A Fast Algorithm for Spherical Filtering on Arbitrary Grids

Martin Böhme[a] and Daniel Potts[b]

[a]University of Lübeck, D–23560 Lübeck, Germany
[b]Institute of Mathematics, University of Lübeck, D–23560 Lübeck, Germany

ABSTRACT

Spherical filters have recently been introduced in order to avoid the spherical harmonic transform. Spherical filtering can be used in a variety of applications, such as climate modelling, electromagnetic and acoustic scattering, and several other areas. However, up to now these methods have been restricted to special grids on the sphere. The main reason for this was to enable the use of FFT techniques. In this paper we extend the spherical filter to arbitrary grids by using the the Nonequispaced Fast Fourier Transform (NFFT).[1] The new algorithm can be applied to a variety of different distributions on the sphere, equidistributions on the sphere being an important example. The algorithm's performance is illustrated with several numerical examples.

Keywords: spherical filter, spherical Fourier transform, spherical harmonics, associated Legendre functions, fast discrete transforms, fast Fourier transform at nonequispaced knots, Wavelets, fast discrete summation, spherical polynomials

1. INTRODUCTION

This paper considers the problem of spherical filtering on nonequispaced grids. The need to filter functions on the sphere arises in a number of applications. The spherical filter, based on the fast multipole method, was suggested by Jakob-Chien and Alpert[2] and improved by Yarvin and Rokhlin.[3] These algorithms require $\mathcal{O}(N^2 \log N)$ operations for $\mathcal{O}(N^2)$ grid points. In[4] we presented an algorithm that avoids the Fast Multipole Method and uses a fast summation algorithm[5] that is based on the NFFT. We believe that this algorithm is conceptually simpler and easier to implement, and the measurements performed in[5] demonstrate that its running times are at least as good as those of FMM-type algorithms. An additional advantage of our approach is that the NFFT builds on the standard (equispaced) Fast Fourier Transform, for which highly optimised libraries are available (see also[6]). The asymptotic complexity of the NFFT-based summation algorithm depends on the distribution of the nodes at which the sum is to be computed. We showed that, for the Legendre nodes used in the spherical filter, the summation algorithm has a complexity of $\mathcal{O}(N \log N)$, which results in a total complexity for the spherical filter algorithm of $\mathcal{O}(N^2 \log N)$.

In this paper, we now generalise the spherical filter to arbitrary grids using the NFFT.

Note that the spherical filter can also be considered as a method for fast approximation of a function on the unit sphere by a spherical polynomial from the space of all spherical polynomials of degree $\leq N$. The spherical filter algorithm can be used to compute the approximation error of the L_2 orthogonal projection on the grid points in a very efficient way. Furthermore a regridding, i.e. computing values of the projected function on an arbitrary grid, is very efficient. These algorithms also require only $\mathcal{O}(N^2 \log N)$ operations for $\mathcal{O}(N^2)$ grid points. One possible application is to compute a spherical wavelet decomposition of the projected function.[4]

The structure of this paper is as follows. Section 2 presents some basic definitions, including the spherical harmonics, introduces the concept of band-limited functions, and defines the spherical filter, concluding with a summary of the complete spherical filter algorithm on arbitrary grids. Section 3 discusses the fast algorithms that are employed. Section 4 discusses an implementation of the algorithm and presents the results of some numerical experiments performed on it. Finally, Section 5 summarises the results that have been obtained.

http://www.students.uni-luebeck.de/~boehme, E-mail: boehme@informatik.uni-luebeck.de,
http://www.math.uni-luebeck.de/potts, E-mail: potts@math.uni-luebeck.de

2. THE SPHERICAL FILTER

We start with several basic definitions.

DEFINITION 2.1. *The* normalised Legendre polynomials *and the* normalised associated Legendre functions *are defined as*

$$P_k(x) := \frac{1}{2^k k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}(x^2-1)^k \qquad (x \in [-1,1]; k \in \mathbb{N}_0),$$

and

$$P_k^n(x) := \sqrt{\left(k+\frac{1}{2}\right)\frac{(k-n)!}{(k+n)!}}(1-x^2)^{\frac{n}{2}}\frac{\mathrm{d}^n}{\mathrm{d}x^n}P_k(x) \quad (x \in [-1,1]; n,k \in \mathbb{N}_0, k \geq n), \tag{2.1}$$

respectively. For $k < n$, we set $P_k^n(x) := 0$. The functions P_k^n satisfy the three-term recurrence

$$xP_k^n = \alpha_k^n P_{k-1}^n + \alpha_{k+1}^n P_{k+1}^n \tag{2.2}$$

with $\alpha_k^n := \left(\frac{(k-n)(k+n)}{(2k-1)(2k+1)}\right)^{\frac{1}{2}}$ for $k \geq n$ and $\alpha_k^n := 0$ otherwise (see e.g.[7]). Further, the P_k^n fulfil the orthogonality relation

$$\int_{-1}^{1} P_k^n(x)P_l^n(x)\mathrm{d}x = \delta_{k,l} \qquad (n \in \mathbb{N}_0; k,l = n, n+1, \ldots).$$

