of $O(n^2)$ [19]. Experimental and theoretical results concerning the effect of finite precision arithmetic on this algorithm are described in Section 6. The algorithm is extended to the associated Legendre polynomials to obtain the full Fourier transform on the sphere in Section 7. Section 8 concludes the paper.

2. Preliminaries

Many problems in physics and engineering possess spherical symmetry, and separation of variables in spherical coordinates reduces these problems in part to the analysis of functions on S^2, the two-dimensional unit sphere (surface of the unit ball) comprised of all unit vectors in three space, $\mathbb{R}^3$. The techniques of Fourier analysis, so familiar and useful for many problems in Euclidean space, are also helpful in the non-Euclidean setting of the sphere. What follows is a brief review of these ideas.

Analysis on the sphere requires the use of coordinates. A familiar choice is the parameterization of the points of S^2 by angles of colatitude and longitude, θ and ϕ, with θ measured down from the z-axis, varying between 0 and π, and ϕ varying between 0 and 2π, measured from the x-axis. Thus a unit vector ω on S^2 may be parameterized as $(\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))$ in these coordinates.

Fourier analysis may be characterized as the systematic use of symmetry to simplify certain linear operators. In the case considered in this paper, the unit sphere admits the special orthogonal group in three variables, $SO(3)$, as a transitive group of symmetries. These are the proper rotations of $\mathbb{R}^3$ about the origin and are characterized as those three-by-three real matrices of determinant one whose inverses are given by their transposes. For example, we have the rotations about the z-axis,

$$\left\{ u(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \middle| \phi \in [0, 2\pi] \right\},$$

or those about the y-axis,

$$\left\{ a(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \middle| \theta \in [0, 2\pi] \right\}.$$

In fact, any rotation g in $SO(3)$ may be written as a product of matrices of these forms in the well known Euler angle decomposition: $g = u(\phi)a(\theta)u(\psi)$ with $\phi \in [0, 2\pi]$, $\theta \in [0, \pi]$, and $\psi \in [0, 2\pi]$ determined uniquely for almost all g [20, 44].

In the present context, the transitivity of the $SO(3)$ action on S^2 means that any point on the sphere may be obtained from any other by a rotation. In particular, the entire sphere is swept out by taking all rotations of the north pole, $\eta = (0, 0, 1)$. It should be noted that the rotation $u(\phi)a(\theta)u(\psi)$ with Euler angles ϕ, θ, and ψ takes the north pole to the point

$$\omega(\theta, \phi) = (\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))$$

with spherical coordinates ϕ and θ. In fact, the sphere is a quotient of the rotation group and inherits its natural coordinate system from that of the group.

Fourier analysis on the sphere amounts to the decomposition of the space of square integrable functions on S^2 into minimal subspaces invariant under all of the rotations in $SO(3)$, thus simplifying the analysis of rotation-invariant operators. The rotations of the sphere induce operators on functions by rotating the graphs over the sphere. Specifically, for each rotation $g \in SO(3)$ we have the operator $\Lambda(g)$ defined on functions on the sphere by

$$\Lambda(g)f(\omega) = f(g^{-1}\omega).$$

The presence of the inverse is required in order that the assignment Λ of rotations to their associated operators on functions respects the group law,

$$\Lambda(g_1 g_2) = \Lambda(g_1)\Lambda(g_2).$$

A vector subspace of functions on the sphere is invariant if all of the operators $\Lambda(g)$ for $g \in SO(3)$ take each function in the space back into the space.

The Hilbert space $L^2(S^2)$ is defined as usual with the inner product $\langle f, h \rangle = \int_{S^2} f\bar{h}\, d\omega$. Here we use the (essentially) unique rotation-invariant

area element on the sphere,

$$\int_{\omega \in S^2} f(\omega)\, d\omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} f(\omega(\theta, \phi)) \sin\theta\, d\theta\, d\phi,$$

in the usual coordinates. Its rotation-invariance,

$$\int_{\omega \in S^2} f(g\omega)\, d\omega = \int_{\omega \in S^2} f(\omega)\, d\omega, \qquad g \in SO(3),$$

follows from the observation that this is the angular part of the polar coordinate decomposition of the (rotation-invariant) Lebesgue measure on $\mathbb{R}^3$. This measure can also be obtained by passing to the quotient with the invariant volume element on $SO(3)$ itself, which may be written $dg = \sin\theta\, d\theta\, d\phi\, d\psi$ in terms of the Euler angle coordinates. We use this measure on $SO(3)$ when we discuss convolution.

It is easy to see that all of the operators $\Lambda(g)$ are unitary on $L^2(S^2)$. The assignment $g \mapsto \Lambda(g)$ is called the left regular representation of $SO(3)$ in $L^2(S^2)$. In this language, the preceding description of Fourier analysis translates as the orthogonal decomposition of the left regular representation into irreducible subrepresentations.

As in the familiar case of periodic functions on the line, or equivalently, functions on the circle, the irreducible subrepresentations are spanned by the restrictions of homogeneous polynomials of a fixed degree which are harmonic in the sense of being annihilated by the Laplace operator. Unlike the circle, the sphere admits a nonabelian group of symmetries, and the invariant subspaces spanned by these harmonic homogeneous polynomials of degree $l \geq 0$ have dimension $2l + 1$, instead of being one-dimensional. The invariant subspace of degree l harmonic polynomials restricted to the sphere is called the space of *spherical harmonics of degree l*. Spherical harmonics of different degrees are orthogonal to one another. Choosing an orthonormal basis of $2l + 1$ spherical harmonics Y_l^m, $-l \leq m \leq l$ for each degree $l \geq 0$ gives an orthonormal basis for all of $L^2(S^2)$.

In coordinates, $Y_l^m(\theta, \phi)$ is

$$(-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}}\, P_l^m(\cos\theta) e^{im\phi}, \tag{1}$$

where the Legendre functions P_l^m are defined as follows. We start with

the polynomials $P_l = P_l^0$,

$$P_l(x) = \frac{1}{2^l l!} D^l\left[(x^2 - 1)^l\right],$$

where D denotes derivative. A suitably normalized version of these, the *Legendre polynomials*,

$$\tilde{P}_l(x) = \sqrt{\frac{2l+1}{2}}\, P_l(x), \tag{2}$$

comprise a familiar orthonormal basis for $L^2[-1, 1]$, obtained from the monomials $1, x, x^2, \ldots$ by the Gram–Schmidt procedure.

The *associated Legendre functions* are then defined by

$$P_l^k(x) = (1 - x^2)^{k/2} D^l P_k^0(x),$$

for $l \geq 0$ and $0 \leq k \leq l$. For $-l \leq k \leq 0$ they are obtained from

$$P_l^{-k}(x) = (-1)^k \frac{(l-k)!}{(l+k)!} P_l^k(x).$$

The effect of this choice is that

$$\overline{Y_l^m} = (-1)^m Y_l^{-m}.$$

The Legendre functions satisfy the recurrence

$$(l - m + 1)P_{l+1}^m(x) - (2l + 1)xP_l^m(x) + (l + m)P_{l-1}^m(x) = 0.$$

This recurrence is known to provide a numerically stable method of computing the values of the Legendre functions and will be essential to the development of the efficient transform algorithm.

The Fourier decomposition of functions on the sphere into these classical spherical harmonics is simply the expansion of a function $f \in L^2(S^2)$ in the orthonormal basis provided by the spherical harmonics

$$f(\theta, \phi) = \sum_{l \in \mathbb{N}} \sum_{|m| \leq l} \hat{f}(l, m) Y_l^m(\theta, \phi) \tag{3}$$

$$\hat{f}(l, m) = \int_{S^2} f \overline{Y}_l^m \, d\omega. \tag{4}$$

Of all the possible bases for $L^2(S^2)$, the spherical harmonics uniquely exploit the symmetries of the sphere. Under a rotation g, each spherical

harmonic of degree l is transformed into a linear combination of only those Y_l^m, $-l \le m \le l$, with the same degree:

$$\Lambda(g)Y_l^m(\omega) = \sum_{|k| \le l} Y_l^k(\omega) D_{k,m}^{(l)}(g).$$

Thus the effect of a rotation on a function expressed in the basis of spherical harmonics is multiplication by a semi-infinite block diagonal matrix, with the $(2l + 1) \times (2l + 1)$ blocks for each $l \ge 0$ given by

$$D^{(l)}(g) = \left(D_{m,n}^{(l)}\right)(g).$$

In technical language, this constitutes the decomposition of the regular representation of $SO(3)$ on $L^2(S^2)$ into irreducible subrepresentations. For future use we note the explicit expression for $D^{(l)}(g)$ when $g = u(\phi)a(\theta)u(\psi)$:

$$D_{m,n}^{(l)}(u(\phi)a(\theta)u(\psi)) = e^{-im\psi} d_{m,n}^{(l)}(\cos\theta) e^{-in\phi},$$

where the $d^{(l)}$ are related to Jacobi polynomials [6, 47]. From this, one may derive the relation between the spherical harmonics and the matrix elements,

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)}{4\pi}}\, \overline{D}_{m,0}^{(l)}(u(\phi)a(\theta)u(\psi)),$$

where the overbar denotes complex conjugation. In particular,

$$d_{m,0}^{(l)}(\cos\theta) = (-1)^m \sqrt{\frac{(l-m)!}{4\pi(l+m)!}}\, P_l^m(\cos\theta).$$

The effect of all this is to block diagonalize rotationally invariant operators; namely, convolution operators obtained as weighted averages of the rotation operators by (generalized) functions or kernels. A well known example is the Laplace–Beltrami operator on smooth functions on S^2,

$$\Delta^* = \frac{1}{\sin\theta}\left(\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) + \frac{1}{\sin\theta}\frac{\partial^2}{\partial\phi^2}\right),$$

which acts diagonally in the spherical harmonic basis, a fact exploited in many problems, including the quantum mechanical analysis of hydrogenic atoms. We now turn to a more detailed analysis of convolution.

3. Convolution on S^2

Each (generalized) function k on the sphere may be used to define a convolution operator. This is accomplished by employing it as a weighting factor for the operators Λ induced by the rotations of the sphere. The operator of left convolution by k is then

$$\begin{aligned}\mathfrak{K}_k f(\omega) &= \left(\int_{g\in SO(3)} dg\, k(g\eta)\Lambda(g)\right) f(\omega) \\ &= \int_{g\in SO(3)} k(g\eta) f(g^{-1}\omega)\, dg \\ &= k * f(\omega).\end{aligned}$$

Here, ω is any point of the sphere and η is the north pole. It is immediate to verify that this sort of definition generalizes the usual one for functions on $\mathbb{R}$, but in this case there is a distinct right convolution obtained by averaging right translations. The analysis of right convolution is analogous to what we do here for left convolution.

Since the operators $\Lambda(g)$ are simultaneously block diagonalized for all $g \in SO(3)$ in the spherical harmonic basis, it follows that the convolution operators obtained as linear combinations of them must also be. This is the true utility of Fourier representations. Here is an explicit statement of this.

Theorem 1. *For functions f, h in $L^2(S^2)$, the transform of the convolution is a pointwise product of the transforms*

$$(f * h)\hat{}\,(l,m) = 2\pi\sqrt{\frac{4\pi}{2l+1}}\,\hat{f}(l,m)\hat{h}(l,0)$$

Proof.

$$(f * h)\hat{}\,(l,m) = \int_{\omega\in S^2} f * h\bar{Y}_l^m(\omega)\, d\omega.$$

Using the definition of convolution, this is

$$\begin{aligned}&\int_{\omega\in S^2}\int_{g\in SO(3)} f(g\eta)h(g^{-1}\omega)\bar{Y}_l^m(\omega)\, dg\, d\omega, \\ &\quad = \int_{SO(3)} f(g\eta)\int_{S^2} h(g^{-1}\omega)\bar{Y}_l^m(\omega)\, d\omega\, dg \\ &\quad = \int_{SO(3)} f(g\eta)\int_{S^2} h(\omega)\bar{Y}_l^m(g\omega)\, d\omega\, dg,\end{aligned}$$

which is

$$\int_{SO(3)} f(g\eta) \int_{S^2} h(\omega) \overline{\sum_{|j| \le l} Y_l^j(\omega) D_{j,m}^{(l)}(g^{-1})}\, d\omega\, dg$$

$$= \sum_{|j| \le l} \int_{SO(3)} f(g\eta) \overline{D_{j,m}^{(l)}(g^{-1})}\, dg\, \hat{h}(l,j).$$

Now observe that the integral is 0 unless $j = 0$. This follows because the measure on $SO(3)$ is invariant; in particular, upon right translation by any rotation about the z-axis, $g \mapsto gu(\psi)$:

$$\int_{SO(3)} f(g\eta) \overline{D_{j,m}^{(l)}(g^{-1})}\, dg$$

$$= \int_{SO(3)} f(gu(\psi)\eta) \overline{D_{j,m}^{(l)}(u(-\psi)g^{-1})}\, dg$$

$$= \int_{SO(3)} f(g\eta) e^{-ij\psi} \overline{D_{j,m}^{(l)}(g^{-1})}\, dg$$

$$= e^{-ij\psi} \int_{SO(3)} f(g\eta) \overline{D_{j,m}^{(l)}(g^{-1})}\, dg.$$

As ψ is arbitrary, this is impossible unless $j = 0$. Because the representation is unitary, $\overline{D_{0,m}^{(l)}(g^{-1})} = D_{m,0}^{(l)}(g)$, so the only nonvanishing integral in the sum is

$$= \int_{SO(3)} f(g\eta) \sqrt{\frac{4\pi}{2l+1}}\, \overline{Y}_l^m(g\eta)\, dg = 2\pi \sqrt{\frac{4\pi}{2l+1}}\, \hat{f}(l,m). \quad \square$$

There is also an important dual convolution result relating pointwise products of functions on S^2 to a "convolution" of their transforms. This follows from the well known Clebsch–Gordon decomposition of tensor products of representations into irreducible representations [6, 47, 44]. This may be thought of as giving the description of the algebraic (semi-ring) structure of the transform domain. As smooth functions on S^2 may be decomposed as rapidly converging sums of spherical harmonics, the description of the transform of a product of such functions follows from the classically known transform of a product of spherical harmonics.

THEOREM 2.

$$Y_{l_1}^{m_1} Y_{l_2}^{m_2} = \sum_{L=|l_1-l_2|}^{l_1+l_2} \sqrt{\frac{(2l_1+1)(2l_2+1)}{4\pi(2L+1)}}\, C_{0,0,0}^{l_1,l_2,L} C_{m_1,m_2,m_1+m_2}^{l_1,l_2,L} Y_L^{m_1+m_2},$$

where the Wigner symbols $C_{\mu_1,\mu_2,\mu}^{\lambda_1,\lambda_2,\lambda}$ *are as defined in* [6] *and* $Y_L^{m_1+m_2}$ *is defined to be* 0 *if* $|m_1 + m_2| > L$.

