Spherical Designs for Function Approximation and Beyond

Yuchen Xiao[1] and Xiaosheng Zhuang[1*]

[1]Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong SAR, China.

*Corresponding author(s). E-mail(s): xzhuang7@cityu.edu.hk;
Contributing authors: yc.xiao@my.cityu.edu.hk;

Abstract

In this paper, we compare two optimization algorithms using full Hessian and approximation Hessian to obtain numerical spherical designs through their variational characterization. Based on the obtained spherical design point sets, we investigate the approximation of smooth and non-smooth functions by spherical harmonics with spherical designs. Finally, we use spherical framelets for denoising Wendland functions as an application, which shows the great potential of spherical designs in spherical data processing.

Keywords: Spherical t-design, variational characterization, restart conjugate gradient, trust region, spherical harmonic transforms, function approximation, Wendland functions, spherical framelets

1 Introduction

Nowadays, the rapid growth of computing power, the massive explosion of data, and the fast advance of modern information technologies have built a new era of artificial intelligence (AI). Especially in deep learning, deep neural networks are used to process extremely large and complex data. Though AI-trained models have many benefits, they require significant computing resources, which can be a challenge for processing certain types of structured data or for certain applications. For example, data in ImageNet [1] are images that can be viewed as samples from a high-dimensional manifold, social network data [2] are with high-dimensional features on a graph-structure domain, data fed into the large language models (LLMs) are typical text type that needs to

be vectorized [3]. How to properly represent such data so that one can process them efficiently is the key to their successful applications. One common way is to use the normalization technique for high-dimensional data. Once normalized, they can be viewed as data on the d-dimensional unit sphere $\mathbb{S}^d := \{\boldsymbol{x} \in \mathbb{R}^{d+1} \,|\, \|\boldsymbol{x}\| = 1\}$, where $\|\cdot\|$ is the Euclidean norm. Besides these data from normalization, many data from the real world are actually spherical data, e.g., navigation and geology data, cosmic microwave background data, panoramic images and videos, etc. Therefore, spherical data processing plays an important role in deep learning as well as many other areas [4–8].

Spherical designs, as one of the very crucial types of point configurations on the sphere, play a very important role in spherical data processing. The increasing demand of processing spherical data has led to increased interest in exploring spherical designs for real-world applications. Delsarte, Goethals, and Seidel established the concept of spherical designs [9] in 1977, where a finite point set $X_N := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{S}^d$ is said to be a *spherical t-design* if for any polynomial $p : \mathbb{R}^{d+1} \to \mathbb{R}$ of degree at most t, the quadrature rule holds:

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

Here, $\mu_d(\boldsymbol{x})$ denotes the surface measure on $\mathbb{S}^d$ such that $\mu_d(\mathbb{S}^d) := \omega_d = \frac{2\pi^{\frac{d+1}{2}}}{\Gamma(\frac{d+1}{2})}$ is the surface area of $\mathbb{S}^d$ with $\Gamma(z) := \int_0^\infty x^{z-1}e^{-x}\mathrm{d}x$ and $\Pi_t := \Pi_t(\mathbb{S}^d)$ is the space of spherical polynomials on $\mathbb{S}^d$ with degree at most t. As we can see, a spherical t-design is an equal weight polynomial-exact quadrature rule in Π_t space, which is a set of points "nicely" distributed on the unit sphere. Spherical designs have significant applications in approximation theory, geometry, and combinatorics. For example, the best packing problems, the minimal energy problems, the optimal configurations related to Smale's 7th Problem, and so on. We refer to a very nice comprehensive survey of Bannai and Bannai [10] for the past five decades of research on spherical designs. Recently, it has been applied in image reconstruction and signal recovery [11–14]. By leveraging the properties of spherical designs, one can develop efficient algorithms for processing spherical data, which can have a wide range of applications in areas such as computer vision, geophysics, and astrophysics.

The existence of spherical t-designs is a deep theoretical problem that has yielded many profound mathematical results. Delsarte et al. [9] gave the lower bound of a spherical t-design on the number of points N for any degree $t \in \mathbb{N}^+$ on $\mathbb{S}^d$: $N \geq \mathcal{O}(t^d)$. Seymour and Zaslavsky [15] proved (non-constructive) that a spherical t-design exists for any t if N is sufficiently large. Wagner [16] proposed the feasible upper bounds with $N = \mathcal{O}(t^{Cd^4})$. Korevaar and Meyers [17] provided spherical t-designs exist in $N = \mathcal{O}(t^{d(d+1)/2})$ and conjectured that $N = \mathcal{O}(t^d)$. Bondarenko, Radchenko and Viazovska [18] verified that spherical t-designs indeed exists for $N \geq c_d t^d$ points, where c_d is constant controlled by d. Furthermore, they 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 $\mathcal{O}(N^{-1/d})$ [19].

Besides the theoretical development, numerical algorithms to produce spherical designs are of great importance for spherical data processing. Computing spherical t-designs can be regarded as non-linear equations and optimization problems. We refer to [13, 20–30] for some of the work on numerical spherical designs. In this paper and in what follows, we pay attention to spherical t-designs on the most important case $d = 2$, i.e., the 2-sphere $\mathbb{S}^2$. Sloan and Womersley [23] presented the variational characterization of spherical t-design as a nonnegative quantity $A_{N,t}(X_N)$ given by

$$A_{N,t}(X_N) := \frac{4\pi}{N^2} \sum_{\ell=0}^{t} \sum_{m=-\ell}^{\ell} \left| \sum_{i=1}^{N} Y_\ell^m(\boldsymbol{x}_i) \right|^2 - 1, \tag{2}$$

where Y_ℓ^m is spherical harmonic with degree ℓ and order m. They proved important properties about the connection of spherical t-designs and $A_{N,t}$. One is that X_N is a spherical t-design if and only if $A_{N,t} = 0$ (cf. Theorem 3 in [23]), which is the equivalent form with *Weyl sums* [9, 23] satisfying $\sum_{i=1}^{N} Y_\ell^m(\boldsymbol{x}_i) = 0$, $\ell = 1, \ldots, t$ and $m = -\ell, \ldots, \ell$. Hence, finding spherical t-designs can be regarded as solving a nonlinear and nonconvex minimization problem:

$$\min_{X_N \subset \mathbb{S}^2} A_{N,t}(X_N). \tag{3}$$

Based on the quantity $A_{N,t}$, Gräf and Potts [31] computed numerical spherical t-designs using nonequispaced fast spherical Fourier transforms (NFSFTs) and manifold optimization techniques for t up to 1000 with $N \approx \frac{t^2}{2}$ points. Following [23, 31], Xiao and Zhuang [13] used the trust-region method with NFSFTs to achieve t up to 3200 numerically. Numerical results in these papers showed that these methods can approximate spherical t-designs with high accuracy.

In this paper, we further investigate the numerical spherical t-designs in several aspects. First, to solve (3), we consider the restart-conjugate gradient method with line search strategy (LS-RCG) in Algorithm 1 and compare it with the trust-region with preconditioned conjugate gradient method (TR-PCG) presented in [13]. Based on algorithms of LS-RCG and TR-PCG , we use two types of point sets on $\mathbb{S}^2$ as initial point sets for computing spherical t-designs and comparing the two algorithms in terms of total iterations, square-root and ℓ_∞-error of $A_{N,t}(X_N)$. With the obtained spherical t-design point sets, we test their performance in function approximation with functions from the combinations of normalized Wendland functions as well as functions with discontinuity. Moreover, we further investigate the applications of spherical t-designs in the construction of spherical (semi)-tight framelets for spherical image/signal denoising.

The paper is organized as follows. In Section 2, we present the algorithms for computing spherical t-designs and provide numerical results for different approaches to compare their performance. In Section 3, we introduce an approximation algorithm using spherical-design point sets and present results for projecting smooth and non-smooth functions onto the sphere. In Section 4, we discuss the application of spherical

design point sets in semi-discrete spherical framelet systems and thresholding techniques. We demonstrate how these systems can be used for spherical data processing. The conclusion is given in Section 5.

2 Spherical designs from optimizations

In this section, we discuss the use of a line-search algorithm with a restart conjugate gradient technique (LS-RCG) for solving the nonlinear and nonconvex minimization problem in (3). We apply such an algorithm to obtain spherical t-designs with various initial point sets and compare them with the trust-region method with preconditioned conjugate gradient method (TR-PCG) in [13].

2.1 Line search with restart conjugate gradient method

A continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient of f at x is defined as

$$\nabla f := [\partial_1 f(x), \ldots, \partial_n f(x)]^\top, \tag{4}$$

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

$$[\mathcal{H}(f)]_{ij} := \left[\nabla^2 f\right]_{ij} := \partial_i \partial_j f(x), \qquad 1 \le i, j \le n, \tag{5}$$

where ∂_i is the partial derivative with respect to the ith coordiante of x. For the general nonlinear and nonconvex optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x), \tag{6}$$

there are mainly two global convergence approaches: one is the line search, and the other is the trust region. The trust region with preconditioned conjugate gradient method (TR-PCG) for spherical designs was discussed in [13]. We refer to [32–34] for the discussion of trust-region and conjugate gradient methods. In this paper, we use the line search approach with the restart-CG method.

The iterative scheme using the conjugate gradient (CG) method for solving (6) is given by $x_{k+1} = x_k + \alpha_k d_k$, where x_0 is a starting point, α_k is determined by a line search strategy, and d_k is a search direction. The first seach direction d_0 is commonly set as $d_0 = -g_0 := -\nabla f(x_0)$. Consecutive d_k is recursively defined as $d_{k+1} = -g_{k+1} + \beta_k d_k$ with $g_k := \nabla f(x_k)$ and a scalar β_k that can be determined with many choices. The conjugate gradient method was originally proposed by Hestenes and Stiefel [35] in the 1950s. Without using the restart strategy, the conjugate gradient method is only linearly convergent [36]. The restart strategy is used to periodically refresh the algorithm by erasing old information that may be redundant. It can lead to n-step quadratic convergence, that is, $\|x_{k+n} - x\| = \mathcal{O}(\|x_k - x\|^2)$ [37]. Fletcher and Reeves [38] extended the CG method for unconstrained nonlinear optimization with restart strategy and exact line search, which is the first nonlinear CG method. In 1967, Daniel [39] proposed a choice of update parameter β_k that required the

evaluation of the Hessian in CG method. In this paper, we use $\beta_k = \frac{g_{k+1}^\top B_{k+1} d_k}{d_k^\top B_{k+1} d_k}$ for updating β_k with restart criterial $g_k^\top g_{k+1}$ being small, where B_k can be the exact or approximation Hessian, see Aglorithm 1. For more discussion on restart procedures for the conjugate gradient method, we refer to [40, 41]. For α_k, we use the commonly used Newton-Raphson method for the line search strategy, see Algorithm 2.

Algorithm 1 Line search with restart conjugate gradient method (LS-RCG)

Input: x_0: initial point set; t: degree; $K_{\max}$: maximum iterations; r: restart orthogonal value; ε_1: termination tolerance;
Initialize $k = 0$, $f_0 = A_{N,t}(x_0)$, $g_0 = \nabla f_0$, $-d_0 = g_0$, $B_0 \approx \mathcal{H}(f_0)$.

while $k \leq K_{\max}$ and $\|g_k\| > \varepsilon_1$ **do**
  compute step size α_k from Algorithm 2
  $x_{k+1} = x_k + \alpha_k d_k$, $f_{k+1} = A_{N,t}(x_{k+1})$, $g_{k+1} = \nabla f_{k+1}$ and $B_{k+1} \approx \mathcal{H}(f_{k+1})$
  if $\|g_{k+1}\| \leq \varepsilon$ **then**
    break
  end if
  if $g_k^\top g_{k+1} \geq r\|g_k\|^2$ **then**
    $d_{k+1} = -g_{k+1}$
  else
    $\beta_k = \frac{g_{k+1}^\top B_{k+1} d_k}{d_k^\top B_{k+1} d_k}$ and $d_{k+1} = -g_{k+1} + \beta_k d_k$
  end if
  $k = k + 1$
end while

Output: minimizer $x^* = x_k \subset \mathbb{S}^2$.

