SIAM J. IMAGING SCIENCES
Vol. 16, No. 4, pp. 2072–2104

Spherical Framelets from Spherical Designs*

Yuchen Xiao† and Xiaosheng Zhuang‡

Abstract. In this paper, we investigate in detail the structures of the variational characterization $A_{N,t}$ of the spherical t-design, its gradient $\nabla A_{N,t}$, and its Hessian $\mathcal{H}(A_{N,t})$ in terms of fast spherical harmonic transforms. Moreover, we propose solving the minimization problem of $A_{N,t}$ using the trust-region method to provide spherical t-designs with large values of t. Based on the obtained spherical t-designs, we develop (semidiscrete) spherical tight framelets as well as their truncated systems and their fast spherical framelet transforms for the practical spherical signal/image processing. Thanks to the large spherical t-designs and localization property of our spherical framelets, we are able to provide signal/image denoising using local thresholding techniques based on a fine-tuned spherical cap restriction. Many numerical experiments are conducted to demonstrate the efficiency and effectiveness of our spherical framelets and spherical designs, including Wendland function approximation, ETOPO data processing, and spherical image denoising.

Key words. tight framelets, spherical framelets, spherical t-designs, fast spherical harmonic transforms, fast spherical framelet transforms, trust-region method, Wendland functions, ETOPO1, spherical signals/images, image/signal denoising

MSC codes. 42C15, 42C40, 58C35, 65D18, 65D32

DOI. 10.1137/22M1542362

1. Introduction and motivation.

Spherical data commonly appear in many real-world applications such as the navigation data in the global positioning system (GPS), the global climate change estimation in geography, the planet study in astronomy, the cosmic microwave background (CMB) data analysis in cosmology, the virus analysis in biology and molecular chemistry, the 360° panoramic images and videos in virtual reality and computer vision, and so on. In many of these real-world application scenarios, the observed spherical data could be large in terms of size, irregular in the sense of function property, or incomplete and noisy due to machine and environment deficiency. How to represent such data "well" so that one can process them "efficiently" is the key to solving these real-world problems successfully. Spherical data are necessarily discrete and can typically be modeled as samples on spherical meshes or spherical point sets [11]. In this paper, we focus on the study of spherical data defined on a special type of structured point sets on the unit sphere, that is, the spherical t-designs, and the construction of multiscale representation systems, namely, the semidiscrete

*Received by the editors December 21, 2022; accepted for publication (in revised form) June 29, 2023; published electronically November 14, 2023.
https://doi.org/10.1137/22M1542362
Funding: The work of the authors was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, project CityU 11309122 and the City University of Hong Kong project 7005603.
†Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong (yc.xiao@my.cityu.edu.hk).
‡Corresponding author. Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong (xzhuang7@cityu.edu.hk).

spherical framelet systems, based on the spherical t-designs, for the sparse representation and efficient processing of spherical data.

How to "nicely" distribute points on the unit sphere lies in the heart of many fundamental problems of mathematics and physics such as the best packing problems [15], the minimal energy problems [33], the optimal configurations related to Smale's 7th Problem [55], and so on. It is well-known that it is highly nontrivial to define a so-called "good" point set on the unit sphere $\mathbb{S}^d := \{\boldsymbol{x} \in \mathbb{R}^{d+1} \,|\, \|\boldsymbol{x}\| = 1\}$ when the dimension $d \geq 2$, where $\|\cdot\|$ is the Euclidean norm. Many real-world problems can be interpreted as a partial differential equation (PDE) or a PDE system and their numerical solutions (e.g., using finite-element methods) are then sought to address the related problems. Numerical integrations (quadrature rules) hence play an important role in such numerical solutions of PDEs. From the viewpoint of numerical integrations on the sphere, that is, finding a quadrature (cubature) rule $Q_N := \{(\boldsymbol{x}_i, w_i) \in \mathbb{S}^d \times \mathbb{R} \,|\, i = 1, \ldots, N\}$ such that $\frac{1}{|\mathbb{S}^d|}\int_{\mathbb{S}^d} f(\boldsymbol{x}) d\mu_d(\boldsymbol{x}) \approx \sum_{i=1}^N w_i f(\boldsymbol{x}_i)$, where μ_d denotes the surface measure on $\mathbb{S}^d$ such that $\mu_d(\mathbb{S}^d) =: |\mathbb{S}^d|$ is the surface area of $\mathbb{S}^d$, one can define a "good" point set $X_N := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{S}^d$ in the sense of requiring the weights $w_i \equiv \frac{1}{N}$ for all i for a certain class of functions f. More precisely, let $\Pi_t := \Pi_t(\mathbb{S}^d)$ denote the space of $(d+1)$-variate polynomials with total degree at most t restricted on $\mathbb{S}^d$. The point set X_N is said to be "good" if it satisfies

$$\frac{1}{N}\sum_{i=1}^N p(\boldsymbol{x}_i) = \frac{1}{|\mathbb{S}^d|}\int_{\mathbb{S}^d} p(\boldsymbol{x})\,\mathrm{d}\mu_d(\boldsymbol{x}) \qquad \forall p \in \Pi_t. \tag{1.1}$$

Such a point configuration X_N, is called a *spherical t-design,* which was established by Delsarte, Goethals, and Seidel [19] in 1977. In other words, a spherical t-design X_N is an equal weight polynomial-exact quadrature rule associated with Π_t. We refer to the excellent survey paper [5] by Bannai and Bannai on the topic of spherical designs.

A natural question immediately follows: Does such a spherical t-design X_N exist? It turns out that such a question leads to many profound mathematical results. Delsarte, Goethals, and Seidel [19] showed that the lower bound of a spherical t-design $X_N \subset \mathbb{S}^d$ on the number N of points for any degree $t \in \mathbb{N}$ satisfies $N \geq N^*(d,t)$, where $N^*(d,t) = 2\binom{d+\frac{t-1}{2}}{d}$ if t is odd and $N^*(d,t) = \binom{d+\frac{t}{2}}{d} + \binom{d+\frac{t}{2}-1}{d}$ if t is even. When the lower bound is attained, it is called a *tight* spherical t-design. Note that on the circle $\mathbb{S}^1$, the vertices of a regular $(t+1)$-gon form a tight spherical t-design, that is, $N^*(1,t) = t+1$. However, it is noteworthy that tight spherical t-designs with $N^*(d,t)$ points exists only for $t = 1,2,3,4,5,7,11$ with different restrictions on the dimension d [5, 48]. On the other hand, Seymour and Zaslavsky [52] proved (nonconstructively) that a spherical t-design exists for any t if N is sufficiently large. Wagner [60] gave the first feasible upper bounds with $N = \mathcal{O}(t^{Cd^4})$. Korevaar and Meyers [33] further showed that spherical t-designs can be done with $N = \mathcal{O}(t^{d(d+1)/2})$ and conjectured that $N = \mathcal{O}(t^d)$. Using topological degree theory, Bondarenko, Radchenko, and Viazovska [6] proved that spherical t-designs indeed exist for $N = \mathcal{O}(t^d)$. They further showed that X_N can be *well-separated* in the sense that the minimal separation distance $\delta_{X_N} := \min_{1\leq i<j\leq N} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is of order $\mathcal{O}(N^{-1/d})$ [7]. Together with $N^*(d,t) = \mathcal{O}(t^d)$, one implies that $c_d t^d \leq N \leq C_d t^d$ for some constants $C_d \geq c_d > 0$ depending on d only and can conclude that the optimal asymptotic order is t^d.

Most of the real-world spherical data mentioned in the beginning are typically spherical signals on the 2-sphere $\mathbb{S}^2$. In this paper and in what follows, we are interested in spherical signal processing coming from many real-world applications. Hence, we restrict ourselves in the case of $d=2$ and consider the spherical t-design X_N on $\mathbb{S}^2$ obtained from numerical optimization methods. Hardin and Sloane [30] have extensively investigated spherical t-designs on $\mathbb{S}^2$ and suggested a sequence of putative spherical t-designs with $\frac{1}{2}t^2+o(t^2)$ points. Numerical calculation of spherical t-designs using multiobjective optimization was studied by Maier [40]. Numerical methods with computer-assisted proofs for computational spherical t-designs have been developed through nonlinear equations and optimization problems in [2, 9, 10]. Note that $\dim(\Pi_t)=(t+1)^2$ on $\mathbb{S}^2$. An *extremal point set* is a set of $(t+1)^2$ points on $\mathbb{S}^2$ that maximizes the determinant of a basis matrix for an arbitrary basis of Π_t. Sloan and Womersley [53] showed that the extremal point set has very nice geometric properties as the points are well-separated. By finding the solutions of systems of underdetermined equations and using the Krawczyk-type interval arithmetic technique, Chen and Womersley in [10] verified the existence of spherical t-designs with $(t+1)^2$ points for small t. In [9], Chen, Frommer, and Lang further improved the interval arithmetic technique and showed that spherical t-designs with $(t+1)^2$ points exist for all degrees t up to 100. The spherical t-designs with $(t+1)^2$ points are called *extremal spherical t-designs* and are also studied in [2]. Womersley [64] constructed *symmetric spherical t-designs* with $N=\frac{t^2+t+4}{2}$ for t up to 325. The interval arithmetic method [9] requires $\mathcal{O}(t^6)$ time complexity and thus prevents it from verifying the existence of spherical t-designs when t is large.

Sloan and Womersley [54] introduced a variational characterization of the spherical t-design via a nonnegative quantity $A_{N,t}(X_N)$ given by

$$A_{N,t}(X_N):=\frac{4\pi}{N^2}\sum_{\ell=1}^{t}\sum_{m=-\ell}^{\ell}\left|\sum_{i=1}^{N}Y_\ell^m(\boldsymbol{x}_i)\right|^2, \tag{1.2}$$

where Y_ℓ^m is the spherical harmonic with degree ℓ and order m. They gave some important properties for the relation between spherical t-designs and $A_{N,t}$. One is that X_N is a spherical t-design if and only if $A_{N,t}(X_N)=0$ (cf. Theorem 3 in [54]). Hence, the search of spherical t-designs is equivalent to finding the roots of the function $f(\boldsymbol{x}_1,\dots,\boldsymbol{x}_N):=A_{N,t}(X_N)$, which can be numerically solved via minimizing a nonlinear and nonconvex problem:

$$\min_{X_N\subset\mathbb{S}^2} A_{N,t}(X_N). \tag{1.3}$$

By using the addition theorem, the quantity can be rewritten in terms of the Legendre polynomials and the three-term recurrence can be used to speed up the numerical evaluations of $A_{N,t}$ (as well as its gradient and Hessian). However, since the formulation of $A_{N,t}$ in [54] essentially uses a full matrix of Legendre polynomial evaluations, the computations of numerical spherical t-designs are only feasible for small t. Gräf and Potts [25] rewrote $A_{N,t}$ using fast matrix-vector evaluations based on optimization techniques on manifold and the nonequispaced fast spherical Fourier transforms (NFSFTs). They computed numerical spherical t-designs for $t\le 1000$ with $N\approx\frac{t^2}{2}$.

Once spherical t-design point sets are obtained, signals on the sphere can be modeled as samples of functions on such point sets. Sparsity is the key to exploiting the underlying structures of the signals for various applications such as signal denoising. It is well-known that sparsity can be well-exploited using multiresolution analysis techniques, which are widely used in terms of wavelet analysis in Euclidean space $\mathbb{R}^d, d \geq 1$. Multiscale representation systems including wavelets, framelets, curvelets, shearlets, etc., have been developed for the sparse representations of data (see, e.g., [13, 17, 26, 27, 29, 35, 41, 65]) over the past four decades, which play an important role in approximation theory, computer graphics, statistical inference, compressed sensing, numerical solutions of PDEs, and so on. Wavelets on the sphere first appeared in [47, 49, 51]. Later, Antonio and Vandergheynst in [3, 4] used a group-theoretical approach to construct continuous wavelets on the spheres. Localized frames on the sphere were studied in [36, 43], which use polynomial-exact quadrature rules on $\mathbb{S}^d$. Based on hierarchical partitions, area-regular spherical Haar tight framelets were constructed in [37]. Extension of wavelets/framelets on the sphere with more desirable properties, such as localized property, tight frame property, symmetry, directionality, etc., were further studied in [20, 32, 63, 42, 45] and many references therein. In [61], based on orthogonal eigenpairs, localized kernels, filter banks, and affine systems, Wang and Zhuang provided a general framework for the construction of tight framelets on a compact smooth Riemannian manifolds and considered their discretizations through polynomial-exact quadrature rules. Fast framelet filter bank transforms are developed and their realizations on the 2-sphere are demonstrated.

In this paper, we further exploit the structure of $A_{N,t}$ and employ the trust-region method together with the NFSFTs to find the numerical spherical t-designs for large values of t beyond 1000. Moreover, we focus on the development of spherical tight framelets on $\mathbb{S}^2$ with fast transform algorithms based on the spherical t-designs for practical spherical signal processing. The contributions of this paper lie in the following aspects. First, we investigate in detail the structures of $A_{N,t}$, its gradient $\nabla A_{N,t}$, and its Hessian $\mathcal{H}(A_{N,t})$ in terms of the fast evaluations of spherical harmonic transforms and their adjoints without the needs of referring to their manifold versions as in [25]. Moreover, we proposed solving the minimization problem (1.3) using the trust-region method to provide spherical t-designs with large values of t. Second, (semidiscrete) spherical tight framelet systems are developed based on the obtained spherical t-design point sets. More importantly, a truncated spherical framelet system is introduced for discrete spherical signal representations and its associated fast spherical framelet (filter bank) transforms are realized for practical signal processing on the sphere. Third, thanks to the high-degree spherical t-designs and localization property of our framelets, we are able to provide signal/image denoising using local thresholding techniques based on a fine-tuned spherical cap [16, 31] restrictions. Last but not least, many numerical experiments are conducted to demonstrate the efficiency and effectiveness of our spherical framelets and spherical designs, including Wendland function approximation, ETOPO data processing, and spherical image denoising.

This paper is organized as follows. We introduce the trust-region method for finding the spherical t-designs in section 2 including the fast evaluations for $A_{N,t}$, its gradient $\nabla A_{N,t}$, and its Hessian $\mathcal{H}(A_{N,t})$. In section 3, we demonstrate the numerical spherical t-designs obtained from various initial point sets and use them for Wendland function approximation. In section 4, based on the spherical t-designs, we provide the construction, characterizations, and algorithmic realizations of the spherical framelet systems as well as their truncated spherical framelet systems. In section 5, numerical experiments to demonstrate the applications of the truncated spherical framelet systems in spherical signal/image denoising are conducted. Finally, the conclusion and final remarks are given in section 6.

2. Spherical t-designs from trust-region optimization.

In this section, we briefly introduce the trust-region method for solving a general optimization problem and show how it can be applied to find spherical t-designs with large values of t.

2.1. Trust-region optimization.

As we mentioned in the introduction, finding X_N to achieve the minimum of $A_{N,t}$ in (1.3) can be regarded as a general nonlinear and nonconvex optimization problem:

$$\min_{x\in X} f(x), \tag{2.1}$$

where $f:\mathbb{R}^d\to\mathbb{R}$ is the objective function to be minimized and $X\subset\mathbb{R}^d$ is a feasible set. There are mainly two global convergence approaches to solve (2.1): one is the line search, and another is the trust region. The line search approach uses the quadratic model to generate a search direction and then find a suitable step size along that direction. Though such a line search method is successful most of the time, it may not exploit the d-dimensional quadratic model sufficiently. Unlike the line search approach, the *trust-region method* obtains a new iterate point by searching in a neighborhood (trust region) of the current iterated point. The trust-region method has many advantages over the line search method such as robustness of algorithms, easier establishment of convergence results, second-order stationary point convergence, and so on. The trust-region method has been developed over 70 years, we briefly give an introduction below. For more details, we refer to the book [57].

Suppose that the objective function f is at least twice differentiable. The gradient of f at $x=(\xi_1,\dots,\xi_d)\in\mathbb{R}^d$ is defined as

$$\nabla f(x) := \left[\frac{\partial}{\partial\xi_1}f(x),\dots,\frac{\partial}{\partial\xi_d}f(x)\right]^\top, \tag{2.2}$$

and the Hessian of f is defined as a $d\times d$ symmetric matrix with elements

$$[\mathcal{H}f(x)]_{ij} := \frac{\partial^2}{\partial\xi_i\partial\xi_j}f(x), \qquad 1\le i,j\le d. \tag{2.3}$$

Suppose that x_k is the current iterated point and consider the quadratic model to approximate the original objective function $f(x)$ at $x=x_k$: $q^{(k)}(s)=f(x_k)+g_k^\top s+\frac{1}{2}s^\top A_k s$, where $g_k=\nabla f(x_k)$ and $A_k=\mathcal{H}f(x_k)$. Then the optimization problem (2.1) is essentially reduced to solve a sequence of trust-region subproblems:

$$\min_s \quad q^{(k)}(s)=f(x_k)+g_k^\top s+\frac{1}{2}s^\top B_k s \quad \text{ s.t.} \quad \|s\|\le\Delta_k, \tag{2.4}$$

where B_k could be exactly equal to A_k or is a symmetric approximation to A_k ($B_k\approx A_k$). This is equivalent to searching a new point x_{k+1} in a region $\Omega_k=\{x:\|x-x_k\|\le\Delta_k\}$ centered at x_k with radius Δ_k. The trust-region algorithm is presented in Algorithm 2.1, where with the initial (x_0,Δ_0) and some parameters $\overline{\Delta},\eta_1,\eta_2,\nu_1,\nu_2$ given, the algorithm iteratively solves s_k in

Algorithm 2.1 Trust-region algorithm

Input: x: initial point; $K_{\max}$: maximum iterations; ε: termination tolerance;
Initialize $k=0$, $x_0=x$, $\overline{\Delta}$, $\Delta_0\in(0,\overline{\Delta})$, $0<\eta_1\le\eta_2<1$, $0<\nu_1<1<\nu_2$.

1: **while** $k\le K_{\max}$ and $\|g_k\|>\varepsilon$ **do**

2: approximately solve the subproblem 2.4 for s_k.

3: compute $f(x_k+s_k)$ and $\tau_k=\frac{f(x_k)-f(x_{k+1})}{q^{(k)}(0)-q^{(k)}(s_k)}$. Set

$$x_{k+1}=\begin{cases}x_k+s_k, & \tau_k\ge\eta_1,\\ x_k & \text{otherwise.}\end{cases}$$

4: Choose Δ_{k+1} satisfies

$$\Delta_{k+1}\in\begin{cases}(0,\nu_1\Delta_k], & \tau_k<\eta_1,\\ [\nu_1\Delta_k,\Delta_k], & \tau_k\in[\eta_1,\eta_2),\\ [\Delta_k,\min\{\nu_2\Delta_k,\overline{\Delta}\}], & \tau_k\ge\eta_2 \text{ and } \|s_k\|=\Delta_k.\end{cases}$$

5: Update $g_{k+1}=\nabla f(x_{k+1})$ and $B_{k+1}\approx(\mathcal{H}f)(x_{k+1})$. Set $k=k+1$.

6: **end while**

Output: minimizer x^*.

(2.4) approximately (line 2) by the preconditioned conjugate gradient (PCG) algorithm given in Algorithm 2.2 and updates (x_{k+1},Δ_{k+1}) from current (x_k,Δ_k) according to the quantities τ_k (lines 3–4). Algorithm 2.2 for solving the subproblem (2.4) is proposed by Steihaug [56] based on a preconditioned and truncated conjugate gradient method. For more details and how to choose the precondition matrix W, we refer to [14].

