CONSTRUCTIVE THEORY OF FUNCTIONS, Sozopol 2013
(K. Ivanov, G. Nikolov and R. Uluchev, Eds.), pp. 115-140
Prof. Marin Drinov Academic Publishing House, Sofia, 2014

Fast Evaluation and Irregular Sampling of Band-limited Functions on the Sphere*

Kamen G. Ivanov and Pencho Petrushev

Algorithms and software for evaluation and sampling (reconstruction) of high degree spherical polynomials (band-limited functions) on the unit 2-d sphere are presented. Our methods rely on highly localized reproducing kernels on the sphere (spherical father needlets). The algorithms are fast, local, memory efficient, numerically stable and with guaranteed (prescribed) accuracy, measured in the uniform norm. Software realization of the algorithms and numerical experiments are described as well.

Keywords and Phrases: spherical harmonics, evaluation at scattered points, needlets, fast computation, irregular sampling, reconstruction of band-limited functions.

Mathematics Subject Classification 2010: 65T99, 42C10, 33C55, 65D32, 41A55.

1. Introduction

This article is a survey of our recent development of robust algorithms for solution of two problems:

(i) Fast, accurate and memory efficient evaluation of high degree (≥ 2000) spherical polynomials at many scattered points on the unit 2-d sphere from [10], and

(ii) Reconstruction (sampling) of high degree spherical polynomials from their values at irregular sampling points on the sphere from [11].

These are fundamental problems that naturally occur in many areas in science and technology ranging from Geopotential Modeling to Quantum Mechanics and Cosmology, where spherical harmonics are the main tool for representation of functions on the sphere.

*This research has been supported by a NURI grant from NGA. The first author has been supported by grant DDVU 02/30 of the Fund for Scientific Research of the Bulgarian Ministry of Education and Science. The second author has been supported by NSF Grant DMS-1211528.

Spherical polynomials (band-limited functions) f on the 2-d unit sphere $\mathbb{S}^2$ in $\mathbb{R}^3$ are usually represented in spherical coordinates (θ, λ) in terms of the standard spherical harmonics basis:

$$f(\theta,\lambda) = \sum_{n=0}^{N} \sum_{m=0}^{n} \big(a_{nm} P_{nm}(\cos\theta)\cos m\lambda + b_{nm} P_{nm}(\cos\theta)\sin m\lambda\big), \qquad (1.1)$$

where P_{nm} are the associated Legendre functions. We shall denote by Π_N the set of all spherical polynomials of degree N as above.

Traditionally, the problems for evaluation and reconstruction of spherical polynomials are stated and solved in terms of their spherical harmonic coefficients, see e.g. [13, 15]. Unlike in the trigonometric case, however, currently there are no satisfactory practical algorithms (like FFT) for fast, stable and accurate evaluation of high degree (≥ 2000) spherical polynomials given by their coefficients. The problem is with the instability of the existing algorithms for evaluation of the associated Legendre functions P_{nm}. This is our motivation for putting forward and utilizing the following principle:

A spherical polynomial $f \in \Pi_N$ is better represented by its values $f(\xi)$ at regular grid points $\xi \in \mathcal{X}$ rather than by its spherical harmonics coefficients.

This principle underlies the way we state and approach the problems for evaluation and reconstruction of spherical polynomials.

We deem a set $\mathcal{X}$ of grid points on $\mathbb{S}^2$ *regular* if it is well structured and serves as a nodal set for a cubature with positive and easy to compute weights which is exact for high degree spherical polynomials. Regular grid points are typically points that are equally spaced or Gaussian with respect to their spherical coordinates (θ, λ). We shall further precise this notion in Subsection 3.3.

We next state explicitly the problems we consider in this article:

Problem 1 (Evaluation). Given the values $f(\xi)$, $\xi \in \mathcal{X}$, of a band-limited function $f \in \Pi_N$ at the points of a regular set $\mathcal{X}$ on $\mathbb{S}^2$ compute the values $f(z)$ at the points z from an arbitrary set $\mathcal{Z} \subset \mathbb{S}^2$.

Problem 2 (Reconstruction). Given the values $f(y)$ of a band-limited function $f \in \Pi_N$ at irregular sampling points $y \in Y$ ($Y \subset \mathbb{S}^2$) compute its values $f(\xi)$ at the points ξ from a regular set $\mathcal{X} \subset \mathbb{S}^2$.

Naturally, Problem 2 has a solution only if the density of sampling points is sufficiently high.

Combining the algorithms for solving Problems 1 and 2 enables us to solve effectively

Problem 3 (Reconstruction). Given the values $f(y)$ of a band-limited function $f \in \Pi_N$ on a set of irregular sampling points $Y \subset \mathbb{S}^2$ compute the values of f at the points z from an arbitrary set $\mathcal{Z} \subset \mathbb{S}^2$.

A targeted application and motivation for our interest in Problem 1 is the task for fast, accurate and memory efficient computation of the values of geodetic quantities such as the geoid undulation, determined from the Earth Gravitational Model EGM2008 of NGA [19].

Our methods rely on the spherical needlets introduced and used in [17, 18] for the purposes of harmonic analysis on the sphere. More explicitly, we employ discrete, reproducing Π_N, operators of the form

$$\Phi_N f(x) = \sum_{\xi \in \mathcal{X}} w_\xi \mathcal{K}_N(x \cdot \xi) f(\xi), \tag{1.2}$$

which rely on highly localized kernels (spherical father needlets) $\mathcal{K}_N(x \cdot \xi)$, and their truncated versions:

$$\Phi_{N,\delta} f(x) = \sum_{\substack{\xi \in \mathcal{X} \\ \rho(x,\xi) \le \delta}} w_\xi \mathcal{K}_N(x \cdot \xi) f(\xi). \tag{1.3}$$

Here $x \cdot \xi$ stands for the inner product of $x, \xi \in \mathbb{R}^3$ and $\rho(\cdot, \cdot)$ denotes the geodesic distance on $\mathbb{S}^2$.

Our algorithms for solving Problems 1 & 2 and hence Problem 3 are fast, local, memory efficient, numerically stable and with guaranteed (prescribed) accuracy, measured in the uniform norm. In this article we place the emphasis on the computational feasibility and practical realization of these algorithms. Robust MATLAB code realizing our algorithms for solving Problems 1 & 2 is developed and examples of effective evaluation and reconstruction of high degree (≥ 2000) spherical polynomials are demonstrated. A detailed description with theoretical justification (proofs) and error analysis of our algorithms and software for evaluation and reconstruction of high degree spherical polynomials are given in [10, 11].

To put our ideas in prospective we develop in [11] algorithms for solving Problems 1 & 2 and hence Problem 3 in the general framework of Dirichlet spaces, proposed in [1, 14]. This allows to extend our methods to various geometric settings, including evaluation and sampling of algebraic polynomials on the ball and simplex and band-limited functions on Lie groups or homogeneous spaces with polynomial volume growth and complete Riemannian manifolds with Ricci curvature bounded from below and satisfying the volume doubling condition.

We next briefly review the existing in the literature algorithms for evaluation and reconstruction of spherical polynomials. A direct evaluation of a spherical polynomial of degree N given by its coefficients at $O(N^2)$ grid points has computational complexity $O(N^4)$. However, by separation of variables the complexity is reduced to $O(N^3)$ and the algorithm is numerically stable. Driscoll and Healy [2] were the first to develop a fast Fourier method on the sphere. Mohlenkamp [16] proposed two algorithms for *approximate* evaluation of spherical polynomials with costs $O(N^{5/2} \log N)$ and $O(N^2 (\log N)^2)$.

Another approach to this problem based on the multi-pole method with complexity $O(N^2 \log N)$ was proposed by Rokhlin and Tygert [20, 22]. Kunis and Potts [15] developed a scheme for evaluation of spherical polynomials based on their representation as trigonometric polynomials in spherical coordinates. This method relies heavily on the excellent computational properties of the algorithm for nonequispaced fast Fourier transform developed by Dutt and Rokhlin in [3]. A common drawback of all these algorithms is that each of them has some degree of instability, rooted in the instability of the discrete Legendre function transform. This makes them problematic when applied to high degree spherical polynomials. In our method we bypass the associated Legendre functions completely and work only with Legendre polynomials.

There is a considerable body of work on sampling; we shall only review the relevant papers. In developing our sampling algorithm we borrow from [4, 5, 6, 7]. The main distinction between our approach to sampling and the one in these papers is in our usage of discrete operators as in (1.3) with highly localized kernels and the recovery of the functions at regular grid points. In [13] the authors apply a least squares approach to the problem for reconstruction of spherical polynomials from scattered sample values. Their algorithm requires dealing with high order associated Legendre functions, which creates instability. As a result, the practical feasibility of this algorithm is problematic when applied to high degree ($\geq$ 2000) spherical polynomials. Moreover, to work properly the algorithms from [13] require much denser sets of scattered points compared with our reconstruction algorithm.

We next outline the main points in this article. In Section 2 we describe in general our methods for evaluation and reconstruction of spherical polynomials. In Section 3 we lay down some of the ground work that is needed for the development of our algorithms. As already indicated, our solution of both Problems 1 and 2 is based on the needlet technology. In Section 4 we present our algorithm for fast, stable and memory efficient solution of Problem 1. The solution of Problem 2 is more involved; it is given in Section 5. Results from numerical tests are reported in Section 6.

Useful notation: In the sequel $|E|$ will stand for the cardinality of a finite set E.

2. Theoretical Underpinning of Our Methods

In this section we present the general principles of our needlet based methods for evaluation and reconstruction of spherical polynomials.