Throughout this paper, we will be dealing with functions that are defined on the unit sphere $\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\|_2 = 1\}$. In particular, we will focus on the space $L^2(\mathcal{S})$ of square-integrable functions on the sphere. We will describe points on the unit sphere by their latitude $\theta \in [0,\pi]$ and longitude $\phi \in [0, 2\pi)$.

DEFINITION 2.2. *The* spherical harmonics *are functions on the unit sphere defined as*

$$Y_k^n(\theta,\phi) := P_k^{|n|}(\cos\theta)\,\mathrm{e}^{\mathrm{i}n\phi}, \tag{2.3}$$

where $\theta \in [0,\pi]$; $\phi \in [0,2\pi)$; $n \in \mathbb{Z}, k \in \mathbb{N}_0, k \geq |n|$. The spherical harmonics form an orthonormal basis of $L^2(\mathcal{S})$ with respect to the scalar product

$$\langle f,g\rangle := \frac{1}{2\pi}\int_0^{\pi}\int_0^{2\pi} f(\theta,\phi)\,\overline{g(\theta,\phi)}\sin\theta\,\mathrm{d}\phi\,\mathrm{d}\theta \qquad (f,g \in L^2(\mathcal{S})).$$

We will now introduce the concept of band-limited functions and use these to define the spherical filter.

DEFINITION 2.3. *A function $f \in L^2(\mathcal{S})$ is said to be* band-limited *with bandwidth N if it can be expressed as*

$$f(\theta,\phi) = \sum_{k=0}^{N}\sum_{n=-k}^{k} a_k^n\,Y_k^n(\theta,\phi) \qquad (a_k^n \in \mathbb{C}).$$

B_N denotes the set of all band-limited functions with bandwidth N.

For any N, B_N is a subspace of $L^2(\mathcal{S})$.

DEFINITION 2.4. *The orthogonal projection of a function $f \in L^2(\mathcal{S})$ onto B_N is called the* spherical filter with bandwidth N. *The result of this operation is denoted by f^N, i.e. if*

$$f = \sum_{n\in\mathbb{Z},k\in\mathbb{N}_0,k\geq|n|} a_k^n\,Y_k^n,$$

then

$$f^N(\theta,\phi) = \sum_{k=0}^{N} \sum_{n=-k}^{k} a_k^n \, Y_k^n(\theta,\phi).$$

This operation is also referred to as a "triangular truncation", since the spherical Fourier coefficients in f^N can be arranged in the shape of a triangle.

The Fourier coefficients

$$\begin{aligned} a_k^n = \langle f, Y_k^n \rangle &= \frac{1}{2\pi} \int_0^{\pi} \int_0^{2\pi} f(\theta,\phi) \; P_k^{|n|}(\cos\theta) \mathrm{e}^{-in\phi} \sin\theta \; \mathrm{d}\phi \; \mathrm{d}\theta \\ &= \int_0^{\pi} P_k^{|n|}(\cos\theta) \sin\theta \; \frac{1}{2\pi} \int_0^{2\pi} f(\theta,\phi) \mathrm{e}^{-in\phi} \; \mathrm{d}\phi \; \mathrm{d}\theta \end{aligned}$$

have to be computed from the function values. For convenience, we denote the inner integral, together with the normalisation constant $\frac{1}{2\pi}$, by $\mathfrak{f}_n(\theta)$ and thus obtain

$$\mathfrak{f}_n(\theta) := \frac{1}{2\pi} \int_0^{2\pi} f(\theta,\phi) \mathrm{e}^{-in\phi} \; \mathrm{d}\phi \tag{2.4}$$

and

$$a_k^n = \int_0^{\pi} P_k^{|n|}(\cos\theta) \mathfrak{f}_n(\theta) \sin\theta \; \mathrm{d}\theta. \tag{2.5}$$