A well-known result regarding the convergence of Algorithm 2.1 is given as follows, which shows that the sequence $\{x_k\}_{k=1}^{\infty}$ converges to a stationary point of f.

Theorem 2.1 (see [57]). *Suppose that $f:\mathbb{R}^d\to\mathbb{R}$ is continuously differentiable on a bounded level set $L=\{x\in\mathbb{R}^d\,|\,f(x)\le f(x_0)\}$, the approximate Hessian B_k is uniformly bounded in norm, and solution s_k of the trust-region subproblem (2.4) is bounded with $\|s_k\|\le\tilde{\eta}\Delta_k$, where $\tilde{\eta}>0$ is a constant. Then, the sequence g_k of Algorithm 2.1 satisfies $\lim_{k\to\infty}g_k=0$.*

Regarding the computational time complexity of the trust-region method, the total cost of Algorithm 2.1 includes the cost from the total outer iteration steps k_{wh} (the while-loop in line 1) in Algorithm 2.1, where each outer iteration has the cost from the inner iteration steps $k_{for,i}$ (the for-loop in line 4) in Algorithm 2.2. The total number of iterations is $K_{TR}=\sum_{i=1}^{k_{wh}}k_{for,i}$. In each iteration of either inner or outer, the main cost comes from the evaluations of f, the gradient $g=\nabla f$, and the Hessian $\mathcal{H}f$ (or its approximation). Denote $C_f,C_g,C_{\mathcal{H}}$ their computational time complexity, respectively. Then, the total computational time complexity of Algorithm 2.1 is of order $\mathcal{O}(K_{TR}\cdot(C_f+C_g+C_{\mathcal{H}}))$. We proceed next to discuss the minimization problem (2.1) under the setting of spherical t-design, i.e., $f=A_{N,t}$, and its related evaluations and complexity.

Algorithm 2.2 PCG algorithm for trust-region subproblem (2.4)

Input: x_k: initial point; $K_{\max}$: maximum iterations; ε_k: kth termination tolerance; W: precondition matrix. $\|g\|_W := \sqrt{g^\top W g}$.
Initialize $z_0 = 0$, $g_0 = \nabla f(x_k)$, $\gamma_0 = -d_0 = W^{-1} g_0$, $\quad B_k \approx (\mathcal{H} f)(x_k)$.

if $\|g_0\| < \varepsilon_k$ **then**
  $s_k = z_0$
else
  for $j = 0, 1, \dots, K_{\max}$ **do**
    if $d_j^\top B_k d_j \le 0$ **then**
      find $\rho > 0$ s.t. $\|z_j + \rho d_j\|_W = \Delta_k$; Set $s_k = z_j + \rho d_j$.
      break
    end if
    Set $\alpha_j = \frac{g_j^\top \gamma_j}{d_j^\top B_k d_j}$ and $z_{j+1} = z_j + \alpha_j d_j$.
    if $\|z_{j+1}\|_W \ge \Delta_k$ **then**
      find $\rho > 0$ s.t. $\|z_j + \rho d_j\|_W = \Delta_k$; Set $s_k = z_j + \rho d_j$.
      break
    end if
    $g_{j+1} = g_j + \alpha_j B_k d_j$.
    if $\|g_{j+1}\|_W < \varepsilon_k \|g_0\|_W$ **then**
      $s_k = z_{j+1}$.
      break
    end if
    Set $\gamma_{j+1} = W^{-1} g_{j+1}$, $\beta_j = \frac{g_{j+1}^\top \gamma_{j+1}}{g_j^\top \gamma_j}$, $d_{j+1} = -\gamma_{j+1} + \beta_j d_j$.
  end for
end if

Output: solution s_k^*.

2.2. Fast evaluations of $A_{N,t}$, $\nabla A_{N,t}$, and $\mathcal{H}(A_{N,t})$.

In what follows, we give details on the evaluations of $A_{N,t}$, $\nabla A_{N,t}$, and $\mathcal{H}(A_{N,t})$.

For $\boldsymbol{x} \in \mathbb{S}^2$, in terms of the spherical coordinate $(\theta, \phi) \in [0, \pi] \times [0, 2\pi)$, we can represent it as $\boldsymbol{x} = \boldsymbol{x}(\theta, \phi) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \in \mathbb{S}^2$. For each $\ell \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and $m = -\ell, \dots, \ell$, the spherical harmonic Y_ℓ^m can be expressed as

$$Y_\ell^m(\boldsymbol{x}) = Y_\ell^m(\theta, \phi) := \sqrt{\frac{(2\ell+1)}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos\theta) \mathrm{e}^{\mathrm{i} m \phi}, \tag{2.5}$$

where $P_\ell^m : [-1, 1] \to \mathbb{R}$ is the associated Legendre polynomial given by $P_\ell^m(z) = (-1)^m (1 - z^2)^{\frac{m}{2}} \frac{\mathrm{d}^m}{\mathrm{d}z^m} P_\ell(z)$ for $\ell \in \mathbb{N}_0$ and $m = 0, \dots, \ell$ with $P_\ell : [-1, 1] \to \mathbb{R}$ being the Legendre polynomial given by $P_\ell(z) = \frac{1}{2^\ell \ell!} \frac{\mathrm{d}^\ell}{\mathrm{d}z^\ell} [(z^2 - 1)^\ell]$ for $\ell \in \mathbb{N}_0$. We use the convention $P_\ell^{-m} := (-1)^m \frac{(\ell-m)!}{(\ell+m)!} P_\ell^m$ to define Y_ℓ^m with negative m. Note that $Y_0^0 = \frac{1}{\sqrt{4\pi}}$. Then, we have $\Pi_t = \operatorname{span}\{Y_\ell^m \mid (\ell, m) \in \mathcal{I}_t\}$ with the index set

$$\mathcal{I}_t := \{(\ell,m) \mid \ell = 0,\dots,t; m = -\ell,\dots,\ell\}. \tag{2.6}$$

Moreover, $\{Y_\ell^m \mid \ell \in \mathbb{N}_0, |m| \le \ell\}$ forms an orthonormal basis for the Hilbert space $L^2(\mathbb{S}^2) := \{f : \mathbb{S}^2 \to \mathbb{C} \mid \int_{\mathbb{S}^2} |f(\boldsymbol{x})|^2 \mathrm{d}\mu_2(\boldsymbol{x}) < \infty\}$ of square-integrable functions on $\mathbb{S}^2$, i.e., $\langle Y_\ell^m, Y_{\ell'}^{m'}\rangle = \delta_{mm'}\delta_{\ell\ell'}$, where the inner product is defined as $\langle f_1, f_2\rangle := \int_{\mathbb{S}^2} f_1(\boldsymbol{x})\overline{f_2(\boldsymbol{x})} d\mu_2(\boldsymbol{x})$ for $f_1, f_2 \in L^2(\mathbb{S}^2)$ and δ_{ij} is the Kronecker delta. Consequently, any function $f \in L^2(\mathbb{S}^2)$ has the L^2-representation $f = \sum_{\ell=0}^\infty \sum_{m=-\ell}^{\ell} \hat{f}_\ell^m Y_\ell^m$, where $\hat{f}_\ell^m := \langle f, Y_\ell^m\rangle$ is its spherical harmonic (Fourier) coefficient with respect to Y_ℓ^m.

In terms of (θ,ϕ), $A_{N,t}(X_N)$ can be regarded as a function of $2N$ variables. In fact, we can identify the point set $X_N = \{\boldsymbol{x}_1,\dots,\boldsymbol{x}_N\} \subset \mathbb{S}^2$ as

$$X_N := (\boldsymbol{\theta}, \boldsymbol{\phi}) := (\theta_1,\dots,\theta_N,\phi_1,\dots,\phi_N) \tag{2.7}$$

with $\boldsymbol{\theta} = (\theta_1,\dots,\theta_N)$, $\boldsymbol{\phi} = (\phi_1,\dots,\phi_N)$, and $\boldsymbol{x}_i := \boldsymbol{x}_i(\theta_i,\phi_i)$ being the ith point determined by its spherical coordinate satisfying $(\theta_i,\phi_i) \in [0,\pi] \times [0, 2\pi)$. In what follows, we identify $\boldsymbol{x}_i = (\theta_i,\phi_i)$ if no ambiguity appears. Denote $[N] := \{1,\dots,N\}$ to be the index set of size N. Then, the variational characterization $A_{N,t}(X_N)$ in (1.2) can be written as a smooth function of $2N$ variables:

$$A_{N,t}(X_N) = A_{N,t}(\boldsymbol{\theta},\boldsymbol{\phi}) = \frac{4\pi}{N^2} \sum_{(\ell,m)\in\mathcal{I}_t} \left| \sum_{i\in[N]} Y_\ell^m(\theta_i,\phi_i)\right|^2 - 1. \tag{2.8}$$

For any point set X_N and degree t, we have $\dim \Pi_t = (t+1)^2$ and the matrix $\boldsymbol{Y}_t := \boldsymbol{Y}_t(X_N) := (Y_\ell^m(\theta_i,\phi_i))_{i\in[N],(\ell,m)\in\mathcal{I}_t}$ is of size $N \times (t+1)^2$:

$$\boldsymbol{Y}_t = \begin{bmatrix} Y_0^0(\boldsymbol{x}_1) & Y_1^{-1}(\boldsymbol{x}_1) & Y_1^0(\boldsymbol{x}_1) & \cdots & Y_t^t(\boldsymbol{x}_1) \\ Y_0^0(\boldsymbol{x}_2) & Y_1^{-1}(\boldsymbol{x}_2) & Y_1^0(\boldsymbol{x}_2) & \cdots & Y_t^t(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_0^0(\boldsymbol{x}_N) & Y_1^{-1}(\boldsymbol{x}_N) & Y_1^0(\boldsymbol{x}_N) & \cdots & Y_t^t(\boldsymbol{x}_N) \end{bmatrix}. \tag{2.9}$$

Its transpose of complex conjugate is $\boldsymbol{Y}_t^\star := \overline{\boldsymbol{Y}_t(X_N)}^\top \in \mathbb{C}^{(t+1)^2\times N}$. Let $\boldsymbol{e} := [1,\dots,1]^\top$ be a vector of size N. We use $\Re(\cdot)$ to denote the (entrywise) operation of taking the real part of a complex object (scalar, vector, or matrix).

We have the following theorem that summarizes the evaluations of $A_{N,t}$ and $\nabla A_{N,t}$ in a concise matrix-vector form in terms of $\boldsymbol{Y}_t$ and $\boldsymbol{Y}_t^\star$.

Theorem 2.2. *Fix $t \in \mathbb{N}_0$. Let $A_{N,t}$ be defined as in (2.8) and define*

$$\hat{c}_\ell^m = \sum_{i=1}^N \overline{Y_\ell^m(\boldsymbol{x}_i)}, \quad a_\ell^m = \sqrt{\frac{\ell^2[(\ell+1)^2 - m^2]}{(2\ell+1)(2\ell+3)}}, \quad b_\ell^m = \sqrt{\frac{(\ell+1)^2(\ell^2-m^2)}{(2\ell-1)(2\ell+1)}}$$

for $(\ell,m) \in \mathcal{I}_{t+2}$ with the convention $a_{-1}^m = b_{t+1}^m = b_{t+2}^m = 0$. Define vectors $\hat{\boldsymbol{c}}_0 \in \mathbb{C}^{(t+2)^2}$, $\hat{\boldsymbol{d}}_0 \in \mathbb{C}^{(t+1)^2}$ and a diagonal matrix $\boldsymbol{D}_{\boldsymbol{\theta}}$ as follows:

$$\hat{\boldsymbol{c}}_0 := \frac{8\pi}{N^2} \cdot (\hat{c}_{\ell-1}^m a_{\ell-1}^m - \hat{c}_{\ell+1}^m b_{\ell+1}^m)_{(\ell,m)\in\mathcal{I}_{t+1}},$$
$$\hat{\boldsymbol{d}}_0 := \frac{8\pi}{N^2}(\mathrm{i}m\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_t}, \quad \boldsymbol{D}_{\boldsymbol{\theta}} := \operatorname{diag}\left(\frac{1}{\sin\theta_1},\dots,\frac{1}{\sin\theta_N}\right).$$

Then, the $A_{N,t}$ and its gradient $\nabla A_{N,t}$ in matrix-vector forms are given by

$$A_{N,t}(X_N) = \frac{4\pi}{N^2}\|\boldsymbol{Y}_t^\star \boldsymbol{e}\|^2 - 1, \tag{2.10}$$

$$\nabla A_{N,t}(X_N) = \Re \begin{bmatrix} \boldsymbol{D_\theta Y}_{t+1}\hat{\boldsymbol{c}}_0 \\ \boldsymbol{Y}_t \hat{\boldsymbol{d}}_0 \end{bmatrix}. \tag{2.11}$$

Proof. The matrix-vector form of $A_{N,t}$ in (2.10) directly follows from (2.8) and (2.9). To rewrite the gradient of $A_{N,t}$ in the matrix-vector form, we have

$$\begin{aligned} \frac{\partial}{\partial \xi_j} A_{N,t}(X_N) &= \frac{8\pi}{N^2}\Re\left[\sum_{(\ell,m)\in\mathcal{I}_t}\left(\sum_{i\in[N]}\overline{Y_\ell^m(\boldsymbol{x}_i)}\right)\frac{\partial}{\partial \xi_j}Y_\ell^m(\boldsymbol{x}_j)\right] \\ &= \frac{8\pi}{N^2}\Re\left[\sum_{(\ell,m)\in\mathcal{I}_t}\hat{c}_\ell^m \frac{\partial}{\partial \xi_j}Y_\ell^m(\boldsymbol{x}_j)\right], \end{aligned} \tag{2.12}$$

where $\xi_j \in \{\theta_j, \phi_j\}$. Based on the following formulae [59],

$$\frac{\partial}{\partial\theta}Y_\ell^m = \frac{1}{\sin\theta}\left[a_\ell^m Y_{\ell+1}^m - b_\ell^m Y_{\ell-1}^m\right], \quad \frac{\partial}{\partial\phi}Y_\ell^m = \mathrm{i}mY_\ell^m, \tag{2.13}$$

we can thus deduce that

$$\frac{\partial}{\partial\theta_i}A_{N,t}(X_N) = \frac{8\pi}{N^2}\cdot\frac{1}{\sin\theta_i}\Re\left[\sum_{(\ell,m)\in\mathcal{I}_{t+1}}\left(\hat{c}_{\ell-1}^m a_{\ell-1}^m - \hat{c}_{\ell+1}^m b_{\ell+1}^m\right)Y_\ell^m(\boldsymbol{x}_i)\right], \tag{2.14}$$

$$\frac{\partial}{\partial\phi_i}A_{N,t}(X_N) = \frac{8\pi}{N^2}\Re\left[\sum_{(\ell,m)\in\mathcal{I}_t}(\mathrm{i}m\hat{c}_\ell^m)Y_\ell^m(\boldsymbol{x}_i)\right], \tag{2.15}$$

which imply the expressions of $\nabla A_{N,t}$ in (2.11). We are done. ∎

The following theorem gives the evaluation of the Hessian in matrix-vector form.

Theorem 2.3. *Retain notation in Theorem 2.2 and further define*

$$\begin{aligned} &\hat{\boldsymbol{d}}_1 := \frac{8\pi}{N^2}\cdot(a_{\ell-1}^m - b_{\ell+1}^m)_{(\ell,m)\in\mathcal{I}_{t+1}}, && \hat{\boldsymbol{d}}_2 := \frac{8\pi}{N^2}\cdot(\mathrm{i}m\cdot 1^{\ell-\ell})_{(\ell,m)\in\mathcal{I}_{t+1}}, \\ &\hat{\boldsymbol{c}}_1 := \frac{8\pi}{N^2}\cdot(\hat{c}_\ell^m\cdot(-m^2))_{(\ell,m)\in\mathcal{I}_t}, && \hat{\boldsymbol{c}}_2 := \frac{8\pi}{N^2}\cdot(\hat{c}_\ell^m\cdot\ell(\ell+1))_{(\ell,m)\in\mathcal{I}_t}, \\ &\hat{\boldsymbol{c}}_3 := \frac{8\pi}{N^2}\cdot(\mathrm{i}m(\hat{c}_{\ell-1}^m a_{\ell-1}^m - \hat{c}_{\ell+1}^m b_{\ell+1}^m))_{(\ell,m)\in\mathcal{I}_{t+1}}, && \boldsymbol{C_\theta} := \mathrm{diag}(\cot\theta_1,\ldots,\cot\theta_N). \end{aligned}$$

Then, the Hessian $\mathcal{H}(A_{N,t})$ can be written as

$$\mathcal{H}(A_{N,t}) = \Re\left(\begin{bmatrix}\boldsymbol{F_{\theta\theta}} & \boldsymbol{F_{\theta\phi}} \\ \boldsymbol{F_{\phi\theta}} & \boldsymbol{F_{\phi\phi}}\end{bmatrix} + \begin{bmatrix}\boldsymbol{E_\theta} \\ \boldsymbol{E_\phi}\end{bmatrix}\begin{bmatrix}\boldsymbol{E_\theta}^\top & \boldsymbol{E_\phi}^\top\end{bmatrix}\right), \tag{2.16}$$

where

$$\boldsymbol{E}_{\boldsymbol{\theta}} = \boldsymbol{D}_\theta \boldsymbol{Y}_{t+1}\hat{\boldsymbol{d}}_1 \quad \textit{and} \quad \boldsymbol{E}_{\boldsymbol{\phi}} = \boldsymbol{Y}_t\hat{\boldsymbol{d}}_2, \tag{2.17}$$

$$\boldsymbol{F}_{\boldsymbol{\theta\theta}} = \operatorname{diag}(\boldsymbol{D}_{\boldsymbol{\theta}}^2\boldsymbol{Y}_t\hat{\boldsymbol{c}}_1 - \boldsymbol{Y}_t\hat{\boldsymbol{c}}_2 - \boldsymbol{C}_{\boldsymbol{\theta}}\boldsymbol{D}_\theta\boldsymbol{Y}_{t+1}\hat{\boldsymbol{c}}_0), \tag{2.18}$$

$$\boldsymbol{F}_{\boldsymbol{\theta\phi}} = \boldsymbol{F}_{\boldsymbol{\phi\theta}} = \operatorname{diag}(\boldsymbol{D}_{\boldsymbol{\theta}}\boldsymbol{Y}_{t+1}\hat{\boldsymbol{c}}_3), \quad \textit{and} \quad \boldsymbol{F}_{\boldsymbol{\phi\phi}} = \operatorname{diag}(\boldsymbol{Y}_t\hat{\boldsymbol{c}}_1). \tag{2.19}$$

Proof. For the Hessian of $A_{N,t}$, from (2.12), we have

$$\frac{\partial^2}{\partial\xi_j\partial\zeta_l}A_{N,t}(X_N) = \frac{8\pi}{N^2}\Re\left[\sum_{(\ell,m)\in\mathcal{I}_t}\left(\frac{\partial}{\partial\xi_j}\overline{Y_\ell^m(\boldsymbol{x}_j)}\frac{\partial}{\partial\zeta_l}Y_\ell^m(\boldsymbol{x}_l) + \hat{c}_\ell^m\delta_{jl}\frac{\partial^2}{\partial\xi_l\partial\zeta_l}Y_\ell^m(\boldsymbol{x}_l)\right)\right]$$

for $\xi_j \in \{\theta_j, \phi_j\}$ and $\zeta_l \in \{\theta_l, \phi_l\}$. Hence, by definition, the Hessian can be written as in (2.16), where each $\boldsymbol{F}_{\boldsymbol{\xi\zeta}}$ is a diagonal matrix and each $\boldsymbol{E}_{\boldsymbol{\xi}}$ is a vector for $\boldsymbol{\xi}, \boldsymbol{\zeta} \in \{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ determined by $\boldsymbol{E}_{\boldsymbol{\xi}} = (\frac{8\pi}{N^2}\sum_{(\ell,m)\in\mathcal{I}_t}\frac{\partial}{\partial\xi_l}Y_\ell^m(\boldsymbol{x}_l))_{l\in[N]} \in \mathbb{C}^{N\times 1}$ and $\boldsymbol{F}_{\boldsymbol{\xi\zeta}} = \operatorname{diag}(\frac{8\pi}{N^2}\sum_{(\ell,m)\in\mathcal{I}_t}\hat{c}_\ell^m \cdot \frac{\partial^2}{\partial\xi_l\partial\zeta_l}Y_\ell^m(\boldsymbol{x}_l))_{l\in[N]}$.