2.1. Spherical Needlets

Let φ be a $C^\infty[0,\infty)$ cutoff function satisfying for some fixed $\tau > 0$ the conditions:

$$\begin{cases} \varphi(t) = 1, & 0 \le t \le 1, \\ 0 \le \varphi(t) \le 1, & 1 < t < 1+\tau, \\ \varphi(t) = 0, & t \ge 1+\tau. \end{cases} \tag{2.1}$$

Consider the univariate kernels

$$\mathcal{K}_N(u) := \sum_{\nu=0}^{\infty} \varphi\Big(\frac{\nu}{N}\Big)(2\nu+1)P_\nu(u), \qquad u \in [-1,1], \quad N \ge 1, \tag{2.2}$$

where P_ν is the ν-th degree Legendre polynomial normalized by $P_\nu(1) = 1$. Then the kernels $\mathcal{K}_N(\xi \cdot \eta)$ are termed *spherical father needlets*.

Two properties of the father needlets make them a valuable tool:

(i) The operator

$$H_N f(\xi) := \frac{1}{4\pi} \int_{\mathbb{S}^2} \mathcal{K}_N(\xi \cdot \eta) f(\eta)\, d\sigma(\eta) \tag{2.3}$$

reproduces the spherical harmonics of degree $\le N$, i.e. $H_N f = f$ for $f \in \Pi_N$, and

(i) The kernels $\mathcal{K}_N(\xi \cdot \eta)$ have nearly exponential localization: For any $s > 0$ there exists a constant $c_s > 0$ such that

$$|\mathcal{K}_N(\xi \cdot \eta)| \le \frac{c_s N}{(1 + N\rho(\xi,\eta))^s}, \qquad \xi, \eta \in \mathbb{S}^2, \quad N \ge 1, \tag{2.4}$$

where $\rho(\xi,\eta) = \arccos(\xi \cdot \eta)$ is the geodesic distance between ξ and η.

As shown in [12], for a cutoff function φ with"small derivatives" the localization of the father needlets $\mathcal{K}_N(\xi \cdot \eta)$ can be improved to sub-exponential. For more details we refer the reader to [10].

Selecting $\mathcal{X}$ to be a regular grid containing sufficiently many points ($O(N^2)$ will do) usually implies the existence of weights $w_\xi > 0$, $\xi \in \mathcal{X}$, such that

$$\frac{1}{4\pi} \int_{\mathbb{S}^2} f(x)\, d\sigma(x) = \sum_{\xi \in \mathcal{X}} w_\xi f(\xi) \qquad \forall f \in \Pi_{(2+\tau)N}. \tag{2.5}$$

This allows to discretize the operator H_N from (2.3), namely, set

$$\Phi_N f(x) := \sum_{\xi \in \mathcal{X}} w_\xi \mathcal{K}_N(x \cdot \xi) f(\xi). \tag{2.6}$$

Clearly, $\Phi_N f = f$ for $f \in \Pi_N$.

We are now prepared to describe our methods for fast evaluation and reconstruction of spherical polynomials.

2.2. Fast Evaluation of Spherical Polynomials

Given the values $f(\xi)$ of a band-limited function $f \in \Pi_N$ at the points of a regular set $\mathcal{X}$ as above we compute the values $f(z)$ at the points z from an arbitrary set $\mathcal{Z} \subset \mathbb{S}^2$ by using that $\Phi_N f = f$, which leads to

$$f(z) = \sum_{\xi \in \mathcal{X}} w_\xi \mathcal{K}_N(z \cdot \xi) f(\xi). \tag{2.7}$$

To make this scheme computationally tractable we truncate the operator Φ_N. Namely, we replace it by the operator

$$\Phi_{N,\delta} f(x) = \sum_{\substack{\xi \in \mathcal{X} \\ \rho(x,\xi) \le \delta}} w_\xi \mathcal{K}_N(x \cdot \xi) f(\xi). \tag{2.8}$$

Due to the superb localization of the kernel $\mathcal{K}_N(x \cdot \xi)$ it can be shown that for a properly selected (small) value of the parameter δ the operator $\Phi_{N,\delta}$ from (2.8) provides an excellent approximation to the operator Φ_N in (2.6), which leads to

$$f(z) \approx \sum_{\substack{\xi \in \mathcal{X} \\ \rho(x,\xi) \le \delta}} w_\xi \mathcal{K}_N(z \cdot \xi) f(\xi) \tag{2.9}$$

with guaranteed accuracy as required. It is an important point that our approach allows to control the error in the uniform norm. Thus, employing (2.7) and (2.9) we arrive at exact and approximate solutions of Problem 1.

The utilization of the operator $\Phi_{N,\delta}$ with a kernel of small support opens the possibility of using simultaneously more than one regular sets $\mathcal{X}$. The point is that part of the nodes in a particular regular set $\mathcal{X}$ are usually not quite well distributed (e.g. nodes concentrate near the poles), which creates problems. This inconvenience can be offset by using two or more different regular sets $\mathcal{X}$ for different subregions of $\mathbb{S}^2$.

2.3. Reconstruction of Spherical Polynomials

Assume that the values $f(y)$ of a band limited function $f \in \Pi_N$ are given at irregular sampling points $y \in Y$ ($Y \subset \mathbb{S}^2$). We let $\mathcal{A} = \{A_y : y \in Y\}$ be the Voronoi tessellation of $\mathbb{S}^2$ induced by Y, i.e. the points from A_y are closer to y than to any other point from Y. Write

$$d := \max_{y \in Y} \sup_{x \in A_y} \rho(x, y). \tag{2.10}$$

Then the caps of radius d centered at the points of Y cover the whole sphere $\mathbb{S}^2$. For every $x \in \mathbb{S}^2$ denote by y_x the closest point from Y. Hence $x \in A_y$.

We shall use the following extension operator for functions g defined on Y:

$$\mathcal{E} g(x) := \sum_{y \in Y} g(y) \mathbb{1}_{A_y}(x), \qquad x \in \mathbb{S}^2, \tag{2.11}$$

where $\mathbb{1}_{A_y}$ is the characteristic function of A_y.

Let $\mathcal{X} \subset \mathbb{S}^2$ be a regular set with associated cubature formula (2.5) and let Φ_N be the operator from (2.6). It is easy to show the following assertion (see [11]):

Under the above conditions, assume that

$$q := d\|\Phi_N\|(1+\tau)N < 1, \tag{2.12}$$

where $\|\Phi_N\|$ *is the norm of* Φ_N *as an* $L^\infty(\mathbb{S}^2)$ *to* $L^\infty(\mathbb{S}^2)$ *operator. Let us set* $\mathcal{R} := \Phi_N - \mathcal{E}\Phi_N$. *Then for every* $f \in \Pi_N$ *the series* $\sum_{k=0}^\infty \mathcal{R}^k(\mathcal{E}f)$ *converges uniformly to* f *and its* n*-th partial sum satisfies*

$$\Big\|f - \sum_{k=0}^{n-1} \mathcal{R}^k(\mathcal{E}f)\Big\|_\infty \le q^n \|f\|_\infty. \tag{2.13}$$

The above statement provides an exact reconstruction algorithm for $f \in \Pi_N$. Indeed, setting $g_k = \mathcal{R}^k(\mathcal{E}f)$ we have by (2.13)

$$f(\xi) = \sum_{k=0}^\infty g_k(\xi), \qquad \xi \in \mathcal{X}. \tag{2.14}$$

Note that the values $g_k(\xi)$ for all $\xi \in \mathcal{X}$ can be iteratively computed by

$$g_0(\xi) = f(y_\xi), \qquad g_{k+1}(\xi) = \mathcal{R}g_k(\xi) = \Phi_N g_k(\xi) - \Phi_N g_k(y_\xi). \tag{2.15}$$

The key observation is that the evaluation of $\Phi_N g(x)$ by (2.6) uses only the values of g at the points from $\mathcal{X}$. Naturally, instead of using the whole series in practice the values $f(\xi)$, $\xi \in \mathcal{X}$, are approximated by $\sum_{k=0}^{n-1} g_k(\xi)$, where n is determined by the target accuracy via (2.13). Furthermore, replacing Φ_N by $\Phi_{N,\delta}$ and using

$$g_{k+1}(\xi) = \mathcal{R}^* g_k(\xi) := \Phi_{N,\delta} g_k(\xi) - \Phi_{N,\delta} g_k(y_\xi), \qquad \xi \in \mathcal{X}, \tag{2.16}$$

instead of (2.15) we arrive at an approximate reconstruction algorithm for $f \in \Pi_N$. The idea of using two or more different regular sets $\mathcal{X}$ for different subregions of $\mathbb{S}^2$ suggested in the previous subsection can be employed for reconstruction of spherical polynomials as well. In sum, we have algorithms for exact and approximate solutions of Problem 2.

The general ideas put forward in this section will be elaborated further in what follows.

