On the computation of spherical designs by a new optimization approach based on fast spherical Fourier transforms

Manuel Gräf Daniel Potts

Spherical t-designs are point sets $\mathcal{X}_M := \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\} \subset \mathbb{S}^2$ which provide quadrature rules with equal weights for the sphere which are exact for polynomials up to degree t. In this paper we consider the problem of finding numerical spherical t-designs on the sphere $\mathbb{S}^2$ for high polynomial degree $t \in \mathbb{N}$. That is, we compute numerically local minimizers of a certain quadrature error $A_t(\mathcal{X}_M)$. The quadrature error A_t was also used for a variational characterization of spherical t-designs by Sloan and Womersley in [25]. For the minimization problem we regard several nonlinear optimization methods on manifolds, like Newton and conjugate gradient methods. We show that by means of the nonequispaced fast spherical Fourier transforms we perform gradient and Hessian evaluations in $\mathcal{O}(t^2 \log t + M \log^2(1/\epsilon))$ arithmetic operations, where $\epsilon > 0$ is a prescribed accuracy. Using these methods we present numerical spherical t-designs for $t \leq 1000$, even in the case $M \approx \frac{1}{2}t^2$.

Math Subject Classifications. 65T40, 65K10 53B21 49M15, 33C55,

Keywords and Phrases. spherical designs, variational characterization, optimization methods on Riemannian manifolds, spherical harmonics, iterative methods, nonequispaced Fourier methods on the sphere.

1 Introduction

Distributing points on the unit sphere $\mathbb{S}^2$ in the Euclidean space $\mathbb{R}^3$ in some optimal sense is a challenging problem, cf. [24]. In this paper, we consider the concept of spherical t-designs, which was introduced by Delsarte, Goethals and Seidel [6] in 1977. There a spherical t-design on $\mathbb{S}^2$ is defined as a finite set $\mathcal{X}_M = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\} \subset \mathbb{S}^2$ satisfying

$$\int_{\mathbb{S}^2} p(\boldsymbol{x}) \mathrm{d}\mu_{\mathbb{S}^2}(\boldsymbol{x}) = \frac{4\pi}{M} \sum_{i=1}^{M} p(\boldsymbol{x}_i), \qquad \text{for all } p \in \Pi_t(\mathbb{S}^2), \tag{1.1}$$

Chemnitz University of Technology, Department of Mathematics, 09107 Chemnitz, Germany, {m.graef,potts}@mathematik.tu-chemnitz.de

where $\mu_{\mathbb{S}^2}$ is the surface measure on $\mathbb{S}^2$ and $\Pi_t(\mathbb{S}^2)$ is the space of all spherical polynomials with degree at most t. Such point sets provide equal weights quadrature formulae on the sphere $\mathbb{S}^2$, which have many applications. In the Hilbert space $\Pi_t(\mathbb{S}^2)$ with standard inner product the worst case quadrature error for the point set $\mathcal{X}_M$ is defined by

$$E_t(\mathcal{X}_M) := \sup_{\substack{p\in\Pi_t(\mathbb{S}^2)\\ \|p\|_2\leq 1}} \left| \int_{\mathbb{S}^2} p(\boldsymbol{x}) \mathrm{d}\mu_{\mathbb{S}^2}(\boldsymbol{x}) - \frac{4\pi}{M} \sum_{i=1}^{M} p(\boldsymbol{x}_i) \right|.$$

For the general setting of quadrature errors in reproducing kernel Hilbert spaces we refer to cf. [23]. Of course, a spherical t-design $\mathcal{X}_M$ is a global minimum of the worst case quadrature error with $E_t(\mathcal{X}_M) = 0$, cf. (1.1). In [25] the authors presents a variational characterization of spherical t-designs which involves a squared quadrature error

$$A_t(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M) := \frac{1}{M^2} \sum_{n=1}^{t} \sum_{k=-n}^{n} \left| \sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right|^2 = \left(\frac{1}{4\pi} E_t(\mathcal{X}_M) \right)^2,$$

where Y_n^k are the spherical harmonics of degree n and order k. In this paper we are interested in finding numerical spherical t-designs, i.e., we compute point sets $\mathcal{X}_M$, such that $A_t(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M) \leq \varepsilon$, where ε is a given accuracy, say $\varepsilon = 1\mathrm{e}-10$. We present optimization algorithms on Riemannian manifolds for attacking this highly nonlinear and nonconvex minimization problem. The proposed methods make use of fast spherical Fourier transforms, which where already successfully applied in [17, 12] for solving high dimensional linear equation systems on the sphere.

It is commonly conjectured that spherical t-designs with $M \approx \frac{1}{2}t^2$ points exists, but there is no proof. Recently, a weaker conjecture was proved in [2], where the authors show the existence of spherical t-designs with $M > ct^2$ points for some unknown constant $c > 0$. Moreover, in [3] it was verified that for $t = 1, \ldots, 100$, spherical t-designs with $(t+1)^2$ points exist, using the characterization of fundamental spherical t-designs and interval arithmetic. For further recent developments in the research of spherical t-designs and related topics we refer to the very nice survey article [1]. We emphasize that the construction of spherical t-designs is a serious challenge even for small polynomial degrees t. The function A_t has many local minima and its hard to decide if a local minimum is a global one. Moreover, even if one computes a point set in a neighborhood of a global one it is hard to decide if its function value is zero or not, due to numerical errors. From this point of view we are satisfied with numerical spherical t-designs which have integration error E_t in the range of machine precision, since such quadrature points are suitable for numerical quadrature on the sphere.