Now we approximate this inner product by application of a quadrature formula for the spherical Fourier coefficients $a_k^n = \langle f, Y_k^n \rangle$ such that $a_k^n \approx \tilde{a}_k^n$ with

$$\tilde{a}_k^n := \sum_{s=0}^{M} \sum_{t=0}^{M_s - 1} \omega_{t,s} f(\theta_s, \phi_{t,s}) Y_k^{-n}(\theta_s, \phi_{t,s}), \tag{2.6}$$

using suitable weights $\omega_{t,s}$ and points $(\theta_s, \phi_{t,s})$ $(s = 0, \dots, M; t = 0, \dots, M_s - 1)$. Now we define the approximation $\tilde{f}^N$ of f^N by

$$\tilde{f}^N(\tilde{\theta},\tilde{\phi}) = \sum_{k=0}^{N} \sum_{n=-k}^{k} \tilde{a}_k^n \, Y_k^n(\tilde{\theta},\tilde{\phi}). \tag{2.7}$$

Our aim is now to avoid the costly computation of the discrete spherical Fourier coefficients[8] $\tilde{a}_k^n$ and compute values of the function $\tilde{f}^N$ on special grids. For this purpose we use the technique of spherical filtering. Up to now this method has only been applied on the equidistant grids.[2, 3] The method is based on the application of the fast multipole method (FMM). In[4] we avoid the FMM, use only FFTs (or NFFTs) and provide a complete complexity estimate. With the help of the NFFT, we can now generalise the spherical filter method and apply this method to nonequispaced grids for the first time.

Now our algorithm can be viewed as a fast method to construct approximations of a function on the unit sphere by a spherical polynomial from the space of all spherical polynomials of degree $\leq N$.[9, 10] This includes the L_2 orthogonal projection and the hyperinterpolation approximation[10] in which the Fourier coefficients are approximated by a positive weight quadrature rule that integrates exactly all polynomials of a certain degree. One main advantage of our method is that we can compute the approximation error $f - f^N$ on the given points in a very fast way, as well as computing values of f^N on other grids.

Note that the main idea of the spherical filter is that the a_k^n or $\tilde{a}_k^n$ need never be calculated explicitly. This means that we combine formula (2.6) and formula (2.7). For this purpose we need

THEOREM 2.5 (CHRISTOFFEL-DARBOUX FORMULA (see e.g.[7])).
The sum

$$S_N^n(x,y) := \sum_{k=n}^{N} P_k^n(x)\; P_k^n(y) \tag{2.8}$$

possesses the closed form

$$S_N^n(x,y) = \begin{cases} \dfrac{\alpha_{N+1}^n \left(P_{N+1}^n(x)P_N^n(y) - P_N^n(x)P_{N+1}^n(y)\right)}{x-y} & \text{if} \quad x \neq y \\ \alpha_{N+1}^n \left(P_{N+1}^{n\,\prime}(x)P_N^n(x) - P_N^{n\,\prime}(x)P_{N+1}^n(x)\right) & \text{if} \quad x = y, \end{cases} \tag{2.9}$$

where the α_k^n *are the constants from the three-term recurrence* (2.2).

The Christoffel-Darboux formula for the case where $x = y$ requires the evaluation of $P_k^{n\,\prime}$, the derivatives of the normalised associated Legendre functions. A recurrence equation for the $P_k^{n\,\prime}$ can be obtained by differentiating the three-term recurrence $xP_k^n = \alpha_k^n P_{k-1}^n + \alpha_{k+1}^n P_{n+1}^n$, yielding

$$P_k^n + xP_k^{n\,\prime} = \alpha_k^n P_{k-1}^{n\,\prime} + \alpha_{k+1}^n P_{k+1}^{n\,\prime}. \tag{2.10}$$

To start the recurrence, we need $P_n^{n\,\prime}$. We have

$$P_n^n(x) = \frac{1}{2^n n!}\left[\frac{2n+1}{2}\,(2n)!\right]^{1/2} (1-x^2)^{n/2}$$

Denoting $c := \frac{1}{2^n n!}\left[\frac{2n+1}{2}\,(2n)!\right]^{1/2}$ we obtain

$$P_n^{n\,\prime}(x) = c \cdot \left[\frac{n}{2}\,(1-x^2)^{(n-2)/2} \cdot (-2x)\right] = -cnx(1-x^2)^{(n-2)/2}.$$