3. Technical Infrastructure

3.1. The Cutoff Function

From the description of our evaluation and reconstruction algorithms in the previous section it becomes clear that the *space* localization of the father

needlets $\mathcal{K}_N(\xi \cdot \eta)$ plays a crucial role. The following approximate identity for cubatures on $\mathbb{S}^2$ with *positive weights* provides a reasonable measure of the localization of $\mathcal{K}_N(\xi \cdot \eta)$:

$$\begin{aligned}\sum_{\substack{\xi\in\mathcal{X}\\ \xi\cdot x<\cos\delta}} w_\xi\ |\mathcal{K}_N(\xi\cdot x)| &\cong \frac{1}{4\pi}\int_{\xi\cdot x<\cos\delta} |\mathcal{K}_N(\xi\cdot x)|\,d\sigma(x)\\ &= \frac{1}{4\pi}\int_\delta^\pi |\mathcal{K}_N(\cos\theta)|\sin\theta\,d\theta \int_0^{2\pi} 1\,d\lambda = \frac{1}{2}\int_{-1}^{\cos\delta} |\mathcal{K}_N(t)|\,dt. \quad (3.1)\end{aligned}$$

Given $0 < \varepsilon < 1$ let δ be determined by

$$\frac{1}{2}\int_{-1}^{\cos\delta} |\mathcal{K}_N(t)|\,dt = \frac{\varepsilon}{2}\left|\int_{-1}^{1} \mathcal{K}_N(t)\,dt\right| = \varepsilon. \quad (3.2)$$

Here the last equality follows by (2.2) and the orthogonality of Legendre polynomials. The above equation determines δ as a function of ε, N, and φ (obeying (2.1) for some $\tau > 0$). Based on (3.1) it will serve as a criterion for measuring the quality of different cutoff functions φ and the respective father needlets. The rule will be, for a given accuracy ε, *the smaller δ the better the cutoff function φ.*

From (2.8), (2.6), (3.1) and (3.2), one can get the following error bound on the truncated operator

$$\|\Phi_N f - \Phi_{N,\delta} f\|_{L^\infty(\mathbb{S}^2)} \le \varepsilon \|f\|_{\ell^\infty(\mathcal{X})} \qquad \forall f \in L^\infty(\mathbb{S}^2). \quad (3.3)$$

Selection of $\boldsymbol{\varphi}$. It is a challenging problem to determine the best or a near best cutoff function φ according to the above rule. As we are interested in cutoff functions φ satisfying the conditions: $\varphi(t) = 1$ for $0 \le t \le 1$ and $\varphi(t) = 0$ for $t \ge 1 + \tau$, we focus on the behavior of φ on $[1, 1 + \tau]$. Write $\varphi(t) = \psi((1 + \tau - t)/\tau)$, where $\psi(u)$ is defined for $u \in [0, 1]$ and satisfies $\psi(0) = 0$, $\psi(1) = 1$.

Our initial selection of a cutoff function φ was based on

$$\begin{aligned}\psi(u) &= \frac{(2m+1)!!}{2(2m)!!}\int_0^{\pi u} \sin^{2m+1} v\,dv\\ &= \frac{1}{2} - \frac{\cos\pi u}{2} - \frac{\cos\pi u}{2}\sum_{k=1}^{m} \frac{(2k-1)!!}{(2k)!!}\sin^{2k}\pi u\end{aligned} \quad (3.4)$$

with m depending on the accuracy ε. For example, we used $m = 5$ whenever $10^{-5} \le \varepsilon \le 10^{-7}$. Not long ago, however, we discovered that cutoff functions induced by the family

$$\psi(u) = \psi_b(u) := \kappa^{-1}\int_0^u e^{b\sqrt{v(1-v)}}\,dv \quad \text{with} \quad \kappa := \int_0^1 e^{b\sqrt{v(1-v)}}\,dv \quad (3.5)$$

are substantially better than the ones based in ψ from (3.4). In (3.5) b is a positive parameter, which for $4 < \log_{10}(1/\varepsilon) < 11$ and $\tau \geq 1$ is given by

$$b = 4.8 \log_{10}(1/\varepsilon) + 3.4 - 0.2 \min\{\tau, 3\}. \tag{3.6}$$

It is straightforward to write code for accurate computation of δ from (3.2) for given φ, ε, and N. Therefore, it is easy to compare the values of δ for different cutoff functions φ's. It turns out that for δ computed from (3.2) the product δN varies slightly with N for our two families of cutoff functions, determined by (3.4) and (3.5). In the following two tables the values of δ are displayed for $N = 1000$ and several choices of τ and ε.

$\tau \setminus \varepsilon$	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
1	0.0386	0.0457	0.0548	0.0624	0.0701	0.0773
2	0.0189	0.0228	0.0268	0.0310	0.0346	0.0386
3	0.0125	0.0150	0.0177	0.0207	0.0232	0.0257
4	0.00935	0.0113	0.0132	0.0154	0.0174	0.0193

Table 1. Values of δ for $N = 1000$ and cutoff function defined via (3.4)

$\tau \setminus \varepsilon$	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
1	0.0278	0.0325	0.0372	0.0419	0.0468	0.0515
2	0.0137	0.0162	0.0185	0.0209	0.0232	0.0257
3	0.00917	0.0107	0.0123	0.0138	0.0155	0.0171
4	0.00685	0.00802	0.00919	0.0103	0.0116	0.0128

Table 2. Values of δ for $N = 1000$ and cutoff function defined via (3.5)

The comparison of Tables 1 and 2 shows definite advantage of using the ψ from (3.5) over the one from (3.4). As a result, the number of terms entering the sum in (2.8) (which is a constant multiple of δ^2) reduces about two times when using the new cutoff function. Furthermore, this advantage becomes more significant when we increase the precision.

3.2. Accurate Kernel Evaluation

The next step in developing our algorithm is the accurate and fast evaluation of $\mathcal{K}_N(\xi \cdot \eta)$ for given $\xi, \eta \in \mathbb{S}^2$, which is a nontrivial task. For every $u \in [-1, 1]$ one can evaluate $\mathcal{K}_N(u)$ using, for instance, the downward Clenshaw recurrence formula. It employs the Legendre recurrence relation

$$(n+1)P_{n+1}(u) = (2n+1)uP_n(u) - nP_{n-1}(u), \qquad n \geq 0,$$
$$P_0(u) = 1, \qquad P_{-1}(u) = 0.$$

This algorithm is numerically stable and fast since it requires only $O(N)$ operations.

The straightforward calculation of $\mathcal{K}_N(\xi \cdot \eta)$, where $u = \xi \cdot \eta$ is obtained via the Spherical Law of Cosines

$$\xi \cdot \eta = \cos \rho(\xi, \eta) = \cos \theta' \cos \theta + \sin \theta' \sin \theta \cos(\lambda' - \lambda)$$

and $\mathcal{K}_N(u)$ is computed by the downward Clenshaw recurrence, looses accuracy when ξ is close to η, that is exactly the case we are interesting in. In order to improve the accuracy by several significant digits we perform the calculations as follows:

(i) We compute the spherical distance ρ between $\xi = (\theta', \lambda')$ and $\eta = (\theta, \lambda)$ via the Haversine Law of Spherical Trigonometry

$$\sin^2 \frac{\rho(\xi, \eta)}{2} = \sin^2 \frac{\theta' - \theta}{2} + \sin \theta' \sin \theta \sin^2 \frac{\lambda' - \lambda}{2}. \tag{3.7}$$

(ii) We compute $\mathcal{K}_N(\xi \cdot \eta) = (\mathcal{K}_N \circ \cos)(\rho)$ via an approximation of $\mathcal{K}_N \circ \cos$.

The Haversine Law (3.7) is well-conditioned for computation of ρ close to 0 and the round-off error is smaller when compared with the Spherical Law of Cosines. This fact has been known since the XIX century. The haversine function is defined by $\operatorname{hav} t := (1 - \cos t)/2 = \sin^2(t/2)$; its latter form is used in (3.7).

The advantage of using the trigonometric polynomial $\mathcal{K}_N \circ \cos$ in step (ii) over the algebraic polynomial $\mathcal{K}_N$ stems from the fact that the derivative of $\mathcal{K}_N \circ \cos$ near the origin is about N times smaller than the derivative of $\mathcal{K}_N$ near 1.

In order to get fast and accurate evaluation of $(\mathcal{K}_N \circ \cos)(\rho)$ for $\rho \in [0, \delta]$ we take the equally spaced points $t_r = \delta r/R$ for $r = -s, -s+1, \ldots, R+s$ and compute $t_r^* = \arccos(\cos t_r)$. Note that in general $t_r^* \neq t_r$ because of the machine arithmetic, while $\cos t_r^*$ and $\cos t_r$ coincide as double precision numbers. Then $\mathcal{K}_N(u_r)$, determined for $u_r = \cos t_r = \cos t_r^*$ via the downward Clenshaw recurrence, is a good approximation to $(\mathcal{K}_N \circ \cos)(t_r^*)$. Thus we have the values of $\mathcal{K}_N \circ \cos$ at the points t_r^*, which are close to equally spaced but not equally spaced. Now $(\mathcal{K}_N \circ \cos)(\rho)$ is computed by Lagrange interpolation of $\mathcal{K}_N \circ \cos$ with nodes t_r^*, $r = m-s, m-s+1, \ldots, m+s+1$, where $m = \lfloor \rho R/\delta \rfloor$. The Lagrange polynomial is of degree $2s+1$.

The choice of R and s depends on the targeted relative error ε and the degree $N_\tau := \lceil (1+\tau)N \rceil - 1$ of $\mathcal{K}_N \circ \cos$. Our experiments show that for $\varepsilon \geq 10^{-11}$ and $N_\tau \leq 16000$ one can take $R = 2000$ for $s = 1$ or $R = 150$ for $s = 2$. The numbers $\mathcal{K}_N(u_r)$ are computed in $O(NR)$ operations and stored at the initial stage of the program. At later stages the evaluation of $(\mathcal{K}_N \circ \cos)(\rho)$ requires only $O(s)$ operations. Of course, the third degree Lagrange interpolation ($s = 1$) is faster but less accurate than the fifth degree Lagrange interpolation ($s = 2$).