2.2 Spherical harmonics and spherical designs

For each spherical coordinate $(\theta, \phi) \in [0, \pi] \times [0, 2\pi)$, it is associated with a point $\boldsymbol{x} := \boldsymbol{x}(\theta, \phi) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \in \mathbb{S}^2$. For each $\ell \in \mathbb{N}_0$ and $m = -\ell, \cdots, \ell$, the spherical harmonic $Y_\ell^m : \mathbb{S}^2 \to \mathbb{C}$ can be expressed as

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

where $P_\ell^m : [-1, 1] \to \mathbb{R}$ is the associated Legendre polynomial with form $P_\ell^m(z) := (1-z^2)^{\frac{m}{2}} \frac{\mathrm{d}^m}{\mathrm{d}z^m}(P_\ell(z))$ and the Legendre polynomial $P_\ell : [-1, 1] \to \mathbb{R}$ is with form $P_\ell(z) = \frac{1}{2^\ell \ell!}\frac{\mathrm{d}^\ell}{\mathrm{d}z^\ell}[(z^2-1)^\ell]$. Note that $Y_0^0 = 1/\sqrt{4\pi}$. The collection $\{Y_\ell^m \,|\, \ell \in \mathbb{N}_0, |m| \leq \ell\}$ forms a complete set of orthonormal bases for the Hilbert space of square-integrable functions $L^2(\mathbb{S}^2) := \{f : \mathbb{S}^2 \to \mathbb{C} \,|\, \int_{\mathbb{S}^2} |f(\boldsymbol{x})|^2 d\mu_2(\boldsymbol{x})\}$, where the L^2-inner product is defined by $\langle f, g\rangle_{L^2(\mathbb{S}^2)} := \int_{\mathbb{S}^d} f(\boldsymbol{x})\overline{g(\boldsymbol{x})}\mathrm{d}\mu_2(\boldsymbol{x})$ for $f, g \in L_2(\mathbb{S}^2)$. The orthogonality $\langle Y_\ell^m, Y_{\ell'}^{m'}\rangle_{L^2(\mathbb{S}^2)} = \delta_{mm'}\delta_{\ell\ell'}$ holds for $\ell, \ell' \in \mathbb{N}_0, |m| \leq \ell, |m'| \leq \ell'$. Here δ_{ij} is the Kronecker delta. With such an orthonormal basis, for any $f \in L_2(\mathbb{S}^2)$, it can be

Algorithm 2 Line search strategy

Input: x_k: current step point set; d_k: descent direction; $I_{\max}$: maximum iterations; ε_2: termination tolerance;

Initialize $n = 0$, $d = d_k$, $f_n = A_{N,t}(x_k)$, $g_n = \nabla A_{N,t}(x_k)$, $E_n = \mathcal{H}(A_{N,t}(x_k))$, $\alpha_n = \frac{g_n^\top d}{d^\top E_n d}$.

while $n \le I_{\max}$ **do**
  $x_{n+1} = x_n + \alpha_n d$ and $f_{n+1} = f(x_{n+1})$
  if $f_{n+1} - f_n > 0$ **then**
    $\alpha_{n+1} = c\alpha_n$, where $c \in (0, 1)$
  else
    $g_{n+1} = g(x_{n+1})$ and $E_{n+1} = \mathcal{H}(f_{n+1})$
    if $\frac{|g_{n+1}^\top d|}{\|g_{n+1}\|\|d\|} < \varepsilon_2$ **then**
      break
    end if
    $\alpha_{n+1} = \alpha_n - \frac{g_{n+1}^\top d}{d^\top E_{n+1} d}$
  end if
  $n = n + 1$
end while

Output: $\alpha_n \in \mathbb{R}$.

represented as $f = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \hat{f}_\ell^m Y_\ell^m$ with $\hat{f}_\ell^m := \langle f, Y_\ell^m \rangle_{L_2(\mathbb{S}^2)}$. Moreover, Π_t is the linear span of the orthonormal basis $\{Y_\ell^m : \ell \in \mathbb{N}_0, \ell \le t, |m| \le \ell\}$ with $\dim \Pi_t = (t+1)^2$.

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 $X_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{S}^2$ as

$$X_N := (\boldsymbol{\theta}, \boldsymbol{\phi}) := (\theta_1, \ldots, \theta_N, \phi_1, \ldots, \phi_N) \tag{7}$$

with $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_N)$, and $\boldsymbol{x}_i := \boldsymbol{x}_i(\theta_i, \phi_i)$ being the i-th point determined by its spherical coordinate satisfying $(\theta_i, \phi_i) \in [0, \pi] \times [0, 2\pi)$. The variational characterization $A_{N,t}(X_N)$ in (2) can be written as a smooth function of $2N$ variables:

$$A_{N,t}(X_N) = A_{N,t}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{4\pi}{N^2} \sum_{\ell=0}^{t} \sum_{m=-\ell}^{\ell} \left| \sum_{i=1}^{N} Y_\ell^m(\theta_i, \phi_i) \right|^2 - 1. \tag{8}$$

The spherical t-design point set X_N can be obtained by solving (8) using Algorithm 1 with the target function $f := A_{N,t}(\boldsymbol{\theta}, \boldsymbol{\phi})$. As discussed in [13], the key is the fast evaluations of $A_{N,t}$, $\nabla A_{N,t}$, and $\mathcal{H}(A_{N,t})$. By [13, Theorems 2.2 and 2.3], they can be evaluated through the fast spherical harmonic transforms with arithmetic complexity of order $\mathcal{O}(t^2 \log^2 t + N \log^2(\frac{1}{\epsilon}))$, where ϵ is prescribed accuracy of the approximate algorithms [42–45].

Moreover, the Hessian $\mathcal{H}(A_{N,t}) = \mathcal{H}_1 + \mathcal{H}_2$ is the sum of a diagonal matrix $\mathcal{H}_1$ and a rank-one matrix $\mathcal{H}_2$, see [13, Theorem 2.3]. In Algorithm 1, for B_k, it is not necessary to be the exact full Hessian. One can use an approximation version of the Hessian for B_k, e.g., $\mathcal{H}_2$.

2.3 Numerical spherical designs from two different approaches

In this section, based on LS-RCG in Algorithm 1 and TR-PCG from [13, Algorithm 2.1] to solve (3), we get numerical spherical t-designs[1]. We use two types of point sets on $\mathbb{S}^2$ as initial point sets for computing spherical t-designs with the setting as follows.

1. Spiral points (SP). To generate spiral points $\boldsymbol{x}_n = (\theta_n, \phi_n)$ on $\mathbb{S}^2$ for $n = 1, \ldots, N$, we set
$$\theta_n := \arccos\left(\frac{2n - (N+1)}{N}\right), \phi_n := \pi(2n - (N+1))\varphi^{-1},$$
where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio, refer to [46] which is Fibonacci spiral on the sphere, same on the setting of the initial distribution of spiral points in [31].
2. Uniformly distributed points (UD). To generate uniformly distributed points on unit sphere $\mathbb{S}^2$, we need to make sure that for each surface area $\mathrm{d}\mu = \sin\theta \mathrm{d}\theta\mathrm{d}\phi$ contain the same number of points. Thus by [47], we generate random variables $k_n \in (0, 1)$ and $p_n \in (0, 1)$ for $n = 1, \ldots, N$, then we have
$$\theta_n := \arccos(1 - 2k_n), \phi_n := 2\pi p_n.$$

We deal with point set $X_N \subset \mathbb{S}^2$ from the beginning by fixing the first point $\boldsymbol{x}_1 = (0, 0) \in X_N$ be the north pole point and the second point $\boldsymbol{x}_2 = (\theta_2, 0) \in X_N$ be on the prime meridian. Then we let $X_N = (\theta_2, \ldots, \theta_N, \phi_3, \ldots, \phi_N)^\top = [\Theta_{N-1}, \Phi_{N-2}]^\top$.

We show in Tables 1 and 2 for various degrees t with $N = (t+1)^2$, including the performance of different methods with the full Hessian $\mathcal{H}(A_{N,t}) = \mathcal{H}_1 + \mathcal{H}_2$ and the approximation Hessian $\mathcal{H}_2$ (the rank-one matrix) under different point sets. We compare the results of different initial point sets and the efficiency and accuracy of different approaches in the two methods.

In terms of the accuracy of function values $A_{N,t}(X_N)$ and $\nabla A_{N,t}(X_N)$ in the two algorithms, TR-PCG and LS-RCG have similar performance (both up to order 12 accuracy). In terms of the total iteration K, TR-PCG uses significantly fewer iterations than those of LS-RCG. The LS-RCG uses a lot of iterations in the line-search step while TR-PCG's main time-consuming step is on the step of finding the trust regions. In terms of the accuracy of using full Hessian and approximation Hessian, there are not many differences between these two approaches. However, in both methods, approximation Hessian takes fewer steps compared to full Hessian. Generally, full Hessian is a bit better than the approximation Hessian in accuracy. In terms of the point sets, from Tables 1 and 2, one can see that structured initial point sets SPs use significantly less iterations than those of the randomly distributed initial point sets UDs.

[1] All numerical experiments in this paper are conducted in MATLAB R2023a on a macOS Sonoma system's iMac (Retina 5K, 27-inch, 2020) with Intel Core i5 10500 CPU and 32 GB DDR4 memory.

Table 1 Computing of spherical t-designs by different approaches from SPD point sets. In point sets X_N, SPD means generated by SP. K is the total iterations in Algorithms (Alg). $\mathcal{H}$ represents the type of Hessian (full or approximation).

X_N	Alg	$\mathcal{H}$	t	N	K	$\sqrt{A_{N,t}(X_N)}$	$\|\nabla A_{N,t}(X_N)\|_\infty$
SPD	LS-RCG	Full	10	121	3551	3.95E-12	1.79E-14
			20	441	4111	1.64E-12	1.25E-14
			30	961	5484	1.49E-12	7.01E-15
			40	1681	7360	1.35E-12	6.77E-15
			50	2601	10397	1.60E-12	6.34E-15
			60	3721	11663	1.09E-12	4.27E-15
			70	5041	13409	1.39E-12	4.50E-15
			80	6561	13709	1.55E-12	5.63E-15
			90	8281	16380	1.27E-12	4.11E-15
			100	10201	18866	1.01E-12	3.11E-15
SPD	LS-RCG	Appr	10	121	3652	3.95E-12	1.87E-14
			20	441	4180	1.64E-12	1.19E-14
			30	961	5336	1.49E-12	7.08E-15
			40	1681	7270	1.35E-12	4.99E-15
			50	2601	10399	1.60E-12	6.49E-15
			60	3721	11670	1.09E-12	4.38E-15
			70	5041	13486	1.39E-12	4.15E-15
			80	6561	13419	1.55E-12	2.64E-15
			90	8281	16546	1.27E-12	2.59E-15
			100	10201	18867	1.01E-12	3.53E-15
SPD	TR-PCG	Full	10	121	169	3.94E-12	2.44E-14
			20	441	379	1.71E-12	9.44E-16
			30	961	480	1.49E-12	9.64E-16
			40	1681	727	1.34E-12	8.77E-16
			50	2601	764	1.58E-12	9.39E-15
			60	3721	939	1.08E-12	2.67E-15
			70	5041	1355	1.39E-12	9.78E-16
			80	6561	1235	1.54E-12	1.13E-15
			90	8281	1371	1.26E-12	6.89E-16
			100	10201	1699	1.00E-12	8.51E-16
SPD	TR-PCG	Appr	10	121	200	3.95E-12	7.77E-16
			20	441	377	1.65E-12	6.03E-16
			30	961	458	1.49E-12	9.76E-16
			40	1681	649	1.35E-12	9.74E-16
			50	2601	743	1.58E-12	1.07E-14
			60	3721	906	1.08E-12	2.83E-15
			70	5041	1037	1.39E-12	1.37E-15
			80	6561	1121	1.54E-12	6.45E-16
			90	8281	1385	1.26E-12	8.76E-16
			100	10201	1461	1.00E-12	9.97E-16

Table 2 Computing of spherical t-designs by different approaches from SUD point set. In point sets X_N, SUD means generated by UD. K is the total iterations in Algorithms (Alg). $\mathcal{H}$ represents the type of Hessian (full or approximation).

X_N	Alg	$\mathcal{H}$	t	N	K	$\sqrt{A_{N,t}(X_N)}$	$\|\nabla A_{N,t}(X_N)\|_\infty$
SUD	LS-RCG	Full	10	121	3633	3.38E-12	1.85E-14
			20	441	5167	1.96E-12	9.65E-15
			30	961	7232	1.79E-12	6.88E-15
			40	1681	8814	1.54E-12	9.50E-15
			50	2601	20221	1.64E-12	5.85E-15
SUD	LS-RCG	Appr	10	121	3701	4.28E-12	1.96E-14
			20	441	5233	1.79E-12	1.32E-14
			30	961	6595	1.81E-12	6.80E-15
			40	1681	8811	1.60E-12	4.86E-15
			50	2601	14260	1.57E-12	5.92E-15
SUD	TR-PCG	Full	10	121	316	4.16E-12	2.92E-15
			20	441	794	1.98E-12	9.15E-16
			30	961	847	1.72E-12	3.17E-15
			40	1681	1546	1.57E-12	2.57E-15
			50	2601	1660	1.44E-12	1.74E-14
SUD	TR-PCG	Appr	10	121	234	3.87E-12	7.54E-16
			20	441	567	1.80E-12	1.19E-15
			30	961	765	1.59E-12	3.18E-15
			40	1681	849	1.61E-12	1.91E-15
			50	2601	1151	1.47E-12	2.71E-14

Based on these tables, we further show the initial point sets and related spherical t-design point sets in Figures 1 and 2 by choosing $t = 50$ and $N = (t+1)^2$ for SP and UD. One can see that the SP initial point set is well distributed already and the pattern of the final spherical design does not change much from such an SP initial point set. On the other hand, one can see that the final spherical design of the UD point set is well distributed compared to its initial stage.