For the $\tilde{f}^N$ in (2.7), we use (2.3), (2.6) and (2.9) to obtain the formula

$$\begin{aligned} \tilde{f}^N(\tilde{\theta},\tilde{\phi}) &= \sum_{n=-N}^{N} \sum_{k=|n|}^{N} \tilde{a}_k^n\, Y_k^n(\tilde{\theta},\tilde{\phi}) \\ &= \sum_{n=-N}^{N} \sum_{k=|n|}^{N} \sum_{s=0}^{M} \sum_{t=0}^{M_s-1} \omega_{t,s} f(\theta_s,\phi_{t,s}) P_k^{|n|}(\cos\theta_s) \mathrm{e}^{-\mathrm{i}n\phi_{t,s}} P_k^{|n|}(\cos\tilde{\theta}) \mathrm{e}^{\mathrm{i}n\tilde{\phi}} \\ &= \sum_{n=-N}^{N} \left(\sum_{s=0}^{M} \left(\left(\sum_{t=0}^{M_s-1} \omega_{t,s} f(\theta_s,\phi_{t,s}) \mathrm{e}^{-\mathrm{i}n\phi_{t,s}} \right) S_N^n(\theta_s,\tilde{\theta}) \right) \right) \mathrm{e}^{\mathrm{i}n\tilde{\phi}}\,. \end{aligned} \tag{2.11}$$

In order to compute $\tilde{f}^N(\tilde{\theta},\tilde{\phi})$ on a nonequidistant grid we use the NFFT and the fast summation algorithm, which is also based on the NFFT (see Section 3).

We summarise the fast spherical filter algorithm and show that it has an asymptotic complexity of $\mathcal{O}(N^2 \log N)$.

Algorithm (Fast Spherical Filter)

Input:	$f(\theta_s,\phi_{t,s}) \in \mathbb{C}$ $(s = 0,\dots,M;\ t = 0,\dots,M_s-1)$ (function values on a spherical grid)
Output:	$\tilde{f}^N(\tilde{\theta}_\sigma,\tilde{\phi}_{\tau,\sigma}) \in \mathbb{C}$ $(\sigma = 0,\dots,S;\ \tau = 0,\dots,S_\sigma-1)$ (approximate values for $f^N(\tilde{\theta}_\sigma,\tilde{\phi}_{\tau,\sigma})$)
Constants:	$M, N, S \in \mathbb{N}$

$\omega_{t,s} \in \mathbb{R}$ $(s = 0, \ldots, M; t = 0, \ldots, M_s - 1)$
(weights for quadrature)

$\theta_s, \phi_{t,s} \in \mathbb{R}$ $(s = 0, \ldots, M; t = 0, \ldots, M_s - 1)$
(given grid points)

$\tilde{\theta}_\sigma, \tilde{\phi}_{\tau,\sigma} \in \mathbb{R}$ $(\sigma = 0, \ldots, S; \tau = 0, \ldots, S_\sigma - 1)$
(grid points for $\tilde{f}^N$)

Precomputation: For $n = 0, \ldots, N$, $s = 0, \ldots, M$ compute $P_N^n(\cos\theta_s)$, $P_{N+1}^n(\cos\theta_s)$, $P_N^{n\prime}(\cos\theta_s)$ and $P_{N+1}^{n\prime}(\cos\theta_s)$ using the recurrences (2.2) and (2.10).

1. For $s = 0, \ldots, M$ compute

$$\tilde{\mathfrak{f}}_n(\theta_s) := \begin{cases} \sum_{t=0}^{M_s-1} \omega_{t,s} f(\theta_s, \phi_{t,s})\, \mathrm{e}^{-\mathrm{i}n\phi_{t,s}} & (n = -\min(N, \lfloor M_s/2 \rfloor), \ldots, \min(N, \lfloor (M_s-1)/2 \rfloor) \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

using an NFFT$^{\mathrm{T}}$. Complexity: $\mathcal{O}(MN \log N)$

2. For $n = -N, \ldots, N$ compute

$$\mathfrak{S}_1(n, \sigma) := \sum_{\substack{s=0 \\ \theta_s \neq \tilde{\theta}_\sigma}}^{M} \frac{\tilde{\mathfrak{f}}_n(\theta_s) P_{N+1}^{|n|}(\cos\theta_s)}{\cos\theta_s - \cos\tilde{\theta}_\sigma} \qquad (\sigma = 0, \ldots, S) \tag{2.13}$$

and

$$\mathfrak{S}_2(n, \sigma) := \sum_{\substack{s=0 \\ \theta_s \neq \tilde{\theta}_\sigma}}^{M} \frac{\tilde{\mathfrak{f}}_n(\theta_s) P_{N}^{|n|}(\cos\theta_s)}{\cos\theta_s - \cos\tilde{\theta}_\sigma} \qquad (\sigma = 0, \ldots, S) \tag{2.14}$$