Now by (2.13), similar to the derivation of (2.14) and (2.15), we can deduce that

$$\boldsymbol{E}_{\boldsymbol{\theta}} = \left(\frac{8\pi}{N^2}\frac{1}{\sin\theta_l}\sum_{(\ell,m)\in\mathcal{I}_{t+1}}(a_{\ell-1}^m - b_{\ell+1}^m)Y_\ell^m(\boldsymbol{x}_l)\right)_{l\in[N]},$$

$$\boldsymbol{E}_{\boldsymbol{\phi}} = \left(\frac{8\pi}{N^2}\sum_{(\ell,m)\in\mathcal{I}_t}(\mathrm{i}m)Y_\ell^m(\boldsymbol{x}_l)\right)_{l\in[N]},$$

which are equivalent to (2.17).

Repeating applying (2.13), we have

$$\frac{\partial^2}{\partial\theta^2}Y_\ell^m = \left[\frac{m^2}{\sin^2\theta} - \ell(\ell+1)\right]Y_\ell^m - \cot\theta\frac{\partial}{\partial\theta}Y_\ell^m, \quad \frac{\partial^2}{\partial\phi^2}Y_\ell^m = -m^2Y_\ell^m, \tag{2.20}$$

$$\frac{\partial^2}{\partial\theta\partial\phi}Y_\ell^m = \frac{\partial^2}{\partial\phi\partial\theta}Y_\ell^m = \mathrm{i}m\frac{\partial}{\partial\theta}Y_\ell^m = \frac{1}{\sin\theta}\left[\mathrm{i}ma_\ell^mY_{\ell+1}^m - \mathrm{i}mb_\ell^mY_{\ell-1}^m\right]. \tag{2.21}$$

Hence, the lth diagonal entries of $\boldsymbol{F}_{\boldsymbol{\xi\zeta}}$ are given by

$$[\boldsymbol{F}_{\boldsymbol{\theta\theta}}]_{l,l} = \frac{8\pi}{N^2}\sum_{(\ell,m)\in\mathcal{I}_t}\hat{c}_\ell^m\cdot\left(\frac{m^2}{\sin^2\theta_l}Y_\ell^m(\boldsymbol{x}_l) - \ell(\ell+1)Y_\ell^m(\boldsymbol{x}_l) - \cot\theta_l\frac{\partial}{\partial\theta_l}Y_\ell^m(\boldsymbol{x}_l)\right),$$

$$[\boldsymbol{F}_{\boldsymbol{\phi\phi}}]_{l,l} = \frac{8\pi}{N^2}\sum_{(\ell,m)\in\mathcal{I}_t}\hat{c}_\ell^m\cdot(-m^2)Y_\ell^m(\boldsymbol{x}_l),$$

$$[\boldsymbol{F}_{\boldsymbol{\theta\phi}}]_{l,l} = \frac{8\pi}{N^2}\sum_{(\ell,m)\in\mathcal{I}_{t+1}}\frac{1}{\sin\theta_l}[\mathrm{i}m(\hat{c}_{\ell-1}^m a_{\ell-1}^m - \hat{c}_{\ell+1}^m b_{\ell+1}^m)]Y_\ell^m(\boldsymbol{x}_l),$$

which imply (2.18) and (2.19). This concludes the proof. ∎

2.3. Computational time complexity. Note that in Theorem 2.2, the coefficient vector $(\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_t} = \boldsymbol{Y}_t^\star \boldsymbol{e}$ and a_ℓ^m, b_ℓ^m can be precomputed. Moreover, the vectors $\hat{\boldsymbol{c}}_k, \hat{\boldsymbol{d}}_k$ and diagonal matrices in Theorems 2.2 and 2.3 can be evaluated in the order of $\mathcal{O}(t^2+N)$. Thanks to the nice structure of $\mathcal{H}(A_{N,t}) = \mathcal{H}_1 + \mathcal{H}_2$ in (2.16), where $\mathcal{H}_1$ is formed by diagonal matrices and $\mathcal{H}_2$ is a rank one matrix, one only needs to implement the matrix-vector multiplication. Moreover, in the trust-region algorithm, the exact Hessian is not required, an approximation of the Hessian could be enough for the desired convergence. In such a case, one can either use $\mathcal{H}_1$ or $\mathcal{H}_2$.

Regarding the evaluations involving $\boldsymbol{Y}_t, \boldsymbol{Y}_t^\star$, fast evaluations have been developed in terms of spherical harmonic transforms (SHTs). We use the package developed by Kunis and Potts [34], where it shows that the NFSFTs $\boldsymbol{y} = \boldsymbol{Y}_t(X_N)\hat{\boldsymbol{c}}$ to obtain $\boldsymbol{y}\in\mathbb{C}^N$ from a given vector $\hat{\boldsymbol{c}}\in\mathbb{C}^{(t+1)^2}$ as well as its adjoint $\hat{\boldsymbol{c}} = \boldsymbol{Y}_t(X_N)^\star\boldsymbol{y}$ can be done in the order of $\mathcal{O}(t^2\log^2 t + N\log^2(\frac{1}{\epsilon}))$ with ϵ being a prescribed accuracy of the algorithms.

From above, we see that the evaluations of $f = A_{N,t}$, the gradient $g = \nabla A_{N,t}$, and the Hessian $\mathcal{H}(A_{N,t})$ only involve diagonal matrices, rank one matrices, NFSFTs $\boldsymbol{y} = \boldsymbol{Y}_t(X_N)\hat{\boldsymbol{c}}$, and their adjoints $\hat{\boldsymbol{c}} = \boldsymbol{Y}_t(X_N)^\star\boldsymbol{y}$. Therefore, the computational time complexity of $C_f + C_g + C_{\mathcal{H}}$ is of order $\mathcal{O}(t^2\log^2 t + N\log^2(\frac{1}{\epsilon}))$. Therefore, the trust-region algorithm in Algorithm 2.1 for computing the spherical t-design point set is with computational time complexity $\mathcal{O}(K_{TR}\cdot(t^2\log^2 t + N\log^2(\frac{1}{\epsilon})))$.

3. Numerical spherical t-designs.

In this section,[1] we show that the numerical spherical t-designs obtained from different initial point sets using Algorithm 2.1. For Algorithm 2.1, in view of the rotation invariance property of the spherical-t design, we preprocess the initial point set $X_N \subset \mathbb{S}^2$ by fixing the first point $\boldsymbol{x}_1 = (\theta_1, \phi_1) = (0,0) \in X_N$ as the north pole point and the second point $\boldsymbol{x}_2 = (\theta_2, 0) \in X_N$ on the prime meridian. Moreover, we set $\varepsilon = 2.2204$E-16 (floating-point relative accuracy of MATLAB) and $K_{\max} = 1$E+7. We introduce four types of initial point sets on $\mathbb{S}^2$ as follows:

(I) Spiral points (SPs). The SPs $\boldsymbol{x}_k = (\theta_k, \phi_k)$ on $\mathbb{S}^2$ for $k\in[N]$ are generated by $\theta_k := \arccos\left(\frac{2k-(N+1)}{N}\right)$ and $\phi_k := \pi(2k-(N+1))/\mathfrak{g}$, where $\mathfrak{g} = \frac{1+\sqrt{5}}{2}$ is the golden ratio [58]. This is the Fibonacci spiral points on the sphere, the same as the initial SPs in [25]. For the SP point sets, we set $N = N(t) = (t+1)^2$ for large $t\in\mathbb{N}$.

(II) Uniformly distributed (UD) points. We generate UD points on the unit sphere $\mathbb{S}^2$ according to the surface area element $\mathrm{d}\mu_2 = \sin\theta\mathrm{d}\theta\mathrm{d}\phi$. By [62], we generate $k_i\in(0,1)$ and $s_i\in(0,1)$ uniformly for $i\in[N]$ and define $\boldsymbol{x}_i = (\theta_i, \phi_i)$ by $\theta_i := \arccos(1-2k_i)$, and $\phi_i := 2\pi s_i$. For UD point sets, we set $N = N(t) = (t+1)^2$ for $t\in\mathbb{N}$.

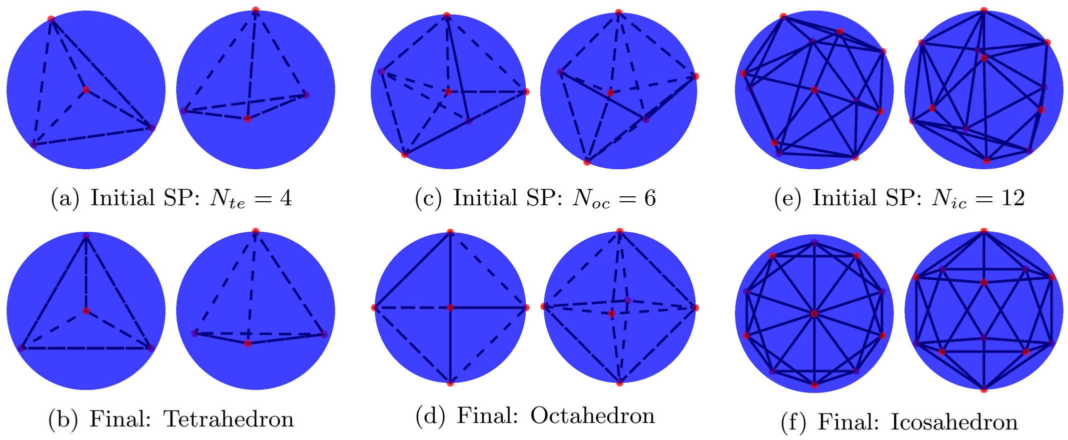
(III) Icosahedron vertices (IVs) mesh points. An icosahedron has 12 vertices, 30 edges, and 20 faces. The faces of the icosahedron are equilateral triangles. Icosahedron is a polyhedron whose vertices can be used as the starting points for sphere tessellation. After generating the IVs, one can get the triangular surface mesh of the Pentakis dodecahedron. The number N of IV points must satisfy $N = N(k) = 4^{k-1}\times 10 + 2$ for $k\in\mathbb{N}$. In this paper, we fix the relation between t and N to be $N \approx (t+1)^2$. That is, we set $t = t(N(k)) = \lfloor\sqrt{4^{k-1}\times 10+2} - 1\rfloor$, where $\lfloor\cdot\rfloor$ is the floor operator.

[1] All numerical experiments in this paper are conducted in MATLAB R2021b on a 64-bit Windows 10 Home desktop computer with Intel Core i9 9820X CPU and 32 GB DDR4 memory.

(IV) HEALpix points (HLs). Hierarchical Equal Area isoLatitude Pixelation points [24] are the isolatitude points on the sphere given by subdivisions of a spherical surface which produce hierarchical equal areas. The number N of HL points must satisfy $N = N(k) = 12 \times (2^{k-1})^2$ for $k \in \mathbb{N}$. Similarly, by requiring $N \approx (t+1)^2$, we set $t = t(N(k)) = \lfloor 2^k\sqrt{3} - 1\rfloor$.

3.1. Spherical t-designs of Platonic solids. We first give a numerical example to show the feasibility of the trust-region method (using the full Hessian $\mathcal{H}(A_{N,t})$) in Algorithm 2.1 to obtain the numerical spherical t-designs that are the famous regular polyhedrons of Platonic solids. We consider the construction of the regular tetrahedron, octahedron, and icosahedron, which are known as the spherical 2-design of 4 points, the spherical 3-design of 6 points, and the spherical 5-design of 12 points, respectively. We generate three spiral point sets with $N_{te} = 4$, $N_{oc} = 6$, and $N_{ic} = 12$, respectively. Then by Algorithm 2.1, we reach $x^* = X_{N_{te}}$, $X_{N_{oc}}$, and $X_{N_{ic}}$, respectively, with (1) $\sqrt{A_{N,t}(X_{N_{te}})} = 2.04\text{E-}16$ and $\|\nabla A_{N,t}(X_{N_{te}})\|_\infty = 7.38\text{E-}16$ for the tetrahedron spherical 2-design, where $\|\cdot\|_\infty$ denotes the l_∞-norm; (2) $\sqrt{A_{N,t}(X_{N_{oc}})} = 4.66\text{E-}13$ and $\|\nabla A_{N,t}(X_{N_{oc}})\|_\infty = 2.37\text{E-}12$ for the octahedron spherical 3-design; and (3) $\sqrt{A_{N,t}(X_{N_{ic}})} = 2.83\text{E-}12$ and $\|\nabla A_{N,t}(X_{N_{ic}})\|_\infty = 2.86\text{E-}13$ for the icosahedron spherical 5-design. The initial point sets and final numerical spherical t-designs are shown in Figure 1.

3.2. Spherical t-designs from four initial point sets. We list in Table 1 the information, including degree t, number of points N, total number of iterations K_{TR}, $\sqrt{A_{N,t}(X_N)}$, $\|\nabla A_{N,t}(X_N)\|_\infty$, and time of running the Algorithm 2.1 with the four initial points sets, that is, the SP points, the UD points, the IV points, and the HL points. The final point sets are named as SPD, SUD, SID, and SHD, respectively. From Table 1, one can see that we can reach significantly small values of $\sqrt{A_{N,t}(X_N)}$ up to the order of 1E-12 as well as near machine precision of $\|\nabla A_{N,t}\|_\infty$ up to the order of 1E-16. The number of total iterations K_{TR} and the computational time increases when t increases.



(a) Initial SP: $N_{te} = 4$ (c) Initial SP: $N_{oc} = 6$ (e) Initial SP: $N_{ic} = 12$

(b) Final: Tetrahedron (d) Final: Octahedron (f) Final: Icosahedron

Figure 1. *Numerical simulations of spherical t-design for Platonic solids on $\mathbb{S}^2$ by using Algorithm 2.1. Top view (left) and side view (right) for each initial SP point set and its resulted final point sets.* (a) *and* (b)*: Tetrahedron.* (c) *and* (d)*: Octahedron.* (e) *and* (f)*: Icosahedron.*

Table 1

Computing spherical t-designs by TR method from different initial point sets. We provide for each initial point sets (SP, UD, IV, HL) and each t, N, the number K_{TR} of iterations to reach the final point sets (SPD, SUD, SID, SHD) with their $\sqrt{A_{N,t}(X_N)}$, $\|\nabla A_{N,t}(X_N)\|_\infty$, $\boldsymbol{\epsilon}_{t,N}$ and the running time, respectively. "s,m,h,d,mo" stand for "second, minute, hour, day, month", respectively.

X_N	t	N	K_{TR}	$\sqrt{A_{N,t}(X_N)}$	$\|\nabla A_{N,t}(X_N)\|_\infty$	$\boldsymbol{\epsilon}_{t,N}$	Time
	16	289	264	2.15E-12	7.04E-16	6.43E-12	10.51 s
	32	1089	567	1.51E-12	7.93E-16	3.33E-12	24.61 s
	64	4225	1087	1.13E-12	1.27E-15	2.52E-12	2.01 m
	128	16641	1929	1.55E-12	1.07E-15	2.91E-12	11.16 m
	256	66049	3234	1.13E-12	1.39E-15	1.26E-11	32.50 m
	512	263169	6049	1.18E-12	8.64E-15	5.55E-12	4.59 h
	1024	1050625	9951	1.28E-12	3.80E-15	3.66E-12	1.02 d
SPD	2048	4198401	20592	2.53E-12	3.44E-15	3.95E-12	13.80 d
	25	676	422	1.73E-12	6.84E-15	3.76E-12	15.38 s
	50	2601	764	1.58E-12	9.39E-15	2.65E-12	46.52 s
	100	10201	1699	1.00E-12	8.51E-16	3.09E-12	3.08 m
	200	40401	2922	1.16E-12	2.30E-15	6.02E-12	26.85 m
	400	160801	4980	1.09E-12	4.22E-15	7.66E-12	2.29 h
	800	641601	8489	1.53E-12	4.18E-14	3.56E-11	21.74 h
	1600	2563601	18274	1.70E-10	9.26E-14	2.52E-10	6.95 d
	3200	10246401	22371	1.07E-09	2.22E-12	3.57E-08	2.07 mo
	25	676	665	1.81E-12	8.49E-16	7.08E-12	41.39 s
	50	2601	1660	1.44E-12	1.74E-14	4.23E-12	1.72 m
	100	10201	3986	1.40E-12	2.15E-14	3.79E-12	8.90 m
SUD	200	40401	12494	1.73E-12	3.71E-14	5.81E-12	2.01 h
	400	160801	24600	6.21E-12	7.32E-14	1.01E-11	12.26 h
	800	641601	86972	2.04E-11	4.85E-13	4.65E-11	6.11 d
	1000	1002001	118693	7.35E-12	4.07E-14	1.47E-11	11.54 d
	11	162	71	1.17E-12	2.93E-15	2.13E-11	27.38 s
	24	642	300	2.17E-12	5.84E-15	9.38E-12	53.56 s
	49	2562	1001	1.58E-12	9.68E-15	5.08E-12	1.72 m
	100	10242	1929	1.61E-12	1.23E-15	4.17E-12	5.24 m
SID	201	40962	3796	1.58E-12	3.65E-15	7.43E-12	32.61 m
	403	163842	8344	1.57E-12	1.25E-15	7.98E-12	3.72 h
	808	655362	22424	2.85E-12	2.44E-14	3.67E-11	2.04 d
	1618	2621442	49262	1.32E-10	5.19E-14	1.64E-10	18.37 d
	12	192	191	3.89E-12	1.71E-14	1.58E-11	8.79 s
	26	768	407	2.58E-12	6.58E-16	5.59E-12	22.34 s
	54	3072	725	1.68E-12	1.50E-15	3.45E-12	1.35 m
	109	12288	1221	1.41E-12	8.80E-16	3.22E-12	3.42 m
SHD	220	49152	2045	1.54E-12	9.08E-16	5.86E-12	17.59 m
	442	196608	3608	1.48E-12	3.48E-15	8.58E-12	1.99 h
	885	786432	5757	1.39E-12	5.53E-15	4.20E-11	10.41 h
	1772	3145728	9814	1.48E-12	1.04E-14	1.37E-10	4.61 d