Proof. The proof of this, the many symmetries and orthogonality properties of the Wigner coefficients, and the recursive techniques for their calculation may be found in standard references [6, 22, 30]. □

4. SAMPLING THEOREM

It is often desirable to sample a band-limited function in such a way that the original function can be exactly recovered from the samples. In the case of functions on the line, the classical sampling theorem was given by Shannon and others and states that a function whose Fourier transform has bounded support may be recovered from its samples provided that these are chosen uniformly with a frequency at least twice the bounding frequency [44]. An important consequence is that the Fourier transform may be computed from the discrete samples.

We desire the ability to sample a band-limited function on the sphere so that the integrals defining the Fourier coefficients can be efficiently evaluated as weighted sums of the samples. The investigation of numerical integration formulas has a long history and there are relevant quadrature results on the sphere, though each has its drawbacks. One, [29], has the unfortunate requirement that the function must be sampled differently for each value of m for which we wish to compute the transform $\hat{f}(l, m)$. This results in the function being significantly oversampled, even in an asymptotic sense. Another, [17], requires that the evaluations of the various $\hat{f}(l, m)$ weight the samples differently, thus precluding a more efficient means of computing the spherical transform. In both cases, only error estimates are determined, and no claim is made that either exactly integrates band-limited signals. Discussions of quadrature on the sphere and some applications may be found in [34, 43].

We, however, exhibit a simple sampling result for a band-limited function that requires an asymptotically optimal number of samples, and which can exactly recover the original function. The main result, given in Theorem 3 below, shows that the transform of a band-limited function can be computed exactly by means of finite sums of the sampled values of that function.

Let $f(\theta, \phi)$ be a band-limited function such that $\hat{f}(l, m) = 0$ for $l \geq b$. We sample the function at the equiangular grid of points (θ_i, ϕ_j), $i = 0, \ldots, 2b - 1$, $j = 0, \ldots, 2b - 1$, where $\theta_i = \pi i/2b$ and $\phi_j = \pi j/b$. The sample points are denser near the poles than at the equator, and we must weight the samples to compensate for this. Define $\{a_k^{(b)}\}$ to be the unique solution vector for the system of equations

$$a_0^{(b)} P_l(\cos(\theta_0)) + a_1^{(b)} P_l(\cos(\theta_1)) + \cdots + a_{2b-1}^{(b)} P_l(\cos(\theta_{2b-1})) = \sqrt{2}\,\delta_{l,0} \tag{5}$$

for $l = 0, \ldots, 2b - 1$. Recall that P_l is the Legendre polynomial, and $\delta_{0,0} = 1$ and $\delta_{m,0} = 0$ for $m \neq 0$. (Existence and uniqueness come down to the well known fact that the Legendre polynomials comprise a Chebychev system [7]. An explicit formula for the weights is given at the end of this section.) Multiply the sampled value of f at (θ_i, ϕ_j) by $a_i^{(b)}$.

The claim is that f can be recovered from these weighted samples and, furthermore, the finitely many nonvanishing Fourier coefficients can be computed in terms of them. The idea is patterned on the standard argument in the case of the line and runs roughly as follows. Sampling is thought of as multiplication of the function f by a "sampling measure" s, a weighted grid of point masses at the sample points on the sphere. It is easy to see that the transform of the measure $f_s = f \cdot s$ is a finite sum involving sampled values of f. On the other hand, if we knew the Fourier coefficients of s, we could determine the transform of f_s a different way, using the dual convolution theorem, Theorem 2, to express $\hat{f}_s$ as a convolution of $\hat{f}$ and $\hat{s}$. In fact, we know that f is band-limited and can show that $\hat{s}(0, 0) = 1$ and $\hat{s}(l, m) = 0$ for $0 < l < 2b$. This allows us to control the blurring caused by the convolution in the transform domain and conclude that $\hat{f}_s(l, m) = \hat{f}(l, m)$ for $l < b$. This gives the desired result, as we already know an expression for $\hat{f}_s$ as a simple sum involving the samples of f.

Although this argument can be understood on an intuitive basis, a rigorous development requires the machinery of distributions on S^2. For background on distributions on manifolds see the standard references [24, 42].

A distribution is a continuous linear functional on the space of test functions, $\mathscr{E}(S^2)$. This is the space of smooth functions, $C^\infty(S^2)$, equipped with a standard Frechet space topology which may be described by the statement that a sequence $f_n \to 0 \Leftrightarrow Df_n \to 0$ uniformly for all differential operators D. Note for example that any integrable function f on S^2 defines a distribution $\phi \mapsto \int_{S^2} f\phi \, d\omega$, for all test functions ϕ. Another example is the point mass measure at a point $\mathbf{x} \in S^2$, $\delta_{\mathbf{x}}$, which simply evaluates a test function at $\mathbf{x}$.

The space of all distributions is denoted $\mathcal{D}'(S^2)$; it is a locally convex topological vector space when equipped with the weak-* topology. Basically, this says that convergence of a sequence of distributions $T_n \to T$ means that $T_n(\phi) \to T(\phi)$ in the complex numbers for any choice of $\phi \in \mathcal{D}(S^2)$.

Many standard operations on functions can be extended to the larger space of distributions by duality. In particular, the Fourier expansion of any distribution T is

$$T = \sum_{l \in \mathbb{N}} \sum_{|m| \le l} \hat{T}(l,m) Y_l^m$$

$$\hat{T}(l,m) = T(\overline{Y_l^m}).$$

Distributional convergence of this expansion obtains precisely because the Fourier series of a test function converges in the test function topology; this follows from standard convergence results for smooth functions, as may be found in Theorem 3.4 on page 21 of Helgason [24], for example.

For any positive integer b, define the *sampling distribution of bandwidth b*,

$$s_b = \frac{\sqrt{2\pi}}{2b} \sum_{j=0}^{2b-1} \sum_{k=0}^{2b-1} a_j^{(b)} \delta_{(\theta_j, \phi_k)}, \tag{6}$$

as a linear combination of point masses at the points on S^2 with coordinates $\theta_j = \pi j / 2b$ and $\phi_k = \pi k / b$. The coefficients $a_k^{(b)}$ are defined in Eq. (5).

LEMMA 1.

$$\hat{s}_b(0,0) = 1$$

and

$$\hat{s}_b(l,m) = 0 \qquad \text{for } 0 < l < 2b.$$

Proof.

$$\hat{s}_b(l,m) = \frac{\sqrt{2\pi}}{2b} \sum_{j=0}^{2b-1} \sum_{k=0}^{2b-1} a_j^{(b)} \delta_{(\theta_j, \phi_k)}(\overline{Y_l^m})$$

$$= \frac{\sqrt{2\pi}}{2b} \sum_{k=0}^{2b-1} e^{-im\pi k/b} \sum_{j=0}^{2b-1} a_j^{(b)} q_l^m P_l^m(\cos\theta_j),$$

where q_l^m denotes the normalization coefficient in Y_l^m;

$$q_l^m = (-1)^m \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}}.$$

The outer sum vanishes unless m is a multiple of $2b$. We are only concerned with the range $l < 2b$ and $|m| < l$, so the transform vanishes in this range unless $m = 0$. In that case,

$$\begin{aligned}\hat{s}_b(l,0) &= \sum_{j=0}^{2b-1} a_j^{(b)} \sqrt{\frac{(2l+1)}{2}} P_l^0(\cos\theta_j) \\ &= \delta_{l,0}\end{aligned}$$

by Eq. (5). □

THEOREM 3. *Let $f(\theta,\phi)$ be a band-limited function on S^2 such that $\hat{f}(l,m) = 0$ for $l \geq b$. Then*

$$\hat{f}(l,m) = \frac{\sqrt{2\pi}}{2b} \sum_{j=0}^{2b-1} \sum_{k=0}^{2b-1} a_j^{(b)} f(\theta_j,\phi_k) \overline{Y_l^m}(\theta_j,\phi_k), \tag{7}$$

for $l < b$ and $|m| \leq l$. Here $\theta_j = \pi j/2b$ and $\phi_k = \pi k/b$, and the coefficients $a_k^{(b)}$ are defined in Eq. (5).

Proof. The product of a smooth function h and a distribution T is a distribution defined on test functions by $h \cdot T(\phi) = T(h\phi)$. The proof proceeds by comparing two computations of the Fourier transform of $f_s = f \cdot s_b$, where the sampling distribution s_b is defined in Eq. (6). This product is easily shown to be

$$f_s = f \cdot s_b = \frac{\sqrt{2\pi}}{2b} \sum_{j=0}^{2b-1} \sum_{k=0}^{2b-1} f(\theta_j,\phi_k) a_j^{(b)} \delta_{(\theta_j,\phi_k)}. \tag{8}$$

Direct computation of the Fourier transform of the *distribution* f_s using the definition $\hat{f}_s(l,m) = f_s(\overline{Y_l^m})$ yields the right-hand side of Eq. (7). It remains to show that this agrees with the Fourier coefficients of the *function* f in the proper range.

To see this, recall from Lemma 1 that

$$s_b = 1 + \sum_{j \geq 2b} \sum_{|k| \leq j} \hat{s}(j,k) Y_j^k(\theta,\phi).$$

Simple distributional convergence arguments justify the following calculation:

$$\begin{aligned} f \cdot s_b &= f + f \cdot \sum_{j \geq 2b} \sum_{|k| \leq j} \hat{s}(j,k) Y_j^k \\ &= f + \sum_{j \geq 2b} \sum_{|k| \leq j} \hat{s}(j,k) f Y_j^k \\ &= f + \sum_{l<b} \sum_{j \geq 2b} \sum_{|m| \leq l} \sum_{|k| \leq j} \hat{s}(j,k) \hat{f}(l,m) Y_l^m Y_j^k. \end{aligned}$$

The latter sum reflects the error, known as aliasing, introduced by sampling f. Its typical term is a multiple of the product of spherical harmonics, $Y_l^m Y_j^k$. This product may itself be expanded in spherical harmonics by means of the Clebsch–Gordon expansion recorded in Theorem 2; in particular, note that no spherical harmonic of degree less than $|l - j|$ can appear. Given the bounds on j and l, it follows that the aliasing is confined to spherical harmonics of degree greater than or equal to that of b. In particular, $\hat{f}_s(l,m) = \hat{f}(l,m)$ in the required range. □

It is also possible to frame this result in an interpolation form, similar to the usual statement of the classical sampling theorems. In order to recover $\hat{f}$ from $\hat{f}_s$ we must multiply $\hat{f}_s(l,m)$ by 1 for $l < b$ and by 0 otherwise. We can, by means of an extension of the convolution theorem, Theorem 1, pull this multiplication back to convolution on the sphere to arrive at the following.

THEOREM 4. *If $\hat{f}(l,m) = 0$ for $l \geq b$ and $f_s = f \cdot s$, where s is the weighted grid of impulses as above, then*

$$f = f_s * \sum_{l=0}^{b-1} \sqrt{\frac{16\pi^3}{2l+1}}\, Y_l^0,$$

where $*$ denotes spherical convolution.

To conclude this section, we now show a closed-form expression for the sampling weights used in the sampling theorem.

LEMMA 2. *For n samples, n a power of two, the sampling weights are*

$$a_j = \frac{2\sqrt{2}}{n} \sin\left(\frac{\pi j}{n}\right) \sum_{l=0}^{n/2-1} \frac{1}{2l+1} \sin\left([2l+1]\frac{\pi j}{n}\right), \qquad j = 0, \ldots, n-1. \tag{9}$$

Proof. We want our weights to satisfy the quadrature rule of Eq. (5):

$$\sum_{l=0}^{n-1} a_l P_k\left(\cos\frac{\pi l}{n}\right) = \sqrt{2}\,\delta_{k,0} = \int_0^{\pi} \sqrt{\frac{2k+1}{2}}\, P_k(\cos\theta)\sin\theta\, d\theta,$$

$$k = 0, \ldots, n-1. \quad (10)$$

The integrand is a trigonometric polynomial with band-limit n. Exact quadrature rules for these on the interval $[-\pi, \pi]$ follow immediately from the discrete orthogonality of the exponentials:

$$\sum_{k=-n}^{n-1} e^{2\pi i k l/2n} = \begin{cases} 2n & \text{if } l \equiv 0 \pmod{2n} \\ 0 & \text{otherwise.} \end{cases}$$

$$= \frac{2n}{2\pi}\int_{-\pi}^{\pi} e^{il\theta}\, d\theta, \qquad \text{for } |l| < 2n. \quad (11)$$

In order to use this, we rewrite the integral in Eq. (10) as an integral over $[-\pi, \pi]$:

$$\int_0^{\pi} P_k(\cos\theta)\sin\theta\, d\theta = \tfrac{1}{2}\left(\int_0^{\pi} P_k(\cos\theta)\sin\theta\, d\theta - \int_{-\pi}^{0} P_k(\cos\theta)\sin\theta\, d\theta\right)$$

$$= \tfrac{1}{2}\int_{-\pi}^{\pi} P_k(\cos\theta)\sin\theta\,\operatorname{sgn}(\theta)\, d\theta,$$

where $\operatorname{sgn}(\theta)$ is 1 for $\theta > 0$ and -1 otherwise. On $[-\pi, \pi]$ this function has the Fourier expansion $(4/\pi)\Sigma_{l \geq 0}(1/(2l+1))\sin(2l+l)\theta$. Our last integral reduces to

$$\tfrac{1}{2}\int_{-\pi}^{\pi} P_k(\cos\theta)\sin\theta\,\Pi_n \operatorname{sgn}(\theta)\, d\theta,$$

$$\text{with } \Pi_n \operatorname{sgn}(\theta) = \frac{4}{\pi}\sum_{l=0}^{n/2-1} \frac{1}{2l+1}\sin(2l+1)\theta$$

for k in the range $0, \ldots, n-1$; this is because the remaining terms in the series for sgn contribute nothing to the integral by orthogonality.