3.3. Regular Point Sets on the Sphere

Given $M \in \mathbb{N}$ we say that $\mathcal{X}$ is a set of *M-regular* points on the sphere if the following two conditions are verified:

1. There exist *non-negative* weights w_ξ, $\xi \in \mathcal{X}$, of a cubature formula with $\mathcal{X}$ as a nodal set which is exact for the polynomials from Π_{M-1}, i.e.
$$\frac{1}{4\pi}\int_{\mathbb{S}^2} f(y)\,d\sigma(y) = \sum_{\xi\in\mathcal{X}} w_\xi f(\xi) \qquad \forall f \in \Pi_{M-1}; \tag{3.8}$$

2. The set $\mathcal{X}$ is *structured* in the sense that for every $x \in \mathbb{S}^2$ and $\delta \in (0,\pi]$ one can determine effectively all points in $\bar{B}_{\mathcal{X}}(x,\delta) = \{\xi \in \mathcal{X} : \rho(x,\xi) \le \delta\}$ using $c|\bar{B}_{\mathcal{X}}(x,\delta)|$ operations, where the constant c is independent of x, δ, M and $|\mathcal{X}|$.

Examples of regular point sets on the sphere are $\mathcal{X}^{(i)} = \{\xi^{(i)}_{k,\ell} = (\theta^{(i)}_k, \lambda^{(i)}_\ell)\}$, $i = 1, 2$, that for given $K, L \ge 1$ are defined by

$$\theta^{(1)}_k = \frac{\pi}{K}\,k, \quad k = 0, 1, \dots, K; \qquad \lambda^{(1)}_\ell = \frac{2\pi}{L}\,\ell, \quad \ell = 0, 1, \dots, L-1;$$

and

$$\theta^{(2)}_k = \frac{\pi}{K}\,k - \frac{\pi}{2K}, \quad k = 1, 2, \dots, K; \qquad \lambda^{(2)}_\ell = \frac{2\pi}{L}\,\ell, \quad \ell = 0, 1, \dots, L-1.$$

Here in $\mathcal{X}^{(1)}$ we consider only one node for $k = 0$ (the North Pole) and only one node for $k = K$ (the South Pole). Another example is the set $\mathcal{X}^{(3)}$ generated by the zeros u_k of the K-th degree Legendre polynomial P_K. In this case we write

$$\theta^{(3)}_k = \arccos u_k, \quad k = 1, 2, \dots, K; \qquad \lambda^{(3)}_\ell = \frac{2\pi}{L}\,\ell, \quad \ell = 0, 1, \dots, L-1.$$

As is well-known the cubatures associated with $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \mathcal{X}^{(3)}$ can be represented as tensor products of one-dimensional algebraic quadrature in the co-latitude direction θ and the rectangular trigonometric quadrature in the latitude direction (see e.g. [10, Subsection 3.4]). The relations between K, L and M are given by (see [10, Theorem 3.11])

$$M \le L, \qquad M \le \begin{cases} 2\,\lfloor (K+1)/2 \rfloor, & i = 1, 2; \\ 2K, & i = 3. \end{cases}$$

Under the above restrictions the sets $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \mathcal{X}^{(3)}$ are M-regular [10, Theorems 3.11 and 3.12].

Other regular point sets can be obtained from $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, or $\mathcal{X}^{(3)}$ by applying rotations or reflections on the sphere. For example, consider the map $T : \mathbb{R}^3 \to \mathbb{R}^3$ given by

$$T(x_1, x_2, x_3) = (x_1, x_3, -x_2).$$

This is $\pi/2$ rotation about the x_1-axis. The restriction of T on the sphere $T|_{\mathbb{S}^2} : \mathbb{S}^2 \to \mathbb{S}^2$ relates the spherical coordinates (θ, λ) and $(\tilde{\theta}, \tilde{\lambda})$ of a point x and its image $\tilde{x} = T(x)$ by

$$(\sin\tilde{\theta}\cos\tilde{\lambda}, \sin\tilde{\theta}\sin\tilde{\lambda}, \cos\tilde{\theta}) = (\sin\theta\cos\lambda, \cos\theta, -\sin\theta\sin\lambda).$$

From the rotation invariance of Π_N it follows that the sets $T^{-1}(\mathcal{X}^{(i)})$ and $T(X^{(i)})$, $i = 1, 2, 3$, are also regular and induce similar cubatures as $\mathcal{X}^{(i)}$.

All of the above regular point sets have one disadvantage – their points congregate near the poles (or the images of the poles). This force us to treat the points near the poles differently compared to the ones away from the poles.

3.4. Utilization of Two Sets of Regular Points on the Sphere

As already mentioned the regular point sets from Subsection 3.3 have the deficiency that the points in each of them concentrate around the poles or the images of the poles via some rotation. This drawback along with the fact that for a given x the value of $\Phi_N f(x)$ is obtained by $O(|\mathcal{X}|)$ operations makes an evaluation algorithm based on Φ_N impractical. To overcome the second deficiency we shall use the truncated version $\Phi_{N,\delta}$ of the operator Φ_N defined in (2.8), and to remedy the first deficiency we shall utilize the rotated version $\tilde{\Phi}_{N,\delta}$ of $\Phi_{N,\delta}$ for the regions around the poles. In this way we will decrease substantially the algorithm's computational cost.

To realize these ideas we first introduce some notation. Given $N \in \mathbb{N}$ and $\varepsilon > 0$ (to be determined) we assume that $\mathcal{X} \subset \mathbb{S}^2$ is one of the M-regular set points $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$, or $\mathcal{X}^{(3)}$ from Subsection 3.3 with $M := \lceil (2+\tau)N \rceil$. In fact, to us the best choice is $\mathcal{X} := \mathcal{X}^{(3)}$. Let $\delta > 0$ be a constant such that (3.2) holds and let $\Phi_{N,\delta}$ be the operator defined in (2.8). We subdivide $\mathbb{S}^2$ into two: The equatorial area (belt) $\mathcal{U}_1$ and its compliment (the polar regions) $\mathcal{U}_2$, defined in spherical coordinates by

$$\mathcal{U}_1 := \{x \in \mathbb{S}^2 : \pi/4 \le \theta(x) \le 3\pi/4\}, \qquad \mathcal{U}_2 := \mathbb{S}^2 \setminus \mathcal{U}_1. \tag{3.9}$$

We also introduce the following sets of nodes on $\mathbb{S}^2$:

$$\begin{aligned} \mathcal{X}_1 &:= \mathcal{X} \cap \{\pi/4 - \delta_0 \le \theta \le 3\pi/4 + \delta_0\}, \\ \mathcal{X}_2 &:= T(\mathcal{X}) \cap \big(\{0 \le \theta \le \pi/4 + \delta_0\} \cup \{3\pi/4 - \delta_0 \le \theta \le \pi\}\big), \\ \mathcal{X}_0 &:= \mathcal{X}_1 \cup \mathcal{X}_2, \end{aligned} \tag{3.10}$$

where $\delta_0 := \delta + d$ with d from (2.10) for the reconstruction problem or with some $d \ge 0$ for the fast evaluation problem. We assume $\delta_0 < \pi/4$.

The rotated by T versions $\tilde{\Phi}_N$, $\tilde{\Phi}_{N,\delta}$ of the operators Φ_N, $\Phi_{N,\delta}$ defined in (2.6) and (2.8) are given by

$$\tilde{\Phi}_N f(x) = \sum_{\xi \in T(\mathcal{X})} \widetilde{w}_\xi \mathcal{K}_N(x \cdot \xi) f(\xi), \qquad \tilde{\Phi}_{N,\delta} f(x) = \sum_{\substack{\xi \in T(\mathcal{X}) \\ \rho(x,\xi) \le \delta}} \widetilde{w}_\xi \mathcal{K}_N(x \cdot \xi) f(\xi), \tag{3.11}$$

where $\widetilde{w}_\xi = w_{T^{-1}(\xi)}$ for $\xi \in T(\mathcal{X})$. Due to the rotation invariance of the spherical harmonic spaces, the operators $\tilde{\Phi}_N$ and $\tilde{\Phi}_{N,\delta}$ have the same properties as Φ_N and $\Phi_{N,\delta}$. Observe also that for every $x \in \mathcal{U}_1$ the value of $\Phi_{N,\delta} f(x)$ depends only on the nodes $\xi \in \mathcal{X}_1$ and, similarly, for every $x \in \mathcal{U}_2$ the value of $\tilde{\Phi}_{N,\delta} f(x)$ depends only on the nodes $\xi \in \mathcal{X}_2$.

4. Fast Evaluation of Spherical Polynomials

In this section we focus on the needlet based algorithm for fast evaluation of high degree spherical polynomials, described in Section 2. First, we outline the properties of the operators Φ_N and $\Phi_{N,\delta}$ from (2.6) and (2.8) that are involved in this algorithm and, second, we describe in detail the consecutive steps in the algorithm.

4.1. Properties of Φ_N and $\Phi_{N,\delta}$

Assume that $\mathcal{X}$ is a set of *ν-regular* points on $\mathbb{S}^2$ with $\nu \geq N$ and let Φ_N and $\Phi_{N,\delta}$ be the operators defined in (2.6) and (2.8) using $\mathcal{K}$ from (2.2). Then Φ_N has these properties:

$$\Phi_N : C(\mathbb{S}^2) \to C(\mathbb{S}^2) \text{ is a bounded linear operator;} \tag{4.1}$$

$$\|\Phi_N\| \leq C, \text{ where } C > 0 \text{ is a constant independent of } N; \tag{4.2}$$

$$\Phi_N f \in \Pi_{(1+\tau)N} \qquad \forall f \in C(\mathbb{S}^2). \tag{4.3}$$

Moreover, if

$$\nu \geq (2+\tau)N, \tag{4.4}$$

then

$$\Phi_N f = f \qquad \forall f \in \Pi_N, \tag{4.5}$$

$$\|f - \Phi_N f\|_\infty \leq (\|\Phi_N\| + 1) E_N(f)_\infty \qquad \forall f \in C(\mathbb{S}^2). \tag{4.6}$$

Here $E_N(f)_\infty$ denotes the best approximation of f from Π_N in the uniform norm and $C(\mathbb{S}^2)$ stands for the space of all continuous functions on $\mathbb{S}^2$.