using the NFFT summation algorithm,[4, 5] and then set

$$\mathfrak{g}_n(\tilde{\theta}_\sigma) := \alpha_{N+1}^{|n|} \Big(P_N^{|n|}(\cos\tilde{\theta}_\sigma) \big(\mathfrak{S}_1(n,\sigma) + \sum_{\substack{s=0 \\ \theta_s = \tilde{\theta}_\sigma}}^{M} \tilde{\mathfrak{f}}_n(\theta_\sigma) P_{N+1}^{|n|\prime}(\cos\tilde{\theta}_\sigma) \big) - P_{N+1}^{|n|}(x_\sigma) \big(\mathfrak{S}_2(n,\sigma) + \sum_{\substack{s=0 \\ \theta_s = \tilde{\theta}_\sigma}}^{M} \tilde{\mathfrak{f}}_n(\theta_\sigma) P_{N_{\lim}}^{|n|\prime}(\cos\theta_\sigma) \big) \Big).$$

Complexity: $\mathcal{O}\big(N(N\log(N))\big)$

3. For $\sigma = 0, \ldots, S$ compute

$$\tilde{f}^N(\tilde{\theta}_\sigma, \tilde{\phi}_{\tau,\sigma}) := \sum_{n=-N}^{N} \mathfrak{g}_n(\tilde{\theta}_\sigma) \mathrm{e}^{\mathrm{i}n\tilde{\phi}_{\tau,\sigma}} \qquad (\tau = 0, \ldots, S_s - 1) \tag{2.15}$$

using an NFFT. Complexity: $\mathcal{O}(SN\log N)$

Note that the reasoning behind the complexity estimate for step 2 is not entirely straightforward (see[4]). The overall complexity of the algorithm is $\mathcal{O}\big((N + M + S)N\log(N)\big)$.

3. THE FAST ALGORITHMS USED IN THE SPHERICAL FILTER

In this section, we give references to the fast algorithms that are used in the spherical filter.

3.1. THE NONEQUISPACED FFT

The NFFT is used for the fast evaluation of sums of the forms

$$f(\omega_j) = \sum_{k=-N/2}^{N/2-1} f_k \, \mathrm{e}^{-\mathrm{i}2\pi k \omega_j} \qquad (j = -M/2, \ldots, M/2-1)$$

and

$$h(k) = \sum_{j=-M/2}^{M/2-1} h_j \, \mathrm{e}^{-\mathrm{i}2\pi k \omega_j} \qquad (k = -N/2, \ldots, N/2-1)$$

($\omega_j \in \mathbb{R}$), known as the NDFT and NDFT$^{\mathrm{T}}$, respectively. The so-called NFFT and its relative, the NFFT$^{\mathrm{T}}$, are approximative algorithms with only $\mathcal{O}(N \log N + M)$ arithmetical operations. Further details including error estimates can be found in.[1, 11] Moreover, free NFFT software packages are available.[6]

3.2. A FAST SUMMATION ALGORITHM BASED ON THE NFFT

As stated already in the introduction we use a fast summation algorithm for sums of the form

$$f(y_j) = \sum_{\substack{k=1 \\ k \neq j}}^{N} \beta_k K(y_j - x_k) \quad (j = 1, \ldots, M),$$

where $\beta_k \in \mathbb{C}$, $x_k \in \mathbb{R}$ $(k = 1, \ldots, N)$, $y_j \in \mathbb{R}$ $(j = 1, \ldots, M)$.

The kernel K must be in C^∞, except for the origin, which may exhibit a singularity.[5] If so, we agree to set $K(0) := 0$ so that we will be able to evaluate K for all $x \in \mathbb{R}$. The kernel that we will use in our application is $K(x) = 1/x$; it was investigated in detail in.[4] Note that, in the paper cited, we present a complete complexity estimate for the spherical filter. This complexity estimate can be generalised to other distributions. The accuracy of the summation algorithms can be controlled using parameters a, p and m, and this is also discussed in.[4]

4. NUMERICAL EXAMPLE

We will use points that are approximately uniformly distributed over the sphere (see[12] and[13]). As a specific example we use the points with spherical polar coordinates $\theta_s, \phi_{t,s}$ $(s = 0, \ldots, M; t = 0, \ldots, M_s - 1)$, given by

$$\begin{aligned}
&\theta_s = s\pi/M, \quad (s = 0, \ldots, M), \\
&M_0 = 1, M_M = 1, \\
&M_s = \left\lfloor \frac{2\pi}{\arccos((\cos(\pi/M) - \cos^2\theta_s)/\sin^2\theta_s)} \right\rfloor, \quad (s = 1, \ldots, M-1), \\
&\phi_{t,s} = 2\pi(t + 1/2)/M_s, \quad (s = 0, \ldots, M;\; t = 0, \ldots, M_s - 1).
\end{aligned}$$