The outline of this paper is as follows. In Section 2 we present the necessary tools. That is we give a brief introduction to Riemannian geometry on the sphere $\mathbb{S}^2$ and the M times product manifold $\mathbb{S}^2 \times \cdots \times \mathbb{S}^2$ in order to describe optimization methods on these manifolds. Afterwards, we define the nonequispaced spherical Fourier transforms and comment on the fast realization. In Section 3 we combine the optimization method with the nonequispaced spherical Fourier transform and show that each iteration step is realized very efficiently with the help of fast spherical Fourier transforms. Finally, we compare the proposed optimization methods and present numerical results in Section 4, where we compute numerically spherical t-designs for $t = 10, 100, 200, 500, 1000$.

2 Prerequisites

The purpose of this section is to define the necessary notations for calculations on Riemannian manifolds in order to describe the optimization algorithms presented in Section 2.3. We are especially interested in the geometry of the sphere $\mathbb{S}^2$ and its M times products $\mathbb{S}^2_M := \mathbb{S}^2 \times \cdots \times \mathbb{S}^2$. So the general Riemannian manifold $\mathcal{M}$ with Riemannian metric $g_{\mathcal{M}}$ holds as substitute for these manifolds. By the famous embedding theorem of Nash every Riemannian manifold $(\mathcal{M}, g_{\mathcal{M}})$ can be seen as a sub-manifold of some d-dimensional Euclidean space $\mathbb{R}^d$. Hence, we use the more extrinsic, but also more intuitive approach to Riemannian geometry. In the following we introduce the general concepts using the example of the sphere. In Section 2.2 we briefly summarize the notations for the product manifold $\mathbb{S}^2_M$. Finally we present in Section 2.4 the basics of the nonequispaced fast spherical Fourier transform.

2.1 Riemannian geometry on the sphere $\mathbb{S}^2$

As a sub-manifold the sphere is embedded in the three-dimensional Euclidean space $\mathbb{R}^3$ by

$$\mathbb{S}^2 := \left\{ \boldsymbol{x} := (x, y, z)^\top \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1 \right\}.$$

From this embedding we obtain a natural understanding of the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$ at a point $\boldsymbol{x} \in \mathbb{S}^2$. It is simply given by the orthogonal complement of the linear subspace $\operatorname{span}\{\boldsymbol{x}\}$, i.e.,

$$\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2 := \{\boldsymbol{v} \in \mathbb{R}^3 : \langle \boldsymbol{v}, \boldsymbol{x} \rangle = 0\},$$

where $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$ is the standard inner product and the induced norm is given by $\|\boldsymbol{x}\|_2 := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$, $\boldsymbol{x} \in \mathbb{R}^3$.

Then the sphere possesses a natural Riemannian metric from the given embedding, which is induced by the Riemannian metric of the ambient space $\mathbb{R}^3$, cf. Figure 2.1. Thus, the Riemannian metric $g_{\mathbb{S}^2} : \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2 \times \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2 \to \mathbb{R}$ is given on the sphere for all $\boldsymbol{x} \in \mathbb{S}^2$ by

$$g_{\mathbb{S}^2}(\boldsymbol{v}, \boldsymbol{w}) := \langle \boldsymbol{v}, \boldsymbol{w} \rangle, \qquad \boldsymbol{v}, \boldsymbol{w} \in \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2.$$

In this metric the geodesic distance between two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^2$ calculates from

$$\mathrm{d}_{\mathbb{S}^2}(\boldsymbol{x}, \boldsymbol{y}) := \arccos\left(\langle \boldsymbol{x}, \boldsymbol{y} \rangle\right).$$

In the following all functions or vector fields are arbitrarily often differentiable. Since, we consider the sphere $\mathbb{S}^2$ as an embedding in the space $\mathbb{R}^3$ it is most natural to consider functions f on the sphere as restrictions of functions $\tilde{f}$ on $\mathbb{R}^3$. The same counts for vector fields $\boldsymbol{X} : \mathbb{S}^2 \to \mathrm{T}_{(\cdot)}\mathbb{S}^2$, where $\tilde{\boldsymbol{X}} : \mathbb{R}^3 \to \mathbb{R}^3$ is an extension of $\boldsymbol{X}$ to the ambient space $\mathbb{R}^3$. From this point of view we easily define the common differential operators on the sphere as restriction of differential operators on $\mathbb{R}^3$. Therefor we introduce for all $\boldsymbol{x} \in \mathbb{S}^2$ the orthogonal projection operator $\mathrm{P}_{\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2} : \mathbb{R}^3 \to \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$ by

$$\mathrm{P}_{\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2}(\boldsymbol{v}) := \boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{x} \rangle \boldsymbol{x}, \qquad \boldsymbol{v} \in \mathbb{R}^3.$$

The spherical gradient $\nabla_{\mathbb{S}^2} f$ of the function f reads as

$$\nabla_{\mathbb{S}^2} f(\boldsymbol{x}) := \mathrm{P}_{\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2}\left(\nabla \tilde{f}(\boldsymbol{x})\right), \qquad \boldsymbol{x} \in \mathbb{S}^2,$$