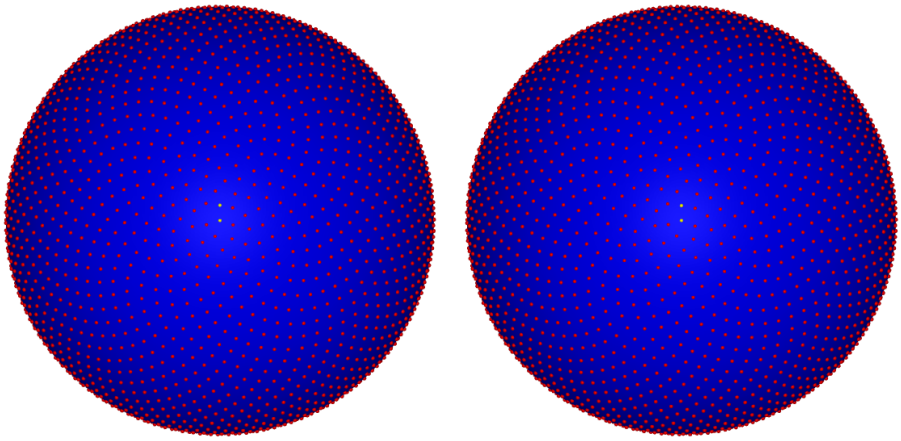
3 Function approximation with spherical designs

In this section, we study the approximation of functions using polynomials with the obtained spherical designs.

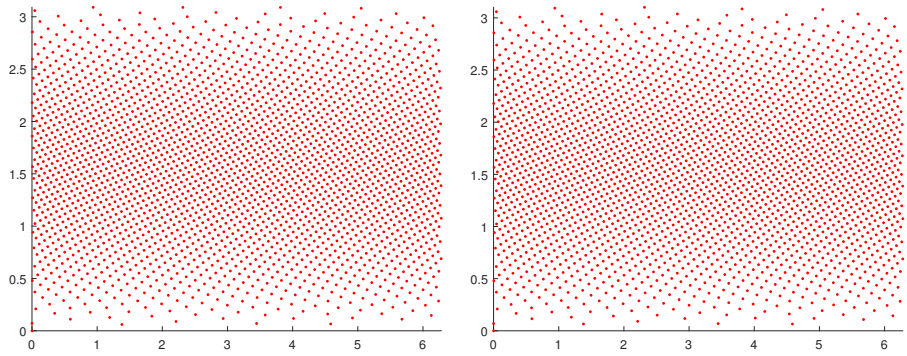
3.1 Polynomial spaces projection

First, we consider the orthogonal projection on Π_t. Given a spherical data $\{(x_i, f(x_i)) \,|\, x_i \in X_N\}$ from some spherical signal $f : \mathbb{S}^2 \to \mathbb{R}$ in $L_2(\mathbb{S}^2)$ and a spherical t-design X_N obtained above, the spherical signal f can be projected on Π_t through orthogonal projection. That is, $f = f_0 + f_1$ as

$$f(x_i) = \sum_{\ell=0}^{t} \sum_{m=-\ell}^{\ell} \hat{c}_\ell^m Y_\ell^m(\theta_i, \phi_i) + f_1(x_i), \tag{9}$$



(a) SP point set 3D. Left: Initial. Right: Final



(b) SP point set 2D. Left: Initial. Right: Final

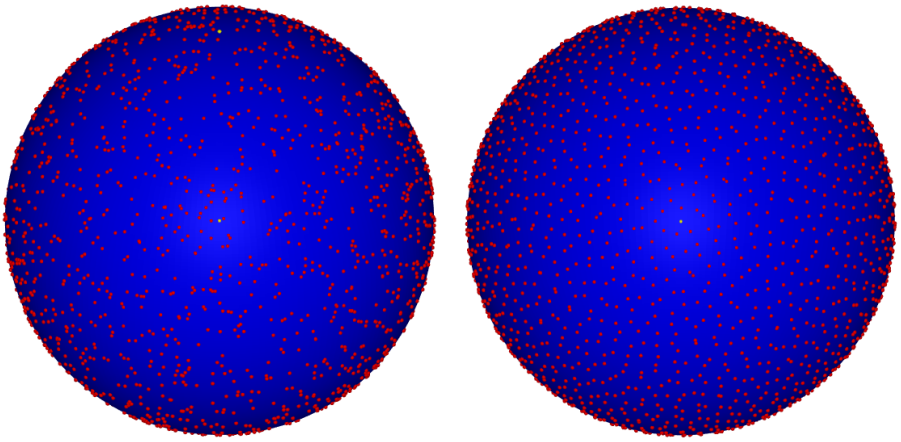
Fig. 1 Numerical simulation of spherical initial point sets (left column) as input point sets and related spherical t-design point sets (right column) as output point sets based on TR-PCG for $t = 50$ and $N = (t+1)^2$ on $\mathbb{S}^2$. (a) on sphere. (b) equirectangular projection.

where $f_0(x) = \sum_{\ell=0}^{t} \sum_{m=-\ell}^{\ell} \hat{c}_\ell^m Y_\ell^m(\theta, \phi) \in \Pi_t$, and the residual $f_1 \notin \Pi_t$. Here we let $\hat{f}_\ell^m := \langle f, Y_\ell^m \rangle$. To obtain f_0, i.e., the coefficient set $\hat{\mathbf{c}} = \{\hat{c}_\ell^m : \ell = 0, \ldots, t, |m| \leq \ell\}$, we can use the least squares method through the minimization problem

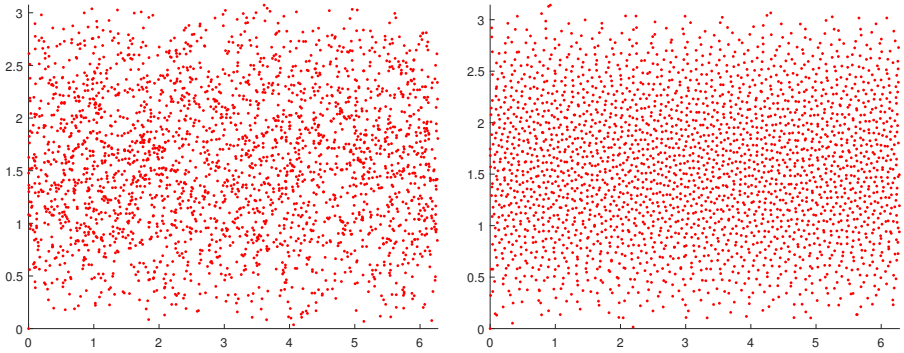
$$\min_{\mathbf{f}_0 \in \Pi_t} \|\mathbf{f} - \mathbf{f}_0\|, \tag{10}$$

where $\mathbf{f} := f|_{X_N}$ and $\mathbf{f}_0 := f_0|_{X_N}$ are the vector representations of f and f_0 on X_N, respectively. To solve (10), by using matrix representation $\mathbf{f}_0 = \mathbf{Y}_t \hat{\mathbf{c}}$, where $\mathbf{Y}_t$ is the $N \times (t+1)^2$ matrix determined by $(Y_\ell^m(x_i))_{x_i \in X_N; |\ell| \leq t, |m| \leq \ell}$, the least squares problem in (10) can be solved through the following equation:

$$\mathbf{Y}_t^*(\mathbf{w} \odot \mathbf{Y}_t \hat{\mathbf{c}}) = \mathbf{Y}_t^*(\mathbf{w} \odot \mathbf{f}) \tag{11}$$



(a) UD point set 3D. Left: Initial. Right: Final



(b) UD point set 2D. Left: Initial. Right: Final

Fig. 2 Numerical simulation of spherical initial point sets (left column) as input point sets and resulted spherical t-design point sets (right column) as output point sets based on TR-PCG for $t = 50$ and $N = (t+1)^2$ on $\mathbb{S}^2$. (a): on sphere. (b): equirectangular projection.

with wight $\mathbf{w}$, where $\odot$ is the Hadamard entrywise product operator. It can be done by using the CG method for the equation $Ax = b$ with $A = \mathbf{Y}_t^* \mathrm{diag}(\mathbf{w})\mathbf{Y}_t$, $x = \hat{\mathbf{c}}$ and $b = \mathbf{Y}_t^*(\mathbf{w} \odot \mathbf{f})$. See Algorithm 3. For convenience, we acquiesce weight function $W(X_N) := \mathbf{w}$ in corresponding operation $(\mathbf{Y}_t^* \cdot)$ on arbitrary point set X_N in Algorithm 3.

3.2 Approximation of spherical functions obtained from Wendland functions

Next, we consider the approximation of some spherical functions using polynomials with different point sets. The functions are the combinations of normalized Wendland functions from a family of compactly supported radial basis functions (RBF). Let

Algorithm 3 Projection by Conjugate Gradient Algorithm

Input: t: polynomial degree; X_N: spherical point set; W: weight function of X_N; $K_{\max}$: maximum iterations; ε: termination tolerance;
Initialize $x = 0$, $k = 0$, $r_0 = \mathbf{Y}_t^*(\mathbf{w} \odot \mathbf{f})$, $A = \mathbf{Y}_t \mathrm{diag}(\mathbf{w}) \mathbf{Y}_t$.

while $\|r_{k+1}\| > \varepsilon$ and $k \le K_{\max}$ **do**
  if $k = 0$ **then**
    $p_1 = r_0$
  else
    $p_{k+1} = r_k + \frac{\|r_k\|^2}{\|r_{k-1}\|^2} p_k$
  end if
  compute step size $\alpha = \frac{\|r_k\|^2}{p_{k+1}^\top A p_{k+1}}$
  $x_{k+1} = x_k + \alpha p_{k+1}$ and $r_{k+1} = r_k - \alpha A p_{k+1}$
  $k = k + 1$
end while

Output: $x^* \in \mathbb{R}^n$.

$(t)_+ := \max\{t, 0\}$ for $t \in \mathbb{R}$. The original Wendland functions are

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

The normalized (equal area) Wendland functions are

$$\phi_k(t) := \tilde{\phi}_k\Big(\frac{t}{\delta_k}\Big), \quad \delta_k := \frac{(3k+3)\Gamma(k+\frac{1}{2})}{2\,\Gamma(k+1)}, \quad k \ge 0.$$

The Wendland functions $\phi_k(t)$ pointwise converge to the Guassian when $k \to \infty$, refer to [48]. 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 [49]

$$f_k(\boldsymbol{x}) := \sum_{i=1}^{6} \phi_k(\|\boldsymbol{z}_i - \boldsymbol{x}\|), \quad k \ge 0. \tag{12}$$

The paper [49] has proved that $f_k \in H^{k+\frac{3}{2}}(\mathbb{S}^2)$, where $H^\sigma(\mathbb{S}^2) := \{f \in L_2(\mathbb{S}^2) : \sum_{\ell=0}^{\infty} \sum_{|m| \le \ell} (1+\ell)^{2\sigma} |\hat{f}_\ell^m|^2 < \infty\}$ is the Sobolev space with the smooth parameter $\sigma > 1$. 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 approximate at these locations, especially for small k.

Given a point set X_N with weight $\mathbf{w}$ (not necessary quadrature point sets), we can sample f_k on X_N as a spherical signal $v : X_N \to \mathbb{R}$, and compute the projection v_1 and residual $r = v - v_1$ based on Algorithm 3, where we set the maximum iterations $K_{\max} = 1000$ and termination tolerance $\varepsilon = 2.2204\text{e-}16$ (floating-point relative accuracy of MATLAB). To check whether the point sets have good approximation properties for such functions, we use the relative projection L_2-error defined by

$$err(v_1, v) = \frac{\|v - v_1\|}{\|v\|} = \frac{\|r\|}{\|v\|}.$$

We show the results in Table 3 under the setting of input polynomial degree $T = \frac{t}{2}$ and weight $W = \mathbf{w}$ on Algorithm 3, and set input polynomial degree $T = \frac{t}{2}$ and weight $W = \sqrt{\mathbf{w}}$ on Algorithm 3 to obtain Table 4. For comparing the initial points, we set SP and UD with equal weight $\mathbf{w} \equiv \frac{4\pi}{N}$. The spherical design point sets are equal-weight quadrature rules. Also in Tables 3 and 4, we give the results of spherical t-designs with degree $t \approx 200, 400$ and $N \approx (t+1)^2$ in different initial points: Spiral points (SP), uniformly distributed points (UD), Icosahedron vertices mesh points (IV), and HEALpix points (HL). ((SP, SPD), (UD, SUD), (IV, SID), and (HL, SHD) are the initial point set and final spherical t-design point set via TR-PCG, respectively). SID means generated by IV, and SHD means generated by HL) For more details on these point sets, we refer to [13]. We shall mention that the paper [13] has shown the partial results for $t \approx 200$ in Table 3.