Moreover, to test the accuracy of these final point sets, we consider the Gram matrix $\boldsymbol{G}_t = \frac{4\pi}{N}\boldsymbol{Y}_t^\star(X_{\tilde{N}})\boldsymbol{Y}_t(X_{\tilde{N}})$. As we know $\boldsymbol{G}_t = \mathbf{I} \in \mathbb{R}^{(t+1)^2\times(t+1)^2}$ theoretically, where $\mathbf{I}$ is identity matrix, for certain $X_{\tilde{N}}$ being a spherical design. Numerically, $\boldsymbol{G}_t \approx \mathbf{I}$. So our goal become to estimate the matrix 2-norm $\|\boldsymbol{G}_t - \mathbf{I}\| = \sup_{\hat{\boldsymbol{c}}\neq 0}\frac{\|\boldsymbol{G}_t\hat{\boldsymbol{c}}-\hat{\boldsymbol{c}}\|}{\|\hat{\boldsymbol{c}}\|}$ using the obtained spherical

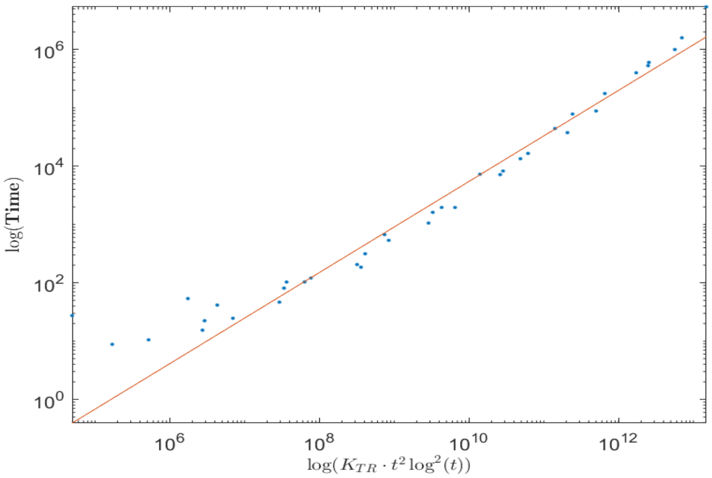


Figure 2. *The log-log plot of Time versus $K_{TR} \cdot t^2 \log^2(t)$. Blue dots are data points of $(K_{TR} \cdot t^2 \log^2(t), \mathrm{Time})$ from Table 1. The red line is the fitted linear line. (Figure in color online.)*

designs to see how close is $\boldsymbol{G}_t$ to $\mathbf{I}$. Note that it is numerically infeasible to explicitly compute $\boldsymbol{G}_t$ and compare it with $\mathbf{I}$ when t is large. Instead, we look at $\frac{\|\boldsymbol{G}_t\hat{\boldsymbol{c}}-\hat{\boldsymbol{c}}\|}{\|\hat{\boldsymbol{c}}\|} = \frac{\|\frac{4\pi}{N}\boldsymbol{Y}_t^\star(\boldsymbol{Y}_t\hat{\boldsymbol{c}})-\hat{\boldsymbol{c}}\|}{\|\hat{\boldsymbol{c}}\|}$, which can be computed fast using NFSFT. Now in Table 1, for each point set X_N being a spherical t-design, we randomly (in the sense of normal distribution) generate 100 vectors $\hat{\boldsymbol{c}}_i$, $i = 1, \dots, 100$, with respect to $\Pi_{\lfloor t/2 \rfloor}$, according to the uniform distribution. For each $\hat{\boldsymbol{c}}_i$, we compute $\epsilon_i = \frac{\|\boldsymbol{G}_{\lfloor t/2 \rfloor}\hat{\boldsymbol{c}}_i-\hat{\boldsymbol{c}}_i\|}{\|\hat{\boldsymbol{c}}_i\|}$ and define $\boldsymbol{\epsilon}_{t,N} := \max\{\epsilon_i\}_{i=1}^{100}$ to estimate $\|\boldsymbol{G}_{\lfloor t/2 \rfloor} - \mathbf{I}\|$. We present $\boldsymbol{\epsilon}_{t,N}$ in Table 1. The results show that $\boldsymbol{G}_t$ is indeed very close to the identity matrix (up to 1E-12) using the obtained spherical design point sets. For a more theoretical study on such a Gram matrix, we refer to [36] for the investigation of the general quadrature rules on the sphere using the summability operator.

For the computational time complexity of Algorithm 2.1, from subsection 2.3, we know it is of order $\mathcal{O}(K_{TR} \cdot (t^2 \log^2(t) + N \log^2(\frac{1}{\epsilon})))$. Since $N \approx (t+1)^2$ in our setting, it is essentially of order $\mathcal{O}(K_{TR} \cdot t^2 \log^2(t))$. To confirm this, from each row of Table 1, we have degree t, total number K_{TR} of iterations, and the Time (in seconds) to generate data points of the form $(K_{TR} \cdot t^2 \log^2(t), \mathrm{Time})$. Then, we use the log-log plot to show all data points and use linear fitting to fit the data points. The result is plotted in Figure 2, where we can see the log-log plot of the data points is close to the fitted straight line. This confirms that the computational time complexity of Algorithm 2.1 does follow the order of $\mathcal{O}(K_{TR} \cdot t^2 \log^2(t))$.

3.3. Spherical t-designs for function approximation. Quadrature formulas with preassigned weights were studied and used in function approximation, e.g., [21, 22, 43, 44, 45]. With the obtained spherical t-design point sets, we can use them to approximate the function as their discrete samples. We demonstrate below how this can be done for a special class of functions, the Wendland functions, defined on the sphere.

A spherical signal is typically sampled from a function $f : \mathbb{S}^2 \to \mathbb{C}$ on certain point set $X_N = \{\boldsymbol{x}_i \,|\, i \in [N]\}$, that is, one only has the sample points $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i)) \,|\, i \in [N]\}$. Note that X_N is not necessarily a spherical t-design point set. We would like to see how well it can be approximated by a polynomial space Π_T. This is equivalent to finding an $f_T \in \Pi_T$

that solves the minimization problem: $f_T = \arg\min_{p\in\Pi_T} \|f|_{X_N} - p|_{X_N}\|$, where $f|_{X_N} := \boldsymbol{f} := [f(\boldsymbol{x}_1),\dots,f(\boldsymbol{x}_N)]^\top$. Then we have $f = f_T + g$ with $g = f - f_T$ being its residual. Note that $f_T = \sum_{(\ell,m)\in\mathcal{I}_T} \hat{c}_\ell^m Y_\ell^m$ for some $\hat{\boldsymbol{c}} := (\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_T}$. Hence, $f_T|_{X_N} =: \boldsymbol{f}_T = \boldsymbol{Y}_T(X_N)\hat{\boldsymbol{c}}$. The minimization problem is equivalent to

$$\min_{\hat{\boldsymbol{c}}} \|\boldsymbol{f} - \boldsymbol{Y}_T\hat{\boldsymbol{c}}\|, \tag{3.1}$$

which can be solved by the least square method. In fact, to find $\hat{\boldsymbol{c}}$ such that $\boldsymbol{Y}_T\hat{\boldsymbol{c}} = \boldsymbol{f}$, one usually uses a diagonal matrix $\boldsymbol{W} = \operatorname{diag}(w_1,\dots,w_N)$ from some weight $\mathbf{w} := w|_{X_N} := \{w_1,\dots,w_N\}$ for the purpose of preconditioning. Define $b := \boldsymbol{Y}_T^\star\boldsymbol{W}\boldsymbol{f}$ and matrix $A = \boldsymbol{Y}_T^\star\boldsymbol{W}\boldsymbol{Y}_T$. Then, $\boldsymbol{Y}_T\hat{\boldsymbol{c}} = \boldsymbol{f}$ can be solved by the normal equation: $A\hat{\boldsymbol{c}} = b$, which can be done by the conjugate gradient (CG) method. See Algorithm 3.1. We remark that we do not need to form $A = \boldsymbol{Y}_T^\star\boldsymbol{W}\boldsymbol{Y}_T$ explicitly but simply the matrix-vector realization $A\hat{\boldsymbol{c}}$, which can be done fast through $\boldsymbol{Y}_T^\star(\boldsymbol{W}(\boldsymbol{Y}_T\hat{\boldsymbol{c}}))$ using the fast spherical harmonic transforms that we discussed in subsection 2.3.

In what follows, we set the maximum iterations $K_{\max} = 1000$ and termination tolerance $\varepsilon = 2.2204\text{E-}16$ in Algorithm 3.1. We use the relative projection l_2-error (with Euclidean norm), i.e., $err(f, f_T) := \frac{\|\boldsymbol{f}-\boldsymbol{f}_T\|}{\|\boldsymbol{f}\|}$, to measure how good the approximation is under different kinds of point sets. We demonstrate our results with f to be the combinations of normalized Wendland functions, which are a family of compactly supported radial basis functions (RBF). Let $(\xi)_+ := \max\{\xi, 0\}$ for $\xi \in \mathbb{R}$. The original Wendland functions are

$$\tilde{\phi}_k(\xi) := \begin{cases} (1-\xi)_+^2, & k=0,\\ (1-\xi)_+^4(4\xi+1), & k=1,\\ (1-\xi)_+^6(35\xi^2+18\xi+3)/3, & k=2,\\ (1-\xi)_+^8(32\xi^3+25\xi^2+8\xi+1), & k=3,\\ (1-\xi)_+^{10}(429\xi^4+450\xi^3+210\xi^2+50\xi+5)/5, & k=4. \end{cases}$$

Algorithm 3.1 Projection by conjugate gradient algorithm

Input: T: polynomial degree; X_N: spherical point set; $\mathbf{w}$: weights of X_N; $K_{\max}$: maximum iterations; ε: termination tolerance;
Initialize $x = 0$, $k = 0$, $r_0 = b = \boldsymbol{Y}_T^\star\boldsymbol{W}\boldsymbol{f}$, $A = \boldsymbol{Y}_T^\star\boldsymbol{W}\boldsymbol{Y}_T$ with $\boldsymbol{W} = \operatorname{diag}(\mathbf{w})$.

```
while ||r_{k+1}|| > ε and k ≤ K_max do
    if k = 0 then
        p_1 = r_0
    else
        p_{k+1} = r_k + (||r_k||^2 / ||r_{k-1}||^2) p_k
    end if
    Compute α = ||r_k||^2 / (p_{k+1}^T A p_{k+1}). Set x_{k+1} = x_k + α p_{k+1} and r_{k+1} = r_k − α A p_{k+1}.
    k = k + 1
end while
```

Output: $\hat{\boldsymbol{c}} =: x^* \in \mathbb{C}^{(T+1)^2}$.

Table 2

Relative l_2-errors $err(f_k, f_T)$ for Wendland functions $f_0, \ldots, f_4$ approximated by Π_T functions for $T = \frac{t}{2}$ and $\mathbf{w} \equiv \frac{4\pi}{N}$ in Algorithm 3.1 with different point sets.

t	N	Q_N	f_0	f_1	f_2	f_3	f_4
200	40401	SP	5.64E-04	3.19E-06	5.25E-08	3.39E-09	3.21E-09
200	40401	SPD	5.78E-04	3.20E-06	5.25E-08	1.69E-09	8.92E-11
200	40401	UD	6.09E-04	3.07E-06	8.50E-08	8.71E-08	8.53E-08
200	40401	SUD	6.99E-04	3.51E-06	5.63E-08	1.79E-09	1.07E-10
201	40962	IV	8.12E-04	3.32E-06	5.28E-08	4.57E-09	6.10E-08
201	40962	SID	6.15E-04	3.11E-06	5.08E-08	1.64E-09	8.74E-11
220	49152	HL	5.98E-04	2.28E-06	3.18E-08	8.68E-09	8.28E-09
220	49152	SHD	5.98E-04	2.28E-06	3.04E-08	8.11E-10	3.59E-11

The normalized (equal area) Wendland functions are $\phi_k(\xi) := \tilde{\phi}_k\left(\frac{\xi}{\Delta_k}\right)$ with $\Delta_k := \frac{(3k+3)\Gamma(k+\frac{1}{2})}{2\,\Gamma(k+1)}$ for $k \geq 0$. The Wendland functions $\phi_k(\xi)$ pointwise converge to Guassian when $k \to \infty$ [12]. Thus the main change as k increases is the smoothness of f. Let $\boldsymbol{z}_1 := (1,0,0), \boldsymbol{z}_2 := (-1,0,0), \boldsymbol{z}_3 := (0,1,0), \boldsymbol{z}_4 := (0,-1,0), \boldsymbol{z}_5 := (0,0,1), \boldsymbol{z}_6 := (0,0,-1)$ be regular octahedron vertices and define (see [23])

$$f_k(\boldsymbol{x}) := \sum_{i=1}^{6} \phi_k(\|\boldsymbol{z}_i - \boldsymbol{x}\|), \quad k \geq 0. \tag{3.2}$$

The function f_k is in $W^{k+\frac{3}{2}}(\mathbb{S}^2)$, where $W^\tau(\mathbb{S}^2) := \{f \in L_2(\mathbb{S}^2) : \sum_{\ell=0}^{\infty} \sum_{|m|\leq \ell} (1+\ell)^{2\tau} |\hat{f}_\ell^m|^2\} < \infty\}$ is the Sobolev space with smooth parameter $\tau > 1$. The function f_k has limited smoothness at the centers $\boldsymbol{z}_i$ and at the boundary of each cap with center $\boldsymbol{z}_i$. These features make f_k relatively difficult to be approximated in these regions, especially for small k.

For all initial point sets SP, UD, IV, HL, and their final point sets SPD, SUD, SID, SHD of spherical t-designs with $t = t(N)$ being determined in section 3, we set the input polynomial degree $T = \frac{t}{2}$ and (equal) weight $\mathbf{w} \equiv \frac{4\pi}{N}$ in Algorithm 3.1. Then for degree $t \approx 200$ and $N \approx (t+1)^2$, we show the projection errors $err(f_k, f_T)$ in Table 2 of the five RBF functions $f_0, \ldots, f_4$ defined in (3.2). We can see that the order of the errors decreases significantly from -4 up to -11 with respect to k in f_k for each of the final spherical t-design point sets, while the order of the errors decreases from -4 up to -9 for the initial point sets. This experiment demonstrates the superiority of spherical t-designs over normal structure point sets in terms of function approximation.

4. Spherical framelets from spherical t-designs.

In this section, we detail the construction and characterizations of the (semidiscrete) spherical framelet systems based on the spherical t-designs. A truncated system is then introduced and the fast spherical framelet transforms in terms of the filter banks and the fast spherical harmonic transforms are then developed.

4.1. Construction and characterizations.

Following the setting of the paper by Wang and Zhuang [61] on framelets defined on manifolds, we first define the (semidiscrete) framelet system on the sphere. Let functions $\Psi := \{\alpha; \beta_1, \ldots, \beta_n\} \subset L^1(\mathbb{R})$ be associated with a filter bank $\eta := \{a; b_1, \ldots, b_n\} \subset l_1(\mathbb{Z}) := \{h = \{h_k\}_{k\in\mathbb{Z}} \subset \mathbb{C} | \sum_{k\in\mathbb{Z}} |h_k| < \infty\}$ with the following relations:

$$\hat{\alpha}(2\xi) = \hat{a}(\xi)\hat{\alpha}(\xi), \quad \hat{\beta}_s(2\xi) = \hat{b}_s(\xi)\hat{\alpha}(\xi), \quad \xi \in \mathbb{R},\, s \in [n], \tag{4.1}$$

where for a function $f \in L^1(\mathbb{R})$, its Fourier transform $\hat{f}$ is defined by $\hat{f}(\xi) := \int_{\mathbb{R}} f(x)e^{-2\pi i x\xi}dx$, and for a filter (mask) $h = \{h_k\}_{k\in\mathbb{Z}} \subset \mathbb{C}$, its Fourier series $\hat{h}$ is given by $\hat{h}(\xi) := \sum_{k\in\mathbb{Z}} h_k e^{-2\pi i k\xi}$ for $\xi \in \mathbb{R}$. The first equation of (4.1) is said to be the refinement equation with α being the refinable function associated with the refinement mask a (low-pass filter). The functions β_s are framelet generators associated with framelet masks b_s (high-pass filter) for $s \in [n]$, which can be derived by extension principles [18, 50].

A *quadrature (cubature) rule* $Q_{N_j} = (X_{N_j}, \mathbf{w}_j)$ on $\mathbb{S}^2$ at scale j is a collection $Q_{N_j} := \{(\boldsymbol{x}_{j,k}, w_{j,k}) \,|\, k \in [N_j]\} \subset \mathbb{S}^2 \times \mathbb{R}$ of point set $X_{N_j} := \{\boldsymbol{x}_{j,k} \,|\, k \in [N_j]\}$ and weight $\mathbf{w}_j := \{w_{j,k} \,|\, k \in [N_j]\}$, where N_j is the number of points at scale j. We said that the quadrature rule Q_{N_j} is *polynomial-exact* up to degree $t_j \in \mathbb{N}_0$ if $\sum_{k=1}^{N_j} w_{j,k} p(\boldsymbol{x}_{j,k}) = \int_{\mathbb{S}^2} p(\boldsymbol{x}) \mathrm{d}\mu_2(\boldsymbol{x})$ for all $p \in \Pi_{t_j}$. We call such a $Q_{N_j} =: Q_{N_j,t_j}$ to be a polynomial-exact quadrature rule of degree t_j. The spherical t-design $X_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ forms a polynomial-exact quadrature rule $Q_{N,t} := (X_N, \mathbf{w})$ of degree t with weight $\mathbf{w} \equiv \frac{4\pi}{N}$.

Now given a sequence $\mathcal{Q} := \{Q_{N_j,t_j}\}_{j\geq J}$ of spherical designs, we can define spherical framelets $\varphi_{j,k}$ and $\psi_{j,k}^{(s)}$ for $s \in [n]$ as follows:

$$\varphi_{j,k}(\boldsymbol{x}) := \sqrt{w_j} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_j}\right) \overline{Y_\ell^m(\boldsymbol{x}_{j,k})} Y_\ell^m(\boldsymbol{x}), \tag{4.2}$$

$$\psi_{j,k}^{(s)}(\boldsymbol{x}) := \sqrt{w_{j+1}} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \hat{\beta}_s\left(\frac{\lambda_{\ell,m}}{t_j}\right) \overline{Y_\ell^m(\boldsymbol{x}_{j+1,k})} Y_\ell^m(\boldsymbol{x}) \tag{4.3}$$

for $\boldsymbol{x} \in \mathbb{S}^2$, where $w_j = \frac{4\pi}{N_j}$ and we set $\lambda_{\ell,m} = \ell$ in this paper. The *(semidiscrete) spherical framelet system* $\mathcal{F}_J(\Psi, \mathcal{Q})$ from the spherical designs starting at a scale $J \in \mathbb{Z}$ is then defined to be

$$\mathcal{F}_J(\Psi, \mathcal{Q}) := \{\varphi_{J,k} : k \in [N_J]\} \cup \{\psi_{j,k}^{(s)} : k \in [N_{j+1}], s \in [n]\}_{j=J}^{\infty}. \tag{4.4}$$

By [61, Corollary 2.6], we immediately have the following characterization result for the system $\mathcal{F}_J(\Psi, \mathcal{Q})$ to be a tight frame for $L^2(\mathbb{S}^2)$.