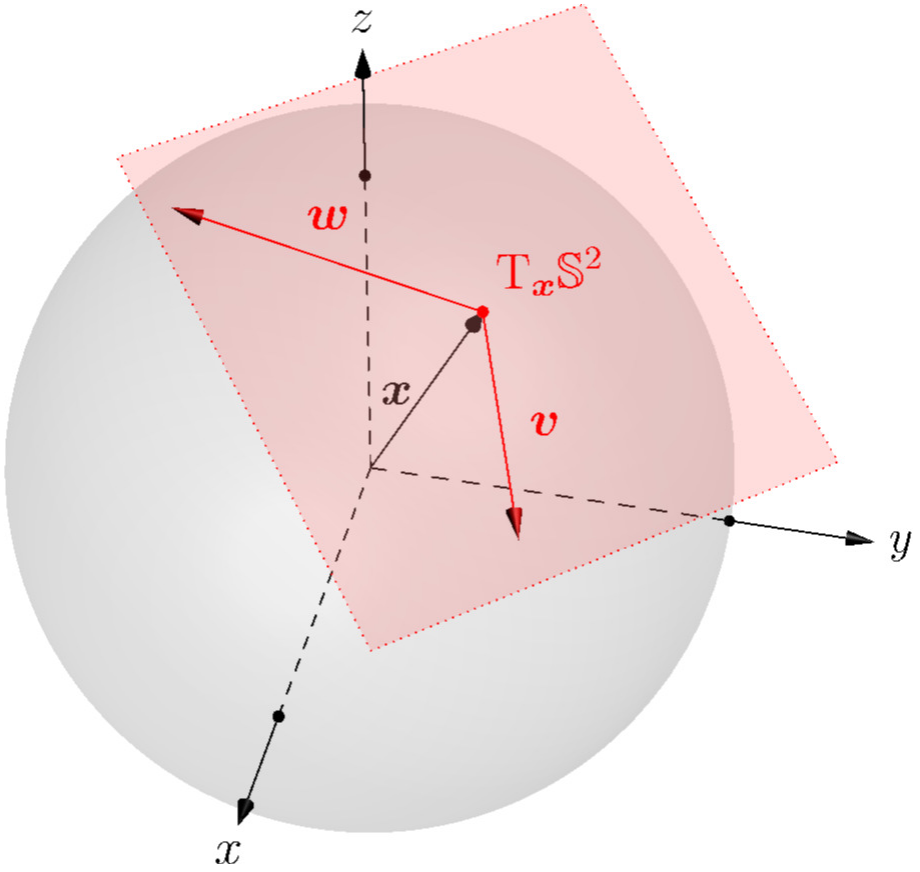

Figure 2.1: The sphere $\mathbb{S}^2$ embedded in $\mathbb{R}^3$ and a tangent space $\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$.

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^\top$ is the usual nabla operator in $\mathbb{R}^3$. Another important notion is given by the Levi-Civita connection $\nabla_{\boldsymbol{Y}} \boldsymbol{X}$ which defines a directional derivative of a vector field $\boldsymbol{X}$ along another vector field $\boldsymbol{Y}$ on manifolds. Using the usual derivative $\mathrm{D}_{\tilde{\boldsymbol{Y}}} \tilde{\boldsymbol{X}}$ of the vector field $\tilde{\boldsymbol{X}}$ with respect to the vector field $\tilde{\boldsymbol{Y}}$ it is expressed by, cf. [4, Sect. 10.1.],

$$\left(\nabla_{\boldsymbol{Y}} \boldsymbol{X}\right)(\boldsymbol{x}) := \mathrm{P}_{\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2}\left(\left(\mathrm{D}_{\tilde{\boldsymbol{Y}}} \tilde{\boldsymbol{X}}\right)(\boldsymbol{x})\right), \qquad \boldsymbol{x} \in \mathbb{S}^2.$$

The Levi-Civita connection is used for defining a concept of parallel transport on manifolds. To this end let $\boldsymbol{s} : [0, T] \to \mathbb{S}^2$, $T > 0$, be a smooth curve on the sphere. We say that the tangent vector $\boldsymbol{v}_0 := \boldsymbol{X}(\boldsymbol{s}(0)) \in \mathrm{T}_{\boldsymbol{s}(0)}\mathbb{S}^2$ is parallel transported along $\boldsymbol{s}$ by $\boldsymbol{X}$ if

$$\left(\nabla_{\dot{\boldsymbol{s}}(t)} \boldsymbol{X}\right)(\boldsymbol{s}(t)) = \boldsymbol{0} \in \mathrm{T}_{\boldsymbol{s}(t)}\mathbb{S}^2, \qquad t \in [0, T].$$

Here $\dot{\boldsymbol{s}}$ denotes the time derivative of $\boldsymbol{s}$, which can be seen as a velocity field on the sphere $\mathbb{S}^2$. An outstanding role play curves $\boldsymbol{g}$, which transport their velocity vectors parallel onto itself, i.e.,

$$\nabla_{\dot{\boldsymbol{g}}} \dot{\boldsymbol{g}} = \boldsymbol{0}.$$

Curves with this property are called geodesics and are the 'straight lines' on the sphere. Given a starting point $\boldsymbol{g}(0) := \boldsymbol{x} \in \mathbb{S}^2$ and a direction $\dot{\boldsymbol{g}}(0) := \boldsymbol{v} \in \mathrm{T}_{\boldsymbol{g}(0)}\mathbb{S}^2$, then the corresponding