Table 3 Relative L_2-errors $err(f_T, f_k)$ for Wendland functions $f_0, \ldots, f_4$ approximated by Π_T functions under the setting $T = \frac{t}{2}$ and $W = \mathbf{w}$ on Algorithm 3.

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
400	160801	SP	1.49E-04	2.05E-07	1.47E-09	1.33E-09	1.45E-09
400	160801	SPD	1.49E-04	2.06E-07	8.61E-10	7.39E-12	2.28E-12
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
400	160801	UD	1.22E-04	1.86E-07	8.07E-08	8.55E-08	9.02E-08
400	160801	SUD	1.68E-04	2.19E-07	1.23E-09	9.33E-10	1.02E-09
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
403	163842	IV	2.05E-04	2.14E-07	3.98E-09	3.88E-09	5.59E-08
403	163842	SID	1.80E-04	2.18E-07	8.81E-10	7.46E-12	2.29E-12
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
442	196608	HL	1.46E-04	1.45E-07	8.19E-10	3.94E-10	3.11E-10
442	196608	SHD	1.46E-04	1.45E-07	4.89E-10	4.07E-12	2.39E-12

Table 4 Relative L_2-errors $err(f_T, f_k)$ for Wendland functions $f_0, \ldots, f_4$ approximated by Π_T functions under the setting $T = \frac{t}{2}$ and $W = \sqrt{\mathbf{w}}$ on Algorithm 3.

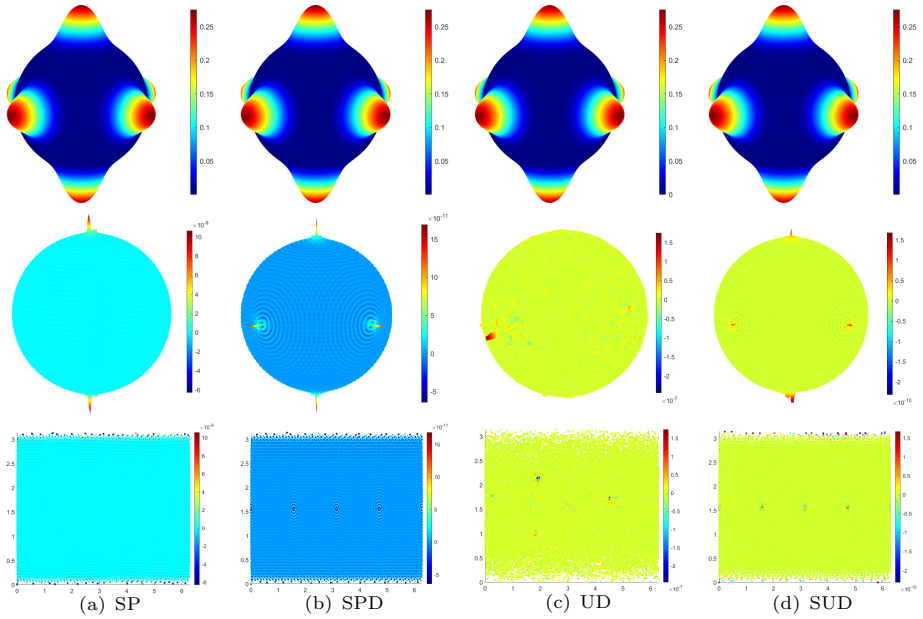
t	N	Q_N	f_0	f_1	f_2	f_3	f_4
200	40401	SP	5.64E-04	3.19E-06	5.24E-08	1.69E-09	9.34E-11
200	40401	SPD	5.78E-04	3.20E-06	5.25E-08	1.69E-09	8.92E-11
400	160801	SP	1.49E-04	2.05E-07	8.58E-10	1.41E-11	1.32E-11
400	160801	SPD	1.49E-04	2.06E-07	8.61E-10	7.39E-12	2.25E-12
200	40401	UD	6.09E-04	3.07E-06	4.80E-08	1.89E-09	1.46E-09
200	40401	SUD	6.99E-04	3.51E-06	5.63E-08	1.79E-09	1.07E-10
400	160801	UD	1.22E-04	1.76E-07	9.04E-10	6.74E-10	6.49E-10
400	160801	SUD	1.68E-04	2.19E-07	8.95E-10	7.35E-12	1.34E-12
201	40962	IV	8.12E-04	3.32E-06	5.26E-08	1.71E-09	2.95E-10
201	40962	SID	6.15E-04	3.11E-06	5.08E-08	1.64E-09	8.74E-11
403	163842	IV	2.05E-04	2.14E-07	9.01E-10	2.45E-10	2.47E-10
403	163842	SID	1.80E-04	2.18E-07	8.81E-10	7.46E-12	2.29E-12
220	49152	HL	5.98E-04	2.28E-06	3.04E-08	8.12E-10	5.82E-11
220	49152	SHD	5.97E-04	2.28E-06	3.04E-08	8.11E-10	3.59E-11
442	196608	HL	1.46E-04	1.45E-07	4.89E-10	3.94E-10	3.11E-10
442	196608	SHD	1.46E-04	1.45E-07	4.89E-10	4.07E-12	2.38E-12

From Tables 3 and 4, all point sets give good approximation error but the point sets with spherical t-design properties give a higher order of approximations. We display the related figures of the projection v_1 and residual r of above point sets for RBF f_4 with degree $t \approx 200$ under the setting of Table 3 in Figures 3 and 4. Comparing to related tables and figures, we can see that projection errors of the initial point sets under the setting of $T = \frac{t}{2}$ and $W = \sqrt{\mathbf{w}}$ are smaller than the setting of $T = \frac{t}{2}$ and $W = \mathbf{w}$, whereas the corresponding spherical t-designs point sets are almost the same (at about 1E-14 difference, but the results of $W = \sqrt{\mathbf{w}}$ are a bit smaller) except SUD at $t = 400$ showed that when $W = \sqrt{\mathbf{w}}$ provides the smaller projection error.

3.3 Denoising of Wendland functions by Projection

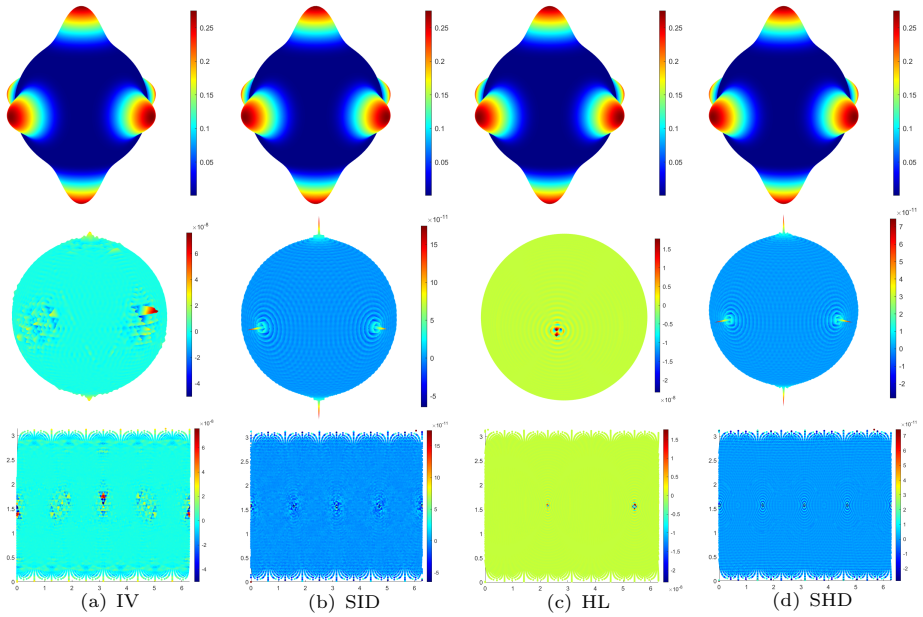
We then consider adding noise on Wendland function f_k to observe the best polynomial degree for projecting noisy function on $\mathbb{S}^2$ since we have known the property of f_k. We generate Wendland function f_k on the spherical t-design point set X_N with $t = 400, N = (t+1)^2$ from SPD. Let $f_{k,\sigma}(X_N) = f_k(X_N) + G_\sigma(X_N)$ be the noisy Wendland function generated by Gaussian white noise G_σ with noise level $\sigma|f_k|_{\max}$. By applying Algorithm 3 under the setting of maximum iterations $K_{\max} = 1000$ and termination tolerance $\varepsilon = 2.2204\text{e-}16$ (floating-point relative accuracy of MATLAB), and choosing projection degree t from 1 to 50, we have $f_{k,\sigma} = f + g$, where $f \in \Pi_t$ and $g \notin \Pi_t$. Consider σ taking from 0.05 to 0.2 with stepsize 0.025, then we test the projection error of f_k on $\mathbb{S}^2$ for $k = 0, 1, 2, 3, 4$. We show relative projection errors for functions f_k and f in Figure 5 and give the minimum error value for f_k and f in Table 5.

We can see that the projection errors for f_k and f descend rapidly and reach the minimum value at certain degrees, then climb up slowly compared with the rate of descent. The projection errors for $f_{k,\sigma}$ and f also descend rapidly for the same degree



(a) SP (b) SPD (c) UD (d) SUD

Fig. 3 Numerical simulation of real part of Projection term (first row), residual term (middle row), and the equirectangular projection of the residual (last row) for RBF f_4 under the setting $T = \frac{t}{2}$ and $W = \mathbf{w}$ on Algorithm 3 for $t = 200, N = (t+1)^2$.



(a) IV (b) SID (c) HL (d) SHD

Fig. 4 Numerical simulation of the real part of Projection term (first row), the residual term (middle row), and the equirectangular projection of the residual (last row) for RBF f_4 under the setting $T = \frac{t}{2}$ and $W = \mathbf{w}$ on Algorithm 3 for $t \approx 200, N \approx (t+1)^2$.

but will converge to 0 slowly as the degree is sufficiently large. From Table 5, we obtain the best projection degrees for projecting f_k from noisy function $f_{k,\sigma}$, which mostly fall on 12 to 16. Thus, we can choose the appropriate spherical t-design point sets with degree t at about 12 to 16.

Table 5 The minimum relative projection L_2-error $err(f, f_k)$ along with corresponding degree t (inside the parentheses) in different noise level σ.

σ	f_0	f_1	f_2	f_3	f_4
0.05	1.28E-02(24)	7.40E-03(16)	7.60E-03(16)	8.40E-03(16)	9.31E-03(16)
0.075	1.68E-02(24)	1.05E-02(16)	1.07E-02(14)	1.20E-02(14)	1.37E-02(16)
0.1	2.03E-02(20)	1.30E-02(14)	1.28E-02(12)	1.53E-02(14)	1.82E-02(16)
0.125	2.38E-02(20)	1.49E-02(12)	1.51E-02(12)	1.84E-02(12)	2.19E-02(14)
0.15	2.64E-02(16)	1.69E-02(12)	1.75E-02(12)	2.08E-02(12)	2.55E-02(14)
0.175	2.91E-02(16)	1.89E-02(12)	2.00E-02(12)	2.34E-02(12)	2.92E-02(14)
0.2	3.19E-02(16)	2.11E-02(12)	2.25E-02(12)	2.61E-02(12)	3.29E-02(14)

3.4 Approximation of non-smooth functions

Finally, instead of considering differentiable functions in Section 3.2, we consider here non-smooth functions approximation with spherical designs. We define a *Spherical* $\mathbf{1}_{\mathcal{D}}$ *function* as follows

$$f_{\mathbf{1}_{\mathcal{D}}} := \begin{cases} 1, & \boldsymbol{x} \in \mathcal{D}, \\ 0, & \boldsymbol{x} \notin \mathcal{D}, \end{cases}$$

where $\mathcal{D} := \{(\theta, \phi) \,|\, \theta \in [0, \frac{\pi}{2}], \phi \in [0, 2\pi); (\theta, \phi) \in \mathbb{S}^2\}$ is the northern hemisphere. Similarly, we set $T = \frac{t}{2}$ and weight $W = \mathbf{w}$, and $T = \frac{t}{2}$ and weight $W = \sqrt{\mathbf{w}}$ on Algorithm 3 to obtain Table 6. We can see from Table 6, that the higher degree t, the lower the error of the projection of spherical $1_{\mathcal{D}}$ function. Besides, there is no significant difference between the two choices of weights. In addition, only a little refinement of the error for point set SPD with degree $t = 400$ and SID with $t = 201, 403$ than their corresponding initial point sets. We also display the example of projection in Figure 6. One can see that the discontinuity of the function $f_{\mathbf{1}_{\mathcal{D}}}$ along the equator brings the main challenge for the polynomial approximation.