The integrand in the last expression is a band-limited trigonometric polynomial with its highest frequency less than or equal to $2n-1$. By the

quadrature formula (11) we therefore have

$$\begin{aligned}\sqrt{2}\,\delta_{k,0} &= \frac{1}{\sqrt{2}}\int_0^{\pi} P_k(\cos\theta)\sin\theta\, d\theta \\ &= \frac{1}{2\sqrt{2}}\int_{-\pi}^{\pi} P_k(\cos\theta)\sin\theta\prod_n \operatorname{sgn}(\theta)\, d\theta \\ &= \frac{1}{2\sqrt{2}}\frac{\pi}{n}\sum_{l=-n}^{n-1} P_k\left(\cos\frac{l\pi}{n}\right)\sin\left(\frac{l\pi}{n}\right)\prod_n \operatorname{sgn}\left(\frac{l\pi}{n}\right) \\ &= \frac{\pi}{2n\sqrt{2}}2\sum_{l=0}^{n-1} P_k\left(\cos\frac{l\pi}{n}\right)\sin\left(\frac{l\pi}{n}\right)\prod_n \operatorname{sgn}\left(\frac{l\pi}{n}\right),\end{aligned}$$

which gives the stated formula for the weights. □

5. Legendre Transforms

As a first step toward developing a spherical transform algorithm, we consider the problem of computing the Legendere transform of a band-limited function over the interval $[-1, 1]$. This is simply the projection of a function $f \in L^2[-1, 1]$ onto the normalized Legendre polynomials,

$$\tilde{f}(k) = \sqrt{\frac{2k+1}{2}}\int_{-1}^{1} f(x)P_k(x)\, dx.$$

We restrict our attention to functions that are sampled at the n points $\cos(\pi i/n)$, for $i = 0, \ldots, n-1$, where n is assumed to be a power of 2.

Lemma 3. *Let f be a function on the interval $[-1, 1]$ which is in the span of $P_i(x)$, $i < n/2$. Let $x_j = \cos(\pi j/n)$ and $a_j^{(n/2)}$ be defined as in the sampling theorem, Theorem 3. Then*

$$\tilde{f}(k) = \sqrt{\frac{2k+1}{2}}\sum_{j=0}^{n-1} a_j^{(n/2)} f(x_j)P_k(x_j)$$

for $k < n/2$, and zero otherwise.

Proof. This follows from the sampling theorem by considering the function $F(\theta, \phi) = f(\cos\theta)$ defined on the sphere. □

The naive method of computing the transform is to evaluate each of the $n/2$ sums by evaluating and adding n terms, which requires $O(n^2)$ time. No asymptotically better algorithm is currently known. Below, we develop

an $O(n(\log n)^2)$ algorithm, for n a power of two, which requires a preprocessed data structure. The overall plan is to first project the function onto the cosine basis by the fast Fourier transform. The representation in this basis is then transformed into one with respect to the Legendre basis. The latter is accomplished efficiently by a sequence of convolutions derived from the recurrence relation for the Legendre polynomials,

$$(l+1)P_{l+1}(x) = (2l+1)xP_l(x) - lP_{l-1}(x).$$

Recall that if h is any complex-valued function on $\mathbb{Z}/N\mathbb{Z}$, the *discrete Fourier transform* of h is the new function $\mathcal{F}h$ defined by

$$\mathcal{F}h(l) = \sum_{m=0}^{N-1} h(m)e^{-2\pi ilm/N},$$

$0 \le l \le N-1$. The classical fast Fourier transform algorithms [5, 11, 13, 46] show how to compute this function in $O(N \log N)$ operations, rather than the $O(N^2)$ operations a naive count would indicate. We henceforth use "FFT" to denote such an algorithm. We use the FFT for the efficient computation of discrete convolutions. If f, h are functions on $\mathbb{Z}/N\mathbb{Z}$, their (periodic) convolution $f * h$ is the function on $\mathbb{Z}/N\mathbb{Z}$ defined by

$$f * h(k) = \sum_{l=0}^{N-1} f(l)h(k-1).$$

One verifies immediately that $\mathcal{F}(f * h) = \mathcal{F}f\mathcal{F}h$; since the pointwise multiplication on the right side is $O(N)$, the complexity of this computation is $O(N \log N)$ if fast Fourier transforms and inverses are used.

We begin with the projection of the weighted samples of f onto the cosine basis. One way to do this uses the FFT. Let

$$h(j) = \begin{cases} \frac{1}{2}a_j f(\cos j\pi/n) & 0 < j < n, \\ a_0 f(0) & j = 0, \\ h(-j) & -n \le j < 0, \end{cases}$$

for $j = -n, \ldots, n-1$. Then the FFT of h gives us the projection in $O(n \log n)$ time, since

$$\begin{aligned} \mathcal{F}h(k) &= \sum_{l=-n}^{n-1} h(l)\exp(-ikl2\pi/2n) \\ &= \sum_{l=0}^{n-1} a_j f(\cos l2\pi/2n)\cos k2\pi l/2n \doteq \langle f, C_k \rangle, \end{aligned}$$

where we have introduced the functions $C_k(\cos t) = \cos(kt)$, and a shorthand notation for the a_j-weighted l^2 inner product.

Define

$$\begin{aligned} Z(k,l) &= \langle f, C_k P_l \rangle \\ &= \sum_{j=0}^{n-1} a_j f(\cos j\pi/n) \cos(jk\pi/n) P_l(\cos j\pi/n), \end{aligned}$$

where $|k| \le n$ and $0 \le l < n$. Note that $Z(0,l) = \tilde{f}(l)$ is the projection onto the Legendre polynomials which we must compute. On the other hand,

$$Z(k,0) = \langle f, C_k P_0 \rangle = \langle f, C_k \rangle$$

can be efficiently obtained from the samples of f using a single discrete Fourier transform, as shown previously.

It is helpful to consider the array $Z(k,l)$ as a collection of sequences Z_l, $l = 0, 1, \ldots, n-1$ with

$$Z_l(k) = Z(k,l) = \langle f, C_k P_l \rangle.$$

We wish to have the values $Z_l(0) = \langle f, P_l \rangle$. Instead, we have the values $Z_0(k) = \langle f, C_k P_0 \rangle$.

Our goal now is to efficiently obtain the former from the latter. We use the recurrence relation for the Legendre polynomials:

$$(l+1)P_{l+1}(\cos t) = (2l+1)\cos t\, P_l(\cos t) - lP_{l-1}(\cos t), \quad (12)$$

as well as the recurrence for the C_k:

$$C_{k+1}(\cos t) = 2\cos t\, C_k(\cos t) - C_{k-1}(\cos t). \quad (13)$$

In terms of our data structure, this recurrence translates into

$$\begin{aligned} Z_{l+1}(k) &= \langle f, C_k P_{l+1} \rangle \\ &= \frac{2l+1}{l+1} \langle f, C_k C_1 P_l \rangle - \frac{l}{l+1} \langle f, C_k P_{l-1} \rangle \\ &= \frac{2l+1}{l+1} \left\langle f, \left(\frac{C_{k+1} + C_{k-1}}{2} \right) P_l \right\rangle - \frac{l}{l+1} Z_{l-1}(k) \\ &= \frac{2l+1}{l+1} \left(\frac{1}{2} Z_l(k+1) + \frac{1}{2} Z_l(k-1) \right) - \frac{l}{l+1} Z_{l-1}(k). \quad (14) \end{aligned}$$

Observe that the weights in this expression are independent of the index

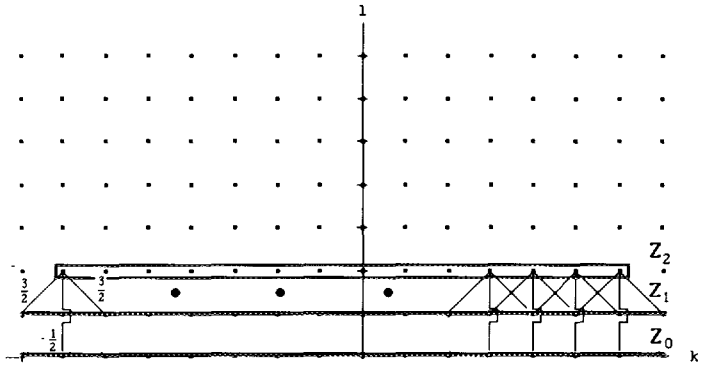

FIG. 1. Computation of Z_2 by convolution of Z_0 and Z_1 with a fixed mask. Each k, l lattice point holds a value of $Z_l(k) = Z(k, l)$. The lines representing the mask are labeled with the appropriate weights. Z_2 values outside the box are computed by periodic extension of Z_0 and Z_1.

k. This implies that the sequence Z_{l+1} can be obtained from the sequences Z_l, Z_{l-1} by (periodic) convolution of each with a fixed convolution mask and then adding the resulting vectors. This is represented schematically in Fig. 1.

In matrix language, this step takes the form of multiplication by a block matrix with circulant blocks:

$$\begin{pmatrix} Z_l \\ Z_{l+1} \end{pmatrix} \begin{pmatrix} \mathrm{O} & \mathrm{Id} \\ -\dfrac{l}{l+1}\mathrm{Id} & \dfrac{2l+1}{l+1}C \end{pmatrix} \begin{pmatrix} Z_{l-1} \\ Z_l \end{pmatrix}, \tag{15}$$

where Id is the $2n \times 2n$ identity matrix and C is the $2n \times 2n$ circulant matrix,

$$C = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \cdots & \cdots & \cdots & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \cdots & \cdots & 0 & \frac{1}{2} & 0 \end{pmatrix}.$$

The convolution of two functions supported on n points can be computed by means of the fast Fourier transform in $O(n \log n)$ time; in matrix language this corresponds to the factorization of the matrix in Eq. (15) as

$$\begin{pmatrix} \mathcal{F}^{-1} & 0 \\ 0 & \mathcal{F}^{-1} \end{pmatrix} \begin{pmatrix} 0 & \mathrm{Id} \\ -\dfrac{l}{l+1}\mathrm{Id} & \dfrac{2l+1}{l+1}\cos\sigma \end{pmatrix} \begin{pmatrix} \mathcal{F} & 0 \\ 0 & \mathcal{F} \end{pmatrix}. \tag{16}$$

Here $\cos\sigma = \mathrm{Diagonal}\{(\cos(\pi k/n) | k = -n, \ldots, n-1\}$, and $\mathcal{F}$ is the $2n \times 2n$ Fourier transform matrix, $\mathcal{F}_{k,l} = 1/\sqrt{2n}\exp(2\pi ikl/2n)$, $-n \le k, l < n$. The key is that $\mathcal{F}$ is itself factored in the FFT so that it may be applied in $O(n \log n)$.

Equation (14) suggests a technique for successively constructing the sequences Z_l for higher indices l from those with lower indices. To begin this, we need the initial data Z_0 and Z_1. As discussed above, we can compute the sequence Z_0 in $O(n \log n)$ as a Fourier transform of the weighted sampling data, as previously discussed. Z_1 is then easily obtained from Z_0 as $Z_1(k) = 1/2\{Z_0(k-1) + Z_0(k+1)\}$. In particular, this operation gives us the values $Z_0(0) = \tilde{f}(0)$ and $Z_1(0) = \tilde{f}(1)$.

We could then apply the operator in Eq. 16 with $l = 1$ to Z_0, and Z_1 and convolve up to Z_2, obtaining in particular $\tilde{f}(2)$. Next, Z_1 and Z_2 can be used to make Z_3, and in particular $\tilde{f}(3)$. Proceeding in this way, we could build up all the sequences Z_l and obtain $\tilde{f}(l)$. This of course would be prohibitively costly, consisting of n $O(n \log n)$ moves. In fact, most of the terms of the higher sequences Z_l are not used, suggesting that we try a more efficient sequence of moves instead. We record the result as the main theorem of the section.

THEOREM 5. *If $n = 2^k < N$, $x_j = \cos(\pi j/n)$, and the coefficients $a_j^{(n/2)}$ are defined as in the sampling theorem, Theorem 3, then*

$$\tilde{f}(l) = \sqrt{\frac{2l+1}{2}} \sum_{j=0}^{n-1} a_j^{(n/2)} f(x_j) P_l(x_j)$$

can be computed for all $l < n/2$ in $O(n(\log n)^2)$ time, given a precomputed data structure of size $O(N(\log N)^2)$.

Proof. First note that we can obtain an m step convolution matrix for obtaining Z_{l+m} and Z_{l+m-1} from Z_l and Z_{l-1} by iterating the one-step convolution in Eq. (15); this amounts to multiplying several of these matrices. Figure 2 gives a graphic representation of this.

In fact, as we show in Lemma 4 at the end of this section, these matrices have a closed form expression in terms of certain trigonometric polynomials called *associated Legendre polynomials*, $P_m(l, x)$ [4]: for any positive

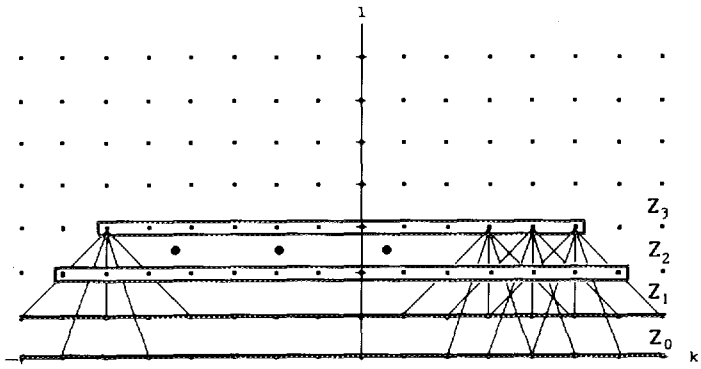

FIG. 2. Z_3 obtained from Z_0 and Z_1 by convolving with a two step mask. This is obtained by composing two one step convolutions. The boxed range may be computed without periodic extension of Z_0 and Z_1. Note that the length of this range will decrease for longer step convolutions.

integer d we define $T_d(l, m) =$

$$\begin{pmatrix} \mathscr{F}^{-1} & \bigcirc \\ \bigcirc & \mathscr{F}^{-1} \end{pmatrix} \begin{pmatrix} -\frac{1}{l+1}P_{m-1}(l+1, \cos\sigma) & P_m(l, \cos\sigma) \\ -\frac{1}{l+1}P_m(l+1, \cos\sigma) & P_{m+1}(l, \cos\sigma) \end{pmatrix} \begin{pmatrix} \mathscr{F} & \bigcirc \\ \bigcirc & \mathscr{F} \end{pmatrix}, \tag{17}$$

$$P_m(l, \cos\sigma) = \text{Diagonal}\{P_m(l, \cos(\pi k/d)) | k = -d, \ldots, d-1\},$$

and $\mathscr{F}$ is the $2d$-dimensional Fourier transform. Then

$$\begin{pmatrix} Z_{l+m} \\ Z_{l+m+1} \end{pmatrix} = T_d(l, m) \begin{pmatrix} Z_{l-1} \\ Z_l \end{pmatrix}. \tag{18}$$

We assume that the appropriate multi-step matrices have been prestored. In fact, it is not difficult to show that they may be derived efficiently from a recurrence relation.