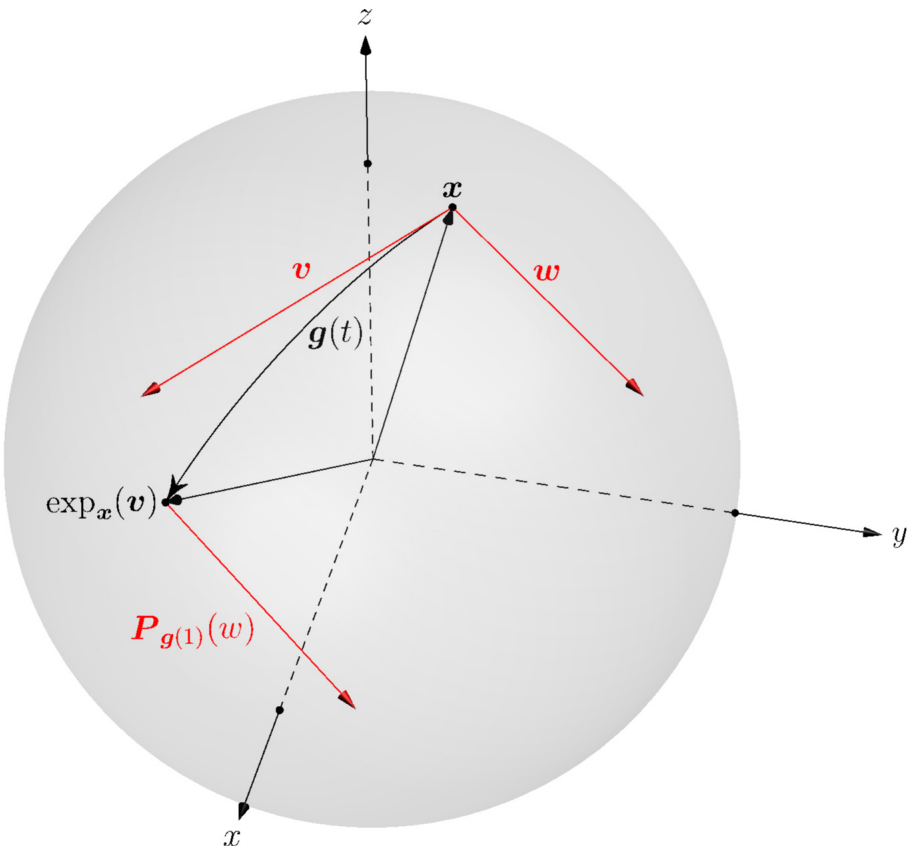

Figure 2.2: A geodesic $\boldsymbol{g}$ and the parallel transported vector $\boldsymbol{P}_{\boldsymbol{g}(1)}(\boldsymbol{w})$ of $\boldsymbol{w} \in \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$.

geodesic $\boldsymbol{g}(t)$ is explicitely parameterized by the exponential map $\exp_{\boldsymbol{x}} : \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2 \to \mathbb{S}^2$, cf. [28, p. 19], due to

$$\boldsymbol{g}(t) := \exp_{\boldsymbol{x}}(t\boldsymbol{v}) := \cos(\|\boldsymbol{v}\|_2 t)\boldsymbol{x} + \sin(\|\boldsymbol{v}\|_2 t)\underbrace{\boldsymbol{v}/\|\boldsymbol{v}\|_2}_{=:\tilde{\boldsymbol{v}}}, \qquad t \geq 0.$$

For an illustration see Figure 2.2. Hence, the geodesic $\boldsymbol{g}$ can also be interpreted as the path of a rotation of the point $\boldsymbol{x}$ about the rotation axis $\boldsymbol{x} \times \boldsymbol{v}$ with rotation angle t. Furthermore the parallel transport of a vector $\boldsymbol{w} \in \mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$ along the geodesic $\boldsymbol{g}$, see Figure 2.2, is realized by

$$\begin{aligned} P_{\boldsymbol{g}(t)}(\boldsymbol{w}) &:= \langle \boldsymbol{w}, \tilde{\boldsymbol{v}} \rangle \dot{\boldsymbol{g}}(t)/\|\boldsymbol{v}\|_2 + \langle \boldsymbol{w}, x \times \tilde{\boldsymbol{v}} \rangle \boldsymbol{g}(t) \times \dot{\boldsymbol{g}}(t)/\|\boldsymbol{v}\|_2, \\ &= \langle \boldsymbol{w}, \tilde{\boldsymbol{v}} \rangle \left(\cos(\|\boldsymbol{v}\|_2 t)\, \tilde{\boldsymbol{v}} - \sin(\|\boldsymbol{v}\|_2 t)\, \boldsymbol{x}\right) + \boldsymbol{w} - \langle \boldsymbol{w}, \tilde{\boldsymbol{v}} \rangle \tilde{\boldsymbol{v}}, \quad t \geq 0. \end{aligned}$$