Let δ be determined by (3.2). Then $\Phi_{N,\delta}$ satisfies:

$$\Phi_{N,\delta} : C(\mathbb{S}^2) \to L^\infty(\mathbb{S}^2) \text{ is a bounded linear operator;} \tag{4.7}$$

$$\|f - \Phi_{N,\delta} f\|_\infty \leq \varepsilon \|f\|_{\ell^\infty(\mathcal{X})} \qquad \forall f \in \Pi_N; \tag{4.8}$$

$$\|f - \Phi_{N,\delta} f\|_\infty \leq (C+1) E_N(f)_\infty + \varepsilon \|f\|_{\ell^\infty(\mathcal{X})} \qquad \forall f \in C(\mathbb{S}^2), \tag{4.9}$$

where C is the constant from (4.2). For more details and proofs we refer the reader to [10].

The values of the $L^\infty \to L^\infty$ norms $\|\Phi_N\|$ and $\|\Phi_{N,\delta}\|$ are given in Subsection 6.3. Inequality (4.8) provides the relative error estimate for band-limited function evaluation by $\Phi_{N,\delta}$.

4.2. Fast Evaluation Algorithm

Here we describe our algorithm for fast evaluation when the values $f(\xi)$ of a polynomial $f \in \Pi_N$ are given at the points of the set $\mathcal{X}_0$ from Subsection 3.4.

Input:

1. Degree N of f and the target accuracy ε.
2. Number of knots K, L and the type of one-dimensional latitude quadrature.
3. Parameter δ_0 for the sets $\mathcal{X}_1$ and $\mathcal{X}_2$ in (3.10).
4. Values $f(\xi)$, $\xi \in \mathcal{X}_0$.

Pre-computation:

1. Compute the knots and weights of the one-dimensional quadratures.
2. Compute the degree of exactness ν of cubature (3.8) and its weights w_ξ as tensor product of the one-dimensional quadratures weights.
3. For the given ν, N determine the largest possible τ satisfying (4.4).
4. For N, ε, τ compute δ from (3.2).
5. For ε, τ compute b, the values of ψ_b from (3.5) and the values $\varphi(k/N)$ from Subsection 3.1.
6. Compute $\mathcal{K}_N(\cos t_r^*)$, $r = -s, -s+1, \ldots, R+s$ with downward Clenshaw recurrence (see Subsection 3.2).

Computation:

For every $z \in \mathcal{Z}$:

If z is in the equatorial belt $\mathcal{U}_1$, i.e. $0.25\pi \le \theta(z) \le 0.75\pi$:

1. Find the indexes of the points ξ in $D_1 := \{\xi \in \mathcal{X}_1 \ : \ \rho(\xi, z) \le \delta\}$.
2. Compute $h_\xi = \mathcal{K}_N(z \cdot \xi)$ for $\xi \in D_1$.
3. Compute $F(z) = \sum_{\xi \in D_1} w_\xi h_\xi f(\xi)$

else (i.e. if z is in the polar regions $\mathcal{U}_2$)

1. Find the indexes of the points ξ in $D_2 := \{\xi \in \mathcal{X}_2 \ : \ \rho(\xi, z) \le \delta\}$.
2. Compute $h_\xi = \mathcal{K}_N(z \cdot \xi)$ for $\xi \in D_2$.
3. Compute $F(z) = \sum_{\xi \in D_2} \widetilde{w}_\xi h_\xi f(\xi)$.

Output: The approximate values $F(z)$ of $f(z)$ at $z \in \mathcal{Z}$.

The algorithm works under the following conditions on the parameters: $\delta \leq \delta_0$, $0 < \varepsilon < 1$ and $\nu > 2N$.

The relative error of the output satisfies

$$\|F - f\|_{\ell^\infty(\mathcal{X})} / \|f\|_{\ell^\infty(\mathcal{X})} < \varepsilon$$

provided $\varepsilon \geq N10^{-15}$ in double precision arithmetic.

Complexity of the algorithm. We next determine the complexity of the *Computation* part in the typical case when $K = O(N)$, $L = O(N)$. The number of operations for step (1) is $O(\delta^2 KL) = O(\delta^2 N^2)$ (see the structure condition (2) for regular points in (3.8) from Subsection 3.3). Steps (2) and (3) require $O(\delta^2 N^2)$ operations as for single ξ and z the computation of $t = z \cdot \xi$ and $\mathcal{K}_N(t)$ requires $O(1)$ operations. Thus, the number of operations at a single point z is $O(\delta^2 N^2)$. Now from $\delta = O(\log(1/\varepsilon)/N)$ we get $O(\delta^2 N^2) = O(\log^2(1/\varepsilon))$. Thus, the total complexity of the *Computation* part is $O(|\mathcal{Z}| \log^2(1/\varepsilon))$.

One of the most demanding steps in the *Pre-computation* part is step (6) which requires $O((R+s)N)$ operations. For the numeric computation of the knots and weights of the Gaussian quadrature in step (1) we use the MATLAB function `legpts` from `Chebfun` software system [21]. It utilizes a fast and accurate algorithm from [9], which shows excellent results for algebraic degree of exactness up to one million. Step (2) is executed in $O(N^2)$ operations and the remaining steps in the *Pre-computation* part require at most $O(N)$ operations.

5. Reconstruction of Spherical Polynomials

This section is devoted to our algorithm for reconstruction of high degree spherical polynomials. We first describe briefly the algorithm for approximate solution of the problem, and then give the details.

5.1. Approximate Solution of Problem 2

We shall employ the linear operator

$$\mathcal{R}g(x) := (\mathcal{I} - \mathcal{E})\Phi_{N,\delta} g(x) \cdot \mathbb{1}_{\mathcal{U}_1}(x) + (\mathcal{I} - \mathcal{E})\tilde{\Phi}_{N,\delta} g(x) \cdot \mathbb{1}_{\mathcal{U}_2}(x), \qquad (5.1)$$

where $\mathcal{E}$ is the extension operator from (2.11), the operator $\Phi_{N,\delta}$ is defined in (2.8), and $\tilde{\Phi}_{N,\delta}$ is from (3.11). The observations we made in the end of Subsection 3.4 imply that the above operator uses only the values of g at points in $\mathcal{X}_0$. Indeed, if $x \in \mathcal{U}_1$ then $\mathcal{E}\Phi_{N,\delta} g(x) = \Phi_{N,\delta} g(y_x)$ uses the values $g(\xi)$ for $\xi \in \mathcal{X}$ with $\rho(y_x, \xi) \leq \delta$, yilding $\rho(x, \xi) \leq \delta + d = \delta_0$, i.e. $\xi \in \mathcal{X}_1$. Let us point out that for such an x the value of $\Phi_{N,\delta} g$ at y_x is determined by values of g on $\mathcal{X}_1$ even in the case when y_x itself belongs to $\mathcal{U}_2$. Similar considerations are valid for $x \in \mathcal{U}_2$.

Our algorithm for approximate solution of Problem 2 is based on the following assertion (see [11, Theorem 4.2]): *Using the notation from above with* δ *determined by* (3.2) *for some* $\varepsilon > 0$ *assume in addition that*

$$q := d(1+\tau)N\|\Phi_N\| + 2\varepsilon < 1. \tag{5.2}$$

Then for any $f : Y \to \mathbb{R}$ *the series* $\sum_{k=0}^{\infty} \mathcal{R}^k(\mathcal{E}f)$ *converges uniformly and for any* $f \in \Pi_N$

$$\Big\| f - \sum_{k=0}^{n-1} \mathcal{R}^k(\mathcal{E}f) \Big\|_\infty \le \Big(q^n + \frac{2\varepsilon}{1-q} \Big) \|f\|_\infty. \tag{5.3}$$

This statement provides an effective reconstruction algorithm for spherical polynomials $f \in \Pi_N$.

5.2. Approximate Reconstruction Algorithm

In solving Problem 2, we would like to find approximate values $F(\xi)$ to the unknown values $f(\xi)$, $\xi \in \mathcal{X}_0$, with prescribed error ε_0, i.e. $|F(\xi) - f(\xi)| \le \varepsilon_0$. We determine the relative error $\varepsilon_1 = \varepsilon_0 / \|f\|_{\ell^\infty(Y)}$ and split it into two parts $\varepsilon_1 = \varepsilon_2 + 2\varepsilon/(1-q)$, where ε_2 will be the iteration accuracy and ε – the needlet accuracy.

We next describe the consecutive steps in our *algorithm for approximate reconstruction*. If we consider Problem 2 as a first step in the solution of Problem 3, then we are free to choose the set $\mathcal{X}_0$ from (3.10) in Subsection 3.4. This case is described below. In the case of fixed $\mathcal{X}_0$ in Problem 2 step (1) from the *Pre-computation* part has to be moved to the *Input* part.

Input:

1. Values $f(y)$, $y \in Y$, at an irregular sampling set Y.
2. Degree N of f, the needlet parameter τ, the target relative accuracy ε_1, the iteration accuracy ε_2 and the needlet accuracy ε.

Pre-computation:

1. Compute the number of knots K, L so that the cubature be exact for polynomials of degree $M-1$ with $M = \lceil (2+\tau)N \rceil$.
2. Compute the knots and weights of the one-dimensional quadratures.
3. Compute the nodes of the set $\mathcal{X} = \mathcal{X}^{(3)}$ (see Subsection 3.3).
4. Compute the weights w_ξ of the cubature (3.8) as tensor product of the one-dimensional quadratures weights.
5. For every $\xi \in \mathcal{X} \cup T(\mathcal{X})$ find the closest point y_ξ in Y.
6. Compute δ for the given N, ε, τ and a cutoff function φ from (3.5)–(3.6).