The same points are used for $\tilde{\theta}_\sigma$ and $\tilde{\phi}_{\tau,\sigma}$. We use the rectangular rule in (2.4) and the Clenshaw-Curtis rule in (2.5), such that the weights are given by

$$\omega_{t,0} = \omega_{t,0} = \frac{1}{\tilde{M} M_s} \sum_{k=0}^{\tilde{M}}{}'' \frac{1}{1-4k^2} \qquad \left(\tilde{M} := \frac{M}{2}\right)$$

$$\omega_{t,s} = \omega_{t,2\tilde{M}-s} = \frac{2}{\tilde{M} M_s} \sum_{k=0}^{\tilde{M}}{}'' \frac{1}{1-4k^2} \cos\frac{ks\pi}{\tilde{M}} \qquad (s = 1, \ldots, \tilde{M}).$$

The double prime on the sum indicates that the first and the last term are to be halved.

The spherical filter algorithm and auxiliary algorithms were implemented in C++ using double-precision arithmetic. The implementation used the FFT library FFTW 2.1.5 and the linear algebra library CLAPACK 3.0. The compiler used was a prerelease of GCC 3.3 (all optimisations turned on). Numerical experiments were run under Linux 2.4.20 on an AMD Atlon XP 2700+ with 1 GB of RAM.

The algorithm was tested on the following functions:

$$\begin{aligned} f_1(\mathbf{x}) &:= x_1 x_2 x_3 \\ f_2(\mathbf{x}) &:= 0.1\mathrm{e}^{x_1+x_2+x_3} \\ f_3(\mathbf{x}) &:= 0.1\|\mathbf{x}\|_1 \\ f_4(\mathbf{x}) &:= 1/\|\mathbf{x}\|_1 \\ f_5(\mathbf{x}) &:= 0.1\sin^2(1+\|\mathbf{x}\|_1) \end{aligned}$$

The functions were taken from.[14] The function f_1 is a polynomial of low degree and the function f_2 is analytic over $\mathcal{S}$. The function of f_3, f_4 and f_5 have only C^0 continuity. In particular, they are not continously differentiable at points where any component of $\mathbf{x}$ is zero.

To test a function which is continuously differentiable, the function $f_6 \in C^1$ with

$$f_6(\theta,\phi) := \begin{cases} 1 & \text{if } \theta \in [0,\pi/2] \\ (1+3\cos^2\theta)^{-1/2} & \text{if } \theta \in (\pi/2,\pi] \end{cases}$$

was used; this function was taken from.[4] If the function values are interpreted as distances from the origin, then this function describes a half-sphere that is joined to a half-ellipsoid. It is smooth everywhere except at the equator. To better test the behaviour of the spherical algorithm at the poles, we rotate the function by $\frac{\pi}{2}$ so that the discontinuity passes through the poles.

To give an idea of the error introduced by the approximate algorithms, we tested the spherical filter on function f_6, computing the maximum relative error as

$$E := \frac{\hat{\|} f^N - \tilde{f}^N \hat{\|}_\infty}{\hat{\|} f^N \hat{\|}_\infty},$$

where

$$\hat{\|} f \hat{\|}_\infty := \max_{s=0,\dots,M;\ t=0,\dots,M_s-1} f(\theta_s, \phi_{t,s})$$

is a discrete approximation to the maximum norm.

We ran the tests with $N = M = 512$. For the parameters $a = p = 5$ in the fast summation algorithm and the NFFT with the truncation parameter $m = 3$ and Kaiser-Bessel functions, we obtained $E \approx 6 \cdot 10^{-3}$; for $a = p = 8$ and $m = 6$ (a medium precision setting that should be sufficient for many applications), we obtained $E \approx 1.4 \cdot 10^{-7}$; and for $a = p = 12$ and $m = 10$, we obtained $E \approx 10^{-12}$.

Next, we will use the spherical filter to examine the projection error

$$E_{\text{proj}} := \frac{\hat{\|} f - \tilde{f}^N \hat{\|}_\infty}{\hat{\|} f \hat{\|}_\infty},$$

for various N and show how the error decreases at different rates as N increases, depending on the smoothness of the function. For these tests, we use $M = 512$, $a = p = 8$ and $m = 6$.

Figure 1 shows plots of the projection error for the smooth functions f_1 and f_2. The error decreases rapidly and quickly reaches a constant level where further improvements in the approximation are masked by the error introduced by the approximate algorithms.

Contrast this to Figure 2, which plots the projection error for the non-smooth functions f_3 to f_6 (note the different scale on the horizontal axis). As expected, these functions require a much greater N to be represented faithfully.