4 Spherical framelets for spherical signal processing using spherical designs

In this section, we discuss the further application of spherical designs in spherical framelet constructions and spherical signal processing. In general, given a group of spherical design point sets, for a noisy function f_σ with Gaussian white noise of noise level $\sigma|f|_{\max}$, following the orthogonal projection procedure of Section 3.1, we have $f_\sigma = f + g$, where the projected function $f \in \Pi_{t_J}$ and residual $g \notin \Pi_{t_J}$. Notice that both projected function f and residual g have noise. Then we use spherical t-design point sets to decompose f by semi-discrete truncated spherical tight framelet system

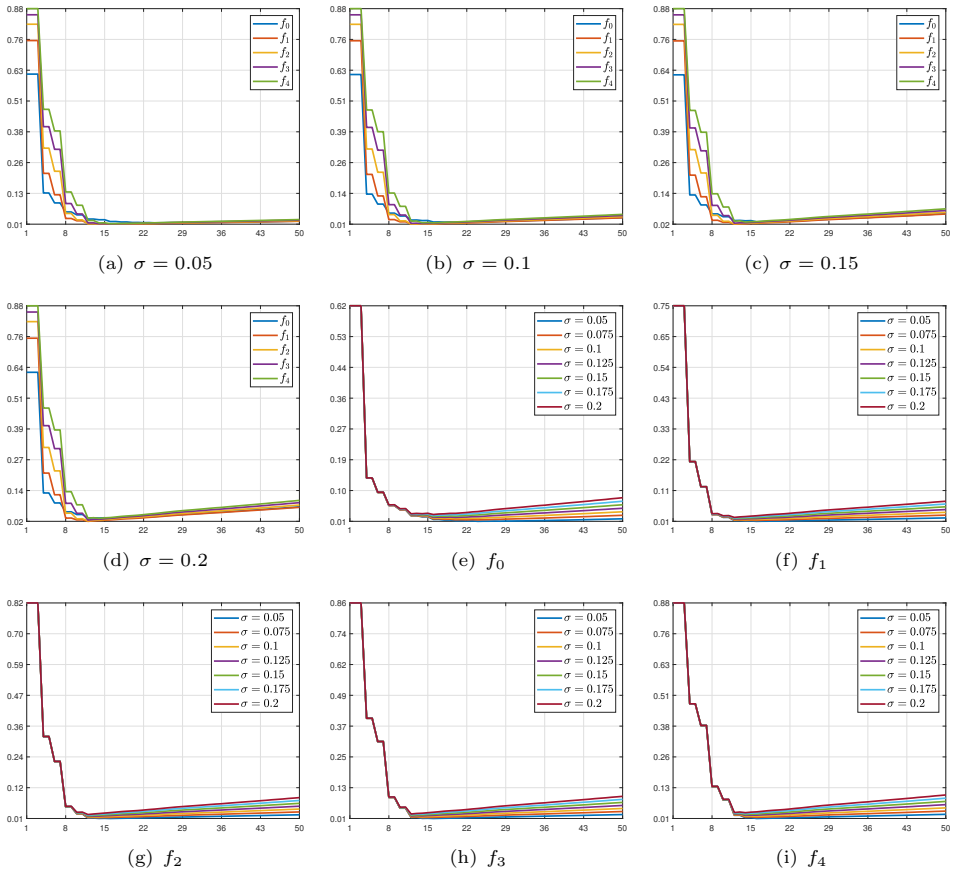


Fig. 5 Relative projection L_2-error with $err(f, f_k)$ for $k = 0, 1, 2, 3, 4$ in different noise level σ. (a)-(d) are the extracted view from $\sigma = 0.05, 0.1, 0.15, 0.2$, $(e) - (i)$ are focusing on Wendland function f_k for $k = 0, 1, 2, 3, 4$.

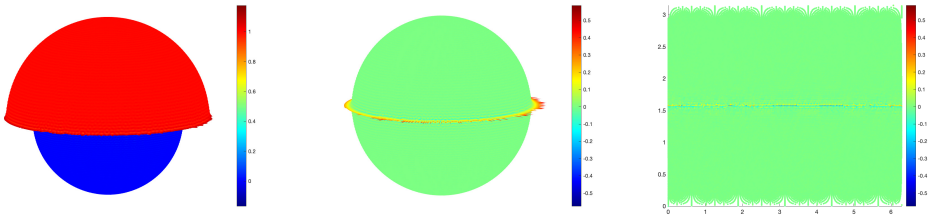
$\mathcal{F}_{J_0}^{J}(\eta, \mathcal{Q})$. After that, we use threshold methods based on spherical caps to denoise the framelet coefficients of f, and then we reconstruct the function by the denoised coefficients to obtain f_{thr}. Similarly, g is also being denosed to obtain g_{thr}. Finally, we get the denoised function $F_{thr} = f_{thr} + g_{thr}$. Below, we introduce the truncated spherical framelet systems, the spherical caps, the thresholding techniques, and the denoising experiments on Wendland functions.

4.1 Semi-discrete spherical framelet systems

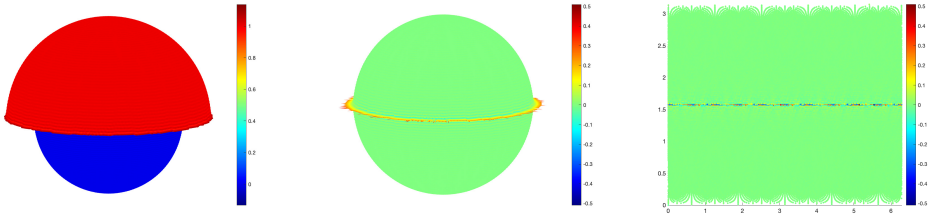
First, we briefly introduce the semi-discrete spherical framelet systems based on spherical harmonics and spherical design point sets.

Table 6 Relative L_2-errors $err(f_w, f_{\mathbf{1}_\mathcal{D}})$ and $err(f_{\sqrt{w}}, f_{\mathbf{1}_\mathcal{D}})$ for projecting spherical $1_\mathcal{D}$ function. f_w and $f_{\sqrt{w}}$ represent the relative error with functions on point sets along with corresponding degree $T = \frac{t}{2}$ take weights $W = \mathbf{w}$ and $W = \sqrt{\mathbf{w}}$, respectively on Algorithm 3.

t	N	Q_N	f_w	$f_{\sqrt{w}}$
200	40401	SP	5.46E-02	5.46E-02
200	40401	SPD	5.46E-02	5.46E-02
400	160801	SP	3.89E-02	3.89E-02
400	160801	SPD	3.88E-02	3.88E-02
200	40401	UD	4.85E-02	4.85E-02
200	40401	SUD	5.21E-02	5.21E-02
400	160801	UD	3.52E-02	3.52E-02
400	160801	SUD	3.68E-02	3.68E-02
201	40962	IV	5.50E-02	5.50E-02
201	40962	SID	5.40E-02	5.40E-02
403	163842	IV	3.84E-02	3.84E-02
403	163842	SID	3.65E-02	3.65E-02
220	49152	HL	5.07E-02	5.07E-02
220	49152	SHD	5.07E-02	5.07E-02
442	196608	HL	3.56E-02	3.56E-02
442	196608	SHD	3.56E-02	3.56E-02



(a) IV



(b) SID

Fig. 6 Numerical simulation of real part of Projection term (left column), residual term (middle column), and the equirectangular projection of the residual (right column) for spherical $1_\mathcal{D}$ function under the setting $T = \frac{t}{2}$ and $W = \sqrt{\mathbf{w}}$ on Algorithm 3 for $t = 403, N = 163842$.

We start with a filter bank $\eta = \{a; b_1, \dots, b_n\} \subset l_1(\mathbb{Z}) := \{h = \{h_k\}_{k\in\mathbb{Z}} \subset \mathbb{C} \,|\, \sum_{k\in\mathbb{Z}} |h_k| < \infty\}$ satisfying the partition of unity condition:

$$|\hat{a}(\xi)|^2 + \sum_{s\in[n]} |\hat{b}_s(\xi)|^2 = 1, \quad \xi \in \mathbb{R}. \tag{13}$$

where for a filter (mask) $h = \{h_k\}_{k\in\mathbb{Z}} \subset \mathbb{C}$, its Fourier series $\hat{h}$ is given by $\hat{h}(\xi) := \sum_{k\in\mathbb{Z}} h_k \mathrm{e}^{-2\pi \mathrm{i} k\xi}$, for $\xi \in \mathbb{R}$, and for a positive integer n, we denote $[n] := \{1, \dots, n\}$.

The *quadrature (cubature) rule* $Q_{N_j} = (X_{N_j}, \mathbf{w}_j)$ on $\mathbb{S}^2$ at scale j is a collection 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. A 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 denote a polynomial-exact quadrature rule of degree t_j as $Q_{N_j} =: Q_{N_j,t_j}$. The spherical t-design $X_N = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ forms a polynomial-exact quadrature rule $Q_{N,t} := (X_N, \mathbf{w})$ of degree t with weight $\mathbf{w} \equiv \frac{4\pi}{N}$. We consider $\mathcal{Q} := \mathcal{Q}_{J_0}^{J+1} := \{Q_{N_j,t_j} : j = J_0, \dots, J+1\}$ be a set of polynomial-exact quadrature rules satisfying $t_{j+1} = 2t_j$.

For a function $f \in L^1(\mathbb{R})$, its Fourier transform $\hat{f}$ is defined by $\hat{f}(\xi) := \int_{\mathbb{R}} f(x) e^{-2\pi \mathrm{i} x\xi} dx$. For a fixed fine scale $J \in \mathbb{Z}$, we set

$$\hat{\alpha}^{(J+1)}\left(\frac{\ell}{t_{J+1}}\right) = \begin{cases} 1 & \text{for } \ell \le t_J, \\ \\ 0 & \text{for } \ell > t_J, \end{cases} \tag{14}$$

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

$$\hat{\alpha}^{(j)}\left(\frac{\ell}{t_j}\right) = \hat{\alpha}^{(j)}\left(2\frac{\ell}{t_{j+1}}\right) = \hat{a}\left(\frac{\ell}{t_{j+1}}\right) \hat{\alpha}^{(j+1)}\left(\frac{\ell}{t_{j+1}}\right), \tag{15}$$

$$\hat{\beta}_s^{(j)}\left(\frac{\ell}{t_j}\right) = \hat{\beta}_s^{(j)}\left(2\frac{\ell}{t_{j+1}}\right) = \hat{b}_s\left(\frac{\ell}{t_{j+1}}\right) \hat{\alpha}^{(j+1)}\left(\frac{\ell}{t_{j+1}}\right), \quad s \in [n], \tag{16}$$

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

$$\Psi = \{\alpha^{(j)}, \beta_s^{(j)} \,|\, j = J_0, \dots, J; s \in [n]\}. \tag{17}$$

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

$$\mathcal{F}_{J_0}^J(\eta, \mathcal{Q}) := \{\varphi_{J_0,k} \,|\, k \in [N_{J_0}]\} \cup \{\psi_{j,k}^{(s)} \,|\, k \in [N_{j+1}], s \in [n]\}_{j=J_0}^J, \tag{18}$$

where $\varphi_{j,k}$ and $\psi_{j,k}^{(s)}$ are defined by

$$\varphi_{j,k}(\boldsymbol{x}) := \sqrt{w_j} \sum_{(\ell,m)\in\mathcal{I}_{t_j}} \hat{\alpha}^{(j)}\left(\frac{\ell}{t_j}\right) \overline{Y_\ell^m(\boldsymbol{x}_{j,k})} Y_\ell^m(\boldsymbol{x}), \tag{19}$$

$$\psi_{j,k}^{(s)}(\boldsymbol{x}) := \sqrt{w_{j+1}} \sum_{(\ell,m)\in\mathcal{I}_{t_{j+1}}} \hat{\beta}_s^{(j)} \left(\frac{\ell}{t_j}\right) \overline{Y_\ell^m(\boldsymbol{x}_{j+1,k})} Y_\ell^m(\boldsymbol{x}). \tag{20}$$

The truncated system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$ is completely determined by the filter bank η and the quadrature rules $\mathcal{Q}$. It was shown in [13, Theorem 4.3] that $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$ 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 $v_{j,k} := \langle f, \varphi_{j,k}\rangle_{L_2(\mathbb{S}^2)}$ and $w_{j,k}^{(s)} := \langle f, \psi_{j,k}^{(s)}\rangle_{L_2(\mathbb{S}^2)}$.