Theorem 4.1. *Let $\alpha \in L^1(\mathbb{R})$ be a band-limited function such that* $\operatorname{supp} \hat{\alpha} \subseteq [0, \frac{1}{2}]$ *and $\Psi := \{\alpha; \beta_s, \ldots, \beta_n\} \subset L^1(\mathbb{R})$ be a set of functions associating with a filter bank $\eta := \{a; b_1, \ldots, b_n\} \subset l_1(\mathbb{Z})$ as in (4.1) . Let $j \in \mathbb{Z}$ and $Q_{N_j,t_j} = \{(\boldsymbol{x}_{j,k}, w_{j,k} \equiv w_j = \frac{4\pi}{N_j}) \,|\, k \in [N_j]\}$ be the quadrature rule determined by a spherical t_j-design $X_{N_j} = \{\boldsymbol{x}_{j,1}, \ldots, \boldsymbol{x}_{j,N_j}\} \subset \mathbb{S}^2$ satisfying $t_{j+1} = 2t_j$. Define $\mathcal{F}_J(\Psi, \mathcal{Q})$ as in (4.4). Let $J_0 \in \mathbb{Z}$ be fixed. Then the following are equivalent:*

(i) *The framelet system $\mathcal{F}_J(\Psi, \mathcal{Q})$ is a tight frame for $L^2(\mathbb{S}^2)$ for all $J \geq J_0$, that is, $f = \sum_{k=1}^{N_J} \langle f, \varphi_{J,k}\rangle \varphi_{J,k} + \sum_{j=J}^{\infty} \sum_{k=1}^{N_{j+1}} \sum_{s=1}^{n} \langle f, \psi_{j,k}^{(s)}\rangle \psi_{j,k}^{(s)}$ for all $f \in L^2(\mathbb{S}^2)$ and $J \geq J_0$,*

(ii) *The generators in Ψ satisfy*

$$\lim_{j\to\infty} \left|\hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right| = 1, \quad \ell \geq 0, |m| \leq \ell, \tag{4.5}$$

$$\left|\hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\right|^2 = \left|\hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right|^2 + \sum_{s=1}^{n} \left|\hat{\beta}_s\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right|^2, \quad \ell \geq 0, |m| \leq \ell, j \geq J_0. \tag{4.6}$$

(iii) *The refinable function* α *satisfies* (4.5) *and the filters in* η *satisfy*

$$\left|\hat{a}\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right|^2+\sum_{s=1}^{n}\left|\hat{b}_s\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right|^2=1 \quad \forall \ell,m\in\mathcal{I}_\alpha^j, \forall j\ge J_0+1, \tag{4.7}$$

where $\mathcal{I}_\alpha^j:=\{\ell\in\mathbb{N}_0,|m|\le\ell:\hat{\alpha}(\frac{\lambda_{\ell,m}}{t_j})\neq 0\}$.

Proof. We need only show that the product rule holds for the spherical harmonics Y_ℓ^m since all other conditions in [61, Corollary 2.6] hold. We next use induction to prove that the product of two spherical harmonics Y_ℓ^m and $Y_{\ell'}^{m'}$ is in $\Pi_{\ell+\ell'}$, that is, $Y_\ell^mY_{\ell'}^{m'}\in\Pi_{\ell+\ell'}$ and it can be written as the linear combination of $\{Y_{\tilde\ell}^{\tilde m}:\tilde\ell\le\ell+\ell',|\tilde m|\le\tilde\ell\}$. By the definition of Y_ℓ^m in (2.5), it suffices to show that the product of two associate Legendre polynomials $P_\ell^m(z)$ and $P_{\ell'}^{m'}(z)$ can be written as the linear combination of $\{P_{\tilde\ell}^{\tilde m}(z):\tilde\ell\le\ell+\ell',m=0,\dots,\ell\}$ for $z\in[-1,1]$. That is,

$$P_\ell^m(z)P_{\ell'}^{m'}(z)=\sum_{0\le\tilde\ell\le\ell+\ell'}\sum_{0\le m\le\tilde\ell}c_{\tilde\ell}^mP_{\tilde\ell}^m(z), \quad z\in[-1,1]. \tag{4.8}$$

We prove it by mathematical induction on $\ell+\ell'$. We omit z in $P_\ell^m(z), P_{\ell'}^{m'}(z)$, and $P_{\tilde\ell}^{\tilde m}(z)$ for convenience.

For $\ell+\ell'=0$, equation (4.8) trivially holds. Suppose (4.8) holds for $\ell+\ell'=k\in\mathbb{N}_0$. We next prove that (4.8) holds for $\ell+\ell'=k+1$. Without loss of generality, we can assume $\ell\ge 2$ (otherwise, we can prove it for $\ell=0,1$ directly). By the recurrence formula of the associated Legendre polynomial: $(\ell-m+1)P_{\ell+1}^m(z)=(2\ell+1)zP_\ell^m(z)-(\ell+m)P_{\ell-1}^m(z)$, we have

$$P_\ell^mP_{\ell'}^{m'}=\frac{2\ell-1}{\ell-m}zP_{\ell-1}^mP_{\ell'}^{m'}-\frac{\ell-1+m}{\ell-m}P_{\ell-2}^mP_{\ell'}^{m'}. \tag{4.9}$$

From the inductive hypothesis and $P_1^0(z)=z$ for $z\in[-1,1]$, (4.9) can be written as

$$\begin{aligned} P_\ell^mP_{\ell'}^{m'}&=z\sum_{\tilde\ell\le k}\sum_{\tilde m\le\tilde\ell}c_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}+\sum_{\tilde\ell\le k-1}\sum_{\tilde m\le\tilde\ell}d_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}\\ &=\sum_{\tilde\ell\le k-1}\sum_{\tilde m\le\tilde\ell}c_{\tilde\ell}^{\tilde m}P_1^0P_{\tilde\ell}^{\tilde m}+\sum_{\tilde m\le k}c_k^{\tilde m}P_1^0P_k^{\tilde m}+\sum_{\tilde\ell\le k-1}\sum_{\tilde m\le\tilde\ell}d_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}\\ &=\sum_{\tilde\ell\le k}\sum_{\tilde m\le\tilde\ell}e_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}+\sum_{\tilde\ell\le k+1}\sum_{\tilde m\le\tilde\ell}h_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}+\sum_{\tilde\ell\le k-1}\sum_{\tilde m\le\tilde\ell}d_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}\\ &=\sum_{\tilde\ell\le k+1}\sum_{\tilde m\le\tilde\ell}a_{\tilde\ell}^{\tilde m}P_{\tilde\ell}^{\tilde m}, \end{aligned} \tag{4.10}$$

where we use $P_1^0P_\ell^m=\frac{1}{2\ell+1}[(\ell-m+1)P_{\ell+1}^m+(\ell+m)P_{\ell-1}^m]$ from the recurrence relation, and $a_{\tilde\ell}^{\tilde m}, c_{\tilde\ell}^{\tilde m}, d_{\tilde\ell}^{\tilde m}, e_{\tilde\ell}^{\tilde m}, h_{\tilde\ell}^{\tilde m}$ are coefficients in different components. That is, (4.8) holds for $\ell+\ell'=k+1$.

Therefore, by mathematical induction, for every $\ell,\ell'\in\mathbb{N}_0$, (4.8) holds for $z\in[-1,1]$. Hence, we complete the proof. ∎

Remark 4.2. By the contract rule and the Wigner $3j$-symbols [59], the product rule holds but it is hard to tell what is the resulting degree of the product of two spherical harmonics.

Besides, it is not explicitly proved in [61] for the product rule of spherical harmonics. Hence, we provide an elementary proof here. By the product rule, given a spherical t-design $X_N = \{\boldsymbol{x}_1,\dots,\boldsymbol{x}_N\}$, we immediately have

$$\frac{4\pi}{N}\sum_{i=1}^{N} Y_\ell^m(\boldsymbol{x}_i)\overline{Y_{\ell'}^{m'}(\boldsymbol{x}_i)} = \langle Y_\ell^m, Y_{\ell'}^{m'}\rangle = \delta_{\ell\ell'}\delta_{mm'} \tag{4.11}$$

for all $\ell+\ell'\le t$, $|m|\le \ell, |m'|\le \ell'$. This implies $\frac{4\pi}{N}\boldsymbol{Y}^\star_{\lfloor t/2\rfloor}(X_N)\boldsymbol{Y}_{\lfloor t/2\rfloor}(X_N) = \mathbf{I}_{(\lfloor t/2\rfloor+1)^2}$ with $\mathbf{I}_k$ being the identity matrix of size k.

4.2. Truncated spherical framelet systems. In practice, the infinite system $\mathcal{F}_{J_0}(\Psi,\mathcal{Q})$ in Theorem 4.1 needs to be truncated at certain scale $J_1 \ge J_0$ and the filter bank $\eta = \{a; b_1,\dots,b_n\}$ plays the key role in the decomposition and reconstruction of a discrete signal on the sphere. We next discuss the truncated systems of spherical framelets for practical spherical signal processing.

We first suppose that the filter bank $\eta = \{a; b_1,\dots,b_n\}$ is designed beforehand and it satisfies the partition of unity condition:

$$|\hat{a}(\xi)|^2 + \sum_{s\in[n]} |\hat{b}_s(\xi)|^2 = 1, \quad \xi\in\mathbb{R}. \tag{4.12}$$

For a fixed fine scale $J\in\mathbb{Z}$, we set

$$\hat{\alpha}^{(J+1)}\left(\frac{\lambda_{\ell,m}}{t_{J+1}}\right) = \begin{cases} 1 & \text{for } \ell\le t_J, \\ 0 & \text{for } \ell > t_J, \end{cases} \tag{4.13}$$

and following (4.1), we recursively define $\hat{\alpha}^{(j)}, \hat{\beta}_s^{(j)}$ from $\hat{\alpha}^{(j+1)}$ by

$$\hat{\alpha}^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right) = \hat{\alpha}^{(j)}\left(2\frac{\lambda_{\ell,m}}{t_{j+1}}\right) = \hat{a}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\hat{\alpha}^{(j+1)}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right), \tag{4.14}$$

$$\hat{\beta}_s^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right) = \hat{\beta}_s^{(j)}\left(2\frac{\lambda_{\ell,m}}{t_{j+1}}\right) = \hat{b}_s\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\hat{\alpha}^{(j+1)}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right), \quad s\in[n], \tag{4.15}$$

for j decreasing from J to J_0. Then, we obtain

$$\Psi = \{\alpha^{(j)}, \beta_s^{(j)} \,|\, j = J_0,\dots,J; s\in[n]\}. \tag{4.16}$$

Let $\mathcal{Q} := \mathcal{Q}_{J_0}^{J+1} := \{Q_{N_j,t_j} : j = J_0,\dots,J+1\}$ be the set of polynomial-exact quadrature rules truncated from the original infinite sequence of spherical t_j-designs satisfying $t_{j+1} = 2t_j$.

With the above Ψ and $\mathcal{Q}$, we can define the *truncated (semidiscrete) spherical framelet system* $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$ from the spherical designs as

$$\mathcal{F}_{J_0}^J(\eta,\mathcal{Q}) := \{\varphi_{J_0,k} \,|\, k\in[N_{J_0}]\} \cup \{\psi_{j,k}^{(s)} \,|\, k\in[N_{j+1}], s\in[n]\}_{j=J_0}^J, \tag{4.17}$$

where the $\varphi_{j,k}$ and $\psi_{j,k}^{(s)}$ are modified as

$$\varphi_{j,k}(\boldsymbol{x}) := \sqrt{w_j} \sum_{(\ell,m)\in\mathcal{I}_{t_j}} \hat{\alpha}^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right) \overline{Y_\ell^m(\boldsymbol{x}_{j,k})} Y_\ell^m(\boldsymbol{x}), \tag{4.18}$$

$$\psi_{j,k}^{(s)}(\boldsymbol{x}) := \sqrt{w_{j+1}} \sum_{(\ell,m)\in\mathcal{I}_{t_{j+1}}} \hat{\beta}_s^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right) \overline{Y_\ell^m(\boldsymbol{x}_{j+1,k})} Y_\ell^m(\boldsymbol{x}). \tag{4.19}$$

Note that in the notation $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$, we emphasize on the role of the filter bank η. The system $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$ is completely determined by the filter bank η and the quadrature rules $\mathcal{Q}$. We have the following result regarding the tightness of $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$ and its relation to Π_{t_J}.

Theorem 4.3. *Let $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$ be the truncated spherical framelet system defined as in (4.17) and assume that the filter bank η satisfies the partition of unity condition (4.12) with $\operatorname{supp}\hat{a}\subseteq[0,\frac{1}{4}]$ and $\operatorname{supp}\hat{b}_s\subseteq[0,\frac{1}{2}]$ for $s\in[n]$. Define $\mathcal{V}_j:=\operatorname{span}\{\varphi_{j,k}\,|\,k\in[N_j]\}$ and $\mathcal{W}_j:=\operatorname{span}\{\psi_{j,k}^{(s)}\,|\,k\in[N_{j+1}], s\in[n]\}$. Then the following results hold:*

(i) $\Pi_{t_J}=\mathcal{V}_{J+1}$ *and thus* $f=\sum_{k=1}^{N_{J+1}}\langle f,\varphi_{J+1,k}\rangle\varphi_{J+1,k}$ *for any* $f\in\Pi_{t_J}$.

(ii) *The decomposition and reconstruction relation* $\mathcal{V}_{j+1}=\mathcal{V}_j+\mathcal{W}_j$ *holds for* $j=J_0,\dots,J$.

(iii) *The truncated spherical framelet system $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$ is a tight frame for Π_{t_J}. That is, for all $f\in\Pi_{t_J}$, we have $f=\sum_{k=1}^{N_0}v_{J_0,k}\varphi_{J_0,k}+\sum_{j=J_0}^{J}\sum_{k=1}^{N_j}\sum_{s=1}^{n}w_{j,k}^{(s)}\psi_{j,k}^{(s)}$, where $v_{j,k}:=\langle f,\varphi_{j,k}\rangle$ and $w_{j,k}^{(s)}:=\langle f,\psi_{j,k}^{(s)}\rangle$.*

Proof. By (4.13), we have $\mathcal{V}_{J+1}\subseteq\Pi_{t_J}$. One the other hand, for $f\in\Pi_{t_J}$, we have

$$f=\sum_{(\ell,m)\in\mathcal{I}_{t_{J+1}}}\hat{f}_\ell^m Y_\ell^m=\sum_{(\ell,m)\in\mathcal{I}_{t_{J+1}}}\hat{f}_\ell^m|\hat{\alpha}^{(J+1)}(\frac{\lambda_{\ell,m}}{t_{J+1}})|^2Y_\ell^m.$$

We next show that the last equation above implies $f=\sum_{k=1}^{N_{J+1}}v_{J+1,k}\varphi_{J+1,k}\in\mathcal{V}_{J+1}$. In fact, more generally, by the orthogonality of Y_ℓ^m for any $f\in L^2(\mathbb{S}^2)$, we have $v_{j,k}=\langle f,\varphi_{j,k}\rangle=\sum_{(\ell,m)\in\mathcal{I}_{t_j}}\hat{f}_\ell^m\bar{\hat{\alpha}}^{(j)}(\frac{\lambda_{\ell,m}}{t_j})\sqrt{w_j}Y_\ell^m(\boldsymbol{x}_{j,k})$. Together with that Q_{N_j,t_j} is a polynomial-exact quadrature rule of degree t_j and $\hat{\alpha}^{(j)}(\frac{\lambda_{\ell,m}}{t_j})\equiv 0$ for $\ell>t_{j-1}$ in view of (4.13) and $\operatorname{supp}\hat{a}\subseteq[0,\frac{1}{2}]$, we can deduce that

$$\begin{aligned}\sum_{k=1}^{N_j}v_{j,k}\varphi_{j,k}&=\sum_{(\ell,m)\in\mathcal{I}_{t_j}}\sum_{(\ell',m')\in\mathcal{I}_{t_j}}\hat{f}_\ell^m\bar{\hat{\alpha}}^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right)\hat{\alpha}^{(j)}\left(\frac{\lambda_{\ell',m'}}{t_j}\right)\mathcal{U}_{\ell,m}^{\ell',m'}(Q_{N_j,t_j})Y_{\ell'}^{m'}\\&=\sum_{(\ell,m)\in\mathcal{I}_{t_j}}\hat{f}_\ell^m|\hat{\alpha}^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right)|^2Y_\ell^m,\end{aligned}$$

where we use $\mathcal{U}_{\ell,m}^{\ell',m'}(Q_{N_j,t_j}):=\sum_{k=1}^{N_j}w_jY_\ell^m(\boldsymbol{x}_{j,k})\overline{Y_{\ell'}^{m'}(\boldsymbol{x}_{j,k})}=\delta_{\ell\ell'}\delta_{mm'}$. Item (i) is proved.

For item (ii), by definition, we obviously have $\mathcal{V}_j+\mathcal{W}_j\subseteq\mathcal{V}_{j+1}$. For the other direction, similarly to above, for any $f\in L^2(\mathbb{S}^2)$, we can deduce that $\sum_{k=1}^{N_{j+1}}w_{j,k}^{(s)}\psi_{j,k}^{(s)}=\sum_{(\ell,m)\in\mathcal{I}_{t_{j+1}}}\hat{f}_\ell^m|\hat{\beta}_s^{(j)}(\frac{\lambda_{\ell,m}}{t_j})|^2Y_\ell^m$. Then, by (4.12), (4.14), and (4.15), we have

$$\begin{aligned}\sum_{k=1}^{N_{j+1}} v_{j+1,k}\varphi_{j+1,k} &= \sum_{(\ell,m)\in\mathcal{I}_{t_{j+1}}} \hat{f}_\ell^m \left|\hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\right|^2 Y_\ell^m \\ &= \sum_{(\ell,m)\in\mathcal{I}_{t_{j+1}}} \hat{f}_\ell^m \left|\hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\right|^2 \left(\left|\hat{a}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\right|^2 + \sum_{s=1}^{n}\left|\hat{b}_s\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\right|^2\right) Y_\ell^m \\ &= \sum_{(\ell,m)\in\mathcal{I}_{t_j}} \hat{f}_\ell^m \left|\hat{\alpha}\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right|^2 Y_\ell^m + \sum_{s=1}^{n}\sum_{(\ell,m)\in\mathcal{I}_{t_{j+1}}} \hat{f}_\ell^m \left|\hat{\beta}_s\left(\frac{\lambda_{\ell,m}}{t_j}\right)\right|^2 Y_\ell^m \\ &= \sum_{k=1}^{N_j} v_{j,k}\varphi_{j,k} + \sum_{k=1}^{N_{j+1}}\sum_{s=1}^{n} w_{j,k}^{(s)}\psi_{j,k}^{(s)}.\end{aligned}$$

Therefore, we have $\mathcal{V}_{j+1}\subseteq\mathcal{V}_j+\mathcal{W}_j$ for all $j=0,\dots,J$. Item (ii) holds.

Item (iii) directly follows from items (i) and (ii). This completes the proof. ∎

4.3. Fast spherical framelet transforms. We next turn to the fast spherical framelet transforms (SFmTs) for the decomposition and reconstruction of a signal on the sphere $\mathbb{S}^2$ using the truncated system $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$.

For a vector $\hat{\boldsymbol{c}}=(\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_{t_{j+1}}}$, define the downsampling operator $\downarrow_j$ by $\hat{\boldsymbol{c}}\downarrow_j:=(\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_{t_j}}$. Similarly, for a vector $\hat{\boldsymbol{c}}=(\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_{t_j}}$, define the upsampling operator $\uparrow_{j+1}$ by $\hat{\boldsymbol{c}}\uparrow_{j+1}:=(\hat{c}_\ell^m)_{(\ell,m)\in\mathcal{I}_{t_{j+1}}}$ with $\hat{c}_\ell^m=0$ for $\ell>t_j$.