We start with the full sequences Z_0 and Z_1, obtained with the FFT at a cost of $O(n \log n)$, as already described. From these, compute $Z_{n/2}$ and $Z_{n/2+1}$ by means of the prestored convolution matrix $T_n(1, n/2)$ at a cost of $O(n \log n)$ operations.

We now own $Z_0, Z_1; Z_{n/2}, Z_{n/2+1}$. Throw away the outer half of each sequence; i.e., keep only the restrictions $Z_j^{(n/2)}$ defined as

$$Z_j^{(n/2)} = \left\{ Z_j(k) \middle| \frac{n}{2} \le k < \frac{n}{2} \right\}.$$

In matrix terms, this is done with projection matrices: $Z_j^{(n/2)} = \Pi_{n/2} Z_j$, where for any $d > 0$ we

$$\Pi_d = \underset{\leftarrow \frac{d}{2} \rightarrow \leftarrow d \rightarrow \leftarrow \frac{d}{2} \rightarrow}{\left(\mathrm{O} \;\vdots\; \mathrm{Id} \;\vdots\; \mathrm{O} \right)} \begin{matrix} \uparrow \\ d \\ \downarrow \end{matrix}.$$

Now apply half-size convolution matrices and projections to obtain

$$\begin{pmatrix} Z^{(n/4)} \\ Z^{(n/4)}_{(n/4)+1} \end{pmatrix} = \begin{pmatrix} \Pi_{n/4} & \mathrm{O} \\ \mathrm{O} & \Pi_{n/4} \end{pmatrix} T_{n/2}\left(1, \frac{n}{4}\right) \begin{pmatrix} Z_0^{(n/2)} \\ Z_1^{(n/2)} \end{pmatrix}$$

and

$$\begin{pmatrix} Z^{(n/4)}_{3n/4} \\ Z^{(n/4)}_{(3n/4)+1} \end{pmatrix} = \begin{pmatrix} \Pi_{n/4} & \mathrm{O} \\ \mathrm{O} & \Pi_{n/4} \end{pmatrix} T_{n/2}\left(\frac{n}{2} + 1, \frac{n}{4}\right) \begin{pmatrix} Z^{(n/2)}_{n/2} \\ Z^{(n/2)}_{(n/2)+1} \end{pmatrix}.$$

One verifies that the correct values of the sequences are obtained in this restricted range despite incorrect wrap-around effects outside of the range. Each of these two computations costs $O((n/2)\log(n/2))$.

At the end of step two, we own the truncated sequences

$$Z_0^{(n/4)}, Z_1^{(n/4)}; Z_{n/2}^{(n/4)}, Z_{n/4}^{(n/4)}, Z_{(n/4)+1}^{(n/4)}; Z_{(n/2)+1}^{(n/4)}; Z_{3n/4}^{(n/4)}, Z_{(3n/4)+1}^{(n/4)}.$$

We apply the four appropriate quarter size convolution matrices and projections to obtain

$$Z_{n/8}^{(n/8)}, Z_{(n/8)+1}^{(n/8)}; Z_{3n/8}^{(n/8)}, Z_{(3n/8)+1}^{(n/8)}; Z_{5n/8}^{(n/8)}, Z_{(5n/8)+1}^{(n/8)}; Z_{7n/8}^{(n/8)}, Z_{(7n/8)+1}^{(n/8)}.$$

Each of these four computations costs $O((n/4)\log(n/4))$.

Continuing in this fashion, as indicated schematically in Fig. 3, we obtain $Z_l(0)$ for $0 \le l < n$ in

$$\sum_{i=0}^{\log n - 1} 2^i \frac{n}{2^i} \log \frac{n}{2^i},$$

which is $O(n(\log n)^2)$. Then the desired coefficients $\tilde{f}(l)$ are obtained as $\sqrt{l + 1/2}\, Z_l(0)$.

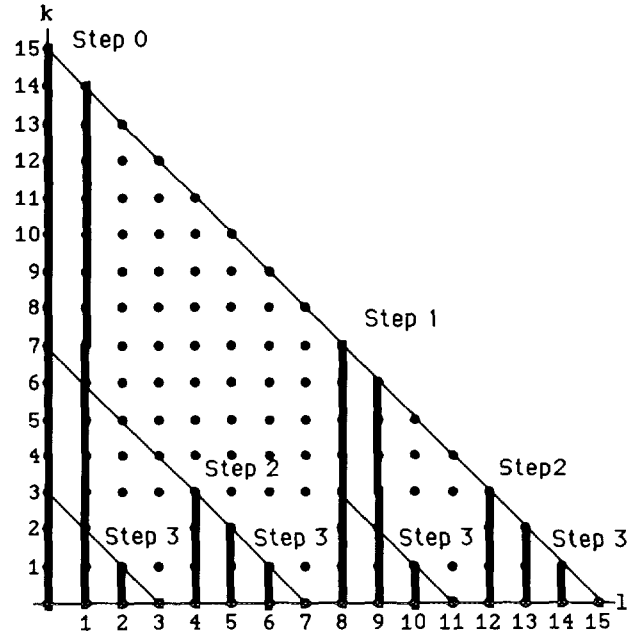

FIG. 3. Computation of the $Z_l(0)$ for $l < n$ by a cascade of circulant block matrix multiplications of decreasing size. The relevant ranges of Z are highlighted, and the step in which they are calculated is indicated.

This calculation assumes prestored convolution masks

$$T_n\left(1, \frac{n}{2}\right)$$

$$T_{n/2}\left(1, \frac{n}{4}\right), T_{n/2}\left(\frac{n}{2} + 1, \frac{n}{4}\right),$$

$$T_{n/4}\left(1, \frac{n}{8}\right), T_{n/4}\left(\frac{n}{4} + 1, \frac{n}{8}\right), T_{n/4}\left(\frac{n}{2} + 1, \frac{n}{8}\right), T_{n/4}\left(\frac{3n}{4} + 1, \frac{n}{8}\right),$$

$$\vdots$$

$$T_4(1,2), \quad T_4(5,2), \quad T_4(9,2) \cdots T_4(n-5,2), \quad T_4(n-3,2).$$

As there are $\log n$ levels in this tree, the total storage requirements for these masks are then $O(n \log n)$. □

By using the recurrence for the associated Legendre functions, we also obtain the following more general result.

THEOREM 6. *If $n = 2^k < N$, and m is a fixed integer with $|m| \le n/2$, then*

$$\hat{f}(l,m) = \sum_{j=0}^{n-1} a_j^{(n/2)} f(x_j) P_l^m(x_j) \qquad |m| \le l \le n/2,$$

can be computed for each fixed m in $O(n(\log n)^2)$ time given a precomputed data structure of size $O(N(\log N)^2)$.

Proof. We need to project f onto the P_l^m for a fixed nonnegative m, and for all $l \ge m$ and below the band-limit $n/2$. The results for m negative are obtained as simple multiples of the cases considered here. As before we use the second-order recurrence for associated Legendre functions,

$$P_{l+1}^m(x) = \frac{2l+1}{l-m+1} x P_l^m(x) - \frac{l+m}{l-m+1} P_{l-1}^m(x).$$

Define

$$Z(k,l) = \sum_{j=0}^{n-1} a_j^{(n/2)} f(\cos j\theta)(\cos kj\theta) P_l^m(\cos j\theta),$$

for $0 \le k < n/2$ and $m \le l \le n/2$, and with $\theta = \pi/n$, as before. Defining $Z_l(k) = Z(k,l)$, then, the initial conditions to start the recurrence are the vectors Z_m and Z_{m+1}.

The vector Z_m can be obtained efficiently by means of the projection onto cosines described earlier. In the present case, the algorithm is applied not to f, but rather to the function $f(\cos\theta)P_m^m(\cos\theta)$, which is simply a multiple of $f(\cos\theta)\sin^m(\theta)$. As before, the recurrence shows that the other initial vector Z_{m+1} is a shifted multiple of Z_m and is thus essentially already known.

Having obtained the initial conditions for the recurrence efficiently, we apply the fast recurrence algorithm as described for the $m = 0$ case. Note that in general this requires extending the range of indices l beyond $n/2$ so that the total range is a power of 2 in length. The result is the fast computation of the $Z_l(0)$, as required. □

It is important to note that the sum in the preceding theorem is not claimed to be the transform of f with respect to any family of associated Legendre functions except for $m = 0$, where the sampling theorem applies. This, however, poses no difficulty for computing spherical transforms.

It remains to consider the problem of inverting the Legendre transform. The inversion is obtained by

$$f(x) = \sum_{k=0}^{n-1} \tilde{f}(k) \sqrt{\frac{2k+1}{2}} P_k(x).$$

It is easily seen that this sum can be calculated in $O(n)$ time, and hence the function can be recovered at the $2n$ sampled points in $O(n^2)$ time.

We conclude by establishing the formula for the $T_d(l, m)$, promised earlier. We also record a bound which is useful in the next section's error analysis of the algorithm.

LEMMA 4. *The m-step convolution matrices in the algorithm of Theorem 5 have the form specified in formula* (17). *This matrix,* $T_d(l, m)$, *has an* l^2 *operator norm bounded by the Euclidean norm of a submatrix,*

$$\|T_d(l,m)\|_2 \le \left\| \begin{pmatrix} -\frac{l}{l+1} P_{m-1}(l+1,1) & P_m(l,1) \\ -\frac{l}{l+1} P_m(l+1,1) & P_{m+1}(l,1) \end{pmatrix} \right\|_E$$

$$\le 2P_{m+1}(l,1) = 2 \sum_{k=0}^{m+1} \frac{l}{l+k}.$$

For brevity, we refer to the sum as $B(l, m)$.

Proof. Recall that $T_d(l, m)$ is a composition of m of the one-step convolutions described in formula (15). Using the Fourier decomposition of these matrices given in formula (16), this product reduces to

$$\begin{pmatrix} \mathcal{F}^{-1} & \mathrm{O} \\ \mathrm{O} & \mathcal{F}^{-1} \end{pmatrix} \begin{pmatrix} \mathrm{O} & \mathrm{Id} \\ -\frac{l+m}{l+m+1}\mathrm{Id} & \frac{2l+2m+1}{l+m+1}\cos\sigma \end{pmatrix} \cdots$$

$$\begin{pmatrix} \mathrm{O} & \mathrm{Id} \\ -\frac{l}{l+1}\mathrm{Id} & \frac{2l+1}{l+1}\cos\sigma \end{pmatrix} \begin{pmatrix} \mathcal{F} & \mathrm{O} \\ \mathrm{O} & \mathcal{F} \end{pmatrix},$$

where σ is the diagonal matrix of sample values $\{(\pi k/n) | k = -d, \ldots, d-1\}$. We find a nice expression for the product of the tridiagonal

matrices between the Fourier matrices. We use the shorthand

$$M_d(l+s) = \begin{pmatrix} \mathrm{O} & \mathrm{Id} \\ -\dfrac{l+s}{l+s+1}\mathrm{Id} & \dfrac{2l+2s+1}{l+s+1}\cos\sigma \end{pmatrix}$$

and

$$\prod_{s=0}^{m} M_d(l+s) = M_d(l+m)\cdots M_d(l+1)M_d(l),$$

where we note that the factors are to be ordered with indices increasing from right to the left.

Note that the product is a 2×2 block matrix whose diagonal blocks are trigonometric polynomials evaluated on the diagonal matrix σ. In fact, these polynomials are determined by a shifted form of the Legendre recurrence relation. Specifically, there are sequences a_k, b_k of trigonometric polynomials with

$$\prod_{s=0}^{k} M_d(l+s) = \begin{pmatrix} a_k(\cos\sigma) & b_k(\cos\sigma) \\ a_{k+1}(\cos\sigma) & b_{k+1}(\cos\sigma) \end{pmatrix}$$

satisfying the recursion implicit in the identity

$$\prod_{s=0}^{k} M_d(l+s) = M_d(l+k)\prod_{s=0}^{k-1} M_d(l+s),$$

namely

$$\begin{pmatrix} a_k(\cos\sigma) & b_k(\cos\sigma) \\ a_{k+1}(\cos\sigma) & b_{k+1}(\cos\sigma) \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{O} & \mathrm{Id} \\ -\dfrac{l+k}{l+k+1}\mathrm{Id} & \dfrac{2l+2k+1}{l+k+1}\cos\sigma \end{pmatrix} \begin{pmatrix} a_{k-1}(\cos\sigma) & b_{k-1}(\cos\sigma) \\ a_k(\cos\sigma) & b_k(\cos\sigma) \end{pmatrix},$$

which says both sequences are solutions of the linear recurrence

$$y_{k+1}(\cos\sigma) = \frac{2l+2k+1}{l+k+1}\cos\sigma\, y_k(\cos\sigma) - \frac{l+k}{l+k+1} y_{k-1}(\cos\sigma).$$

This, taken with the initial conditions

$$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = -\frac{l}{l+1}\begin{pmatrix} 0 \\ \mathrm{Id} \end{pmatrix} \qquad \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} \mathrm{Id} \\ \dfrac{2l+1}{l+1}\cos\sigma \end{pmatrix},$$

defines the sequences uniquely as the associated Legendre polynomials [4], so that

$$\prod_{s=0}^{m} M_d(l+s) = \begin{pmatrix} -\dfrac{l}{l+1}P_{m-1}(l+1,\cos\sigma) & P_m(l,\cos\sigma) \\ -\dfrac{l}{l+1}P_m(l+1,\cos\sigma) & P_{m+1}(l,\cos\sigma) \end{pmatrix}.$$

Many facts are known about the trigonometric polynomials, $P_m(l,\cos\theta)$, including the one we need for our bound; they take their maximum absolute value at $\theta = 0$:

$$\sup_{\theta\in[0,2\pi]} |P_m(l,\cos\theta)| = P_m(l,1) = \sum_{k=0}^{m} \frac{l}{l+k}. \tag{19}$$

This follows from their representation in terms of the (unnormalized) Legendre polynomials,

$$P_m(l,\cos\theta) = \sum_{k=0}^{m} \frac{l}{l+k} P_k(\cos\theta) P_{m-k}(\cos\theta),$$

for the $P_k(\cos\theta)$ are all bounded in absolute value by 1, with $P_k(1) = 1$ for all k.