For implementing a second order optimization method on the sphere one also needs a notion for the Hessian $\mathrm{H}_{\mathbb{S}^2}$ of a function f on manifolds, cf. [28]. It is given by the bilinear form

$$\mathrm{H}_{\mathbb{S}^2} f(\boldsymbol{x})(\boldsymbol{Y}, \boldsymbol{X}) = g_{\mathbb{S}^2}(\nabla_{\boldsymbol{Y}} \nabla_{\mathbb{S}^2} f(\boldsymbol{x}), \boldsymbol{X}(\boldsymbol{x}))$$

on the tangent spaces $\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$, $\boldsymbol{x} \in \mathbb{S}^2$. For a coordinate representation let $\boldsymbol{E}_1, \boldsymbol{E}_2$ be vector fields, which form an orthonormal frame in a small neighborhood $U \subset \mathbb{S}^2$ of $\boldsymbol{x}$, i.e., for every $\boldsymbol{y} \in U$ the vectors $\boldsymbol{E}_1(\boldsymbol{y})$, $\boldsymbol{E}_2(\boldsymbol{y})$ form an orthonormal basis of the tangent space $\mathrm{T}_{\boldsymbol{y}}\mathbb{S}^2$. Then

the components of the Hessian with respect to the vector fields $\boldsymbol{E}_1, \boldsymbol{E}_2$ are given for all $\boldsymbol{x} \in \mathbb{S}^2$ by

$$(\mathrm{H}_{\mathbb{S}^2} f)_{i,j}(\boldsymbol{x}) := (\mathrm{H}\tilde{f})_{i,j}(\boldsymbol{x}) - \left(\nabla_{\nabla_{\boldsymbol{E}_i}\boldsymbol{E}_j}\tilde{f}\right)(\boldsymbol{x}), \qquad i,j = 1,2, \tag{2.1}$$

where $\mathrm{H}\tilde{f}(\boldsymbol{x})$ is the usual Hessian restricted to the subspaces spanned by $\boldsymbol{E}_i(\boldsymbol{x})$, $i = 1,2$, and $\nabla_{\boldsymbol{v}}\tilde{f}$ is the directional derivative towards $\boldsymbol{v}$ in $\mathbb{R}^3$ of the extension $\tilde{f}$ respectively. For the sake of completeness we define the Laplace-Beltrami operator on the sphere

$$\Delta_{\mathbb{S}^2} f := \operatorname{tr}(\mathrm{H}_{\mathbb{S}^2} f) \tag{2.2}$$

via the trace of the Hessian.

For a local parameterization of $\mathbb{S}^2$ we use as usual spherical coordinates $(\theta, \varphi) \in [0, \pi] \times [0, 2\pi)$ with $\boldsymbol{x} := \boldsymbol{x}(\theta, \varphi) := (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)^\top$. Then the vectors

$$\boldsymbol{x}_\theta := \frac{\partial}{\partial\theta}\boldsymbol{x}(\theta,\varphi), \qquad \boldsymbol{x}_\varphi := \frac{\partial}{\partial\varphi}\boldsymbol{x}(\theta,\varphi)$$

form an orthogonal basis of the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$ for $\boldsymbol{x} \in \mathbb{S}^2 \setminus \{\pm(0,0,1)^\top\}$. In the literature on Riemannian geometry the quantities which define differential operators on manifolds are expressed in terms of this canonical basis. However, we prefer to use an orthonormal basis of the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathbb{S}^2$. For that reason we introduce the unit vectors

$$\boldsymbol{e}_\theta := \boldsymbol{x}_\theta, \qquad \boldsymbol{e}_\varphi := \frac{1}{\sin\theta}\boldsymbol{x}_\varphi.$$

We remark, due to the singularities at the poles $\boldsymbol{e}_z := (0,0,1)^\top$ and $-\boldsymbol{e}_z$ there is no basis in spherical coordinates of the corresponding tangent spaces. By the frame $\{\boldsymbol{e}_\theta, \boldsymbol{e}_\varphi\}$ the spherical nabla operator reads as

$$\nabla_{\mathbb{S}^2} := \left(\frac{\partial}{\partial\theta}, \frac{1}{\sin\theta}\frac{\partial}{\partial\varphi}\right)^\top := \boldsymbol{e}_\theta\frac{\partial}{\partial\theta} + \boldsymbol{e}_\varphi\frac{1}{\sin\theta}\frac{\partial}{\partial\varphi}, \tag{2.3}$$

and the Hessian is parameterized by

$$\mathrm{H}_{\mathbb{S}^2} = \begin{pmatrix} \frac{\partial^2}{\partial\theta^2} & \frac{1}{\sin\theta}\frac{\partial^2}{\partial\theta\partial\varphi} - \frac{\cot\theta}{\sin\theta}\frac{\partial}{\partial\varphi} \\ \frac{1}{\sin\theta}\frac{\partial^2}{\partial\varphi\partial\theta} - \frac{\cot\theta}{\sin\theta}\frac{\partial}{\partial\varphi} & \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\varphi^2} + \cot\theta\frac{\partial}{\partial\theta} \end{pmatrix}. \tag{2.4}$$

Furthermore the Laplace-Beltrami operator given by (2.2) implies

$$\Delta_{\mathbb{S}^2} = \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\varphi^2}.$$