We have the following theorem regarding the decomposition of reconstruction using the truncated spherical framelet system $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$.

Theorem 4.4. *Given a truncated system $\mathcal{F}_{J_0}^J(\eta,\mathcal{Q})$ as in Theorem 4.3. Define*

$$\boldsymbol{v}_j:=(v_{j,k})_{k\in[N_j]}\in\mathbb{C}^{N_j},\qquad \boldsymbol{w}_j^{(s)}:=(w_{j,k})_{k\in[N_{j+1}]}\in\mathbb{C}^{N_{j+1}}, \tag{4.20}$$

$$\hat{\boldsymbol{a}}_j:=\left(\hat{a}\left(\frac{\lambda_\ell^m}{t_{j+1}}\right)\right)_{(\ell,m)\in\mathcal{I}_{t_{j+1}}},\qquad \hat{\boldsymbol{b}}_j^{(s)}:=\left(\hat{b}_s\left(\frac{\lambda_\ell^m}{t_{j+1}}\right)\right)_{(\ell,m)\in\mathcal{I}_{t_{j+1}}} \tag{4.21}$$

for $j=J_0,\dots,J$. Let $w_j:=\frac{4\pi}{N_j}$. Then, for $j=J_0,\dots,J$, we have the one-level framelet decomposition that obtains $\{\boldsymbol{v}_j,\boldsymbol{w}_j^{(s)}\,|\,s\in[n]\}$ from $\boldsymbol{v}_{j+1}$:

$$\boldsymbol{v}_j=\sqrt{w_j}\boldsymbol{Y}_{t_j}\left[\left[(\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1})\odot\bar{\hat{\boldsymbol{a}}}_j\right]\downarrow_j\right], \tag{4.22}$$

$$\boldsymbol{w}_j^{(s)}=\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}\left[(\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1})\odot\bar{\hat{\boldsymbol{b}}}_j^{(s)}\right],\quad s\in[n], \tag{4.23}$$

and the one-level framelet reconstruction of $\boldsymbol{v}_{j+1}$ from $\{\boldsymbol{v}_j,\boldsymbol{w}_j^{(s)}\,|\,s\in[n]\}$:

$$\boldsymbol{v}_{j+1}=\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}\left[[\sqrt{w_j}\boldsymbol{Y}_{t_j}^\star\boldsymbol{v}_j]\uparrow_{j+1}\odot\hat{\boldsymbol{a}}_j+\sum_{s=1}^{n}[(\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{w}_j^{(s)})\odot\hat{\boldsymbol{b}}_j^{(s)}]\right], \tag{4.24}$$

where the symbol $\odot$ denotes the Hadamard entrywise product operator.

Proof. Given $f \in \Pi_{t_J}$, by item (i) of Theorem 4.3, it is uniquely determined by its Fourier coefficient sequence $\hat{f}_\ell^m$, i.e., $f = \sum_{(\ell,m)\in\mathcal{I}_{t_J}} \hat{f}_\ell^m Y_\ell^m$, and we can represent it in $\mathcal{V}_{J+1}$ as $f = \sum_{k=1}^{N_{J+1}} v_{J+1,k}\varphi_{J+1,k}$, which is associated with the spherical t-design point set $X_{N_{J+1}}$. Define $\hat{\boldsymbol{f}}_j := (\hat{f}_\ell^m)_{(\ell,m)\in\mathcal{I}_{t_j}}$ and $\hat{\boldsymbol{\alpha}}_j := (\hat{\alpha}^{(j)}(\lambda_\ell^m/t_j))_{(\ell,m)\in\mathcal{I}_{t_j}}$ for $j = J_0,\dots,J+1$ with the convention that $\hat{f}_\ell^m = 0$ for $(\ell,m)\notin\mathcal{I}_{t_J}$.

By (4.14) and (4.15), we have

$$v_{j,k} = \sum_{(\ell,m)\in\mathcal{I}_{t_j}} \hat{f}_\ell^m \bar{\hat{\alpha}}^{(j)}\left(\frac{\lambda_{\ell,m}}{t_j}\right)\sqrt{w_j}Y_\ell^m(\boldsymbol{x}_{j,k})$$
$$= \sum_{(\ell,m)\in\mathcal{I}_{t_j}} \hat{f}_\ell^m \bar{\hat{a}}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\bar{\hat{\alpha}}^{(j)}\left(\frac{\lambda_{\ell,m}}{t_{j+1}}\right)\sqrt{w_j}Y_\ell^m(\boldsymbol{x}_{j,k}).$$

This implies that

$$\boldsymbol{v}_{j+1} = \sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}(\hat{\boldsymbol{f}}_{j+1}\odot\bar{\hat{\boldsymbol{\alpha}}}_{j+1}), \quad \boldsymbol{v}_j = \sqrt{w_j}\boldsymbol{Y}_{t_j}\left[[(\hat{\boldsymbol{f}}_{j+1}\odot\hat{\boldsymbol{\alpha}}_{j+1})\odot\bar{\hat{\boldsymbol{a}}}_j]\downarrow_j\right], \tag{4.25}$$

where we use $w_j = \frac{4\pi}{N_j}$. Note that, by $\hat{\alpha}^{(j)}(\frac{\lambda_{\ell,m}}{t_{j+1}}) \equiv 0$ for $\ell > t_j$ and the polynomial-exact quadrature rule $Q_{N_{j+1}}$ of degree t_{j+1}, we have

$$[\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star \boldsymbol{v}_{j+1}]|_{\mathcal{I}_{t_j}} = [w_{j+1}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{Y}_{t_{j+1}}(\hat{\boldsymbol{f}}_{j+1}\odot\bar{\hat{\boldsymbol{\alpha}}}_{j+1})]|_{\mathcal{I}_{t_j}} = (\hat{\boldsymbol{f}}_{j+1}\odot\bar{\hat{\boldsymbol{\alpha}}}_{j+1})|_{\mathcal{I}_{t_j}},$$

where $|_{\mathcal{I}_{t_j}}$ denotes the restriction on the index set $\mathcal{I}_{t_j}$. Consequently, replacing the above expression of $(\hat{\boldsymbol{f}}_{j+1}\odot\bar{\hat{\boldsymbol{\alpha}}}_{j+1})$ into $\boldsymbol{v}_j$ in (4.25), we have (4.22). Similarly, we have $\boldsymbol{w}_j^{(s)} = \sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}[(\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1})\odot\bar{\hat{\boldsymbol{b}}}_j^{(s)}]$. Hence, we obtain the one-level framelet decomposition.

For the reconstruction, by (4.22) and $\operatorname{supp}\hat{a}\subseteq[0,\frac{1}{4}]$, we have $[\sqrt{w_j}\boldsymbol{Y}_{t_j}^\star\boldsymbol{v}_j]\uparrow_{j+1}\odot\hat{\boldsymbol{a}}_j = ([\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1}]\odot\bar{\hat{\boldsymbol{a}}}_j)\odot\hat{\boldsymbol{a}}_j = [\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1}]\odot[\bar{\hat{\boldsymbol{a}}}_j\odot\hat{\boldsymbol{a}}_j]$. Similarly, $(\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{w}_j^{(s)})\odot\hat{\boldsymbol{b}}_j^{(s)} = [\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1}]\odot[\bar{\hat{\boldsymbol{b}}}_j^{(s)}\odot\hat{\boldsymbol{b}}_j^{(s)}]$. Consequently, by the partition of unity condition in (4.12) and the support constrains of $\hat{a},\hat{b}_s$ ($\operatorname{supp}\hat{a}\subset[0,\frac{1}{4}],\operatorname{supp}\hat{b}_s\subset[0,\frac{1}{2}]$), we have $[\sqrt{w_j}\boldsymbol{Y}_{t_j}^\star\boldsymbol{v}_j]\uparrow_{j+1}\odot\hat{\boldsymbol{a}}_j + \sum_{s=1}^n[(\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{w}_j^{(s)})\odot\hat{\boldsymbol{b}}_j^{(s)}] = [\sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}^\star\boldsymbol{v}_{j+1}]\odot([\bar{\hat{\boldsymbol{a}}}_j\odot\hat{\boldsymbol{a}}_j] + \sum_{s=1}^n[\bar{\hat{\boldsymbol{b}}}_j^{(s)}\odot\hat{\boldsymbol{b}}_j^{(s)}]) = \hat{\boldsymbol{f}}_{j+1}\odot\hat{\boldsymbol{\alpha}}_{j+1}$. Now (4.24) follows from (4.25), which completes the proof. ∎

Based on Theorem 4.4, we have the pseudo-code of multilevel spherical framelet transforms as in Algorithms 4.1 and 4.2. Since each step in the decomposition or reconstruction involves only the fast spherical harmonic transforms or the down- and up-sampling operators, the computational time complexity of the multilevel spherical framelet transforms is of order $\mathcal{O}(t^2\log^2(t) + N\log^2(\frac{1}{\epsilon}))$.

The procedure of spherical framelet decomposition and reconstruction is illustrated in Figure 3.

5. Spherical framelets for spherical signal denoising.

In this section, we provide numerical experiments for illustrating the efficiency and effectiveness of spherical signal denoising using the spherical framelet systems developed in section 4.

Algorithm 4.1 Multilevel spherical framelet transforms: Decomposition

Input: $\{Q_{N_j,t_j} = (X_{N_j}, w_j = \frac{4\pi}{N_j})\}_{j=J_0}^{J+1}$: polynomial-exact quadrature rules; $\boldsymbol{f}_{J+1} = f|_{X_{N_{J+1}}}$: samples of $f \in \Pi_{t_J}$ on the spherical point set $X_{N_{J+1}}$; η: filter bank.

Initialize $\hat{\boldsymbol{f}}_{\boldsymbol{J+1}} = w_{j+1}\boldsymbol{Y}^{\star}_{t_{J+1}}\boldsymbol{f}_{J+1}$.

for j from J to J_0 **do**
  for s from 1 to n **do**
    $\boldsymbol{w}_j^{(s)} = \sqrt{w_{j+1}}\boldsymbol{Y}_{t_{j+1}}[\hat{\boldsymbol{f}}_{j+1} \odot \bar{\hat{\boldsymbol{b}}}_j^{(s)}]$.
  end for
  $\hat{\boldsymbol{f}}_j = [\hat{\boldsymbol{f}}_{j+1} \odot \bar{\hat{\boldsymbol{a}}}_j]{\downarrow_j}$.
end for
$\boldsymbol{v}_{J_0} = \sqrt{w_{J_0}}\boldsymbol{Y}_{t_{J_0}}\hat{\boldsymbol{f}}_{J_0}$.

Output: $\{\boldsymbol{v}_{J_0}, \boldsymbol{w}_j^{(s)} \,|\, j = J_0, \dots J; s \in [n]\}$.

Algorithm 4.2 Multilevel spherical framelet transforms: Reconstruction

Input: $\{Q_{N_j,t_j} = (X_{N_j}, w_j = \frac{4\pi}{N_j})\}_{j=J_0}^{J+1}$: polynomial-exact quadrature rules; $\{\boldsymbol{v}_{J_0}, \boldsymbol{w}_j^{(s)} \,|\, j = J_0, \dots J; s \in [n]\}$: coefficient sequences; η: filter bank.

Initialize $\hat{\boldsymbol{f}}_{\boldsymbol{J_0}} = \sqrt{w_{J_0}}\boldsymbol{Y}_{t_{J_0}}\boldsymbol{v}_{J_0}$.

for j from J_0 to J **do**
  $\hat{\boldsymbol{f}}_{j+1} = \hat{\boldsymbol{f}}_j{\uparrow_{j+1}} \odot \bar{\hat{\boldsymbol{a}}}_j$
  for s from 1 to n **do**
    $\hat{\boldsymbol{f}}_{j+1} = \hat{\boldsymbol{f}}_{j+1} + [\sqrt{w_{j+1}}\boldsymbol{Y}^{\star}_{t_{j+1}}\boldsymbol{w}_j^{(s)}] \odot \hat{\boldsymbol{b}}_j^{(s)}$.
  end for
end for
$\boldsymbol{f}_{\boldsymbol{J+1}} = w_{j+1}\boldsymbol{Y}_{t_{J+1}}\hat{\boldsymbol{f}}_{J+1}$.

Output: $\boldsymbol{f}_{J+1}$: samples of $f \in \Pi_{t_J}$ on the spherical point set $X_{N_{J+1}}$;

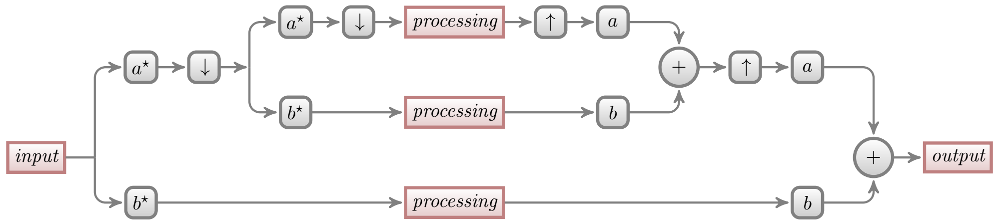
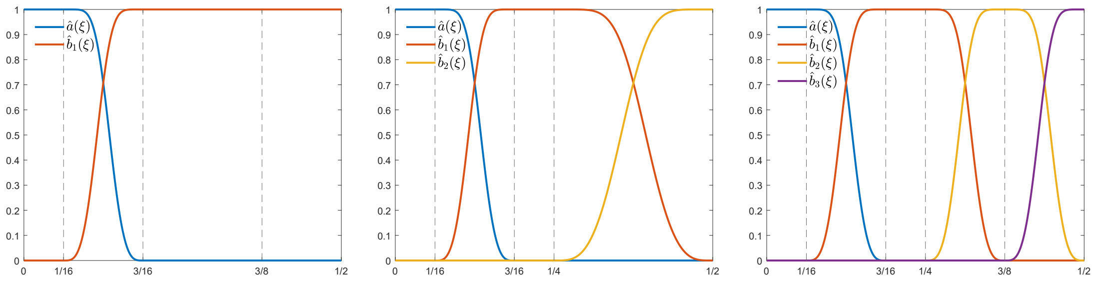


Figure 3. *Two-level framelet filter bank decomposition and reconstruction based on the filter bank* $\eta = \{a; b_1, \dots, b_n\}$. *Here the node with respect to* b *(or* $b^\star$*) runs from* b_1 *to* b_n *while the node with respect to* $\oplus$ *sums all* $b_s, s \in [n]$.

5.1. Three framelet systems. We first discuss the ingredients for the system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$. For $\mathcal{Q} = \{Q_{N_j,t_j}\}_{j=J_0}^{J+1} = \{(X_{N_j}, w_j = \frac{4\pi}{N_j})\}_{j=J_0}^{J+1}$, it is the set of spherical designs obtained in section 3 satisfying $t_{j+1} = 2t_j$. For η, we construct three different filter banks η_1, η_2, and η_3 with 1, 2, and 3 high-pass filters, respectively.



(a) $\eta_1 = \{\hat{a}; \hat{b}_1\}$ (b) $\eta_2 = \{\hat{a}; \hat{b}_1, \hat{b}_2\}$ (c) $\eta_3 = \{\hat{a}; \hat{b}_1, \hat{b}_2, \hat{b}_3\}$

Figure 4. *Filter banks* η_1, η_2, η_3 *on* $[0, \frac{1}{2}]$.

(1) The filter bank $\eta_1 = \{a; b_1\}$ is determined by $\hat{a} := \chi_{[-\frac{3}{16}, \frac{1}{8}]; \frac{1}{16}, \frac{1}{16}}$ and $\hat{b}_1 := \chi_{[\frac{1}{8}, \frac{9}{16}]; \frac{1}{16}, \frac{1}{16}}$. Note that $\operatorname{supp} \hat{a} \subset [0, \frac{1}{4}]$.
(2) The filter bank $\eta_2 = \{a; b_1, b_2\}$ is determined by the same $\hat{a}$ as in item (1), and $\hat{b}_1 := \chi_{[\frac{1}{8}, \frac{3}{8}]; \frac{1}{16}, \frac{1}{8}}$ and $\hat{b}_2 := \chi_{[\frac{3}{8}, 1]; \frac{1}{8}, \frac{1}{8}}$.
(3) The filter bank $\eta_3 = \{a; b_1, b_2, b_3\}$ is determined by the same $\hat{a}$ as in item (1), and $\hat{b}_1 := \chi_{[\frac{1}{8}, \frac{5}{16}]; \frac{1}{16}, \frac{1}{16}}$, $\hat{b}_2 := \chi_{[\frac{5}{16}, \frac{7}{16}]; \frac{1}{16}, \frac{1}{16}}$, and $\hat{b}_3 := \chi_{[\frac{7}{16}, \frac{9}{16}]; \frac{1}{16}, \frac{1}{16}}$.

Here the bump function $\chi_{[c_L, c_R]; \epsilon_L, \epsilon_R}$ is the continuous function supported on $[c_L - \epsilon_L, c_R + \epsilon_R]$ as defined in [28, 61] and is given by

$$\chi_{[c_L, c_R]; \epsilon_L, \epsilon_R}(\xi) := \begin{cases} 0, & \xi \le c_L - \epsilon_L \text{ or } \xi \ge c_R + \epsilon_R, \\ \sin(\frac{\pi}{2} \nu(\frac{\xi - c_L + \epsilon_L}{2\epsilon_L})), & c_L - \epsilon_L < \xi < c_L + \epsilon_L, \\ 1, & c_L + \epsilon_L \le \xi \le c_R - \epsilon_R, \\ \cos(\frac{\pi}{2} \nu(\frac{\xi - c_R + \epsilon_R}{2\epsilon_R})), & c_R - \epsilon_R < \xi < c_R + \epsilon_R, \end{cases}$$

where c_L, c_R are control points, ϵ_L, ϵ_R are shape parameters, $\nu(t)$ is the elementary function [17] such that $\nu(t) = t^4(35 - 84t + 70t^2 - 20t^3)$ for $0 < t < 1$, $\nu(t) = 1$ for $t \ge 1$, and $\nu(t) = 0$ for $t < 0$. Note that $\nu(t)$ satisfies $\nu(t) + \nu(1-t) = 1$. Each filter bank η_k corresponds to a truncated tight framelet system $\mathcal{F}_{J_0}^J(\eta_k, \mathcal{Q})$ on the sphere. We show in Figure 4 the filter banks η_k for $k = 1, 2, 3$. It can be verified that $|\hat{a}(\xi)|^2 + \sum_{s=1}^n |\hat{b}_s(\xi)|^2 = 1$ for $\xi \in [0, \frac{1}{2}]$, which implies (4.12).