Now for the operator norm bound,

$$\|T_d(l,m)\|_2 = \left\| \begin{pmatrix} -\dfrac{l}{l+1}P_{m-1}(l+1,\cos\sigma) & P_m(l,\cos\sigma) \\ -\dfrac{l}{l+1}P_m(l+1,\cos\sigma) & P_{m+1}(l,\cos\sigma) \end{pmatrix} \right\|_2,$$

we consider

$$\left\| \begin{pmatrix} -\dfrac{l}{l+1}P_{m-1}(l+1,\cos\sigma) & P_m(l,\cos\sigma) \\ -\dfrac{l}{l+1}P_m(l+1,\cos\sigma) & P_{m+1}(l,\cos\sigma) \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \right\|_2^2, \tag{20}$$

for column vectors

$$\mathbf{x} = \begin{pmatrix} x_{-d} \\ \vdots \\ x_{d-1} \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_{-d} \\ \vdots \\ y_{d-1} \end{pmatrix}.$$

By reordering the sum we see that the squared norm is

$$\sum_{k=-d}^{d-1} \left\| \begin{pmatrix} -\frac{l}{l+1} P_{m-1}\left(l+1, \cos\frac{\pi k}{d}\right) & P_m\left(l, \cos\frac{\pi k}{d}\right) \\ -\frac{l}{l+1} P_m\left(l+1, \cos\frac{\pi k}{d}\right) & P_{m+1}\left(l, \cos\frac{\pi k}{d}\right) \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \right\|_2^2$$

$$\leq \sum_{k=-d}^{d-1} \left\| \begin{pmatrix} -\frac{l}{l+1} P_{m-1}\left(l+1, \cos\frac{\pi k}{d}\right) & P_m\left(l, \cos\frac{\pi k}{d}\right) \\ -\frac{l}{l+1} P_m\left(l+1, \cos\frac{\pi k}{d}\right) & P_{m+1}\left(l, \cos\frac{\pi k}{d}\right) \end{pmatrix} \right\|_E^2 \times \left\| \begin{pmatrix} x_k \\ y_k \end{pmatrix} \right\|_2^2.$$

By our bound on the associated Legendre polynomials, Eq. (19), this is bounded by

$$\left\| \begin{pmatrix} -\frac{l}{l+1} P_{m-1}\left(l+1, \cos\frac{\pi 0}{d}\right) & P_m\left(l, \cos\frac{\pi 0}{d}\right) \\ -\frac{l}{l+1} P_m\left(l+1, \cos\frac{\pi 0}{d}\right) & P_{m+1}\left(l, \cos\frac{\pi 0}{d}\right) \end{pmatrix} \right\|_E^2 \sum_{k=-d}^{d-1} \left\| \begin{pmatrix} x_k \\ y_k \end{pmatrix} \right\|_2^2.$$

One quickly checks that the largest matrix element is $P_{m+1}(l, 1)$, and the result follows. □

6. Fast Legendre Transforms in Finite Precision Arithmetic

In this section, we offer simple a priori error bounds and experimental results concerning the numerical properties of the Legendre polynomial

algorithm, when implemented in fixed precision floating point. The main theoretical result is found in Theorem 7 below. This is followed by plots of the bounds presented in that theorem, and a presentation of the experimental results.

Our findings are roughly similar to analogous results for the standard abelian fast Fourier transform algorithms, whose numerical properties have been studied for decades; see for example [8, 9, 26–28, 33, 35, 36, 38, 39, 45]. In addition to theoretical studies of the abelian FFT, there is a huge body of experiment indicating its numerical stability. It seems fair to say that the widespread confidence in the numerical reliability of the FFT algorithms derives largely from actual computational experience obtained over many years.

In light of this, we present our results and experiments to date as initial evidence supporting the position that the Legendre transform algorithm, or a modification of it, may be used for reliable computations with existing hardware and software in the near future.

Good references for the type of a priori analysis used in this section include [21, 39, 48, 49]. One must take this sort of analysis for what it is worth; Wilkinson's remarks quoted on page 65 of Golub and van Loan's book [21] are very much to the point.

The standard model of floating point errors upon rounding of intermediate results is discussed in the references; for the current discussion, the main issue is the difference between floating point and exact computation of matrix vector products. Suppose that A is an $n \times n$ matrix and $\mathbf{x}$ an n-vector, both with floating point entries. Let $fl(A\mathbf{x})$ denote the approximation of the matrix vector product $A\mathbf{x}$ when the latter quantity is actually computed in floating point arithmetic. Then we have the bound ([48, p. 83])

$$\| fl(A\mathbf{x}) - A\mathbf{x} \|_2 \leq \epsilon n \|A\|_E \|\mathbf{x}\|_2, \tag{21}$$

where $\|A\|_E$ is the Euclidean norm of A (square root of the sum of the squares of its entries) and ϵ is a machine dependent constant arising from the rounding in floating point. On many current workstations, it is around 10^{-16} for double precision floating point.

Each output of our algorithm is obtained from the initial input vectors Z_0, Z_1 by a sequence of basic linear operations. Each of these linear operations is either a multi-step convolution or the identity map, followed by elimination of half of the result, $P_{d/2}$. Recall that, in the algorithm, the basic convolutions $T_d(l, m)$ are actually effected by first taking the FFT of two input sequences, then multiplying by a 2×2 block matrix with diagonal blocks to produce two new sequences, followed by inverse Fourier transform of the two sequences.

So we see that each "basic" convolution operation is actually performed by a chain of floating point linear operations on vectors, corresponding to a particular matrix factorization of the operator. There are several sources of floating point error here. First of all, each of the matrices in the factorization of $T_d(l, m)$ must be replaced by its floating point approximation. Then, each of these is applied in floating point to the vector of partial results from the preceding stages. Each stage produces a bit of error, resulting in a cumulative difference between the exact and floating point applications of $T_d(l, m)$.

Once we have a handle on the error incurred in the basic operations, we must deal with the fact that each point of the output of the algorithm is obtained from a particular sequence of such operations and must then seek to bound the resulting floating point error of that sequence.

We therefore require a floating point error analysis of the basic operations and then an analysis of the error incurred by composing them. As a first step, we record a lemma giving bounds on the floating point error of a sequence of linear operations in terms of error bounds for the individual operations.

LEMMA 5. *Let A_i, $i = 1, \ldots, n$, be a sequence of finite-dimensional linear operators. Suppose we have specified algorithms for applying each of these to floating point vectors, and that we have differential bounds on the l^2 floating point error of applying them by means of these algorithms*:

$$\| fl(A_k \mathbf{x}_k) - A_k \mathbf{x}_k \|_2 \leq \epsilon B(A_k) \| \mathbf{x}_k \|_2 + O(\epsilon^2).$$

Then a bound on the l^2 error for floating point application of the whole sequence to floating point vector $\mathbf{x}$ is given by

$$\epsilon \| \mathbf{x} \|_2 \sum_k B(A_k) \prod_{j \neq k} \| A_j \|_2 + O(\epsilon^2),$$

where $\| A_k \|_2$ denotes the l^2 operator norm of A_k.

Proof. Let $\mathbf{x}$ be a floating point vector. For this proof, it is convenient to represent the result of the floating point algorithm for applying A to $\mathbf{x}$ by the symbol $A \circ \mathbf{x}$ rather than by $fl(A\mathbf{x})$. $A\mathbf{x}$ denotes the exact arithmic result. The overall calculation performed is then a sequence of these individual operations; namely,

$$A_n \circ (A_{n-1} \circ \cdots A_2 \circ (A_1 \circ \mathbf{x}) \cdots)),$$

evaluated from the innermost expression on out. We omit the parentheses from now on.

The error incurred, $A_n \circ A_{n-1} \circ \cdots \circ A_1 \circ \mathbf{x} - A_n A_{n-1} \cdots A_1 \mathbf{x}$, may be written as the telescoping expansion

$$\begin{aligned} & A_n \circ (A_{n-1} \circ A_{n-2} \circ \cdots \circ A_1 \circ \mathbf{x}) - A_n (A_{n-1} \circ A_{n-2} \circ \cdots \circ A_2 \circ \mathbf{x}) \\ & \quad + A_n A_{n-1} \circ (A_{n-2} \circ \cdots \circ A_1 \circ \mathbf{x}) - A_n A_{n-1} (A_{n-2} \circ \cdots \circ A_1 \circ \mathbf{x}) \\ & \quad + \cdots \\ & \quad + A_n A_{n-1} \cdots A_2 \circ (A_1 \circ \mathbf{x}) - A_n A_{n-1} \cdots A_2 (A_1 \circ \mathbf{x}) \\ & \quad + A_n A_{n-1} \cdots A_2 A_1 \circ \mathbf{x} - A_n A_{n-1} \cdots A_2 A_1 \circ x. \end{aligned}$$

We now obtain bounds for each of the terms of this expansion starting with the simplest term:

$$\| A_n \cdots A_2 A_1 \circ \mathbf{x} - A_n \cdots A_1 \mathbf{x} \|_2 \leq \| A_n \cdots A_2 \|_2 \, \| A_1 \circ \mathbf{x} - A_1 \mathbf{x} \|_2.$$

By our presumed bound on the floating point error of applying A_1, we have

$$\| A_1 \circ \mathbf{x} - A_1 \mathbf{x} \|_2 \leq \epsilon B(A_1) \| \mathbf{x} \|_2 + O(\epsilon^2), \tag{22}$$

which gives the desired bound for the term:

$$\epsilon \| A_n \cdots A_2 \|_2 B(A_1) \| \mathbf{x} \|_2 + O(\epsilon^2).$$

A similar analysis bounds the next term,

$$\begin{aligned} & \| A_n \cdots A_3 (A_2 \circ [A_1 \circ \mathbf{x}] - A_2 [A_1 \circ \mathbf{x}]) \|_2 \\ & \quad \leq \| A_n \cdots A_3 \|_2 \, \| A_2 \circ [A_1 \circ \mathbf{x}] - A_2 [A_1 \circ \mathbf{x}] \|_2. \end{aligned}$$

The error bound for A_2 shows

$$\| A_2 \circ [A_1 \circ \mathbf{x}] - A_2 [A_1 \circ \mathbf{x}] \|_2 \leq \epsilon B(A_2) \| A_1 \circ \mathbf{x} \|_2 + O(\epsilon^2). \tag{23}$$

Now Eq. (22) above implies

$$\| A_1 \circ \mathbf{x} \|_2 = \| A_1 \mathbf{x} \|_2 + O(\epsilon). \tag{24}$$

Thus we have the desired bound for the second term

$$\epsilon \| A_n \cdots A_3 \|_2 B(A_2) \| A_1 \|_2 \| \mathbf{x} \|_2 + O(\epsilon^2).$$

Additionally, Eqs. (24) and (25) imply

$$\begin{aligned}\|A_2 \circ A_1 \circ \mathbf{x}\|_2 &= \|A_2(A_1 \circ \mathbf{x})\|_2 + O(\epsilon) \le \|A_2\|_2 \|A_1 \circ \mathbf{x}\|_2 + O(\epsilon) \\ &\le \|A_2\|_2 \|A_1 \mathbf{x}\|_2 + O(\epsilon) \le \|A_2\|_2 \|A_1\|_2 \|\mathbf{x}\|_2 + O(\epsilon),\end{aligned}$$

which is needed in the estimate of the next term.

Proceeding in this fashion, we come to the bound of the kth term, and we assume that from the bound of the $(k-1)$-st term we have determined that

$$\|A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}\|_2 \le \|A_{k-1}\|_2 \cdots \|A_1\|_2 \|\mathbf{x}\|_2 + O(\epsilon). \quad (25)$$

Then the bound for the k th term is

$$\begin{aligned}&\|A_n \cdots A_{k+1}(A_k \circ [A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}] - A_k[A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}])\|_2 \\ &\quad \le \|A_n \cdots A_{k+1}\|_2 \|A_k \circ [A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}] \\ &\qquad\qquad - A_k[A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}]\|_2 \\ &\quad \le \|A_n \cdots A_{k+1}\|_2 \, \epsilon B(A_k) \|A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}\|_2 + O(\epsilon^2) \\ &\quad \le \epsilon \|A_n \cdots A_{k+1}\|_2 B(A_k) \|A_{k-1}\|_2 \cdots \|A_1\|_2 \|\mathbf{x}\|_2 + O(\epsilon^2),\end{aligned}$$

with the last step by the induction hypothesis, Eq. (25).

The error bound used in the middle step of the preceding shows that

$$\|A_k \circ [A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}]\|_2 = \|A_k A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}\|_2 + O(\epsilon)$$

and the induction hypothesis then gives the bound

$$\begin{aligned}\|A_k \circ A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}\|_2 &\le \|A_k\|_2 \|A_{k-1} \circ \cdots \circ A_1 \circ \mathbf{x}\|_2 + O(\varepsilon) \\ &\le \|A_k\|_2 \|A_{k-1}\|_2 \cdots \|A_1\|_2 \|\mathbf{x}\|_2 + O(\epsilon)\end{aligned}$$

required for the next step of the procedure.

The result now follows by induction. □

Now we analyze the floating point error of one of the steps $P_{d/2} T_d(l, m)$ as it would be applied in the algorithm.

LEMMA 6. *A bound for the floating point error of the operation $P_{d/2} T_d(l, m)$ on the vector $\mathbf{z}$ is given by $(2K \log_2 d + 3) B(l, m) \epsilon \|\mathbf{z}\|_2 + O(\epsilon^2)$, where $B(l, m)$ is defined as the sum defined in Lemma 3, and K is a (small) constant determined by the accuracy with which the trigonometric functions in the Fourier transform are calculated.*

Proof. The projection operator obviously introduces no floating point error. Recall that $T_d(l, m)$ is applied to a pair of $2d$-length floating point

vectors, say $\mathbf{x}$ and $\mathbf{y}$, as a sequence of three successive operations on vectors, applied from right to left:

$$\begin{pmatrix} \mathscr{F}^{-1} & \mathrm{O} \\ \mathrm{O} & \mathscr{F}^{-1} \end{pmatrix} \begin{pmatrix} -\frac{l}{l+1} P_{m-1}(l+1, \cos\sigma) & P_m(l, \cos\sigma) \\ -\frac{l}{l+1} P_m(l+1, \cos\sigma) & P_{m+1}(l, \cos\sigma) \end{pmatrix} \times \begin{pmatrix} \mathscr{F} & \mathrm{O} \\ \mathrm{O} & \mathscr{F} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \tag{26}$$

where $\mathscr{F}, \mathscr{F}^{-1}$ are calculated by means of the abelian fast Fourier transform and inverse. We use the shorthand $F^{-1}MF$ for the matrix decomposition in Eq. (26).

We need error bounds on each of the three operations in order to use Lemma 5. We may use the existing analyses of the floating point error for the fast Fourier transform and its inverse. Here, we use the bound of Ramos [39]. For the $2d$-dimensional radix-2 fast Fourier transform, $\mathscr{F}$, there is a constant K such that

$$\|\mathscr{F} \circ \mathbf{x} - \mathscr{F}\mathbf{x}\|_2 \leq \epsilon K \log_2 d \|\mathbf{x}\|_2.$$

We have once again used the " $\circ$ " notation introduced in the last lemma for the floating point matrix-vector product. K depends on the accuracy of evaluation of the trigonometric functions for a given size problem. Ramos lists examples where it varies in the range 8–25.