2.2 Riemannian geometry on $\mathbb{S}^2 \times \cdots \times \mathbb{S}^2$

For computing spherical t-designs we aim to optimize over the product manifold

$$\mathbb{S}^2_M := \underbrace{\mathbb{S}^2 \times \cdots \times \mathbb{S}^2}_{M \text{ times}}$$

of M spheres $\mathbb{S}^2$. We briefly summarize the necessary notations for the geometric objects on this manifold. The tangent space at the point $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathbb{S}^2_M$ is simply defined by

$$\mathrm{T}_{\vec{\boldsymbol{x}}}\mathbb{S}^2_M := \mathrm{T}_{\boldsymbol{x}_1}\mathbb{S}^2 \times \cdots \times \mathrm{T}_{\boldsymbol{x}_M}\mathbb{S}^2$$

with its canonical Riemannian metric

$$g_{\mathbb{S}^2_M}(\boldsymbol{v},\boldsymbol{w}) := \sum_{i=1}^{M} g_{\mathbb{S}^2}(\boldsymbol{v}_i,\boldsymbol{w}_i), \qquad \boldsymbol{v} := (\boldsymbol{v}_1,\dots,\boldsymbol{v}_M), \boldsymbol{w} := (\boldsymbol{w}_1,\dots,\boldsymbol{w}_M) \in \mathrm{T}_{\vec{\boldsymbol{x}}}\mathbb{S}^2_M.$$

Since the tangent subspaces $\mathrm{T}_{\boldsymbol{x}_i}\mathbb{S}^2$, $\mathrm{T}_{\boldsymbol{x}_j}\mathbb{S}^2$ are orthogonal for $i \neq j$ the distance is given by the Pythagorean sum

$$\mathrm{d}_{\mathbb{S}^2_M}(\vec{\boldsymbol{x}},\vec{\boldsymbol{y}}) := \left(\sum_{i=1}^{M} \mathrm{d}^2_{\mathbb{S}^2}(\boldsymbol{x}_i,\boldsymbol{y}_i)\right)^{\frac{1}{2}}, \quad \vec{\boldsymbol{x}},\vec{\boldsymbol{y}} \in \mathbb{S}^2_M.$$

In the same manner we obtain for every $\vec{\boldsymbol{x}} \in \mathbb{S}^2_M$ the exponential map $\exp : \mathbb{S}^2_M \to \mathbb{S}^2_M$ by

$$\exp_{\vec{\boldsymbol{x}}}(\boldsymbol{v}) := \big(\exp_{\boldsymbol{x}_1}(\boldsymbol{v}_1),\dots,\exp_{\boldsymbol{x}_M}(\boldsymbol{v}_M)\big) \in \mathbb{S}^2_M, \qquad \boldsymbol{v} \in \mathrm{T}_{\vec{\boldsymbol{x}}}\mathbb{S}^2_M.$$

We denote by $\nabla^i_{\mathbb{S}^2} f(\vec{\boldsymbol{x}}) \in \mathrm{T}_{\boldsymbol{x}_i}\mathbb{S}^2$, $i = 1,\dots,M$ the spherical gradient of f with respect to $\boldsymbol{x}_i$, then the gradient of f is expressed by

$$\nabla_{\mathbb{S}^2_M} f(\vec{\boldsymbol{x}}) := \big(\nabla^1_{\mathbb{S}^2} f(\vec{\boldsymbol{x}}),\dots,\nabla^M_{\mathbb{S}^2} f(\vec{\boldsymbol{x}})\big) \in \mathrm{T}_{\vec{\boldsymbol{x}}}\mathbb{S}^2_M. \tag{2.5}$$

Similarly, we denote by $\mathrm{H}^i_{\mathbb{S}^2} f$ the spherical Hessian with respect to the coordinate $\boldsymbol{x}_i$ and obtain by (2.3), (2.4) and (2.1) the formula

$$\mathrm{H}_{\mathbb{S}^2_M} f(\vec{\boldsymbol{x}}) := \begin{pmatrix} \mathrm{H}^1_{\mathbb{S}^2} f(\vec{\boldsymbol{x}}) & \nabla^1_{\mathbb{S}^2}\nabla^2_{\mathbb{S}^2}{}^{\top} f(\vec{\boldsymbol{x}}) & \dots & \nabla^1_{\mathbb{S}^2}\nabla^M_{\mathbb{S}^2}{}^{\top} f(\vec{\boldsymbol{x}}) \\ \nabla^2_{\mathbb{S}^2}\nabla^1_{\mathbb{S}^2}{}^{\top} f(\vec{\boldsymbol{x}}) & \mathrm{H}^2_{\mathbb{S}^2} f(\vec{\boldsymbol{x}}) & \dots & \nabla^2_{\mathbb{S}^2}\nabla^M_{\mathbb{S}^2}{}^{\top} f(\vec{\boldsymbol{x}}) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^M_{\mathbb{S}^2}\nabla^1_{\mathbb{S}^2}{}^{\top} f(\vec{\boldsymbol{x}}) & \nabla^M_{\mathbb{S}^2}\nabla^2_{\mathbb{S}^2}{}^{\top} f(\vec{\boldsymbol{x}}) & \dots & \mathrm{H}^M_{\mathbb{S}^2} f(\vec{\boldsymbol{x}}) \end{pmatrix}. \tag{2.6}$$

In Section 3 we do all the computations in the basis of the tangent spaces $\mathrm{T}_{\boldsymbol{x}_i}\mathbb{S}^2$ given by $\{\boldsymbol{e}_\theta(\boldsymbol{x}_i), \boldsymbol{e}_\varphi(\boldsymbol{x}_i)\}$, $i = 1,\dots,M$. Hence, $\boldsymbol{x}_i = \pm\boldsymbol{e}_z$ is not a feasible point. Furthermore we express the tangent vectors $\boldsymbol{v}_i \in \mathrm{T}_{\boldsymbol{x}_i}\mathbb{S}^2$, $i = 1,\dots,M$ by the representation

$$\boldsymbol{v}_i := v_{\theta_i}\boldsymbol{e}_\theta(\boldsymbol{x}_i) + v_{\varphi_i}\boldsymbol{e}_\varphi(\boldsymbol{x}_i)$$

and write for simplicity

$$\boldsymbol{v}_i := (v_{\theta_i}, v_{\varphi_i})^{\top} \in \mathbb{R}^2. \tag{2.7}$$