5.2. Denoising procedure. We next discuss the denoising procedure for a given noisy signal f_σ using the spherical framelet systems. Given a noisy function $f_\sigma = f_o + G_\sigma$ on $X_{N_{J+1}}$, where f_o is an unknown ground truth and G_σ is the Gaussian white noisy, we project it onto Π_{t_J} (using Algorithm 3.1) to obtain $f_\sigma = f + g$ such that $f \in \Pi_{t_J}$ is the projection part and $g = f_\sigma - f$ is the residual part. Note that all f_σ, f, g are sampled on $X_{N_{J+1}}$. We then use the spherical tight framelet system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$ to decompose f (more precisely, $\boldsymbol{f}_{J+1} = f|_{X_{N_{J+1}}}$; see Algorithm 4.1) into the framelet coefficient sequences $\{\boldsymbol{v}_{J_0}\} \cup \{\boldsymbol{w}_j^{(s)} \,|\, j = J_0, \dots, J; s \in [n]\}$. We apply the thresholding techniques for denoising the framelet coefficient sequences $\boldsymbol{w}_j^{(s)}$ of f and the residual g. After that, we apply the framelet reconstruction (Algorithm 4.2) to the denoised framelet coefficient sequences and obtain the denoised reconstruction signal f_{thr} (cf. Figure 3). Finally, together with the denoised residual g_{thr}, we can obtain a denoised signal $f_{\sigma,thr} = f_{thr} + g_{thr}$. To quantify the performance of the framelet denoising, we use the signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) to measure the quality of denoising.

We next detail the denoising procedure of f and g for obtaining f_{thr} and g_{thr}.

Given the framelet coefficient sequence $\boldsymbol{w}_j^{(s)} = (w_j^{(s)})_{k\in[N_{j+1}]}$, note that $w_{j,k}^{(s)}$ is associated with the point $\boldsymbol{x}_{j+1,k}$. We first normalize it according to the norm $\|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}$ by $\tilde{w}_{j,k}^{(s)} = w_j^{(s)}/\|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}$. In practice, such a norm $\|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}$ can be computed by setting all coefficient sequences in $\{\boldsymbol{v}_{J_0}\} \cup \{\boldsymbol{w}_j^{(s)} \,|\, j = J_0,\dots,J; s \in [n]\}$ to be 0 except $w_j^{(s)} = 1$, applying the framelet reconstruction Algorithm 4.2 obtaining a reconstruction signal with respect to $\psi_{j,k}^{(s)}$, and calculating its l_2-norm to obtain $\|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}$. We then perform the local-soft (LS) thresholding method which updates $\tilde{w}_{j,k}^{(s)}$ to be

$$\check{w}_{j,k}^{(s)} = \begin{cases} \tilde{w}_{j,k}^{(s)} - \operatorname{sgn}(\tilde{w}_{j,k}^{(s)})\tau_{j,k,r}^{(s)}, & |\tilde{w}_{j,k}^{(s)}| \ge \tau_{j,k,r}^{(s)}, \\ 0, & |\tilde{w}_{j,k}^{(s)}| < \tau_{j,k,r}^{(s)}, \end{cases} \tag{5.1}$$

where $\tau_{j,k,r}^{(s)}$ is a thresholding value determined by

$$\tau_{j,k,r}^{(s)} = \frac{c\cdot\sigma^2}{\sqrt{(\bar{w}_{j,k,r}^{(s)} - \sigma^2)_+}} \tag{5.2}$$

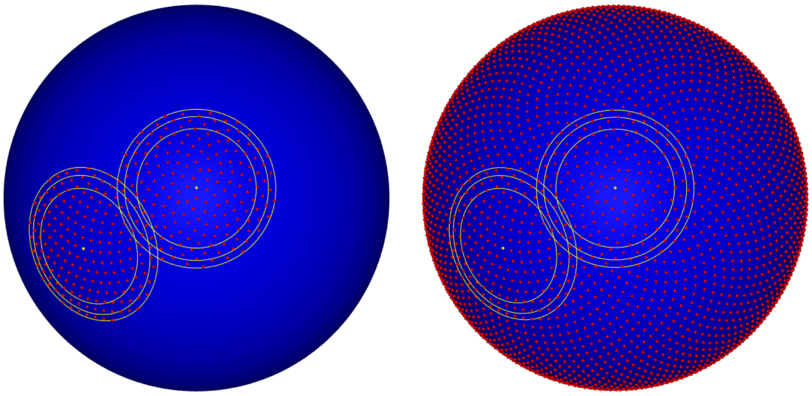
with c being a constant that is tuned by hand to optimize the performance. Here, $\bar{w}_{j,k,r}^{(s)}$ is the average of the coefficients near $\tilde{w}_{j,k}^{(s)}$ determined by a spherical cap $C(\boldsymbol{x}, r) := \{\boldsymbol{y} \in \mathbb{S}^2 : \|\boldsymbol{x}\times\boldsymbol{y}\| \le r\}$ of radius r and centered at $\boldsymbol{x} = \boldsymbol{x}_{j+1,k}$. The symbol $\times$ denotes cross product. More precisely, we can obtain the neighborhood $\mathcal{N}_{j+1,k,r}$ of $\boldsymbol{x}_{j+1,k}$ in $C(\boldsymbol{x}_{j+1,k}, r)$ as $\mathcal{N}_{j+1,k,r} := X_{N_{j+1}} \cap C(\boldsymbol{x}_{j+1,k}, r)$. Then, $\bar{w}_{j,k,r}^{(s)} = \frac{1}{|\mathcal{N}_{j+1,k,r}|}\sum_{i:\boldsymbol{x}_i\in\mathcal{N}_{j+1,k,r}} |\tilde{w}_{j,i}^{(s)}|^2$, where $|\mathcal{N}_{j+1,k,r}|$ denotes the cardinality of the set $\mathcal{N}_{j+1,k,r}$. After the thresholding procedure, we denormalize $\check{w}_{j,k}^{(s)}$ to obtain the updated coefficient $w_{j,k}^{(s)} = \check{w}_{j,k}^{(s)} \cdot \|\psi_{j,k}^{(s)}\|_{L^2(\mathbb{S}^2)}$. Finally, framelet reconstruction is applied to the updated coefficient sequences.

Similarly, the LS thresholding method for g is

$$g_{thr}(\boldsymbol{x}_{J+1,k}) = \begin{cases} g(\boldsymbol{x}_{J+1,k}) - \operatorname{sgn}(g(\boldsymbol{x}_{J+1,k}))\tau_{J+1,k}, & |g(\boldsymbol{x}_{J+1,k})| \ge \tau_{J+1,k,r}, \\ 0, & |g(\boldsymbol{x}_{J+1,k})| < \tau_{J+1,k,r}. \end{cases} \tag{5.3}$$

where $\tau_{J+1,k,r} = \frac{c_1\cdot\sigma^2}{\sqrt{(\bar{g}(\boldsymbol{x}_{J+1,k})-\sigma^2)_+}}$ with $\bar{g}(\boldsymbol{x}_{J+1,k}) = \frac{1}{|\mathcal{N}_{J+1,k,r}|}\sum_{i:\boldsymbol{x}_i\in\mathcal{N}_{J+1,k,r}} |g(\boldsymbol{x}_{j+1,i})|^2$. Then, we obtain g_{thr} after the LS thresholding.

In practice, the neighborhood $\mathcal{N}_{j+1,k,r}$ of $\boldsymbol{x}_{j+1,k}$ in $X_{N_{j+1}}$ can be found through the nearest neighborhood search algorithm (rnn-search). During our numerical experiments, we choose different radius r for $\mathcal{N}_{j+1,k,r}$ according to $r_i = \frac{\rho_i}{(t_{j+1}+1)^2}$, where ρ_i is a constant for the ith spherical cap layer, which roughly gives points near the center within the layer defined by the boundary $\partial C(\boldsymbol{x}, r_i)$ of $C(\boldsymbol{x}, r_i)$. After running some tests, we set $\rho_i = 13.84\cdot i$. With the above definition, we can precompute the set $\mathcal{N}_{r_i}(X_N) = \{\mathcal{N}_{r_i}(\boldsymbol{x}_k) := \{\boldsymbol{x} \in C(\boldsymbol{x}_k, r) \cap X_N\} \,|\, k \in [N]\}$



(a) Partial view (b) General view

Figure 5. *Spherical caps (rnn-search) from an SPD spherical 64-design point set.* (a) *Partial view with points inside the caps.* (b) *Full view with all caps and points.*

for some fixed $i \in \mathbb{N}$ and for a given point set X_N to speed up the LS thresholding process. In Figure 5, we shows an example of a spherical cap boundary $\partial C(\boldsymbol{x}, r_i)$ for $i = 15, 22, 27$ which centroids are $\boldsymbol{x} = \boldsymbol{x}_1 = (0,0,1)^\top$ and $\boldsymbol{x} = \boldsymbol{x}_{500} = (0.3018, -0.5854, 0.7525)^\top$, respectively, from a SPD spherical 64-design point set; see Table 1.

Note that the denoising procedure can be modeled as a solution to the following optimization problem:

$$\min_{f_{thr}\in\Pi_{t_J}, g_{thr}\in(\Pi_{t_J})^\perp} \frac{1}{2}\|f_\sigma - f_{thr} - g_{thr}\|_2^2 + \|\Gamma_f \mathcal{W} f_{thr}\|_1 + \|\Gamma_g g_{thr}\|_1, \tag{5.4}$$

where $\mathcal{W}$ is the framelet transform operator associated with $\mathcal{F}^J_{J_0}(\eta, \mathcal{Q})$, and Γ_f and Γ_g are the diagonal weight operators associated with (5.1) and (5.3), respectively. We refer to [8, 38, 39, 46] and many references therein for more details on the study of related ℓ_1-optimization models.

We next consider the denoising of two types of data: the ETOPO1 data and the spherical images.

5.3. ETOPO. We next discuss the denoising of the ETOPO1 data [1]. It is an elevation dataset of the earth, which includes the elevation information on $\mathbb{S}^2$ sampled on a grid X_E of 10800×21600 points. The ETOPO1 is a spherical geometry data formed by an equally distributed position. That is, the grid is given by $X_E := \{(\theta_i, \phi_j) \in [0,\pi] \times [0, 2\pi) \,|\, i = 1, \ldots, 10800, j = 1, \ldots, 21600\}$ with $\theta_i = (i-1)\Delta$, $\phi_j = (j-1)\Delta$ and $\Delta = \frac{\pi}{10800}$. For a spherical point set X_N, we can easily resample the ETOPO1 data on X_E to a data on X_N by finding the $\boldsymbol{x}(\theta, \phi) \in X_N$ with respect to the nearest ETOPO1 index according to $i_{\boldsymbol{x}} = \lceil \frac{\phi}{\Delta} \rceil$ and $j_{\boldsymbol{x}} = \lceil \frac{\theta}{\Delta} \rceil$, where $\lceil \cdot \rceil$ is the ceiling operator. Thus, for a given $X_{N_{J+1}}$, we can obtain a ETOPO1 data on $X_{N_{J+1}}$ by $f_o(\boldsymbol{x}) = \text{ETOPO1}(i_{\boldsymbol{x}}, j_{\boldsymbol{x}})$, $\boldsymbol{x} \in X_{N_{J+1}}$, where ETOPO1$(i,j)$ denotes the (i,j)-entry of the ETOPO1 data.

We generate the noisy ETOPO1 data $f_\sigma = f_o + G_{\sigma_{f_o}}$ on $X_{N_{J+1}}$ with noise level $\sigma_{f_o} = \sigma |f_0|_{\max}$ for $\sigma \in \{0.050, 0.075, \ldots, 0.175, 0.200\}$. Given a group of spherical t-design point sets

X_{N_j} (SPD) with degrees $t_0 = 256, t_1 = 512, t_2 = 1024$, we have the spherical framelet system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$ ($J_0 = 0, J = 1$) with $\eta = \eta_3$. We do a lot of experiments to fix $c = c_1 = 0.6$ and the spherical cap layer orders $i = 12$ in the cap radius r_i. After the denoising procedure as described in subsection 5.2, we obtained a denoised signal $f_{\sigma,thr}$. We use $\mathrm{SNR}(f_o, f_{\sigma,thr}) = 10\log_{10}(\frac{\|f_o\|}{\|f_{\sigma,thr}-f_o\|})$ for measuring the quality of denoising. The results are presented in Table 3.

From Table 3, we can see that the behavior of the denoising procedure shows that $\eta_3 > \eta_2 > \eta_1$, and it does increase the SNR of the denoised signal up to 11.6 dB. We demonstrate in Figure 6 the figures for the ground truth signal f_o, its noisy version f_σ for $\sigma = 0.05$, and the reconstruction denoised signal $f_{\sigma,thr}$. The SNR of the final denoised data η_3 is 5.98 dB greater than that of the initial noisy data SNR_0. We further show in Figure 7 the framelet coefficient sequences $\{\boldsymbol{v}_0, \boldsymbol{w}_j^{(s)} \,|\, j = 0, 1; s = 1, 2, 3\}$ of f in the projection decomposition of $f_\sigma = f + g$ for some $f \in \Pi_{t_J}$ with $t_J = 512$ by the truncated system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$. One can see

Table 3

ETOPO1 denoising results with respect to different noise level σ with filter bank η_3. The row SNR_0 is the initial SNR between f_σ and f_o. The row η_i is the final $\mathrm{SNR}(f_o, f_{\sigma,thr})$ with respect to the denoising using the filter bank η_i, $i = 1, 2, 3$.

σ	0.05	0.075	0.1	0.125	0.15	0.175	0.2
SNR_0	**16.38**	**12.85**	**10.36**	**8.42**	**6.83**	**5.50**	**4.34**
η_1	22.23	20.27	18.88	17.85	17.06	16.42	15.85
η_2	22.34	20.38	18.97	17.91	17.09	16.44	15.88
η_3	**22.36**	**20.41**	**19.01**	**17.95**	**17.12**	**16.45**	**15.92**

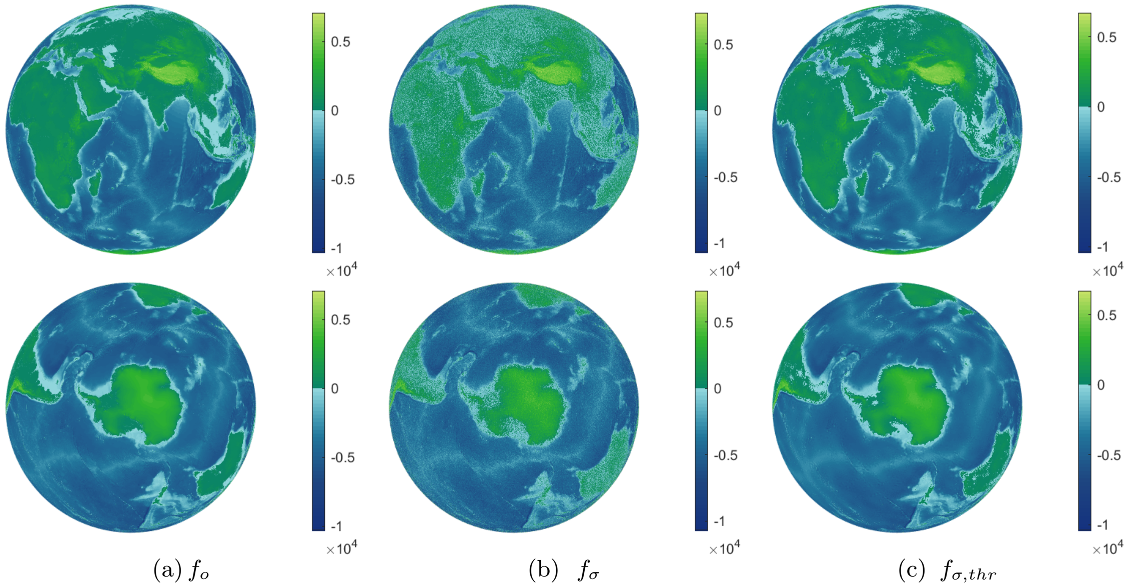


Figure 6. *The behavior of denoising ETOPO1 f_σ for $\sigma = 0.05$ by η_3 on SPD with $t_0 = 256, t_1 = 512, t_2 = 1024$. Top 3: north view. Bottom 3: south view.*

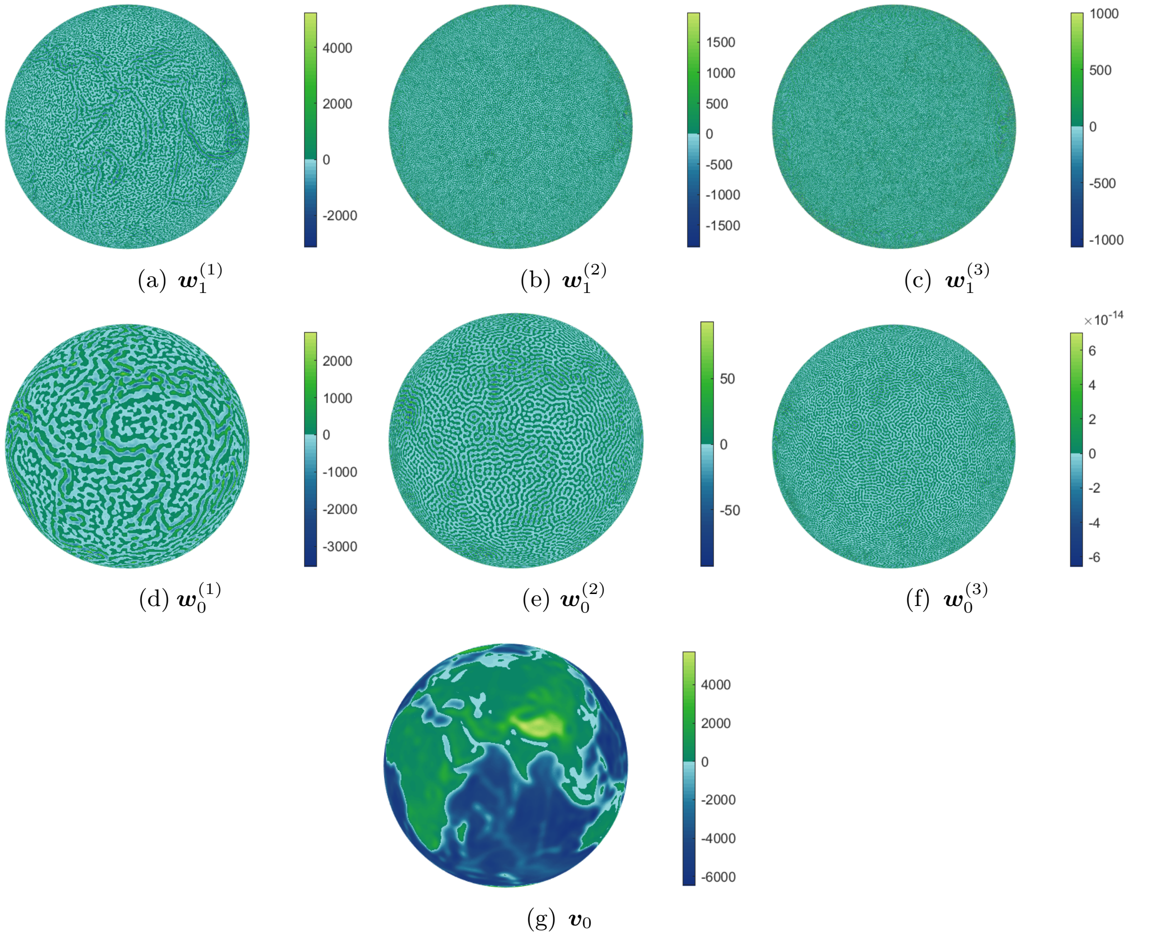


Figure 7. *The 2-levels framelet decomposition for ETOPO f_σ with $\sigma = 0.05$ by η_3 on SPD with $t_0 = 256, t_1 = 512, t_2 = 1024$. $\boldsymbol{w}_j^{(s)}$ is with respect to the sth high pass filter at the jth level decomposition. $\boldsymbol{v}_0$ is with respect to the low-pass filter decomposition in the coarsest level.*

that the coefficient sequence $\boldsymbol{w}_j^{(s)}$ for $j = 0, 1; s = 1, 2, 3$ do contain significant noise from the original data. This confirms the effectiveness of using the multiscale system to extract noise from noisy data on the sphere.