One easily checks that the error bound for the direct sum fast Fourier matrix, F, which appears in the decomposition (26) is also $\epsilon K \log_2 d$. It is easy to see that this holds likewise for its inverse.

The error bound for applying the tridiagonal matrix, M, may be obtained in a manner similar to its l^2 operator norm bound in Lemma 3. First note that we are only able to store and use a floating point representation of this matrix, and this differs slightly from the matrix we wrote down. In fact, M varies from its stored version, which we denote $fl(M)$, by an error matrix E, with entries satisfying

$$|E_{i,j}| \doteq \left| M_{i,j} - fl(M_{i,j}) \right| \leq \epsilon |M_{i,j}|. \tag{27}$$

With this notation, we are concerned with the error $\|fl(M) \circ \mathbf{w} - M\mathbf{w}\|_2$, where $\mathbf{w} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}$ is composed of two length $2d$ vectors. This is bounded

by

$$\| fl(M) \circ \mathbf{w} - fl(M)\mathbf{w} \|_2 + \| fl(M)\mathbf{w} - M\mathbf{w} \|_2.$$

Using Eq. (27), the latter term is bounded by

$$\| fl(M) - M \|_2 \| \mathbf{w} \|_2 \leq \epsilon \| M \|_2 \| \mathbf{w} \|_2 \leq B(l, m) \| \mathbf{w} \|_2,$$

by Lemma 3, where we obtained the bound $B(l, m)$ for the operator norm of M.

The square of the first term may be expanded and re-ordered as

$$\sum_{k=-d}^{d-1} \left\| \begin{pmatrix} -fl\left[\frac{l}{l+1} P_{m-1}\left(l+1, \cos\frac{\pi k}{d}\right)\right] & fl\left[P_m\left(l, \cos\frac{\pi k}{d}\right)\right] \\ -fl\left[\frac{l}{l+1} P_m\left(l+1, \cos\frac{\pi k}{d}\right)\right] & fl\left[P_{m+1}\left(l, \cos\frac{\pi k}{d}\right)\right] \end{pmatrix} \circ \begin{pmatrix} u_k \\ v_k \end{pmatrix} \right.$$

$$\left. - \begin{pmatrix} -fl\left[\frac{l}{l+1} P_{m-1}\left(l+1, \cos\frac{\pi k}{d}\right)\right] & fl\left[P_m\left(l, \cos\frac{\pi k}{d}\right)\right] \\ -fl\left[\frac{l}{l+1} P_m\left(l+1, \cos\frac{\pi k}{d}\right)\right] & fl\left[P_{m+1}\left(l, \cos\frac{\pi k}{d}\right)\right] \end{pmatrix} \begin{pmatrix} u_k \\ v_k \end{pmatrix} \right\|_2^2;$$

applying the formula (21) for the error of matrix multiplication to each summand shows this is bounded by

$$\sum_{k=-d}^{d-1} (2\epsilon)^2 \left\| \begin{pmatrix} -fl\left[\frac{l}{l+1} P_{m-1}\left(l+1, \cos\frac{\pi k}{d}\right)\right] & fl\left[P_m\left(l, \cos\frac{\pi k}{d}\right)\right] \\ -fl\left[\frac{l}{l+1} P_m\left(l+1, \cos\frac{\pi k}{d}\right)\right] & fl\left[P_{m+1}\left(l, \cos\frac{\pi k}{d}\right)\right] \end{pmatrix} \right\|_E^2$$

$$\times \left\| \begin{pmatrix} \mathbf{u}_k \\ \mathbf{v}_k \end{pmatrix} \right\|_2^2.$$

In Lemma 4, we considered the infinite precision versions of the 2×2 floating point matrices appearing in the previous sum. There we saw that the largest Euclidean norm of these matrices occurred at $k = 0$, with a

value $B(l, m)$. Applying Eq. (27), we can bound the previous expression by

$$2\epsilon B(l, m)(1 + \epsilon)\left\|\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}\right\|_2.$$

Putting the two pieces together results in an error bound for the application of M to $\mathbf{w}$ of $3\epsilon B(l, m)\|\mathbf{w}\|_2 + O(\epsilon^2)$.

Finally, recall that the l^2 operator norms of the Fourier transform and its inverse are both 1. We now apply the result of Lemma 5 with these various bounds, and see that the l^2 error bound for applying $T_d(l, m)$ to $\mathbf{z}$ is indeed $(3K \log_2 d + 1)B(l, m)\epsilon\|\mathbf{z}\|_2 + O(\epsilon^2)$. □

The error accumulated by the algorithm after the construction of Z_0 and Z_1 accounts for most of the error of the algorithm. The error of the algorithm at a particular Legendre coefficient depends on the particular sequence of operations utilized to reach that point from the initial vectors. These operator sequences may be parameterized by paths in a tree describing the computation from the initial pair of vectors, as shown in Fig. 4.

Each path consists of a particular sequence of left or right moves in the tree. Left moves are merely projections and incur no floating point error. Right moves incur the floating point error of the convolution operators, as described in Lemma 6. To obtain a bound on the error of a given pair of output coefficients, one applies Lemmas 5 and 6 to the sequence of operators required to reach it.

These operator sequences, and therefore the associated error bounds, can be parameterized by the paths in the tree or, equivalently, by the binary representation of the leaves in the tree, which correspond to pairs of output coefficients. For instance, the first output pair is $00 \cdots 0$, corresponding to all left moves in the tree. The last output pair is $11 \cdots 1$, corresponding to all right moves.

LEMMA 7. *Consider the output pair* $(Z^{(2)}_{2\beta}, Z^{(2)}_{2\beta+1})$ *of the size* $n = 2^k$ *Legendre transform algorithm, where* β *has binary representation* $\sum_{l=1}^{k-1}\beta_l 2^{k-l-1}$. *Then we have the floating point error bound*

$$\sqrt{\left|fl\left(Z_{2\beta}(0)\right)\right| - Z_{2\beta}(0)|^2 + |fl\left(Z_{2\beta+1}(0)\right) - Z_{2\beta+1}(0)|^2}$$
$$\leq \epsilon A(\beta)\sqrt{\|Z_0\|_2^2 + \|Z_1\|_2^2} + O(\epsilon^2).$$

The factor $A(\beta)$ *is* $\prod_{l=1}^{k-1}M_l(\beta)\sum_{l=1}^{k-1}N_l(\beta)$ *with*

$$M_l(\beta) = \begin{cases} 1 & \text{if } \beta_l = 0 \\ B\left(1 + \sum_{j=1}^{l-1}\beta_j 2^{k-j}, \frac{n}{2^l}\right) & \text{if } \beta_l = 1, \end{cases}$$

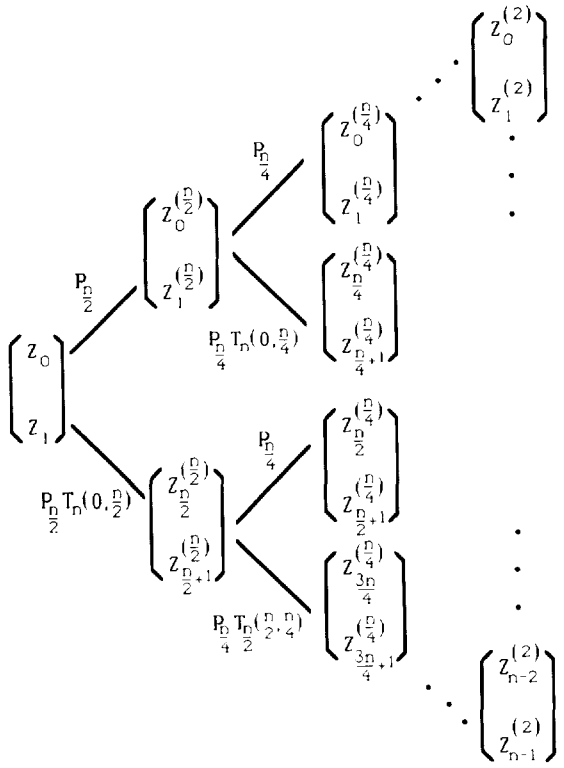

FIG. 4. A tree describing the computations of the Legendre transform algorithm.

$B(l, m)$ as defined in Lemma 4, *and*

$$N_l(\beta) = \begin{cases} 0 & \text{if } \beta_l = 0 \\ 2K(k-l) + 3 & \text{if } \beta_l = 1, \end{cases}$$

where the constant K is defined in Lemma 6.

Proof. The output pair $(Z^{(2)}_{2\beta}, Z^{(2)}_{2\beta+1})$ is obtained from the input pair Z_0, Z_1 by means of the operator sequence $\Lambda(\beta) = \Lambda_{k-1}(\beta) \cdots \Lambda_1(\beta)$ applied in floating point, where

$$\Lambda_l(\beta) = \begin{cases} P_{n/2^l} & \text{if } \beta_l = 0 \\ P_{n/2^l} T_{n/2^{l-1}}\left(1 + \sum_{j=1}^{l-1} \beta_j 2^{k-j}, \dfrac{n}{2^l}\right) & \text{if } \beta_l = 1. \end{cases}$$

By Lemma 5, we can compute a floating point error bound for applying this sequence, given floating point bounds and operator norm bounds on the elements of the sequence $\Lambda_l(\beta)$. The projections have operator norm 1, and introduce no floating point errors. Therefore, by Lemma 4,

$$\|\Lambda_l(\beta)\|_2 = \begin{cases} 1 & \text{if } \beta_l = 0 \\ B\left(1 + \sum_{j=1}^{l-1} \beta_j 2^{k-j}, \frac{n}{2^l}\right) & \text{if } \beta_l = 1, \end{cases}$$

and Lemma 6 gives the error bound

$$\|fl(\Lambda_l \mathbf{z}) - \Lambda_l \mathbf{z}\|_2 \begin{cases} = 0 & \text{if } \beta_l = 0 \\ \leq \epsilon(2K \log_2 d + 3) B\left(1 + \sum_{j=1}^{l-1} \beta_j 2^{k-j}, \frac{n}{2^l}\right) & \\ \quad \times \|\mathbf{z}\|_2 + O(\epsilon^2) & \text{if } \beta_l = 1. \end{cases}$$

The final form of the estimate follows from the observation that the floating point error bound may be rewritten

$$\|fl(\Lambda_l \mathbf{z}) - \Lambda_l \mathbf{z}\|_2 \leq \epsilon N(\beta) \|\Lambda_l(\beta)\|_2 \|\mathbf{z}\|_2. \quad \square$$

We may now state and prove a relative error bound for the Legendre transform.

THEOREM 7. *For the size $n = 2^k$ Legendre transform algorithm applied to the samples $f_j = f(\cos \pi j/2n)$ of a band-limited function f, the relative error*

$$|fl(\tilde{f}(l)) - \tilde{f}(l)| / \|f\|_{L^2[-1,1]} = |fl(\tilde{f}(l)) - \tilde{f}(l)| / \|\tilde{f}\|_{l^2(n)}$$

is bounded by

$$\epsilon \sqrt{\frac{(2l+1)\pi}{n}} \prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right) \times \left[\frac{15}{2} + K \log_2 n + \sum_{s=1}^{k-1} N_s\left(\text{Floor}\left[\frac{l}{2}\right]\right)\right] + O(\epsilon^2),$$

with M_r and N_s as in Lemma 7 and Floor$[t]$ *the greatest integer less than or equal to t. For some cases of interest these are described by the plots in Fig. 5 below.*

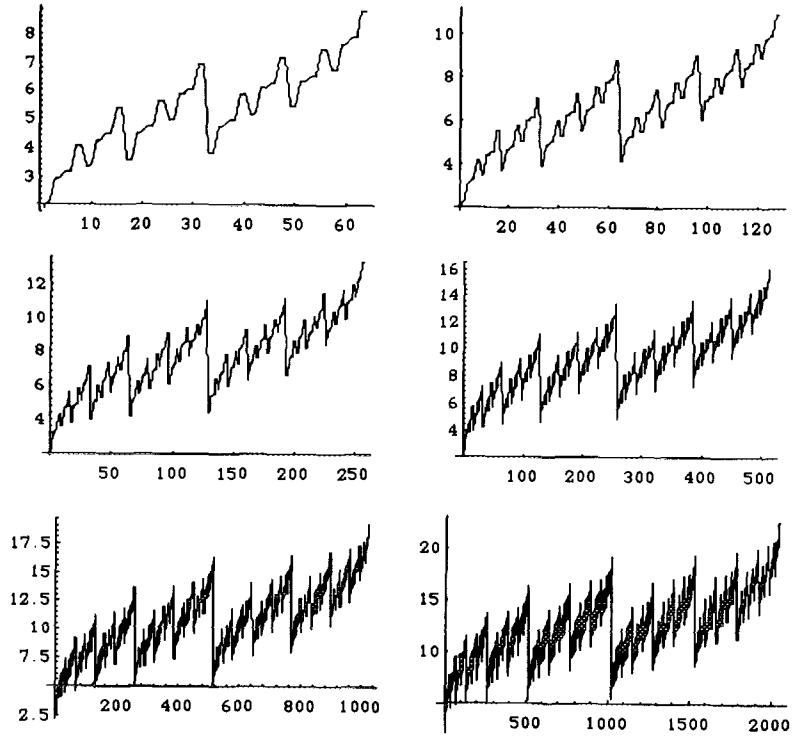

FIG. 5. Plots of the bounds on the relative error, $\log_{10}((1/\epsilon)(|fl(\tilde{f}(l)) - \tilde{f}(l)|/\|f\|_{L^2[-1,1]}))$ for $k = 7, \ldots, 11$ and $K = 6$.

Proof. It remains to combine the estimate of Lemma 7 with bounds on a few additional sources of roundoff error. One of these is incurred upon normalizing the output $Z_l(0)$ by multiplying with the factor $\sqrt{(2l+1)/2}$. There is also error incurred in the startup, namely, the construction of sequences Z_0 and Z_1 from sampled data. This includes the error in producing a floating point approximation to the vector of weighted samples. Next, one has the error incurred by cosine transforming it to create Z_0 and then creating Z_1 by adding two shifted versions of Z_0 together. It turns out that all of these have some small impact on the error.