7. Compute the values $\varphi(k/N)$ for the given N, ε, τ and φ from (3.5)–(3.6).
8. Compute $\mathcal{K}_N(\cos t_r^*)$, $r = -s, -s+1, \dots, R+s$ with downward Clenshaw recurrence (see [10, Subsection 3.3]).
9. Compute $d = \max_{\xi\in\mathcal{X}} \rho(\xi \cdot y_\xi)$ and form the sets $\mathcal{X}_i$, $i = 0, 1, 2$, with parameter $\delta_0 = \delta + d$ (see (3.10)).
10. Compute the matrices:
$V^{(1)} = \{v_{\xi,\eta}^{(1)} : \xi \in \mathcal{X}_1 \cap \mathcal{U}_1, \eta \in \mathcal{X}\}$, $V^{(2)} = \{v_{\xi,\eta}^{(2)} : \xi \in \mathcal{X}_2 \cap \mathcal{U}_2, \eta \in T(\mathcal{X})\}$,
$V^{(3)} = \{v_{\xi,\eta}^{(3)} : \xi \in \mathcal{X}_1 \cap \mathcal{U}_2, \eta \in T(\mathcal{X})\}$, $V^{(4)} = \{v_{\xi,\eta}^{(4)} : \xi \in \mathcal{X}_2 \cap \mathcal{U}_1, \eta \in \mathcal{X}\}$,
defined by

$$\begin{aligned} v_{\xi,\eta}^{(j)} &= w_\eta\big(\widetilde{\mathcal{K}}_N(\xi \cdot \eta) - \widetilde{\mathcal{K}}_N(y_\xi \cdot \eta)\big), \qquad j = 1, 4, \\ v_{\xi,\eta}^{(j)} &= \widetilde{w}_\eta\big(\widetilde{\mathcal{K}}_N(\xi \cdot \eta) - \widetilde{\mathcal{K}}_N(y_\xi \cdot \eta)\big), \qquad j = 2, 3, \end{aligned} \tag{5.4}$$

where $\widetilde{\mathcal{K}}_N(t) = \mathcal{K}_N(t)$ for $t \geq \cos\delta$ and $\widetilde{\mathcal{K}}_N(t) = 0$ for $t < \cos\delta$.

Iterations:

1. Initial values: $g_0(\xi) = f(y_\xi)$, $F(\xi) = g_0(\xi)$, $\xi \in \mathcal{X}_0$.
2. Iteration steps: For $k = 0, 1, \dots$ do
   (a)
   $$g_{k+1}(\xi) = \sum_{\eta\in\mathcal{X}_1} v_{\xi,\eta}^{(1)} g_k(\eta), \qquad \xi \in \mathcal{X}_1 \cap \mathcal{U}_1; \tag{5.5}$$
   (b)
   $$g_{k+1}(\xi) = \sum_{\eta\in\mathcal{X}_2} v_{\xi,\eta}^{(2)} g_k(\eta), \qquad \xi \in \mathcal{X}_2 \cap \mathcal{U}_2; \tag{5.6}$$
   (c)
   $$g_{k+1}(\xi) = \sum_{\eta\in\mathcal{X}_2} v_{\xi,\eta}^{(3)} g_k(\eta), \qquad \xi \in \mathcal{X}_1 \cap \mathcal{U}_2; \tag{5.7}$$
   (d)
   $$g_{k+1}(\xi) = \sum_{\eta\in\mathcal{X}_1} v_{\xi,\eta}^{(4)} g_k(\eta), \qquad \xi \in \mathcal{X}_2 \cap \mathcal{U}_1; \tag{5.8}$$
   (e)
   $$F(\xi) = F(\xi) + g_{k+1}(\xi), \qquad \xi \in \mathcal{X}_0;$$
3. Stopping criterion: $\|g_{k+1}\| \leq \varepsilon_2 \|g_0\|$.

Output: The approximate values $F(\xi)$ of $f(\xi)$ at all points $\xi \in \mathcal{X}_0$.

The only condition imposed on the parameters is (5.2). Under this condition the algorithm converges at a geometric rate and we have (see [11, Proposition 5.1])

Proposition 5.1. *The relative error of the algorithm output is given by*

$$\frac{\|F - f\|_{\ell^\infty(\mathcal{X}_0)}}{\|f\|_{\ell^\infty(Y)}} < \varepsilon_2 + \frac{2\varepsilon}{1-q} = \varepsilon_1, \tag{5.9}$$

$$\max\Big\{\frac{\|\Phi_{N,\delta}F - f\|_{\ell^\infty(Y\cap\mathcal{U}_1)}}{\|f\|_{\ell^\infty(Y)}}, \frac{\|\tilde{\Phi}_{N,\delta}F - f\|_{\ell^\infty(Y\cap\mathcal{U}_2)}}{\|f\|_{\ell^\infty(Y)}}\Big\} < \|\Phi_{N,\delta}\|\varepsilon_1. \tag{5.10}$$

Inequality (5.9) shows that the prescribed accuracy is achieved by the algorithm, while (5.10) give us a tool to verify whether the computed values $F(\xi)$, $\xi \in \mathcal{X}_0$, reconstruct the spherical polynomial f, known by its values at the scattered points $y \in Y$.

Complexity of the algorithm. We determine the complexity for the best choice of K and L in step (1), which means $K = O(N)$, $L = O(N)$. Steps (1)–(4) and (6)–(8) are analyzed in [10, Subsection 3.7]. On account of the structure condition for regular points in Subsection 3.3 step (5) requires $O(N^2 + |Y|)$ operations. The complexity of step (9) is $O(N^2)$.

Step (10) is the most demanding one on both memory and number of operations (i.e. speed) in the whole algorithm. The "matrices of influence" $V^{(j)}$, $j = 1, 2, 3, 4$, express the relative distances between the elements of the two sets $\mathcal{X}_0$ and Y. Their size is huge: $V^{(1)}$ and $V^{(2)}$ have $O(N^4)$ elements and $V^{(3)}$ and $V^{(4)}$ have $O(N^4\delta_0)$ elements. If one works with the complete "matrices of influence" then polynomial degrees exceeding 200 will be practically prohibitive. For comparison, for degree 1000 we work with a set $\mathcal{X}_0$ with close to 6 000 000 points and the nodal sets $\mathcal{X} = \mathcal{X}^{(3)}$ and $T(\mathcal{X})$ contain 8 000 000 points each. This makes a total of 4.8×10^{13} elements in the "matrices of influence" and only the storage of such amount of data on a "hard disk" as 8 bytes numbers will require 350 TB of memory!

Using the superb localization of the father needlet kernel $\mathcal{K}_N(x \cdot \xi)$ we make the "matrices of influence" sparse by setting $\widetilde{\mathcal{K}}_N(t) = 0$ for $t < \cos\delta$ in (5.4). Thus, the total number of non-zero elements in these matrices is $O(N^2\bar{n}) = O(N^4\delta^2) = O(N^2 \ln^2(1/\varepsilon))$, where $\bar{n}$ is the average number per point ξ of non-zero elements in (5.4). Several values of $\bar{n}$ are given in Table 4 of Subsection 6.2. For $\tau = 2$ they range from 267 for $\varepsilon = 10^{-5}$ to 1150 for $\varepsilon = 10^{-11}$. Other important parameters of the problem as memory requirements and time of execution are also given in Subsection 6.2. In sum, step (10) requires $O(N^2 \ln^2(1/\varepsilon))$ operations but the O constant is quite large.

Every step in the *Iterations* part executes a matrix-times-vector multiplication, where every non-zero element of the "matrices of influence" is used once. This requires $O(N^2 \ln^2(1/\varepsilon))$ operations. The number of iterations is $\ln(1/\varepsilon_2)/\ln(1/q)$. Hence the choice $\varepsilon_2 = \varepsilon_1/3$ and $\varepsilon = (1-q)\varepsilon_1/3$ will result in $O(N^2 \ln^3(1/\varepsilon_1))$ operations for the complexity of the algorithm.

Memory requirements. For $N = 1000$ and $\varepsilon = 10^{-7}$ the values of the elements of the sparse "matrices of influence" will occupy some 21 GB memory (see Table 4). With additional 12 GB for the indexes of the non-zero elements we arrive at 33 GB of memory for storage of these matrices. This fact made us decide to save the "matrices of influence" in pieces on the hard disk. Then the operations in (5.5)–(5.8) are executed by reading one piece at a time from HD, performing the multiplication and clearing the matrix piece from the memory before reading another piece. In this way the execution time for 20 iterations is comparable to the time necessary to compute matrix element values in (5.4) and to save them on HD (see Table 5 in Subsection 6.2).

Each of the other input, work, and output variables as F, old and new g (i.e. g_k and g_{k+1}), spherical coordinates of the irregular sample points and the polynomial values requires $O(N^2)$ bites of memory. In view of the small number of such variables this is easily manageable for N in the range of a several thousand.

Optimal choice of the needlet parameter τ. For $M = \lceil (2+\tau)N \rceil$, $K = \lceil M/2 \rceil$, $L = M$ we have:

- The number of nodes in $\mathcal{X}_0$ is proportional to M^2;
- The average number of nodes from $\mathcal{X}_0$ in a δ neighborhood is proportional to $\delta^2 M^2$.