5.4. Spherical images. We finally discuss the denoising of spherical images. For a given gray scale image IMG (pixel value range in $[0, 255]$) of size $m \times n$, similar to the ETOPO1, we identify it as a spherical data on the grid $X_G = \{(\theta_i, \phi_j) : i = 1, \dots, m, j = 1, \dots, n\} \subset [0, \pi] \times [0, 2\pi)$ with $\theta_i = (i-1)\Delta_\theta$, $\phi_j = (j-1)\Delta_\phi$ and $\Delta_\theta = \frac{\pi}{m}$, $\Delta_\phi = \frac{2\pi}{n}$. For a spherical point sets X_N, we can easily resample the image data on X_G to a data on X_N by finding the $\boldsymbol{x}(\theta, \phi) \in X_N$ with respect to the nearest image index by $i_{\boldsymbol{x}} = \lceil \frac{\phi}{\Delta_\theta} \rceil$ and $j_{\boldsymbol{x}} = \lceil \frac{\theta}{\Delta_\phi} \rceil$. Thus, for a given $X_{N_{J+1}}$, we can obtain a spherical image data on $X_{N_{J+1}}$ by $f_o(\boldsymbol{x}) = \mathrm{IMG}(i_{\boldsymbol{x}}, j_{\boldsymbol{x}})$, $\boldsymbol{x} \in X_{N_{J+1}}$, where $\mathrm{IMG}(i, j)$ is the (i, j)-entry of the image.

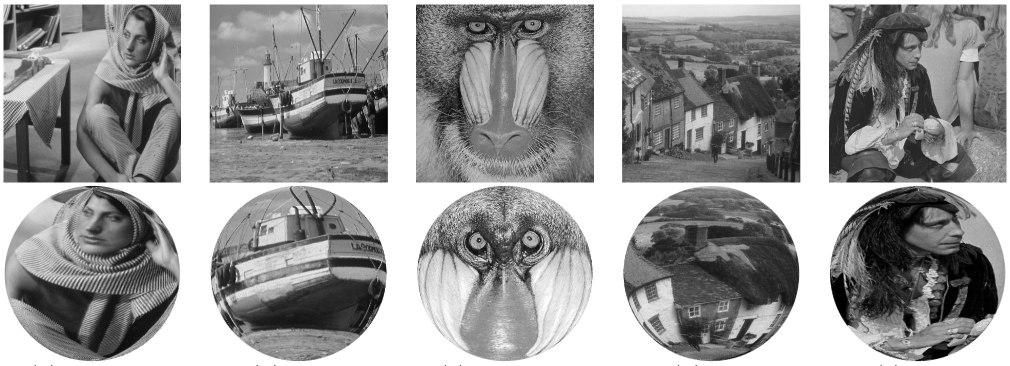
We use 512×512 pixels classical images Barbara, Boat, Mandrill, Hill, and Man as the input data to generate spherical image data f_o by the above procedure; see Figure 8. Given spherical t-design point sets X_{N_j} (SPD) corresponding to degree $t_0 = 256, t_1 = 512, t_2 = 1024$. Let $J_0 = 0$ and $J = 1$. The noisy spherical image data $f_\sigma = f_0 + G_{\sigma\cdot 255}$ on $X_{N_{J+1}}$ with $\sigma \in \{0.05, 0.075, \dots, 0.175, 0.2\}$. We have the spherical framelet system $\mathcal{F}^J_{J_0}(\eta, \mathcal{Q})$ with filter banks $\eta = \eta_3$ (since η_1, η_2 do not perform as good as η_3, we omit their results for spherical images) and LS thresholding method with the setting of $c = 0.6, c_1 = 0.5$ and the spherical cap layer order $i = 23$. We apply the denoising procedure as above to obtain the denoised signal $f_{\sigma,thr}$. We use PSNR to measure the quality of image denoising, which is $\mathrm{PSNR}(f_o, f_{\sigma,thr}) := 10\log_{10}(\frac{255^2}{\mathrm{MSE}})$ and MSE is the mean squared error which defined as $\mathrm{MSE} = \frac{1}{N_{J+1}}\sum_{\boldsymbol{x}\in X_{N_{J+1}}} |f_o(\boldsymbol{x}) - f_{\sigma,thr}(\boldsymbol{x})|^2$. We show the results in Table 4.

From the table, we conclude that the (semidiscrete) spherical tight framelets with LSthreshold method based on spherical t-design point sets do provide effective results in denoising and reconstruction.

Table 4

Images denoising results. For each images, the first row is $\mathrm{PSNR}_0 := \mathrm{PSNR}(f_o, f_\sigma)$, *and the second row is* $\mathrm{PSNR}(f_o, f_{\sigma,thr})$ *value using* η_3.

Image	σ	0.05	0.075	0.1	0.125	0.15	0.175	0.2
Barbara	PSNR_0	**26.34**	**22.81**	**20.31**	**18.38**	**16.79**	**15.45**	**14.29**
	η_3	**30.84**	**28.56**	**27.07**	**25.97**	**25.12**	**24.45**	**23.87**
Boat	PSNR_0	**26.02**	**22.50**	**20.00**	**18.06**	**16.48**	**15.14**	**13.98**
	η_3	**31.45**	**29.39**	**27.90**	**26.66**	**25.62**	**24.74**	**24.05**
Mandrill	PSNR_0	**28.18**	**24.66**	**22.16**	**20.22**	**18.63**	**17.30**	**16.14**
	η_3	**30.43**	**27.90**	**26.23**	**25.00**	**24.08**	**23.40**	**22.89**
Hill	PSNR_0	**26.70**	**23.17**	**20.68**	**18.74**	**17.15**	**15.81**	**14.65**
	η_3	**31.71**	**29.66**	**28.21**	**27.16**	**26.39**	**25.81**	**25.35**
Man	PSNR_0	**26.51**	**22.99**	**20.49**	**18.55**	**16.97**	**15.63**	**14.47**
	η_3	**32.18**	**29.97**	**28.46**	**27.28**	**26.35**	**25.61**	**25.02**



(a) Barbara (b) Boat (c) Mandrill (d) Hill (e) Man

Figure 8. *Project images by spherical* 1024-*design point set (SPD) on* $\mathbb{S}^2$. *Top: original image. Bottom: spherical image.*

6. Conclusions and final remarks. In this paper, starting from numerically solving a minimization problem, we use a variational characterization of the spherical t-design $A_{N,t}$ to find spherical t-designs with large value t using the trust-region method. We use the obtained spherical t-designs for function approximation and build spherical tight framelet systems. Especially, we construct truncated spherical tight framelet systems for discrete spherical signal processing. Several numerical experiments demonstrate the efficiency and effectiveness of our spherical framelet systems in processing signals or images on the sphere. We remark that the truncated systems are not studied in [61], which plays the key role for discrete signal processing here. Compared to [25], we use the trust-region method instead of the line-search method and do not need to refer to the manifold versions of the gradient and Hessian.

The polynomial-exactness of the spherical t-designs plays a key role in the construction of spherical tight framelet systems and their truncated versions. The fast framelet transforms and the multiscale structure of the framelet systems provide efficient separation of noise from the noisy spherical signals. As one can see from our numerical experiments, the noise spreads in both f and g in the decomposition $f_\sigma = f + g$. In practice, one can only process f up to certain polynomial approximation space Π_t by the truncated system, while the part g could be spread over the higher frequency spectrum. The noise might not be well-suppressed in the part g in our denoising procedure. We shall consider in the future the further improvement of the denoising of g. Moreover, the quadrature rule sequence $\mathcal{Q}$ is not nested in general. It would be nice to have nested quadrature rule sequences for spherical tight framelets in view of the multilevel structure of the traditional framelet systems on the Euclidean domain for the usual image processing (of grid data).

Acknowledgments. We would like to thank anonymous reviewers for their valuable comments and suggestions that greatly helped us to improve the presentation of this paper.

REFERENCES

[1] C. Amante and B. W. Eakins, *ETOPO1 arc-minute global relief model: Procedures, data sources and analysis*, in NOAA Technical Memorandum NESDIS NGDC-24, Vol. 10, National Geophysical Data Center, NOAA, 2009, V5C8276M.

[2] C. An, X. Chen, I. H. Sloan, and R. S. Womersley, *Well conditioned spherical designs for integration and interpolation on the two-sphere*, SIAM J. Numer. Anal., 48 (2010), pp. 2135–2157, https://doi.org/10.1137/100795140.

[3] J.-P. Antoine and P. Vandergheynst, *Wavelets on the n-sphere and related manifolds*, J. Math. Phys., 39 (1998), pp. 3987–4008.

[4] J.-P. Antoine and P. Vandergheynst, *Wavelets on the 2-sphere: A group-theoretical approach*, Appl. Comput. Harmon. Anal., 7 (1999), pp. 262–291.

[5] E. Bannai and E. Bannai, *A survey on spherical designs and algebraic combinatorics on spheres*, European J. Combin., 30 (2009), pp. 1392–1425.

[6] A. Bondarenko, D. Radchenko, and M. Viazovska, *Optimal asymptotic bounds for spherical designs*, Ann. of Math. (2), 178 (2013), pp. 443–452.

[7] A. Bondarenko, D. Radchenko, and M. Viazovska, *Well-separated spherical designs*, Constr. Approx., 41 (2015), pp. 93–112.

[8] F. Chen, L. Shen, B. W. Suter, and Y. Xu, *A fast and accurate algorithm for ℓ_1 minimization problems in compressive sampling*, EURASIP J. Adv. Signal Process., 2015 (2015), pp. 1–12.

[9] X. Chen, A. Frommer, and B. Lang, *Computational existence proofs for spherical t-designs*, Numer. Math., 117 (2011), pp. 289–305.

[10] X. Chen and R. S. Womersley, *Existence of solutions to systems of underdetermined equations and spherical designs*, SIAM J. Numer. Anal., 44 (2006), pp. 2326–2341, https://doi.org/10.1137/050626636.
[11] X. Chen and R. S. Womersley, *Spherical designs and nonconvex minimization for recovery of sparse signals on the sphere*, SIAM J. Imaging Sci., 11 (2018), pp. 1390–1415, https://doi.org/10.1137/17M1147378.
[12] A. Chernih, I. H. Sloan, and R. S. Womersley, *Wendland functions with increasing smoothness converge to a Gaussian*, Adv. Comput. Math., 40 (2014), pp. 185–200.
[13] C. K. Chui, *An Introduction to Wavelets*, Vol. 1, Academic Press, 1992.
[14] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445, https://doi.org/10.1137/0806023.
[15] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Grundlehren Math. Wiss. 290, Springer-Verlag, New York, 2013.
[16] F. Dai and H. Wang, *Positive cubature formulas and Marcinkiewicz-Zygmund inequalities on spherical caps*, Constr. Approx., 31 (2010), pp. 1–36.
[17] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992, https://doi.org/10.1137/1.9781611970104.
[18] I. Daubechies, B. Han, A. Ron, and Z. Shen, *Framelets: MRA-based constructions of wavelet frames*, Appl. Comput. Harmon. Anal., 14 (2003), pp. 1–46.
[19] P. Delsarte, J. Goethals, and J. Seidel, *Spherical codes and designs*, Geom. Dedicata, 6 (1977), pp. 363–388.
[20] L. Demanet and P. Vandergheynst, *Directional Wavelets on the Sphere*, Technical report R-2001-2, Signal Processing Laboratory (LTS), EPFL, Lausanne, Switzerland, 2001.
[21] M. Ehler, U. Etayo, B. Gariboldi, G. Gigante, and T. Peter, *Asymptotically optimal cubature formulas on manifolds for prefixed weights*, J. Approx. Theory, 271 (2021), 105632.
[22] M. Ehler, M. Gräf, and C. J. Oates, *Optimal Monte Carlo integration on closed manifolds*, Stat. Comput., 29 (2019), pp. 1203–1214.
[23] Q. L. Gia, I. H. Sloan, and H. Wendland, *Multiscale analysis in sobolev spaces on the sphere*, SIAM J. Numer. Anal., 48 (2010), pp. 2065–2090, https://doi.org/10.1137/090774550.
[24] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, *HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere*, Astrophys. J., 622 (2005), pp. 759–771.
[25] M. Gräf and D. Potts, *On the computation of spherical designs by a new optimization approach based on fast spherical Fourier transforms*, Numer. Math., 119 (2011), pp. 699–724.
[26] K. Guo and D. Labate, *The construction of smooth parseval frames of shearlets*, Math. Model. Nat. Phenom., 8 (2013), pp. 82–105.
[27] B. Han, *Framelets and Wavelets: Algorithms, Analysis, and Applications*, Springer, Cham, 2017.
[28] B. Han, Z. Zhao, and X. Zhuang, *Directional tensor product complex tight framelets with low redundancy*, Appl. Comput. Harmon. Anal., 41 (2016), pp. 603–637.
[29] B. Han and X. Zhuang, *Algorithms for matrix extension and orthogonal wavelet filter banks over algebraic number fields*, Math. Comput., 82 (2013), pp. 459–490.
[30] R. H. Hardin and N. J. Sloane, *Mclaren's improved snub cube and other new spherical designs in three dimensions*, Discrete Comput. Geom., 15 (1996), pp. 429–441.
[31] K. Hesse and R. S. Womersley, *Numerical integration with polynomial exactness over a spherical cap*, Adv. Comput. Math., 36 (2012), pp. 451–483.
[32] I. Iglewska-Nowak, *Frames of directional wavelets on n-dimensional spheres*, Appl. Comput. Harmon. Anal., 43 (2017), pp. 148–161.
[33] J. Korevaar and J. Meyers, *Spherical Faraday cage for the case of equal point charges and Chebyshev-type quadrature on the sphere*, Integral Transforms Spec. Funct., 1 (1993), pp. 105–117.
[34] S. Kunis and D. Potts, *Fast spherical Fourier algorithms*, J. Comput. Appl. Math., 161 (2003), pp. 75–98.
[35] G. Kutyniok and D. Labate, *Shearlets: Multiscale Analysis for Multivariate Data*, Birkhäuser Boston, Boston, 2012.

[36] Q. T. Le Gia and H. N. Mhaskar, *Localized linear polynomial operators and quadrature formulas on the sphere*, SIAM J. Numer. Anal., (2008), pp. 440–466, https://doi.org/10.1137/060678555.
[37] J. Li, H. Feng, and X. Zhuang, *Convolutional neural networks for spherical signal processing via area-regular spherical haar tight framelets*, IEEE Trans. Neural Netw. Learn. Syst., (2022), https://doi.org/10.1109/TNNLS.2022.3160169.
[38] Y.-R. Li, R. H. Chan, L. Shen, and X. Zhuang, *Regularization with multilevel non-stationary tight framelets for image restoration*, Appl. Comput. Harmon. Anal., 53 (2021), pp. 332–348.
[39] Y.-R. Li, L. Shen, and X. Zhuang, *A tailor-made 3-dimensional directional haar semi-tight framelet for pMRI reconstruction*, Appl. Comput. Harmon. Anal., 60 (2022), pp. 446–470.
[40] U. Maier, *Numerical calculation of spherical designs*, Adv. Multivariate Approx., (1999), pp. 213–226.
[41] S. Mallat, *A Wavelet Tour of Signal Processing*, Elsevier/Academic Press, Amsterdam, 1999.
[42] J. D. McEwen, C. Durastanti, and Y. Wiaux, *Localisation of directional scale-discretised wavelets on the sphere*, Appl. Comput. Harmon. Anal., 44 (2018), pp. 59–88.
[43] H. N. Mhaskar, *On the representation of smooth functions on the sphere using finitely many bits*, Appl. Comput. Harmon. Anal., 18 (2005), pp. 215–233.
[44] H. N. Mhaskar, *Approximate quadrature measures on data-defined spaces*, in Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan, Springer, Cham, 2018, pp. 931–962.
[45] H. N. Mhaskar and J. Prestin, *Polynomial frames: A fast tour*, in Approximation Theory XI: Gatlinburg, 2004, Mod. Methods Math., Nashboro Press, Brentwood, TN, 2005, pp. 287–318.
[46] C. A. Micchelli, L. Shen, and Y. Xu, *Proximity algorithms for image models: Denoising*, Inverse Problems, 27 (2011), 045009.
[47] F. J. Narcowich and J. D. Ward, *Nonstationary Spherical Wavelets for Scattered Data*, Ser. Approx. Decompos. 6, World Sci. Publ., River Edge, NJ, 1995, pp. 301–308.
[48] G. Nebe and B. Venkov, *On tight spherical designs*, St. Petersburg Math. J., 24 (2013), pp. 485–491.
[49] D. Potts and M. Tasche, *Interpolatory Wavelets on the Sphere*, Ser. Approx. Decompos. 6, World Sci. Publ., River Edge, NJ, 1995, pp. 335–342.
[50] A. Ron and Z. Shen, *Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operator*, J. Funct. Anal., 148 (1997), pp. 408–447.
[51] P. Schröder and W. Sweldens, *Spherical wavelets: Efficiently representing functions on the sphere*, in Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 1995, pp. 161–172.
[52] P. Seymour and T. Zaslavsky, *Averaging sets: A generalization of mean values and spherical designs*, Adv. Math., 52 (1984), pp. 213–240.
[53] I. H. Sloan and R. S. Womersley, *Extremal systems of points and numerical integration on the sphere*, Adv. Comput. Math., 21 (2004), pp. 107–125.
[54] I. H. Sloan and R. S. Womersley, *A variational characterisation of spherical designs*, J. Approx. Theory, 159 (2009), pp. 308–318.
[55] S. Smale, *Mathematical problems for the next century,* in Mathematics: Frontiers and Perspectives, Amer. Math. Soc., Providence, RI, 2000, pp. 271–294.
[56] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
[57] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Vol. 1, Springer, New York, 2006.
[58] R. Swinbank and R. James Purser, *Fibonacci grids: A novel approach to global modelling*, Quart. J. R. Meteorol. Soc., 132 (2006), pp. 1769–1793.
[59] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii, *Quantum Theory of Angular Momentum*, World Scientific, 1988.
[60] G. Wagner and B. Volkmann, *On averaging sets*, Monatsh. Math., 111 (1991), pp. 69–78.
[61] Y. G. Wang and X. Zhuang, *Tight framelets and fast framelet filter bank transforms on manifolds*, Appl. Comput. Harmon. Anal., 48 (2020), pp. 64–95.
[62] E. W. Weisstein, *Sphere Point Picking*, 2004, https://mathworld.wolfram.com/SpherePointPicking.html; from MathWorld-A Wolfram Web Resource.

[63] Y. Wiaux, J. D. McEwen, and P. Vielva, *Complex data processing: Fast wavelet analysis on the sphere*, J. Fourier Anal. Appl., 13 (2007), pp. 477–493.

[64] R. S. Womersley, *Efficient spherical designs with good geometric properties*, in Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan, Springer, Cham, 2018, pp. 1243–1285.

[65] X. Zhuang, *Digital affine shear transforms: Fast realization and applications in image/video processing*, SIAM J. Imaging Sci., 9 (2016), pp. 1437–1466, https://doi.org/10.1137/15M1048318.