Let **s** denote the vector of weighted samples, $s_j = a_j f_j$. The algorithm is applied to the floating point approximation of this, $fl(\mathbf{s})$. This produces an error

$$|fl(s_j) - s_j| = |fl[fl(a_j)fl(f_j)] - a_j f_j|,$$

which is bounded by $3\epsilon|s_j|$ using now familiar arguments.

The algorithm proceeds by computing Z_0 and Z_1 from $fl(\mathbf{s})$ by applying the operator sequence

$$\begin{pmatrix} Z_0 \\ Z_1 \end{pmatrix} = FMD\, fl(\mathbf{s}).$$

Here

$$D\mathbf{x} = \begin{pmatrix} \mathbf{h} \\ \mathbf{h} \end{pmatrix}, \qquad \text{with } h(j) = \begin{cases} \frac{1}{2}\mathbf{x}_j & 0 < j < 2n \\ x_0 & j = 0 \\ h(-j) & -2n \le j < 0; \end{cases}$$

$$M\begin{pmatrix} \mathbf{h} \\ \mathbf{h} \end{pmatrix} = \begin{pmatrix} \mathbf{h} \\ C\mathbf{h} \end{pmatrix}, \qquad \text{where } (C\mathbf{h})_j = \cos(j\pi/2n)h(j);$$

and F acts on each of the two components of its input by FFT: $F = \mathscr{F} \oplus \mathscr{F}$.

One verifies easily the operator norms

$$\|D\|_2 = \sqrt{2}, \qquad \|M\|_2 = 1, \qquad \|F\|_2 = 1,$$

and hence the operator norm of the composition, $\|FMD\|_2 = \sqrt{2}$. We now bound the floating point error for the application of each of these operators to a floating point vector.

We have a floating point error bound for F computed by FFT as in the proof of Lemma 6; $\|fl(F\mathbf{z}) - F\mathbf{z}\|_2 \le \epsilon K \log_2 n \|\mathbf{z}\|_2$. Next,

$$fl\left[M\begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix}\right] - M\begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} 0 \\ fl(C\mathbf{x}) - C\mathbf{x} \end{pmatrix};$$

the nonzero components have absolute values $|fl[fl(\cos(\pi j/2n))x_j] - \cos(\pi j/2n)x_j|$

$$\le \left|fl\left[fl(\cos(\pi j/2n))\, x_j\right] - fl(\cos(\pi j/2n))\, x_j\right| + \left|fl(\cos(\pi j/2n)) - \cos(\pi j/2n)\right| |x_j|.$$

The latter term can be bounded by $\epsilon|x_j|$ if the cosine is (pre)computed with sufficient precision. From this, we conclude

$$\left\| fl\left[M\begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix}\right] - M\begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix} \right\|_2 \le 2\epsilon \left\| \begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix} \right\|_2 + O(\epsilon^2).$$

Similarly, one may show that $\|fl(D\mathbf{x}) - D\mathbf{x}\|_2 \le \epsilon\sqrt{2}/2\|\mathbf{x}\|_2$. Lemma 5

now gives the bound

$$\| fl(FMDfl(\mathbf{s})) - FMD(fl(\mathbf{s}))\|_2 \le \epsilon\sqrt{2}\,(K \log_2 n + 5/2)\| fl(\mathbf{s})\|_2.$$

Consider next the normalization. We need to bound

$$fl\Big[fl\Big(\sqrt{(2l+1)/2}\Big)z\Big] - \sqrt{(2l+1)/2}\,z,$$

for z a floating point number. This is bounded in absolute value by

$$\begin{aligned}
&\le \Big|fl\Big[fl\Big(\sqrt{(2l+1)/2}\Big)z\Big] - fl\Big(\sqrt{(2l+1)/2}\Big)z\Big| \\
&\quad + \Big|fl\Big(\sqrt{(2l+1)/2}\Big)z - \sqrt{(2l+1)/2}\,z\Big| \\
&\le \epsilon\Big|fl\Big(\sqrt{(2l+1)/2}\Big)\Big||z| + \Big|fl\Big(\sqrt{(2l+1)/2}\Big) - \sqrt{(2l+1)/2}\Big||z| \\
&\le 2\epsilon\sqrt{(2l+1)/2}\,|z| + O(\epsilon^2).
\end{aligned}$$

The operator norm bound for this multiplication is clearly $\sqrt{(2l+1)/2}$.

We may now combine these bounds with the bounds from Lemma 7 by means of Lemma 5. Let T_l denote the operator applied in the algorithm to obtain the lth Legendre coefficient of f from its weighted samples,

$$T_l = \sqrt{\frac{2l+1}{2}}\,\Lambda\left(\text{Floor}\left[\frac{l}{2}\right]\right)FMD,$$

where Λ is described in the proof of Lemma 7. Then

$$\begin{aligned}
&|fl[T_l(fl(\mathbf{s}))] - T_l(fl(\mathbf{s}))| \\
&\quad \le 2\epsilon\sqrt{\frac{2l+1}{2}}\prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right)\sqrt{2}\|\mathbf{s}\|_2 \\
&\qquad + \sqrt{\frac{2l+1}{2}}\,\epsilon\prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right)\left[\sum_{s=1}^{k-1} N_s\left(\text{Floor}\left[\frac{l}{2}\right]\right)\right]\sqrt{2}\|\mathbf{s}\|_2 \\
&\qquad + \sqrt{\frac{2l+1}{2}}\prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right)\sqrt{2}\,\epsilon(K \log_2 n + 5/2)\|\mathbf{s}\|_2 + O(\epsilon^2)
\end{aligned}$$

so that $|fl[T_l(fl(\mathbf{s}))] - T_l(fl(\mathbf{s}))|$

$$\begin{aligned}
&\le \epsilon\sqrt{2l+1}\prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right) \\
&\quad \times\left[\frac{9}{2} + K \log_2 n + \sum_{s=1}^{k-1} N_s\left(\text{Floor}\left[\frac{l}{2}\right]\right)\right]\| fl(\mathbf{s})\|_2 + O(\epsilon^2).
\end{aligned}$$

Now we can apply this result to help bound

$$\begin{aligned}\left|fl\left[\tilde{f}(l)\right] - \tilde{f}(l)\right)\right| &= \left|fl\left[T_l(fl(\mathbf{s}))\right] - T_l(\mathbf{s})\right| \\ &\le \left|fl\left[T_l fl(\mathbf{s})\right] - T_l(fl(\mathbf{s}))\right| + \left|T_l(fl(\mathbf{s})) - T_l(\mathbf{s})\right|.\end{aligned}$$

The latter term is bounded by

$$\|T_l\|_2 \| fl(\mathbf{s}) - \mathbf{s}\|_2 \le \|T_l\|_2 3\epsilon \|\mathbf{s}\|_2 \le 3\epsilon \|\mathbf{s}\|_2 \sqrt{2l+1} \prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right).$$

Using this, and observing $\|fl(\mathbf{s})\|_2 = \|\mathbf{s}\|_2 + O(\epsilon^2)$, we have

$$\begin{aligned}\left|fl\left[\tilde{f}(l)\right] - \tilde{f}(l)\right)\right| \le \epsilon\sqrt{2l+1} \prod_{r=1}^{k-1} M_r\left(\text{Floor}\left[\frac{l}{2}\right]\right) \\ \times \left[\frac{15}{2} + K \log_2 n + \sum_{s=1}^{k-1} N_s\left(\text{Floor}\left[\frac{l}{2}\right]\right)\right] \|\mathbf{s}\|_2 \\ + O(\epsilon^2).\end{aligned}$$

To obtain the desired relative error bound, we note that

$$\|\mathbf{s}\|_2^2 = \sum_{j=0}^{2n-1} a_j^2 f^2(\cos(\pi j/2n)) \le \frac{\pi}{n} \sum_{j=0}^{2n-1} a_j f^2(\cos(\pi j/2n)),$$

by our expression (9) for the sampling weights and a simple bound on the overshoot of the partial Fourier sum in that formula. By the assumption on the band-limit, $f = \sum_{l=0}^{n-1} \tilde{f}(l)\sqrt{(2l+1)/2}\, P_l$. Using this and the special case of the sampling theorem in Lemma 3,

$$\begin{aligned}\|\mathbf{s}\|_2^2 &\le \frac{\pi}{n} \sum_{j=0}^{2n-1} a_j f(\cos(\pi j/2n)) \sum_{l=0}^{n-1} \tilde{f}(l) \sqrt{\frac{2l+1}{2}}\, P_l(\cos(\pi j/2n)) \\ &= \frac{\pi}{n} \sum_{l=0}^{n-1} \tilde{f}(l) \sqrt{\frac{2l+1}{2}} \sum_{j=0}^{2n-1} a_j f(\cos(\pi j/2n)) P_l(\cos(\pi j/2n)) \\ &= \frac{\pi}{n} \sum_{l=0}^{n-1} \tilde{f}(l) \sqrt{\frac{2l+1}{2}} \sqrt{\frac{2}{2l+1}}\, \tilde{f}(l),\end{aligned}$$

so $\|\mathbf{s}\|_2 \le \sqrt{\pi/n}\, \|f\|_{L^2[-1,1]}$ by standard properties of orthonormal expansions, and the result follows. □

The algorithm has been coded in C and tested experimentally on a variety of test inputs. The test bed for the experiments is a standard Decstation 5000/200 workstation, with ϵ about 2.2×10^{-16}. The usual

experiment began with the generation of a table of Legendre transform coefficients. These were converted into a corresponding table of function samples on $[-1, 1]$ by means of a naive (slow) inverse transform. Our algorithm was applied to these samples and the result was compared to the initial coefficients.

To insure as much reliability as possible for the samples, the inversion was also done on a Cray, with a double precision wordlength of 128 bits as opposed to the Decstation's 64. In fact, this seemed to make little difference in the precision of the samples for the cases we observed, with agreement of 14 decimal places or better in the several cases for which we made the comparison.

One coefficient test set we considered at the size-1024 level consisted of 30 samples of simulated mean 0 variance 1 Gaussian white noise. Figure 6

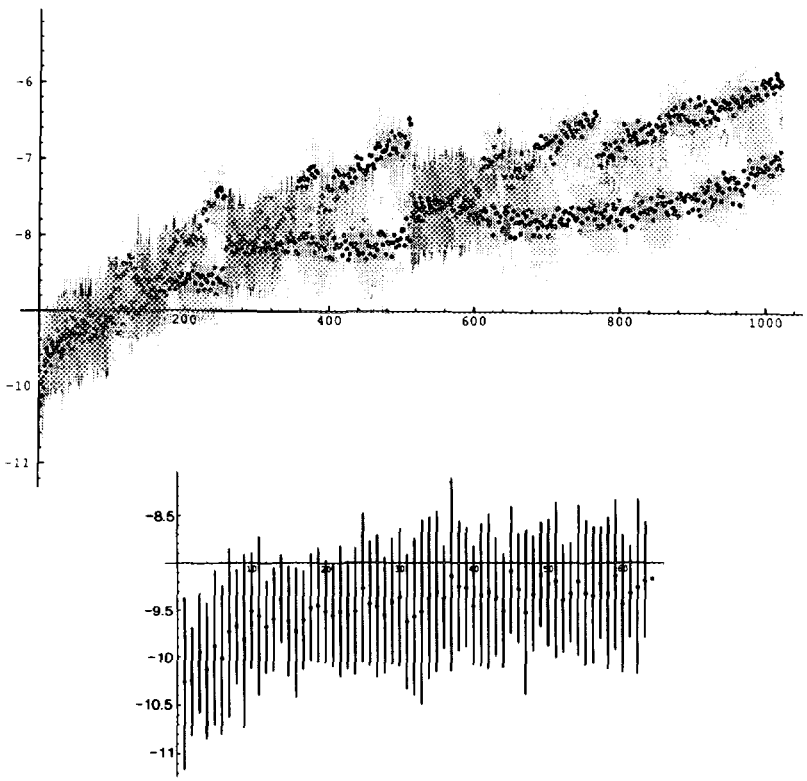

FIG. 6. The dots plot the sample mean value of $\log_{10}(|fl(\tilde{f}(l)) - \tilde{f}(l)| / |\tilde{f}(l)|)$ over 30 samples of $\tilde{f}$ taken from simulated $N(0, 1)$ white noise. The grey error bars indicate the sample standard deviation. The lower portion of the figure shows expanded detail of the plot.

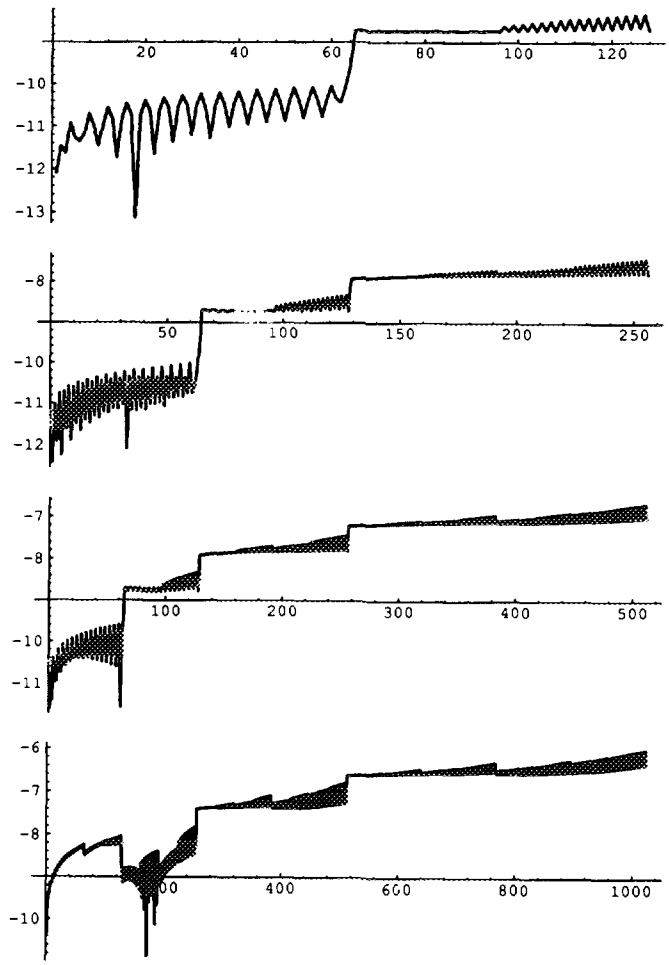

FIG. 7. Values of $\log_{10}(|fl(\tilde{f}(l) - \tilde{f}(l)|/|\tilde{f}(l)|)$ for the function with Legendre coefficients $\tilde{f}(l) = 1$, $l = 0, 1, \ldots, 2^k - 1$. Shown here are the cases $k = 7$, 8, 9, and 10.

shows the mean relative error plots for this set with error bars at plus and minus the sample standard deviation.