Therefore, we can use the spherical tight framelet system to decompose f into framelet coefficient sequences $\{v_{j,k}; w_{j,k}^{(s)} : s = 1, \ldots, n\}$. We then apply the thresholding techniques for denoising the framelet coefficient sequence.

4.2 Spherical caps

Given a point set $X_N \subset \mathbb{S}^2$, for each point $\boldsymbol{x} \in X_N$, We want to get geodesic information for the connection of points on X_{N_j}. Naturally, we consider the spherical cap. In one way, we can generate index sets of $\boldsymbol{x}$ based on *radius nearest neighbor (rnn)* $\mathcal{C}_R(\boldsymbol{x})$. To get this, we first define a spherical cap $C(\boldsymbol{x}, r) \subset \mathbb{S}^2$ centred at $\boldsymbol{x} \in \mathbb{S}^2$ with radius $r \in \mathbb{N}_0$ by

$$C(\boldsymbol{x}, r) := \{\boldsymbol{y} \in \mathbb{S}^2 : \|\boldsymbol{x} \times \boldsymbol{y}\| \le r\},$$

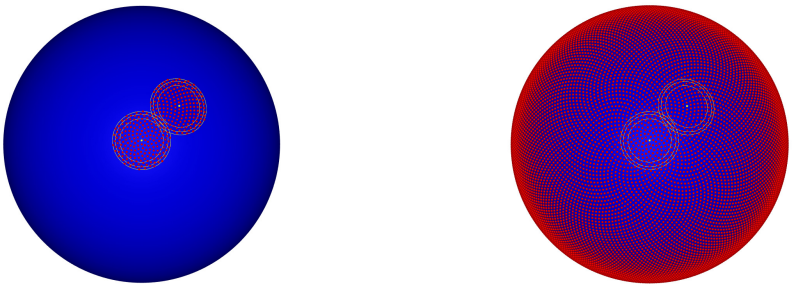
where $\|\boldsymbol{x} \times \boldsymbol{y}\|$ is the magnitude of the cross product of $\boldsymbol{x}$ and $\boldsymbol{y}$. Then the boundary of $C(\boldsymbol{x}, r)$ is

$$\partial C(\boldsymbol{x}, r) := \{\boldsymbol{y} \in \mathbb{S}^2 : \|\boldsymbol{x} \times \boldsymbol{y}\| = r\}.$$

Then for each point $\boldsymbol{x} \in X_N$, we have $C(\boldsymbol{x}, r) \cap X_N = \{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_s}\}$ being the s points in X_N that are within the cap $C(\boldsymbol{x}, r)$. We use $\mathcal{C}_R(\boldsymbol{x}) := \{i_1, \ldots, i_s\}$ to denote such an index set for the rnn of $\boldsymbol{x}$. Furthermore, we use $\mathcal{C}_R(X_N) = \{\mathcal{C}_R(\boldsymbol{x}) : \boldsymbol{x} \in X_N\}$ to denote the collection of all such index sets. We show Figure 7 an example of a spherical cap and the point set inside it.

Additionally, we can generate index sets of $\boldsymbol{x} \in X_N$ based on k *points nearest neighbor (knn)* $\mathcal{C}_K(\boldsymbol{x})$ by finding k nearest points $\{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_k}\} := C(\boldsymbol{x}, k) \subset X_N$ that are most closed to $\boldsymbol{x}$. We can use $\mathcal{C}_K(\boldsymbol{x}) := \{i_1, \ldots, i_k\}$ to denote the corresponding index set for x. Similarly, the index set collection $\mathcal{C}_K(X_N) := \{\mathcal{C}_K(\boldsymbol{x}) : \boldsymbol{x} \in X_N\}$.

The neighborhood point sets given by $C(\boldsymbol{x}, r)$ or $C(\boldsymbol{x}, k)$ can be used in fine-tuned denoising for spherical signal processing.



(a) Partial view (b) General view

Fig. 7 Spherical caps (rnn-based) of a spherical 128-design point set $X_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ for $N = (t+1)^2$ and $t = 128$. Here we select two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_{600}$ and choose three different radius for each of the two points. Left: Partial view (points outside caps are not shown). Right: General view (points within and outside caps are shown).

4.3 Thresholding techniques

Given a noisy signal f_σ with Gaussian white noise $N(0, \sigma^2)$, we have $f_\sigma = f + g$ with $f \in \Pi_{t_J}$ and residual $g \notin \Pi_{t_J}$. Given the truncated spherical framelet system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$ as in (18), since $f \in \Pi_{t_J}$, we have $f = \sum_{k=1}^{N_0} v_{J_0,k}\varphi_{J_0,k} + \sum_{j=J_0}^{J}\sum_{k=1}^{N_j}\sum_{s=1}^{n} w_{j,k}^{(s)}\psi_{j,k}^{(s)}$, with coefficients $v_{j,k} := \langle f, \varphi_{j,k}\rangle$ and $w_{j,k}^{(s)} := \langle f, \psi_{j,k}^{(s)}\rangle$.

For f, the coefficients need to be normalized first. we compute the energy norm $\|\varphi_{j,k}\|_{L_2(\mathbb{S}^2)}$ and $\|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}$ for $s = 1, \ldots, n$, then the coefficients become

$$\tilde{v}_{j,k} = \frac{v_{j,k}}{\|\varphi_{j,k}\|_{L_2(\mathbb{S}^2)}}, \quad \tilde{w}_{j,k}^{(s)} = \frac{w_{j,k}^{(s)}}{\|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}}, \quad s = 1, \ldots, n.$$

Second, the coefficients can be processed with various thresholding techniques. There are four ways for thresholding in this paper: Global hard (GH) and global soft (GS) thresholding techniques and local hard (LH) and local soft thresholding techniques. We represent $\{\breve{v}_{j,k}; \breve{w}_{j,k}^{(s)} : s = 1, \ldots, n\}$ as global thresholds and $\{\breve{v}_{j,k}; \breve{w}_{j,k}^{(s)} : s = 1, \ldots, n\}$ as local thresholds.

The global hard threshold (GH) and the global soft (GS) threshold are given by

$$\breve{w}_{j,k}^{(s)} = \begin{cases} \tilde{w}_{j,k}^{(s)}, & |\tilde{w}_{j,k}^{(s)}| \geq \tau, \\ 0, & |\tilde{w}_{j,k}^{(s)}| < \tau, \end{cases} \breve{v}_{j,k} = \begin{cases} \tilde{v}_{j,k}, & |\tilde{v}_{j,k}| \geq \tau, \\ 0, & |\tilde{v}_{j,k}| < \tau, \end{cases} \tag{21}$$

and

$$\breve{w}_{j,k}^{(s)} = \begin{cases} \tilde{w}_{j,k}^{(s)} - \operatorname{sgn}(\tilde{w}_{j,k}^{(s)})\sigma, & |\tilde{w}_{j,k}^{(s)}| \geq \tau, \\ 0, & |\tilde{w}_{j,k}^{(s)}| < \tau, \end{cases} \breve{v}_{j,k} = \begin{cases} \tilde{v}_{j,k} - \operatorname{sgn}(\tilde{v}_{j,k})\sigma, & |\tilde{v}_{j,k}| \geq \tau, \\ 0, & |\tilde{v}_{j,k}| < \tau, \end{cases} \tag{22}$$

respectively, where $\tau = c\sigma$ is the threshold value and c is a constant.

The local hard (LH) spherical cap threshold and local soft (LS) spherical cap threshold are introduced below. For each $\boldsymbol{x}_k^j \in X_{N_j}$, let it be the centroid of a spherical cap $C(\boldsymbol{x}_k^j, m) \subset \mathbb{S}^2$, where m could be radius or number of points. We can find point set $\mathcal{X}_M(\boldsymbol{x}_k^j) := \mathcal{X}_{M_j^k} \subset C(\boldsymbol{x}_k^j, m)$, where elements $\boldsymbol{x} \in \mathcal{X}_{M_j^k}$ are selected in X_{N_j} by the index of $\mathcal{C}_m(X_{N_j})$, we assume that there are M_j^k points in $\mathcal{X}_{M_j^k}$. Now we have the coefficients on spherical cap to $\boldsymbol{x}_k^j$, which are $\{\tilde{v}_{j,i} : \boldsymbol{x}_i \in \mathcal{X}_{M_j^k}\}$ and $\{\tilde{\boldsymbol{w}}_{j,i}^{(1)}, \ldots, \tilde{\boldsymbol{w}}_{j,i}^{(n)} : \boldsymbol{x}_i \in \mathcal{X}_{M_{j+1}^k}\}$, then we averaging each coefficients, which are

$$\bar{\boldsymbol{w}}_{j,k}^{(s)} = \frac{1}{M_{j+1}^k} \sum_{i=1}^{M_{j+1}^k} |\tilde{\boldsymbol{w}}_{j,i}^{(s)}|^2, \quad s = 1, \ldots, n,\ \boldsymbol{x}_i \in \mathcal{X}_{M_{j+1}^k} \tag{23}$$

$$\bar{v}_{j,k} = \frac{1}{M_j^k} \sum_{i=1}^{M_j^k} |\tilde{v}_{j,i}|^2, \quad \boldsymbol{x}_i \in \mathcal{X}_{M_j^k}. \tag{24}$$

We define

$$\sigma_{j,k}^{(s)} = \sqrt{\max\{\bar{\boldsymbol{w}}_{j,k}^{(s)} - \sigma^2, 0\}}, \quad s = 1, \ldots, n, \tag{25}$$

$$\sigma'_{j,k} = \sqrt{\max\{\bar{v}_{j,k} - \sigma^2, 0\}}. \tag{26}$$

Then we do the threshold value estimation, which is

$$\tau_{j,k}^{(s)} = \frac{c\sigma^2}{\sigma_{j,k}^{(s)}}, \quad s = 1, \ldots, n, \tag{27}$$

$$\tau'_{j,k} = \frac{c\sigma^2}{\sigma'_{j,k}}, \tag{28}$$

where c is a constant. The local hard (LH) shrinkage is given by

$$\check{\boldsymbol{w}}_{j,k}^{(s)} = \begin{cases} \tilde{\boldsymbol{w}}_{j,k}^{(s)}, & |\tilde{\boldsymbol{w}}_{j,k}^{(s)}| \geq \tau_{j,k}^{(s)}, \\ 0, & |\tilde{\boldsymbol{w}}_{j,k}^{(s)}| < \tau_{j,k}^{(s)}, \end{cases} \tag{29}$$

$$\check{v}_{j,k} = \begin{cases} \tilde{v}_{j,k}, & |\tilde{v}_{j,k}| \geq \tau'_{j,k}, \\ 0, & |\tilde{v}_{j,k}| < \tau'_{j,k}. \end{cases} \tag{30}$$

Similarly, the shrinkage for local soft (LS) thresholds is given by

$$\check{\boldsymbol{w}}_{j,k}^{(s)} = \begin{cases} \tilde{\boldsymbol{w}}_{j,k}^{(s)} - \operatorname{sgn}(\tilde{\boldsymbol{w}}_{j,k}^{(s)})\tau_{j,k}^{(s)}, & |\tilde{\boldsymbol{w}}_{j,k}^{(s)}| \geq \tau_{j,k}^{(s)}, \\ 0, & |\tilde{\boldsymbol{w}}_{j,k}^{(s)}| < \tau_{j,k}^{(s)}, \end{cases} \tag{31}$$

$$\check{v}_{j,k} = \begin{cases} \tilde{v}_{j,k} - \operatorname{sgn}(\tilde{v}_{j,k})\tau'_{j,k}, & |\tilde{v}_{j,k}| \geq \tau'_{j,k}, \\ 0, & |\tilde{v}_{j,k}| < \tau'_{j,k}. \end{cases} \tag{32}$$

Finally, we need to denormalized the coefficients, which is

$$\dot{\boldsymbol{w}}_{j,k}^{(s)} = \mathring{\boldsymbol{w}}_{j,k}^{(s)} \|\psi_{j,k}^{(s)}\|_{L_2(\mathbb{S}^2)}, \quad s = 1, \ldots, n, \tag{33}$$

$$\dot{v}_{j,k} = \mathring{v}_{j,k} \|\varphi_{j,k}\|_{L_2(\mathbb{S}^2)}, \tag{34}$$

where $\{\mathring{v}_{j,k}; \mathring{\boldsymbol{w}}_{j,k}^{(1)}, \ldots, \mathring{\boldsymbol{w}}_{j,k}^{(n)}\}$ is one of thresholds (global hard, global soft, local hard and local soft) we mentioned above. Thus, we obtain new coefficients $\{\dot{v}_{j,k}; \dot{\boldsymbol{w}}_{j,k}^{(1)}, \ldots, \dot{\boldsymbol{w}}_{j,k}^{(n)}\}$, which has been denoised by local soft spherical cap threshold.