Hence, both the size of the "matrices of influence" and the number of operation in (5.5)–(5.8) for a single iteration step is $O(\delta^2 M^4)$. Using the estimate $\delta = O(\ln(1/\varepsilon)\tau^{-1}N^{-1})$ we get $\delta^2 M^4 = O((2+\tau)^4 \tau^{-2} N^2)$ and the minimal value of the last expression is attained for $\tau = 2$. Therefore, the best choice of the needlet parameter τ relative to memory usage as well as speed is $\tau = 2$.

6. Numerical Examples

The algorithms described in this article have been implemented in software written in MATLAB with double-precision variables. The code does not relay on variable precision arithmetic. It was intensively tested for degrees between 60 and 4000.

6.1. Undulation

The geoid undulation G is approximated by a spherical polynomial of degree and order $N = 2189$, computed in the official Earth Gravitational Model EGM2008 and publicly released by the U.S. National Geospatial-Intelligence Agency (NGA). The polynomial coefficients have been taken from

`http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/egm08_wgs84.html`.

This website also contains the values of the geoid undulation on two mesh grids of type $\mathcal{X}^{(1)}$ (see Subsection 3.3): $2.5' \times 2.5'$ (i.e. $K = 4320$, $L = 8640$) and $1' \times 1'$ (i.e. $K = 10800$, $L = 21600$). The $1' \times 1'$ grid points are $233,301,600$. The geoid undulation values as single precision numbers occupy 890MB on the hard disk and range from -106.9 m to 85.8 m.

The table given below summarizes the results of the testing on a 2.4 GHz PC with 16 GB of RAM. The following programs are compared:

- `hsynth_WGS84` – the NGA spherical harmonic synthesis program computing G directly from its coefficients; written in FORTRAN; for comparison purposes assumed to be exact, so the error reported in Table 3 is 0.
- `interp_1min` – the NGA spherical harmonic synthesis program computing G by spline interpolation of the $1' \times 1'$ undulation data; written in FORTRAN.
- `interp_2p5min` – the NGA spherical harmonic synthesis program computing G by spline interpolation of the $2.5' \times 2.5'$ undulation data; written in FORTRAN.
- `needlet3` – implementation of our algorithm which uses the $3' \times 3'$ undulation data; written in MATLAB.
- `needlet4` – implementation of our algorithm which uses the $4' \times 4'$ undulation data; written in MATLAB.

Program	Size HD (MB)	Size RAM (MB)	Values/ second	Error (mm)
`hsynth_WGS84`	71.2	53.9	16	0
`interp_1min`	890.0	1 814.0	640 000	0.84
`interp_2p5min`	142.5	287.6	630 000	8.65
`needlet3`	70.9	132.4	11 000	0.40
`needlet4`	41.1	100.2	4 400	0.36

Table 3. Program comparison by memory size, speed (on a 2.4 GHz PC) and error

The NGA programs `interp_1min` and `interp_2p5min` require approximately 20 and 7 seconds for loading into the memory and initialization, while each of `needlet3` or `needlet4` requires approximately 2 seconds. The total run time should be formed as the sum of these values plus the time for proper point evaluation computed using column "values/second" above.

The sampling interval for the Nyquist frequency is $\pi/2160 = 5'$ and programs `needlet3` and `needlet4` work with $3' \times 3'$ and $4' \times 4'$ mesh grids producing results with relative errors approximately $4 \cdot 10^{-6}$. The programs demonstrate how the needlets can also be successfully used as *an approximation tool for data compression.*

The test results described in Table 3 show that `needlet3` and `needlet4` are memory efficient and, therefore, they can be effectively used for fast compressed and accurate computation of the geoid undulation at scattered points on the sphere. This is the main advantage of `needlet3` and `needlet4` over `interp_1min`. Of course, as usual here there is a trade-off between memory size and speed.

6.2. Reconstruction

We have implemented our reconstruction algorithm in a MATLAB R2012b code and have extensively tested it on a 2.4 GHz PC, CPU Intel Core i7 with 16 GB of RAM. The code does not relay on variable precision arithmetic.

For irregular points we have taken the HEALPix pixel centers and their rotations on the sphere.

The optimal speed and memory requirements were achieved for $\tau = 2$ according to the theory. Hence, we report in this subsection results only for this value of the needlet parameter. In the latitude direction the quadrature is Gaussian.

For $K = 2N$, $L = 4N$ and φ from (3.5)–(3.6) we get the following values for the size of the "influence matrices", i.e. number of points in $\mathcal{X}_0$ and average number of non-zero elements in (5.4).

$N \setminus \varepsilon$	10^{-5}	10^{-7}	10^{-9}	10^{-11}
250	384728×270	396300×494	408016×786	419836×1150
500	1470932×268	1493560×488	1516332×773	1539128×1127
1000	5749660×267	5794400×485	5839244×767	5884128×1117

Table 4. Size of the "influence matrices": number of points in $\mathcal{X}_0$ and average number of non-zero elements

For different irregular sampling sets Y the average number of non-zero elements may slightly vary. The number of points in $\mathcal{X}_0$ grows slowly when ε decreases due to the $\log \varepsilon$ enlargement of the adjacent sets $\mathcal{X}_1 \cap \mathcal{U}_2$ and $\mathcal{X}_2 \cap \mathcal{U}_1$.

The polynomial values were provided by several low and high degree polynomials including the polynomials G_N and $\widetilde{G}_N$, which are given by

$$\begin{aligned} G_N(\theta,\lambda) &:= \sum_{m=1}^{N} m^{-1/3} q_{m,N} P_{m,N}(\cos\theta) \sin m\lambda \\ &\quad + \sum_{m=1}^{N-3} m^{-1/3} q_{m,N-3} P_{m,N-3}(\cos\theta) \sin m\lambda, \\ \widetilde{G}_N(\theta,\lambda) &:= q_{0,N} P_{0,N}(\cos\theta) + 2 \sum_{m=1}^{N} q_{m,N} P_{m,N}(\cos\theta) \cos m\lambda, \end{aligned}$$

where $P_{m,n}$ are the associated Legendre functions and the coefficients $q_{m,n}$ are selected so that they normalize to 1 in $L^2(\mathbb{S}^2, \frac{1}{4\pi}d\sigma)$ each spherical harmonic term.

The uniform norms of G_N and $\widetilde{G}_N$ for selected values of N are given in Tables 6 and 7, respectively. The global extrema of G_N and $\widetilde{G}_N$ are localized around the points $(\frac{\pi}{2}, \frac{\pi}{2})$ and $(\frac{\pi}{2}, \frac{3\pi}{2})$. We believe that polynomials G_N and $\widetilde{G}_N$ are good for testing of our reconstruction algorithm since they have relatively large spherical harmonic coefficients and highly oscillatory behavior.

Degree N	250	500	1000	2000
Pre-computation part	6.2	23.0	92.7	363.5
Iterations part (20 iterations)	6.2	24.2	96.1	384.4
Total	12.4	47.2	188.8	747.9

Table 5. Execution times (in minutes) of the reconstruction algorithm

Table 5 contains the execution times of the *Pre-computation* and *Iterations* parts of the reconstruction algorithm. The *Pre-computation* time is the total of the times for execution of all steps of *Pre-computation* from Subsection 4.2 plus the "matrices of influence" saving time on HD. The *Iterations* time includes the execution times for 20 steps of *Iterations* plus the "matrices of influence" loading time from HD. The number of irregular sampling points is approximately 8 times larger than the number of points in $\mathcal{X}_0$, but their influence on the times reported below is minimal (apart from the influence on the number of iterations for achieving the target accuracy). The values of the other parameters are $\varepsilon = 10^{-7}$, $K = 2N$, $L = 4N$, and the number of iterations in the table is 20.

We see that the execution times are proportional to N^2 according to the theory given in Subsection 4.2. The saving time is approximately 27% of the *Pre-computation* time, while the loading time is approximately 63% of the *Iterations* time.

The relative errors defined in (5.9) for $f = G_N$ and $f = \widetilde{G}_N$ at the points $\mathcal{X}_0$ are given in Tables 6 and 7, respectively. These errors are obtained from the algorithm in Subsection 4.2 with accuracy parameters $\varepsilon = 10^{-7}$ and $\varepsilon_2 = 10^{-8}$.

Degree N	250	500	1000	2000
$\|G_N\|_\infty$	76.45	121.35	192.65	305.86
Relative error	8.4667e-09	7.8133e-09	5.7893e-09	5.8170e-09

Table 6. Uniform norms and relative errors from (5.9) for G_N

As a rule the observed relative errors are 10 to 15 times smaller than the target relative accuracy ε_1! Our experiments also show that the relative errors from (5.10) at the sampling points Y are very close to the respective errors at the points $\mathcal{X}_0$.

Degree N	250	500	1000	2000
$\|\widetilde{G}_N\|_\infty$	480.60	958.99	1915.4	3828.0
Relative error	5.6226e-09	5.6581e-09	5.4573e-09	5.3932e-09

Table 7. Uniform norms and relative errors from (5.9) for $\widetilde{G}_N$

6.3. Norms of Operators

The operator norms in this subsection are $\infty \to \infty$ norms. The norm of the integral needlet operator (2.3) is given by

$$\|H_N\| = \sup_{x\in\mathbb{S}^2} \frac{1}{4\pi} \int_{\mathbb{S}^2} |\mathcal{K}_N(x\cdot y)|\, d\sigma(y) = \frac{1}{2}\int_{-1}^{1} |\mathcal{K}_N(t)|\, dt. \tag{6.1}$$

For φ from (3.5)–(3.6) and for various τ and ε the numerical values of the norm from (6.1) for $N = 40$, $N = 400$, $N = 4000$ are displayed in Table 8.