The interpolating function with Legendre coefficients all 1 proved to be an interesting test function, with relative error profiles whose shape is easily discerned. Figure 7 shows these profiles for a range of problem sizes.

These examples are representative of our numerical experiments to date. In particular, we have never observed fewer than 4 significant digits

at the size 1024 level. We continue to run numerical experiments on our algorithm. We feel that the simple theoretical and experimental results we have at this time suggest that our algorithm, or a descendent, will actually prove useful on real machines.

7. Fourier Transform on the Sphere

THEOREM 8. *If $f(\theta, \phi)$ is in the span of $\{Y_l^m | l < b/2, |m| \le l\}$, then the Fourier coefficients $\hat{f}(l, m)$ for $l < b/2$, $|m| \le l$ can be computed in $O(n(\log n)^2)$ operations from the $n = b^2$, $n = 2^k < N$, sampled values $f(k\pi/b, j2\pi/b)$, $0 \le j, k < b - 1$, using a preprocessed data structure of size $O(N(\log N)^2)$.*

Proof. From the sampling theorem we know that the transform may be computed by means of finite sums,

$$\hat{f}(l, m) = \sum_{j=0}^{b-1} \sum_{k=0}^{b-1} a_k^{(b/2)} f(k\theta, j\phi) Y_l^m(k\theta, j\phi),$$

where $\theta = \pi/b$, $\phi = 2\pi/b$, and the $a_k^{(b/2)}$ are as defined in the sampling theorem, Theorem 3. Rewriting, we get

$$\hat{f}(l, m) = q_l^m \sum_{k=0}^{b-1} a_k^{(b/2)} P_l^m(\cos k\theta) \sum_{j=0}^{b-1} e^{-imj\phi} f(k\theta, j\phi),$$

where the q_l^m are the normalization coefficients for the spherical harmonics. The inner sum is computed for each fixed k and for all m in the appropriate range by means of the Fast Fourier transform; the resulting table $\check{f}(k\theta, m)$ is obtained in $O(b^2 \log b)$.

It remains to compute

$$\hat{f}(l, m) = q_l^m \sum_{k=0}^{b-1} a_k^{(b/2)} \check{f}(k\theta, m) P_l^m(\cos k\theta),$$

which has the form of an associated Legendre transform for each fixed m. Each transform can be accomplished in $O(b(\log b)^2)$ time, and since m ranges over $O(b)$ values, the total time to compute all values of $\hat{f}(l, m)$ is $O(b^2(\log b)^2)$. □

THEOREM 9. *If $\hat{f}(l, m) = 0$ for $l \ge b$, then the inverse transform of $\hat{f}$ can be computed at $n = O(b^2)$ points, $n = 2^k$, in $O(n^{1.5})$ time.*

Proof. The transform is inverted by the sum

$$f(k\theta, j\phi) = \sum_{l=0}^{b-1} \sum_{m=-l}^{l} \hat{f}(l, m) Y_l^m(k\theta, j\phi),$$

where $\theta = \pi/b$ and $\phi = 2\pi/b$. If we rearrange and exchange sums by extending $\hat{f}$ to be 0 at inappropriate indices, we get

$$f(k\theta, j\phi) = \sum_{m=1-b}^{b-1} e^{-imj\phi} \sum_{l=0}^{b-1} \hat{f}(l,m) q_l^m P_l^m(\cos k\theta).$$

Take $g(k\theta, m)$ to be the sum

$$\sum_{l=0}^{b-1} \hat{f}(l,m) q_l^m P_l^m(\cos k\theta)$$

The $O(b^2)$ values for g can each be determined by summing the $O(b)$ terms, for a total of $O(b^3)$ time. Then

$$f(k\theta, j\phi) = \sum_{m=1-b}^{b-1} e^{-imj\phi} g(k\theta, m)$$

can be determined also by a naive summation in $O(b^3)$ time. □

Combining the previous theorems and the convolution theorem, Theorem 1, we obtain the following.

THEOREM 10. *If f and g are band-limited functions on the sphere (i.e., $\hat{f}(l,m)$ and $\hat{g}(l,m) = 0$ for $l \geq b$), then $f * g$, the spherical convolution of f and g, can be computed in $O(n^{1.5})$ time, for $n = O(b^2)$, $n = 2^k < N$, using a precomputed data structure of size $O(N(\log N)^2)$.*

8. CONCLUSION

We have given natural convolution theorems for functions on the sphere and used these to produce an asymptotically optimal sampling theorem for band-limited functions. That is to say, we require a fixed multiple of the minimum number of samples required by linear algebra.

The sampling result was applied to the problem of efficiently computing spherical harmonic expansions, and an $O(n(\log n)^2)$ time algorithm was given to compute these for functions sampled at $n = 2^k$ points on an equiangular grid. This improves the naive bound of $O(n^2)$, the best previously known algorithm.

A special case of the algorithm was studied theoretically and experimentally for the effects of finite-precision arithmetic. The results are generally encouraging and suggest that the algorithm or a modification may be useful in actual problems.

We have also described an $O(n^{1.5})$ inversion algorithm, which is then used to efficiently compute convolutions of functions on the sphere.

Issues under investigation include the following.

• Our sampling theorem requires an asymptotically optimal number of sampled points for a given band-limited function. Is it possible to exhibit an "optimal" sampling? That is, if a function has no energy for $l \geq b$, then there are at most $(b-1)(b-2)+b$ nonzero coefficients in the transform, and we ask if there is a sampling of the function at $(b-1)(b-2)+b$ points that can exactly recover the transform?

• The algorithms presented here are composed of iterated convolutions. We have described an FFT approach to computing these convolutions, but there are others. For example, use of fast cosine transforms rather than the FFT will cut the running time in many instances.

• In the current work, the inversion of the Legendre transform is done naively in $O(n^2)$ time, making the inverse transform the rate determining step for convolution. We are currently considering a more efficient approach to the inverse transform.

• We have presented a study of the impact of finite precision arithmetic on the algorithm for the Legendre polynomial transform. Our analysis suggests that the worst errors are rather localized in the transform domain, and we are considering modifications to reduce these errors. We are also studying similar results for the associated Legendre functions and the inverse transform.

• The technique presented here makes use of the three-term recurrence formula for the Legendre functions. Many other special functions are described by such formulas. The extension of our results to these other transforms will require proper sampling schemes.

Acknowledgments

We thank Audrey Terras for suggesting this problem to us and for many pleasant discussions over the years. We thank Persi Diaconis and Dan Rockmore for inspiration and encouragement. Dick Askey taught us about associated Legendre polynomials. Joe Gregorio and Joe DeStefano provided beautiful C code. Finally, we offer our appreciation of the very careful reading and many helpful suggestions provided by the referees.

References

1. B. Alpert and V. Rokhlin, "A Fast Algorithm for the Evaluation of Legendre Expansions," *Research Report YALEU/DCS/RR-671*, Yale University, Department of Computer Science, 1989.
2. A. Aho, J. Hopcroft, and J. Ullman, "The Design and Analysis of Computer Algorithms," Addison Wesley, Reading, MA, 1976.

3. L. AUSLANDER AND R. TOLMIERI, Is computing with the Fast Fourier transform pure or applied mathematics?, *Bull. Amer. Math. Soc.* (N. S.) **1** (1979), 847–897.
4. P. BARRUCAND AND D. DICKINSON, On the associated Legendre polynomials, *in* "Orthogonal Expansions and Their Continuous Analogues," Southern Illinois University Press, Carbondale, IL, 1968.
5. T. BETH, On the computational complexity of the general discrete Fourier transform, *Theoret. Comput. Sci.* **51** (1987), 331–339.
6. L. C. BIEDENHARN AND J. D. LOUCK, "Angular Momentum in Quantum Mechanics," Addison Wesley, Reading, MA, 1981.
7. R. BULRISCH AND J. STOER, "Introduction to Numerical Analysis," Springer-Verlag, New York, 1983.
8. D. CALVETTI, A stochastic roundoff error analysis for the fast Fourier transform, *Math. Comput.* **56**, No. 194 (1991), 755–774.
9. D. V. S. CHANDRA, Accumulation of coefficient roundoff error in fast Fourier transforms implemented with logarithmic number system, *IEEE Trans. Acoust. Speech Signal Process*, **35**, No. 11 (1987), 1633–1636.
10. M. CLAUSEN, Spectral transforms for the symmetric group, *in* "Proc., Third International Workshop on Spectral Techniques, 1988."
11. J. W. COOLEY AND J. W. TUKEY, An algorithm for machine calculation of complex Fourier series, *Math. Comput.* **19** (1965), 297–301.
12. H. S. M. COXETER, "Regular Polytopes," Dover, New York, 1973.
13. P. DIACONIS, Average running time of the fast Fourier transform, *J. Algorithms* **1** (1980), 187–208.
14. P. DIACONIS, "Group Representations in Probability and Statistics," IMS, Hayward, CA, 1988.
15. P. DIACONIS AND D. ROCKMORE, Efficient computation of the Fourier transform on finite groups, *J. Amer. Math. Soc.* **3**, No. 2 (1990), 297–332.
16. G. A. DILTS, Computation of spherical harmonic expansion coefficients via FFTs, *J. Comput. Phys.*, **57**, No. 3 (1985), 439–453.
17. W. FREEDEN, On integral formulas of the (unit) sphere and their application to numerical computation of integrals, *Computing* **25** (1980), 131–146.
18. F. A. GILBERT, Inverse problems for the Earth's normal modes, *in* "Mathematical Problems in the Geophysical Sciences," Vol. I, American Mathematical Society, Providence, RI. 1971.
19. N. C. GALLAGHER, G. L. WISE, AND J. W. ALLEN, A novel approach for the computation of Legendre polynomial expansions, *IEEE Trans. Acoust. Speech Signal Process.* **26**, No. 1 (1978), 105–106.
20. H. GOLDSTEIN, "Classical Mechanics," Addison Wesley, Reading, MA, 1980.
21. G. H. GOLUB AND C. F. VAN LOAN, "Matrix Computations," 2nd ed., Johns Hopkins University Press, Baltimore, 1989.
22. M. HAMMERMESH, "Group Theory," Addison Wesley, Reading, MA, 1964.
23. D. M. HEALY, "A Relationship between Harmonic Analysis on SU(2) and on SL(2, C)/SU(2)," Ph.D. thesis, U.C. San Diego, 1986.
24. S. HELGASON, "Groups and Geometric Analysis," Academic Press, New York, 1984.
25. G. T. HERMAN (Ed.), "Image Reconstruction from Projections," Springer-Verlag, New York, 1979.
26. D. V. JAMES, Quantization errors in the fast Fourier transform, *IEEE Trans. Acoust. Speech Signal Process.* **23**, No. 3 (1975), 277–283.
27. T. KANEKO AND B. LIU, Accumulation of round-off error in fast Fourier transforms, *J. Assoc. Comput. Mach.* **17**, No. 4 (1970), 637–654.

28. W. KNIGHT AND R. KAISER, A simple fixed-point error bound for the fast Fourier transform, *IEEE Trans. Acoust. Speech Signal Process.* **27**, No. 6 (1979), 615–620.
29. K. KOBAYASHI, Solution of multi-dimensional neutron transport equation of the spherical harmonics method using the finite Fourier transformation and quadrature formula, *Transport Theory Statist. Phys.* **14** (1985), 63–81.
30. A. MESSIAH, "Quantum Mechanics," Vol. II, Wiley, New York, 1958.
31. J. MORGENSTERN, Note on a lower bound of the linear complexity of the fast Fourier transform, *J. Assoc. Comput. Mach.* **20** (1973), 305–306.
32. S. D. MORGERA, Efficient synthesis and implementation of large discrete Fourier transformations, *SIAM J. Comput.* **9** (1980), 251–272.
33. D. C. MUNSON, Floating point roundoff error in the fast Fourier transform, *IEEE Trans. Acoust. Speech Signal Process.* **29**, No. 4 (1981), 877–882.
34. W. NEUTSCH, Optimal spherical designs and numerical integration on the sphere, *J. Comput. Phys.* **51** (1983), 313–325.
35. A. V. OPPENHEIM AND C. J. WEINSTEIN, Effects of finite register length in digital filtering and the fast Fourier transform, *Proc. IEE-E* **60**, No. 8 (1972), 957–976.
36. S. A. ORZAG, Fast eigenfunction transforms, *in* "Science and Computers," Academic Press, Orlando, 1986.
37. G. PANDA, R. N. PAL, AND B. CHATTERJEE, Error analysis of Good–Winograd algorithm assuming correlated truncation errors, *IEEE Trans. Acoust. Speech Signal Process.* **31**, No. 2 (1983), 508–512.
38. C. H. PAPADIMITRIOU, Optimality of the fast Fourier transform, *J. Assoc. Comput. Mach.* **26** (1979), 95–102.
39. S. PRAKASH AND V. V. RAO, Vector radix FFT error analysis, *IEEE Trans. Acoust. Speech Signal Process.* **30**, No. 5 (1982), 808–811.
40. G. U. RAMOS, Roundoff error analysis of the fast Fourier transform, *Math. Comp.* **25** (1971), 757–768.
41. D. ROCKMORE, Computation of Fourier transforms on the symmetric group, *in* "Computers and Mathematics," Springer-Verlag, New York, 1989.
42. J. T. SCHWARTZ, Mathematics addresses problems in computer vision for advanced robotics, *SIAM News* **18** (1985), 3.
43. L. SCHWARTZ, "Theorie des Distributions," Hermann, Paris, 1966.
44. A. TAM, "Optimal Choice of Directions for the Reconstruction of an Object from Its Plane Integrals, Ph.D. thesis, U.C. Berkeley, 1982.
45. A. A. TERRAS, "Harmonic Analysis on Symmetric Spaces I," Springer-Verlag, New York, 1985.
46. TRAN-THONG AND B. LIU, Fixed point fast Fourier transform error analysis, *IEEE Trans. Acoust. Speech Signal Process.* **24**, No. 6 (1976), 563–573.
47. C. VAN LOAN, "Computational Framework for the Fast Fourier Transform," SIAM, Philadelphia, 1992.
48. N. J. VILENKIN, "Special Functions and the Theory of Group Representations," American Mathematical Society, Providence, RI, 1968.
49. J. H. WILKINSON, "Rounding Errors in Algebraic Processes," Prentice–Hall, Englewood Cliffs, NJ, 1963.
50. J. H. WILKINSON, "The Algebraic Eigenvalue Problem," Oxford Univ. Press, Oxford, 1988.
51. S. WINOGRAD, On computing the discrete Fourier transform, *Math. Comp.* **32** (1978), 175–199.