Following the procedure of the spherical framelet system, we reconstruct the function with $\{\dot{v}_{j,k}; \dot{\boldsymbol{w}}_{j,k}^{(1)}, \ldots, \dot{\boldsymbol{w}}_{j,k}^{(n)}\}$, which is

$$f_{thr}(\boldsymbol{x}) = \sum_{k=1}^{N_j} \dot{v}_{j,k}\varphi_{j,k} + \sum_{k=1}^{N_{j+1}} \sum_{s=1}^{n} \dot{\boldsymbol{w}}_{j,k}^{(s)} \psi_{j,k}^{(s)}. \tag{35}$$

For g, on the point set $\boldsymbol{x}_k^j \in X_{N_j}$, both the hard and soft thresholds are similar to f. That is, the global hard (GH) threshold is given as

$$g_{\tau_1}(\boldsymbol{x}_k^j) = \begin{cases} g(\boldsymbol{x}_k^j), & |g(\boldsymbol{x}_k^j)| \geq \tau_1, \\ 0, & |g(\boldsymbol{x}_k^j)| < \tau_1, \end{cases} \tag{36}$$

and the global soft (GS) threshold is given as

$$g_{\tau_1}(\boldsymbol{x}_k^j) = \begin{cases} g(\boldsymbol{x}_k^j) - \operatorname{sgn}(g(\boldsymbol{x}_k^j))\sigma, & |g(\boldsymbol{x}_k^j)| \geq \tau_1, \\ 0, & |g(\boldsymbol{x}_k^j)| < \tau_1, \end{cases} \tag{37}$$

where $\tau_1 = c_1\sigma$ is the threshold value and c_1 is a constant. The local hard (LH) spherical cap threshold for $\boldsymbol{x}_k^j \in X_{N_j}$ uses $\mathcal{X}_{M_j^k} \subset C(\boldsymbol{x}_k^j, m)$, which is

$$g_{thr}(\boldsymbol{x}_k^j) = \begin{cases} g(\boldsymbol{x}_k^j), & |g(\boldsymbol{x}_k^j)| \geq \tau_{j,k}, \\ 0, & |g(\boldsymbol{x}_k^j)| < \tau_{j,k}, \end{cases} \tag{38}$$

where $\tau_{j,k} = \frac{c_1\sigma^2}{\sqrt{\max\{\bar{g}(\boldsymbol{x}_k^j) - \sigma^2, 0\}}}$ and $\bar{g}(\boldsymbol{x}_k^j)$ with the form

$$\bar{g}(\boldsymbol{x}_k^j) = \frac{1}{M_j^k} \sum_{i=1}^{M_j^k} |g(\boldsymbol{x}_i^j)|^2, \quad \boldsymbol{x}_i^j \in \mathcal{X}_{M_j^k}, \tag{39}$$

while the local soft (LS) spherical cap threshold for $\boldsymbol{x}_k^j \in X_{N_j}$ uses $\mathcal{X}_{M_j^k} \subset C(\boldsymbol{x}_k^j, m)$, which is

$$g_{thr}(\boldsymbol{x}_k^j) = \begin{cases} g(\boldsymbol{x}_k^j) - \operatorname{sgn}(g(\boldsymbol{x}_k^j))\tau_{j,k}, & |g(\boldsymbol{x}_k^j)| \geq \tau_{j,k}, \\ 0, & |g(\boldsymbol{x}_k^j)| < \tau_{j,k}. \end{cases} \tag{40}$$

Hence, we obtain g_{thr} after local soft thresholding.

4.4 Denoising of Wendland function

In this section, we give some experiments to illustrate the efficiency of denoising. That is, after projecting noisy function f_σ with noise level σ on Π_t space on spherical t-design point sets, we have $f_\sigma = f + g$. We use spherical t-design point sets along with its equal weight quadrature rules for framelet decomposition and reconstruction on $\mathbb{S}^2$. Finally, we apply the thresholds introduced in Section 4.3 for denoising. Therefore, we have a new function $F_{thr} = f_{thr} + g_{thr}$.

Based on the spherical t-design point sets (collected by TR-PCG method) in [13] and the experiments of the best polynomial degrees of projection errors of Wendland functions in Section 3.2, we choose spherical t_j-design point sets with degree $t_j = 16, 32, 64$ both in SPD and SUD, $t_j = 11, 24, 49$ in SID, and $t_j = 12, 26, 54$ in SHD ($j = 1, 2, 3$). Then we use X_{N_J} to project noisy data $f_\sigma = f_4 + G_\sigma$ (generated by normalized Wendland function f_4 in (12) and Gaussian white noise G_σ with noise level $\sigma|f_4|_{\max}$ and $\sigma = [0.05 : 0.025 : 0.2]$) on Π_{t_J} space with maximum degree t_J ($J = 3$) by Algorithm 3 under the setting of maximum iterations $K_{\max} = 1000$ and termination tolerance $\varepsilon = 2.2204\text{e-}16$ (floating-point relative accuracy of MATLAB) to obtain $f_\sigma = f + g$, then decompose and reconstruct using a 2-level spherical framelet system $\mathcal{F}_{J_0}^J(\eta, \mathcal{Q})$ with four kinds of thresholds for denoising: global hard (GH), global soft (GS), local hard (LH) and local soft (LS).

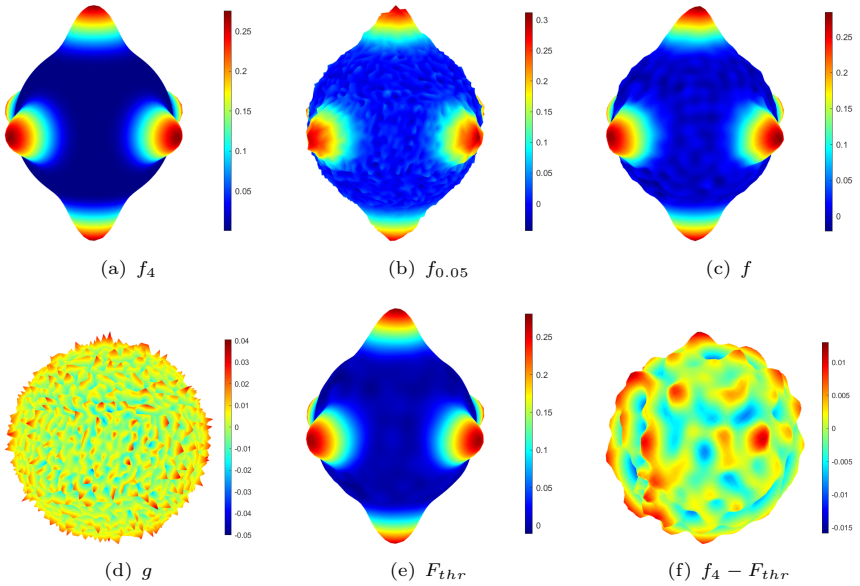
We use SNR for measuring original signal f_4 and the rest of noise $F_{thr} - f_4$, which is $\mathrm{SNR}(f_4, F_{thr}) := 10 \log_{10}(\frac{\|f_4\|}{\|F_{thr} - f_4\|})$ in different filter banks η_k for $k = 1, 2, 3$ (see [13, Section 5.1]), and different point sets X_{N_j} corresponding with degree t_j. For finding the suitable constants in Section 4.1, we do a lot experiments by changing the values of c and c_1 to see the variation of $\mathrm{SNR}(f_4, F_{thr})$, we give a general view for the behavior of SNR in using η_3 at $\sigma = 0.05$ in Figure 9. Then we set $c = 2.5, c_1 = 3$ in GH and LH, $c = 1, c_1 = 3$ in GS and LS. For local thresholds, the spherical cap we take cap layer orders $i = 15, 22, 27$ for filters η_1, η_2, η_3, respectively. Notice that all the settings of thresholds may not obtain the best values of $\mathrm{SNR}(f_4, F_{thr})$, but are the mildest values. We show the results in Table 7 for different initial point sets of spherical t-designs. In tables, we can see: SNR_0 are the $\mathrm{SNR}(f_4, f_\sigma)$ for different σ values; For each thresholding method, the first row to the third row are: $\mathrm{SNR}(f_4, F_{thr})$ values of using η_1, η_2, η_3, respectively.

As we can see from the tables, local threshold methods (especially local soft (LS)) are more beneficial than global thresholds, which are reasonable in construction. The behavior of different filters is $\eta_3 > \eta_2 > \eta_1$, which means the more high pass filters, the better behavior in denoising. One interesting discovery is that the initial icosahedron

Table 7 Wendland f_4 denoising results

Q_{N_j}	Thr	σ	0.05	0.075	0.1	0.125	0.15	0.175	0.2
SPD	SNR_0		**13.63**	**10.11**	**7.61**	**5.67**	**4.09**	**2.75**	**1.59**
	GH	η_1	18.51	15.72	13.90	13.08	12.50	11.99	11.51
		η_2	21.50	17.64	15.11	13.89	13.23	12.74	12.28
		η_3	23.25	19.13	15.75	14.31	13.55	13.09	12.67
	GS	η_1	19.10	16.45	14.65	13.26	12.13	11.21	10.43
		η_2	20.50	17.48	15.43	13.89	12.66	11.64	10.80
		η_3	21.29	18.09	15.90	14.24	12.93	11.86	10.98
	LH	η_1	19.94	16.97	15.29	14.29	13.80	12.84	12.27
		η_2	22.80	19.53	17.43	15.91	14.55	13.28	12.81
		η_3	24.36	21.04	18.82	17.15	15.52	13.89	**13.21**
	LS	η_1	20.67	18.06	16.42	15.21	14.19	13.24	12.31
		η_2	23.11	20.05	18.03	16.47	15.18	14.02	12.88
		η_3	**24.48**	**21.25**	**19.03**	**17.30**	**15.82**	**14.49**	13.19
SUD	SNR_0		**13.63**	**10.11**	**7.61**	**5.67**	**4.09**	**2.75**	**1.59**
	GH	η_1	18.56	15.81	13.69	12.77	12.15	11.58	10.87
		η_2	21.57	17.80	14.74	13.61	13.07	12.50	11.90
		η_3	22.97	19.27	15.30	14.07	13.47	12.92	12.34
	GS	η_1	19.05	16.29	14.47	13.07	11.91	10.96	10.17
		η_2	20.44	17.40	15.30	13.71	12.45	11.43	10.58
		η_3	21.15	17.92	15.73	14.07	12.75	11.68	10.81
	LH	η_1	19.77	16.85	15.39	14.49	13.92	13.03	12.49
		η_2	22.51	19.34	17.25	15.82	14.51	13.36	12.68
		η_3	23.91	20.48	18.36	16.84	15.36	13.84	**13.04**
	LS	η_1	20.70	18.02	16.39	15.23	14.22	13.20	12.18
		η_2	22.98	19.98	17.95	16.38	15.09	13.90	12.70
		η_3	**24.15**	**20.90**	**18.73**	**17.07**	**15.63**	**14.29**	12.97
SID	SNR_0		**13.63**	**10.11**	**7.61**	**5.67**	**4.09**	**2.75**	**1.59**
	GH	η_1	18.61	14.60	12.60	11.26	9.35	7.69	7.10
		η_2	22.22	18.39	15.40	12.96	10.18	8.32	7.69
		η_3	23.51	20.02	16.94	14.18	10.92	8.69	7.96
	GS	η_1	17.07	14.04	12.04	10.57	9.41	8.43	7.58
		η_2	18.62	15.37	13.12	11.42	10.06	8.94	7.98
		η_3	19.25	15.94	13.62	11.84	10.42	9.24	8.26
	LH	η_1	19.92	16.44	14.03	12.65	11.61	10.88	10.25
		η_2	23.44	19.98	17.43	15.78	14.30	13.10	11.94
		η_3	**24.66**	**21.21**	**18.82**	**17.00**	**15.39**	**14.02**	**12.79**
	LS	η_1	20.13	16.86	14.70	13.15	11.91	10.82	9.77
		η_2	23.34	19.90	17.44	15.51	13.90	12.43	11.05
		η_3	24.54	21.03	18.51	16.46	14.68	13.09	11.59
SHD	SNR_0		**13.58**	**10.05**	**7.55**	**5.62**	**4.03**	**2.69**	**1.53**
	GH	η_1	17.83	14.74	12.29	10.17	8.97	8.14	7.52
		η_2	21.19	17.42	14.05	11.19	9.70	8.81	8.10
		η_3	22.70	18.63	15.29	11.95	10.09	9.16	8.41
	GS	η_1	17.02	14.04	12.03	10.52	9.32	8.33	7.51
		η_2	18.36	15.11	12.88	11.21	9.87	8.79	7.91
		η_3	18.93	15.61	13.31	11.57	10.18	9.04	8.11
	LH	η_1	19.54	15.86	14.02	12.60	11.40	10.32	9.34
		η_2	22.14	18.56	16.06	14.40	13.09	11.77	10.55
		η_3	**23.23**	19.69	17.16	15.25	13.85	**12.55**	**11.25**
	LS	η_1	19.91	16.68	14.64	13.07	11.76	10.60	9.50
		η_2	22.21	18.80	16.42	14.57	13.00	11.59	10.28
		η_3	23.21	**19.70**	**17.22**	**15.26**	**13.59**	12.07	10.64

point sets of spherical t-designs (SID) starting from $t = 49$ (the smallest degree as finest scale) are adapted with the local hard threshold (LH) with good performance for noise level with $\sigma = 0.05$ to 0.1. Moreover, the initial spiral point sets of spherical t-designs (SPD) are reliable in more general applications since we can generate it for what we require with various relations of t and N. Figur 8 shows the behavior of denoised and reconstructed functions.