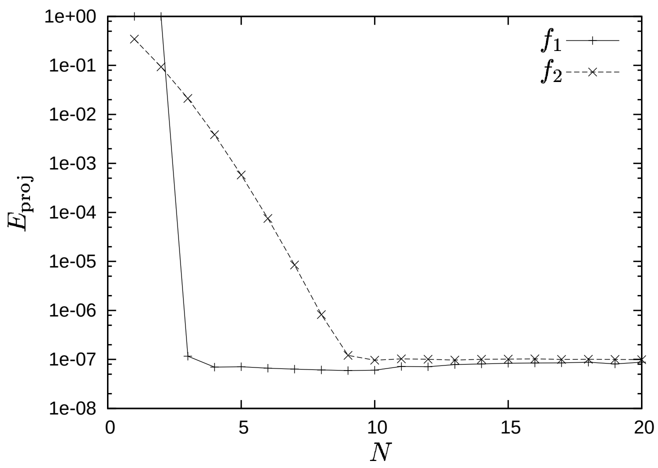

Figure 1. Projection error E_{proj} for f_1 and f_2.

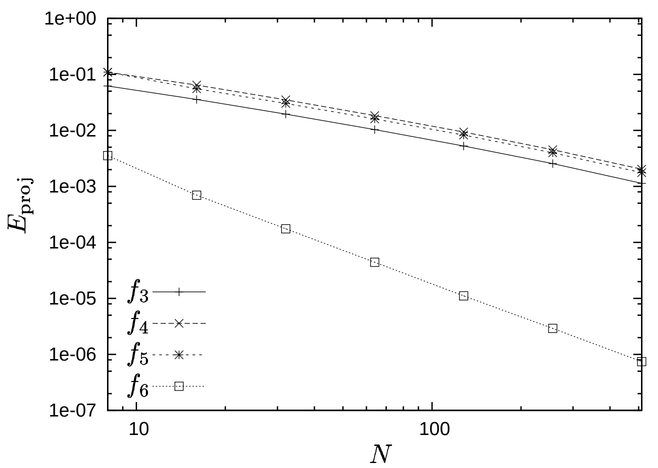

Figure 2. Projection error E_{proj} for f_3 to f_6.

An interesting question is how the performance of the spherical filter on the points we use here compares with the Gauß-Legendre setting, where the point density at the poles is much greater. Figure 3 shows the projection error for the same functions as in Figure 2 but computed using equidistributed points with Gauss-Legendre nodes and weights in the direction of θ (see[4]). It is apparent that the projection errors are virtually the same for both point sets. The results indicate that the approximately uniformly distributed points we use in this paper give results comparable to those of using the Gauß-Legendre setting while requiring less memory to store.

Note that, for the Gauß-Legendre setting, we used $M = 511$, since it requires the number of points to be a power of two, but we believe that this reduction of M by one should not have a significant impact on the results. Note also that the running times for the two point sets do not differ greatly, in spite of the fact that the Gauß-Legendre setting has more points at the poles. This is because the bulk of the running time is consumed by step 2 of the algorithm, where the complexity is independent of the number of points in the direction of ϕ. In fact, the running time for the Gauß-Legendre setting was slightly lower – 21.59 seconds at $M = 1023, N = 1024$ compared to 22.19 seconds for the approximately uniformly distributed points at $M = N = 1024$.

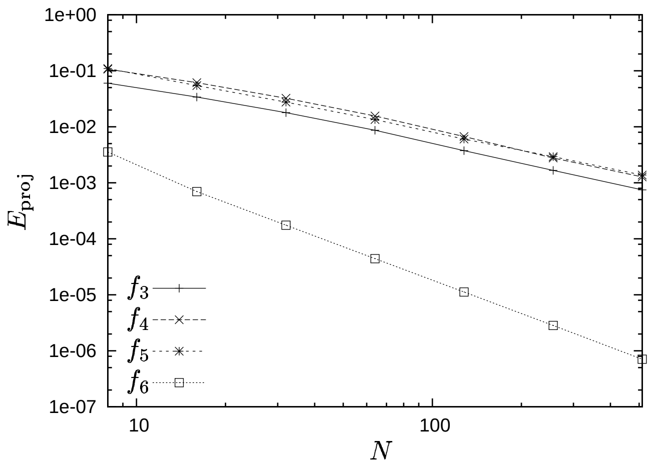

Figure 3. Projection error E_{proj} for f_3 to f_6, computed at the Gauß-Legendre setting as defined in.[4]

Finally, we present the results of timing tests performed on the spherical filter (for the approximately uniformly distributed points). The tests were run at a range of resolutions from $M = 32$ to $M = 1024$ using $N = M$ in all cases. The parameters were set to $a = p = 8$ and $m = 6$. Figure 4 shows the results of the tests. For comparison, the dashed line shows a time complexity of $\mathcal{O}(N^2)$. The results confirm the predicted complexity of $\mathcal{O}(N^2 \log N)$.