2.3 Optimization methods on Riemannian manifolds

In this section we describe the Newton method and the method of conjugate gradients for nonlinear problems on Riemannian manifolds. For a nice survey article with applications of optimization on manifolds see [7]. We shortly recapitulate these standard methods in the Euclidean space.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be the objective function and $\boldsymbol{x}_* \in \mathbb{R}^d$ a minimum point, i.e., $\nabla f(\boldsymbol{x}_*) = 0$ with positive definite Hessian $\mathrm{H}f$ in a neighborhood of $\boldsymbol{x}_*$. Then Newton's method is defined for an initial guess $\boldsymbol{x}_0$ close to $\boldsymbol{x}_*$ by the following iteration

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \mathrm{H}f(\boldsymbol{x}_k)^{-1}\nabla f(\boldsymbol{x}_k), \qquad k = 0,1,\dots,$$

and its well-know that for smooth functions it converges quadratically in a neighborhood of $\boldsymbol{x}_*$, i.e.,

$$\|\boldsymbol{x}_* - \boldsymbol{x}_{k+1}\|_2 \leq c\|\boldsymbol{x}_* - \boldsymbol{x}_k\|_2^2.$$

Some drawback in large dimensions is the difficulty to invert or even calculate the Hessian. Hence efficient first order optimization methods are preferred for high dimensional objective functions. Here, we consider conjugate gradient (CG) algorithms for nonlinear optimization. For a nice survey in the Euclidean case see [13]. The general scheme of a nonlinear CG method uses the recurrence

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k, \qquad k = 0, 1, \ldots,$$

where α_k is a positive step size and $\boldsymbol{d}_k$ are the search directions given by the rule

$$\boldsymbol{d}_{k+1} := -\boldsymbol{g}_{k+1} + \beta_k \boldsymbol{d}_k, \qquad \boldsymbol{d}_0 := -\boldsymbol{g}_0, \qquad \boldsymbol{g}_k := \nabla f(\boldsymbol{x}_k).$$

Various CG methods are known, which differ only in the choices for β_k, e.g., the one for exact conjugacy proposed by Daniel in [5]

$$\beta_k := \frac{\langle \boldsymbol{g}_{k+1}, \mathrm{H}f(\boldsymbol{x}_{k+1})\boldsymbol{d}_k\rangle}{\langle \boldsymbol{d}_k, \mathrm{H}f(\boldsymbol{x}_{k+1})\boldsymbol{d}_k\rangle}.$$

The step size α_k is determined by the search of the nearest local minimum to $\boldsymbol{x}_k$ along the line $\boldsymbol{x}_k + t\boldsymbol{d}_k$, $t > 0$, hence it has to satisfy

$$\nabla f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k)\boldsymbol{d}_k = 0.$$

The above algorithms generalize in a natural way to Riemannian manifolds $(\mathcal{M}, g_{\mathcal{M}})$. For that reason we remark that all the geometric objects defined in the last section for the cases of the sphere $\mathbb{S}^2$ and its products $\mathbb{S}^2_M$ are defined in a rigorous manner for general Riemannian manifolds $(\mathcal{M}, g_{\mathcal{M}})$, cf. [26, 7]. Hence, we just replace the subscripts $\mathbb{S}^2$ and $\mathbb{S}^2_M$ by $\mathcal{M}$ for the general description of the algorithms.

In Riemannian geometry the addition of a tangent vector to the base point $\boldsymbol{x}$ is replaced by the exponential map $\exp_{\boldsymbol{x}} : \mathrm{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$. Moreover the translation of tangent vectors is replaced by the notion of parallel transport $\boldsymbol{P}_{\boldsymbol{g}(t)}(\boldsymbol{v})$ along geodesics $\boldsymbol{g}$. By doing so the Newton method reads as, cf. [28, Sec. 7.5],

$$\boldsymbol{x}_{k+1} := \exp_{\boldsymbol{x}_k}\left(-\mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}_k)^{-1} \nabla_{\mathcal{M}} f(\boldsymbol{x}_k)\right), \qquad k = 0, 1, \ldots,$$

where $f : \mathcal{M} \to \mathbb{R}$ is the objective function and $\boldsymbol{x}_0 \in \mathcal{M}$ is close to a minimum $\boldsymbol{x}_* \in \mathcal{M}$. As in the Euclidean case it was shown in [26] that this scheme is also quadratically convergent

$$\mathrm{d}_{\mathcal{M}}(\boldsymbol{x}_*, \boldsymbol{x}_{k+1}) \leq c\, \mathrm{d}^2_{\mathcal{M}}(\boldsymbol{x}_*, \boldsymbol{x}_k).$$