(a) f_4 (b) $f_{0.05}$ (c) f

(d) g (e) F_{thr} (f) $f_4 - F_{thr}$

Fig. 8 The behavior of denoising Wendland $f_{0.05}$ by η_3 with LS on SPD with $t_j = 16, 32, 64$

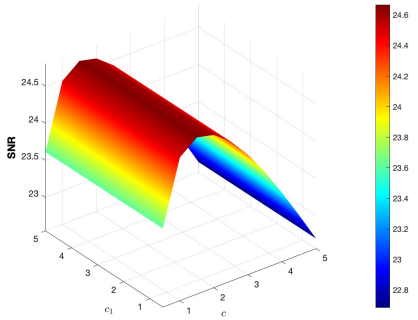


Fig. 9 The general view for the behavior under different setting of c and c_1 in denoising $f_{0.05}$ by η_3 on SPD for $t_j = 64, 32, 16$ at cap layer order $i = 27$.

5 Conclusions

In this paper, we present the line search with the restart conjugate gradient method (LS-RCG) for obtaining numerical spherical t-design point sets. We also compare the different approaches (LS-RCG and TR-PCG, full Hessian $\mathcal{H}$ and approximation Hessian $\mathcal{H}_2$) from the results of their numerical spherical t-designs. In general, the LS-RCG consumes a lot in the line search iterations, whereas the TR-PCG costs fewer iterations in finding the trust regions. However, there are no significant differences in the accuracy of full Hessian and approximation Hessian. Then, we study the approximation of spherical functions (Wendland functions with certain smoothness) and non-smooth functions (spherical $1_{\mathcal{D}}$ function) by projecting them with orthogonal polynomial using spherical t-designs point sets on $\mathbb{S}^2$. The results show that choosing different weights and different types of spherical t-designs will affect the error accuracy of projects. Moreover, we introduce the spherical signal processing based on the best projection degree of polynomials in Wendland functions. We use spherical framelets in [13] from spherical designs under four different types of thresholding techniques which the fine-tuned spherical caps may demand. We show that all the point sets perform well under the spherical framelets in the denoising procedure, indicating the spherical designs have a great potential in spherical data processing.

Acknowledgments. The authors thank the anonymous reviewers for their constructive comments and valuable suggestions that greatly help the improvement of the quality of this article. This work was supported in part by the Research Grants Council of Hong Kong (Project no. CityU 11309122, CityU 11302023, and CityU 11301224) and a grant from the Innovation and Technology Commission of Hong Kong (Project no. MHP/054/22).

Declarations

- Funding. This work was supported in part by the Research Grants Council of Hong Kong (Project no. CityU 11309122, CityU 11302023, and CityU 11301224) and a grant from the Innovation and Technology Commission of Hong Kong (Project no. MHP/054/22).
- Conflict of interest/Competing interests. The authors declare that they have no conflict of interest and have no competing interests.
- Ethics approval and consent to participate. The authors claim the submitted work is original and does not have been published elsewhere in any form or language.
- Consent for publication. The authors understand that Springer Nature reserves the right to remove an article, chapter, book, or other content from its online platforms at any time.
- Data availability. Not applicable.
- Materials availability. Not applicable.
- Code availability. Not applicable.
- Author contribution. The authors state that they contribute equally to this article.

References

[1] Deng, J., Russakovsky, O., Krause, J., Bernstein, M.S., Berg, A., Fei-Fei, L.: Scalable multi-label annotation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3099–3102 (2014)

[2] Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM **35**(12), 61–70 (1992)

[3] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26** (2013)

[4] Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical cnns. In: International Conference on Learning Representations (2018)

[5] Esteves, C., Allen-Blanchette, C., Makadia, A., Daniilidis, K.: Learning so (3) equivariant representations with spherical cnns. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 52–68 (2018)

[6] Kondor, R., Lin, Z., Trivedi, S.: Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. Advances in Neural Information Processing Systems **31** (2018)

[7] Fang, Z., Feng, H., Huang, S., Zhou, D.-X.: Theory of deep convolutional neural networks ii: Spherical analysis. Neural Networks **131**, 154–162 (2020)

[8] Li, J., Feng, H., Zhuang, X.: Convolutional neural networks for spherical signal processing via area-regular spherical haar tight framelets. IEEE Transactions on Neural Networks and Learning Systems (2022)

[9] Delsarte, P., Goethals, J., Seidel, J.: Spherical codes and designs. Geometriae Dedicata **6**(3), 363–388 (1977)

[10] Bannai, E., Bannai, E.: A survey on spherical designs and algebraic combinatorics on spheres. European Journal of Combinatorics **30**(6), 1392–1425 (2009)

[11] Chen, X., Womersley, R.S.: Spherical designs and nonconvex minimization for recovery of sparse signals on the sphere. SIAM Journal on Imaging Sciences **11**(2), 1390–1415 (2018)

[12] Wang, Y.G., Zhuang, X.: Tight framelets and fast framelet filter bank transforms on manifolds. Applied and Computational Harmonic Analysis **48**(1), 64–95 (2020)

[13] Xiao, Y., Zhuang, X.: Spherical framelets from spherical designs. SIAM Journal on Imaging Sciences **16**(4), 2072–2104 (2023) https://doi.org/10.1137/22M1542362

[14] Zhuang, X., Xiao, Y.: Denoising on sphere via large spherical t-designs and spherical framelets. In: 2023 International Conference on Sampling Theory and Applications (SampTA), pp. 1–5 (2023). https://doi.org/10.1109/SampTA59647.2023.10301196

[15] Seymour, P., Zaslavsky, T.: Averaging sets: a generalization of mean values and spherical designs. Advances in Mathematics **52**(3), 213–240 (1984)

[16] Wagner, G., Volkmann, B.: On averaging sets. Monatshefte für Mathematik **111**(1), 69–78 (1991)

[17] Korevaar, J., Meyers, J.: Spherical faraday cage for the case of equal point charges and chebyshev-type quadrature on the sphere. Integral Transforms and Special Functions **1**(2), 105–117 (1993)

[18] Bondarenko, A., Radchenko, D., Viazovska, M.: Optimal asymptotic bounds for spherical designs. Annals of mathematics, 443–452 (2013)

[19] Bondarenko, A., Radchenko, D., Viazovska, M.: Well-separated spherical designs. Constructive approximation **41**(1), 93–112 (2015)

[20] Hardin, R.H., Sloane, N.J.: Mclaren's improved snub cube and other new spherical designs in three dimensions. Discrete & Computational Geometry **15**(4), 429–441 (1996)

[21] Maier, U.: Numerical calculation of spherical designs. Advances in Multivariate Approximation, 213–226 (1999)

[22] Chen, X., Womersley, R.S.: Existence of solutions to systems of underdetermined equations and spherical designs. SIAM Journal on Numerical Analysis **44**(6), 2326–2341 (2006)

[23] Sloan, I.H., Womersley, R.S.: A variational characterisation of spherical designs. Journal of Approximation Theory **159**(2), 308–318 (2009)

[24] Chen, X., Frommer, A., Lang, B.: Computational existence proofs for spherical t-designs. Numerische Mathematik **117**(2), 289–305 (2011)

[25] Le Gia, Q.T., Mhaskar, H.N.: Localized linear polynomial operators and quadrature formulas on the sphere. SIAM Journal on Numerical Analysis, 440–466 (2008)

[26] Mhaskar, H., Narcowich, F., Ward, J.: Spherical marcinkiewicz-zygmund inequalities and positive quadrature. Mathematics of computation **70**(235), 1113–1130 (2001)

[27] Womersley, R.S.: Efficient spherical designs with good geometric properties. In: Contemporary Computational Mathematics-A Celebration of the 80th Birthday

of Ian Sloan, pp. 1243–1285. Springer, Cham, Switzerland (2018)

[28] An, C., Xiao, Y.: Numerical construction of spherical t-designs by barzilai-borwein method. Applied Numerical Mathematics **150**, 295–302 (2020)

[29] An, C., Chen, X., Sloan, I.H., Womersley, R.S.: Well conditioned spherical designs for integration and interpolation on the two-sphere. SIAM journal on numerical analysis **48**(6), 2135–2157 (2010)

[30] Zheng, R., Zhuang, X.: On the existence and estimates of nested spherical designs. Applied and Computational Harmonic Analysis **73**, 101672 (2024) https://doi.org/10.1016/j.acha.2024.101672

[31] Gräf, M., Potts, D.: On the computation of spherical designs by a new optimization approach based on fast spherical fourier transforms. Numerische Mathematik **119**(4), 699–724 (2011)

[32] Sun, W., Yuan, Y.-X.: Optimization Theory and Methods: Nonlinear Programming vol. 1. Springer, New York, NY (2006)

[33] Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pacific journal of Optimization **2**(1), 35–58 (2006)

[34] Shewchuk, J.R., et al.: An introduction to the conjugate gradient method without the agonizing pain. Carnegie-Mellon University. Department of Computer Science Pittsburgh (1994)

[35] Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving. Journal of research of the National Bureau of Standards **49**(6), 409–436 (1952)

[36] Crowder, H., Wolfe, P.: Linear convergence of the conjugate gradient method. IBM Journal of Research and Development **16**(4), 431–433 (1972)

[37] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York, NY (1999)

[38] Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. The computer journal **7**(2), 149–154 (1964)

[39] Daniel, J.W.: The conjugate gradient method for linear and nonlinear operator equations. SIAM Journal on Numerical Analysis **4**(1), 10–26 (1967)

[40] Powell, M.J.D.: Restart procedures for the conjugate gradient method. Mathematical programming **12**, 241–254 (1977)

[41] Dai, Y.-H., Liao, L.-Z., Li, D.: On restart procedures for the conjugate gradient method. Numerical Algorithms **35**, 249–260 (2004)

[42] Kunis, S., Potts, D.: Fast spherical fourier algorithms. Journal of Computational and Applied Mathematics **161**(1), 75–98 (2003)

[43] Potts, D., Steidl, G., Tasche, M.: Fast fourier transforms for nonequispaced data: A tutorial. Modern sampling theory, 247–270 (2001)

[44] Ware, A.F.: Fast approximate fourier transforms for irregularly spaced data. SIAM review **40**(4), 838–856 (1998)

[45] Keiner, J., Potts, D.: Fast evaluation of quadrature formulae on the sphere. Mathematics of computation **77**(261), 397–419 (2008)

[46] Swinbank, R., James Purser, R.: Fibonacci grids: A novel approach to global modelling. Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography **132**(619), 1769–1793 (2006)

[47] Weisstein, E.W.: Sphere Point Picking. From MathWorld–A Wolfram Web Resource (2004). https://mathworld.wolfram.com/SpherePointPicking.html

[48] Chernih, A., Sloan, I.H., Womersley, R.S.: Wendland functions with increasing smoothness converge to a gaussian. Advances in Computational Mathematics **40**(1), 185–200 (2014)

[49] Gia, Q.L., Sloan, I.H., Wendland, H.: Multiscale analysis in sobolev spaces on the sphere. SIAM journal on numerical analysis **48**(6), 2065–2090 (2010)