τ	$N \setminus \varepsilon$	10^{-5}	10^{-7}	10^{-9}	10^{-11}
1	40	3.1364	3.4067	3.6306	3.8230
	400	3.1280	3.3996	3.6251	3.8194
	4000	3.1267	3.3982	3.6236	3.8179
2	40	2.4559	2.6774	2.8613	3.0197
	400	2.4487	2.6700	2.8538	3.0123
	4000	2.4478	2.6691	2.8529	3.0114
3	40	2.1905	2.3927	2.5606	2.7054
	400	2.1849	2.3867	2.5545	2.6991
	4000	2.1842	2.3861	2.5538	2.6984
4	40	2.0510	2.2421	2.4010	2.5380
	400	2.0465	2.2373	2.3960	2.5328
	4000	2.0460	2.2368	2.3954	2.5323

Table 8. Numerical evaluation of norm from (6.1)

As Table 8 shows the norm practically does not depend on the degree N. This fact is in compliance with the theory which states that these norms have majorants, which are independent of N. The slight decrease of the norm with N is predictable and is due to the increased smoothness of the kernel $\mathcal{K}_N$. The variations of the norm with τ and ε are due to the different functions φ defined in (3.5)–(3.6).

The norms of the discrete operators Φ_N and $\Phi_{N,\delta}$ are given by

$$\|\Phi_N\| = \sup_{x\in\mathbb{S}^2} \sum_{\xi\in\mathcal{X}} w_\xi |\Phi_N(x\cdot\xi)| \tag{6.2}$$

and

$$\|\Phi_{N,\delta}\| = \sup_{x\in\mathbb{S}^2} \sum_{\substack{\xi\in\mathcal{X}\\ \rho(x,\xi)\le\delta}} w_\xi |\Phi_N(x\cdot\xi)|. \tag{6.3}$$

Let us recall that due to (3.3) the two norms are quite close, namely,

$$0 < \|\Phi_N\| - \|\Phi_{N,\delta}\| \leq \varepsilon.$$

The norms in (6.2) and (6.3) depend on N, δ, φ, τ, ε, K, L, and the type of the regular nodes used. As in the case of the norm in (6.1) the relative variation of these norms with respect to N is less than one percent.

For $N = 500$ and for various τ and ε the numerical values of the norm from (6.2) are displayed in Table 9. The other parameters for the computations are: Gaussian quadrature with $K = 2\lceil(2+\tau)N/4\rceil$, $L = 2K$, and φ from (3.5)–(3.6).

$\tau \setminus \varepsilon$	10^{-5}	10^{-7}	10^{-9}	10^{-11}
1	4.2324	4.6610	5.0166	5.3227
2	3.1562	3.5077	3.7990	4.0497
3	2.7355	3.0577	3.3245	3.5540
4	2.5137	2.8193	3.0724	3.2901

Table 9. Numerical evaluation of $\|\Phi_N\|$ for $\mathcal{X} = \mathcal{X}^{(3)}$

In the solution of Problem 1 we in fact use the operators $\Phi_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_1}+\tilde{\Phi}_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_2}$ instead of $\Phi_{N,\delta}$. Their norms are given by

$$\|\Phi_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_1}+\tilde{\Phi}_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_2}\| = \sup_{\substack{x\in\mathbb{S}^2\\ \pi/4\leq\theta\leq 3\pi/4}} \sum_{\substack{\xi\in\mathcal{X}\\ \rho(x,\xi)\leq\delta}} w_\xi |\Phi_N(x\cdot\xi)|. \tag{6.4}$$

For the same values of the parameters as in Table 9 we have these norms:

$\tau \setminus \varepsilon$	10^{-5}	10^{-7}	10^{-9}	10^{-11}
1	3.7265	4.0423	4.3022	4.5245
2	2.9159	3.1833	3.4026	3.5898
3	2.5899	2.8396	3.0438	3.2178
4	2.4152	2.6549	2.8507	3.0174

Table 10. Numerical evaluation of $\|\Phi_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_1}+\tilde{\Phi}_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_2}\|$ for $\mathcal{X} = \mathcal{X}^{(3)}$

According to (6.3) and (6.4) the inequality

$$\|\Phi_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_1}+\tilde{\Phi}_{N,\delta}\cdot\mathbb{1}_{\mathcal{U}_2}\| \leq \|\Phi_{N,\delta}\| \tag{6.5}$$

holds. For the nodes $\mathcal{X} = \mathcal{X}^{(3)}$ generated by the Gaussian quadrature one has strict inequality in (6.5) as evidenced by Tables 9 and 10. The reason for this is that the supremums in (6.2) and (6.3) are attained for x at one of the poles, while the supremum in (6.4) is attained for x at the equator.

For $\mathcal{X} = \mathcal{X}^{(1)}$ or $\mathcal{X} = \mathcal{X}^{(2)}$ all supremums above are attained for x's at the equator and, hence, in (6.5) we have an equality. For these types of nodes and

minimal possible K and L the norm values are approximately in the middle between the norm in (6.1) given in Tables 8 and the norm in (6.4) given in Table 10. The main reason for the decrease of the norm is that the number of knots in latitude direction is doubled. The general rule is that for a fixed cutoff function φ whenever the nodes get denser then the norm becomes smaller and tends to the value given in Table 8. Note that the parameters K and L are optimized for speed, but not to minimize $\|\Phi_N\|$.

The results in this subsection show that the norms of our needlet-type operators are quite small, which in turn guarantees the stability of the described algorithms.

Bibliography

[1] T. Coulhon, G. Kerkyacharian and P. Petrushev, Heat kernel generated frames in the setting of Dirichlet spaces, *J. Fourier Anal. Appl.* **18** (2012), 995–1066.

[2] J. R. Driscoll and D. M. Healy, Computing Fourier transforms and convolutions on the 2-sphere, *Adv. in Appl. Math.* **15** (1994), 202–250.

[3] A. Dutt and V. Rokhlin, Fast Fourier transforms for nonequispaced data, *SIAM J. Sci. Stat. Comput.* **14** (1993), 1368–1393.

[4] H. Feichtinger and K. Gröchenig, Irregular sampling theorems and series expansions of band-limited functions, *J. Math. Anal. Appl.* **167** (1992), 530–556.

[5] H. Feichtinger and K. Gröchenig, Iterative reconstruction of multivariate band-limited functions from irregular sampling values, *SIAM J. Math. Anal.* **231** (1992), 244–261.

[6] H. Feichtinger and K. Gröchenig, Theory and practice of irregular sampling, *in* "Wavelets: Mathematics and Applications" (J. Benedetto and M. Frazier, Eds.), pp. 305–363, Stud. Adv. Math., CRC, Boca Raton, FL, 1994.

[7] H. Feichtinger, K. Gröchenig and T. Strohmer, Efficient numerical methods in nonuniform sampling theory, *Numer. Math.* **69** (1995), 423–440.

[8] H. Feichtinger and I. Pesensen, Iterative recovery of band-limited functions on manifolds, *Contemp. Math.* **345** (2004), 137–153.

[9] A. Glaser, X. Liu and V. Rokhlin, A fast algorithm for the calculation of the roots of special functions, *SIAM J. Sci. Comput.* **29** (2007), 1420–1438.

[10] K. G. Ivanov and P. Petrushev, Fast memory efficient evaluation of spherical polynomials at scattered points, *Adv. Comput. Math.* (2014), DOI: 10.1007/s10444-014-9354-3.

[11] K. G. Ivanov and P. Petrushev, Irregular sampling of band-limited functions on the sphere, *Appl. Comput. Harmon. Anal.* **37** (2014), 545–562.

[12] K. G. Ivanov, P. Petrushev and Yuan Xu, Sub-exponentially localized kernels and frames induced by orthogonal expansions, *Math. Z.* **264** (2010), 361–397.

[13] J. Keiner, S. Kunis and D. Potts, Efficient reconstruction of functions on the sphere from scattered data, *J. Fourier Anal. Appl.* **13** (2007), 435–458.

[14] G. Kerkyacharian and P. Petrushev, Heat kernel based decomposition of spaces of distributions in the framework of Dirichlet spaces, *Trans. Amer. Math. Soc.* **367** (2015), 121–189.

[15] S. Kunis and D. Potts, Fast spherical Fourier algorithms, *J. Comput. Appl. Math.* **161** (2003), 75–98.

[16] M. Mohlenkamp, A fast transform for spherical harmonics, *J. Fourier Anal. Appl.* **5** (1999), 159–184.

[17] F. J. Narcowich, P. Petrushev and J. D. Ward, Localized tight frames on spheres, *SIAM J. Math. Anal.* **38** (2006), 574–594.

[18] F. J. Narcowich, P. Petrushev and J. D. Ward, Decomposition of Besov and Triebel-Lizorkin spaces on the sphere, *J. Funct. Anal.* **238** (2006), 530–564.

[19] N. K. Pavlis, S. A. Holmes, S. C. Kenyon and J. K. Factor, An Earth Gravitational Model to Degree 2160: EGM2008, presented at the 2008 General Assembly of the European Geosciences Union, Vienna, Austria, April 13–18, 2008.

[20] V. Rokhlin and M. Tygert, Fast algorithms for spherical harmonic expansions, *SIAM J. Sci. Comput.* **27** (2006), 1903–1928.

[21] L. N. Trefethen and others, Chebfun Version 4.2, The Chebfun Development Team, 2011, `http://www.maths.ox.ac.uk/chebfun/`.

[22] M. Tygert, Fast algorithms for spherical harmonic expansions II, *J. Comput. Phys.* **227** (2008), 4260–4279.

K. G. Ivanov

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
bl. 8 Acad. G. Bonchev Str.
1113 Sofia
BULGARIA
E-mail: `kamen@math.bas.bg`

P. P. Petrushev

Department of Mathematics
University of South Carolina
Columbia, SC 29208
USA
E-mail: `pencho@math.sc.edu`