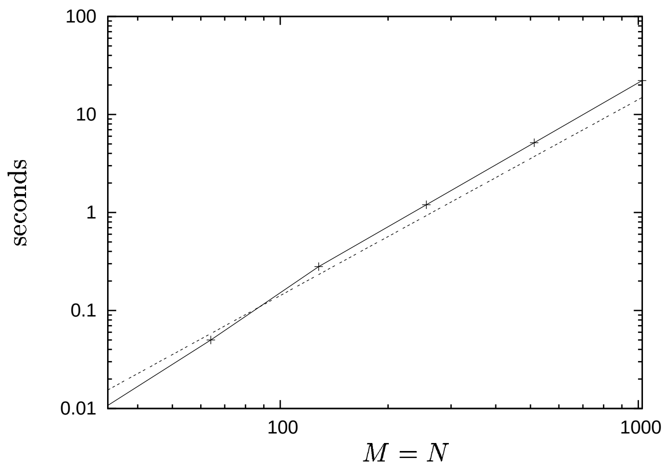

Figure 4. Running times of the spherical filter for various $M = N$. The parameters used were $a = p = 8$ and $m = 6$. The dashed line shows a time complexity of $\mathcal{O}(N^2)$ and intercepts the plot at $N = 64$.

5. CONCLUSION

In this paper we generalised the spherical filters to arbitrary grids. Our algorithm can be viewed as a fast method to construct approximations of a function on the unit sphere by a spherical polynomial from the space of all spherical polynomials of degree $\leq N$. In[14] the authors conclude that the quality of the polynomial interpolation is critically dependent on the choice of interpolation points if $N < 30$. Our numerical experiments demonstrate that the projection errors are virtually the same for the Gauß-Legendre grid and the approximately uniformly distributed grid for greater N.

REFERENCES

1. D. Potts, G. Steidl, and M. Tasche, "Fast Fourier transforms for nonequispaced data: A tutorial," in *Modern Sampling Theory: Mathematics and Applications*, J. J. Benedetto and P. J. S. G. Ferreira, eds., pp. 247 – 270, Birkhäuser, (Boston), 2001.
2. R. Jakob-Chien and B. K. Alpert, "A fast spherical filter with uniform resolution," *J. Comput. Phys.* **136**, pp. 580 – 584, 1997.
3. N. Yarvin and V. Rokhlin, "A generalized one-dimensional fast multipole method with application to filtering of spherical harmonics," *J. Comput. Phys.* **147**, pp. 549 – 609, 1998.
4. M. Böhme and D. Potts, "A fast algorithm for filtering and wavelet decomposition on the sphere," *Electron. Trans. Numer. Anal.* **16**, pp. 70 – 92, 2003.
5. D. Potts and G. Steidl, "Fast summation at nonequispaced knots by NFFTs," SIAM *J. Sci. Comput.* , to appear.
6. S. Kunis and D. Potts, "NFFT, Softwarepackage, C subroutine library." http://www.math.uni-luebeck.de/potts/nfft, 2002.
7. P. N. Swarztrauber and W. F. Spotz, "Generalized discrete spherical harmonic transforms," *J. Comput. Phys.* **159**, pp. 213 – 230, 2000.
8. S. Kunis and D. Potts, "Fast spherical Fourier algorithms," *J. Comput. Appl. Math* , to appear.
9. H. Mhaskar, F. Narcowich, and J. Ward, "Quadrature formulas on spheres using scattered data," *Math. Comput.* , to appear.
10. I. H. Sloan and R. S. Womersley, "Constructive polynomial approximation on the sphere," *J. Approx. Theory* **103**, pp. 91 – 118, 2000.
11. G. Steidl, "A note on fast Fourier transforms for nonequispaced grids," *Adv. Comput. Math.* **9**, pp. 337 – 353, 1998.
12. W. Freeden, T. Gervens, and M. Schreiner, *Constructive Approximation on the Sphere*, Oxford University Press, Oxford, 1998.
13. J. Cui and W. Freeden, "Equidistribution on the sphere," SIAM *J. Sci. Comput.* **18**, pp. 595 – 609, 1997.
14. R. S. Womersley and I. H. Sloan, "How good can polynomial interpolation on the sphere be?," *Adv. Comput. Math.* **14**, pp. 195 – 226, 2001.