The CG method on Riemannian manifolds is given by

$$\boldsymbol{x}_{k+1} := \exp_{\boldsymbol{x}_k}(\alpha_k \boldsymbol{d}_k), \qquad k = 0, 1, \ldots,$$

with

$$\boldsymbol{d}_{k+1} := -\boldsymbol{g}_{k+1} + \beta_k \boldsymbol{P}_{\boldsymbol{g}(\alpha_k)}(\boldsymbol{d}_k), \qquad \boldsymbol{d}_0 := -\boldsymbol{g}_0, \qquad \boldsymbol{g}_k := \nabla_{\mathcal{M}} f(\boldsymbol{x}_k),$$

where $\boldsymbol{g}$ is the geodesic from $\boldsymbol{g}(0) = \boldsymbol{x}_k$ to $\boldsymbol{g}(\alpha_k) = \boldsymbol{x}_{k+1}$ in direction $\boldsymbol{d}_k$. Furthermore the scalar β_k is obtained by

$$\beta_k := \frac{\langle \boldsymbol{g}_{k+1}, \mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}_{k+1}) \boldsymbol{P}_{\boldsymbol{g}(\alpha_k)}(\boldsymbol{d}_k) \rangle}{\langle P_{\boldsymbol{g}(\alpha_k)}(\boldsymbol{d}_k), \mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}_{k+1}) P_{\boldsymbol{g}(\alpha_k)}(\boldsymbol{d}_k) \rangle}$$

and the step size α_k is determined by

$$\nabla_{\mathcal{M}} f(\boldsymbol{g}(\alpha_k)) \boldsymbol{P}_{\boldsymbol{g}(\alpha_k)}(\boldsymbol{d}_k) = 0. \tag{2.8}$$

For an illustration of one iteration step of the CG method on the sphere $\mathbb{S}^2$ see Figure 2.3.

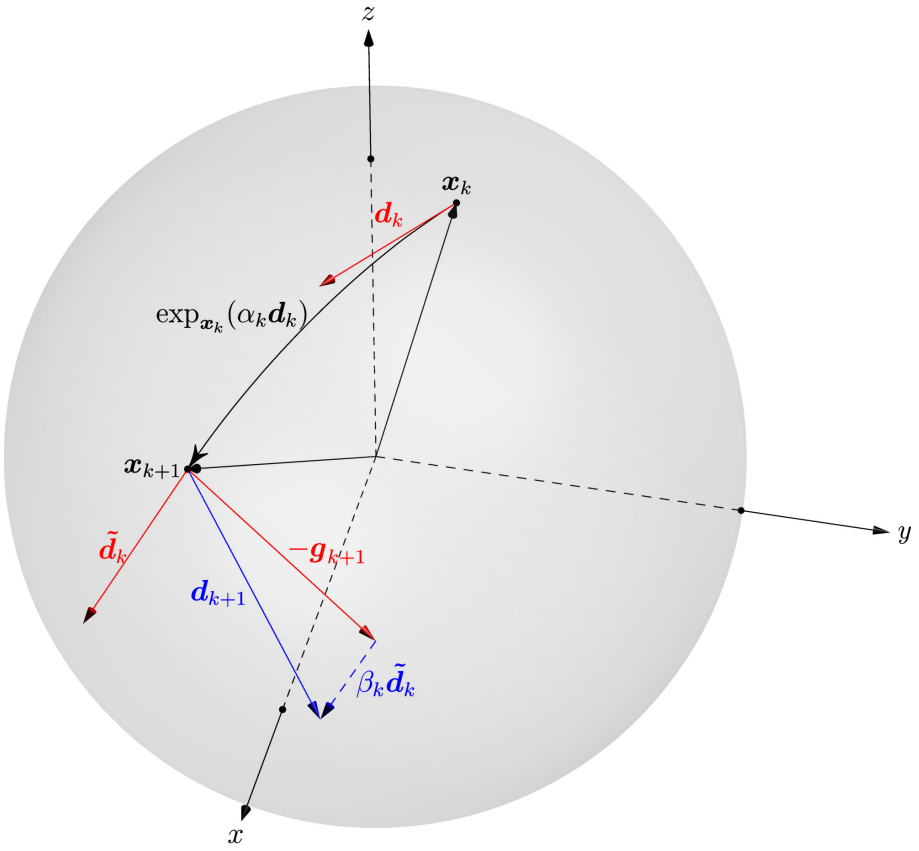

Figure 2.3: An iteration step of the nonlinear CG method on the sphere $\mathbb{S}^2$.

2.4 Fast spherical Fourier transforms

It is well known that the eigenfunctions of the spherical Laplace-Beltrami operator $\Delta_{\mathbb{S}^2}$ are the spherical harmonics Y_n^k of degree n and order k, cf. [22],

$$Y_n^k(\boldsymbol{x}) = Y_n^k(\theta, \varphi) := \sqrt{\frac{2n+1}{4\pi}} P_n^{|k|}(\cos\theta) \mathrm{e}^{\mathrm{i}k\varphi}, \qquad \boldsymbol{x} = \boldsymbol{x}(\theta, \varphi) \in \mathbb{S}^2,$$

where the associated Legendre functions $P_n^k : [-1,1] \to \mathbb{R}$ and the Legendre polynomials $P_n : [-1,1] \to \mathbb{R}$ are given by

$$P_n^k(x) := \left(\frac{(n-k)!}{(n+k)!}\right)^{1/2} \left(1-x^2\right)^{k/2} \frac{\mathrm{d}^k}{\mathrm{d}x^k} P_n^k(x), \qquad n \in \mathbb{N}_0,\ k = 0,\ldots,n,$$

$$P_n(x) := \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left(x^2-1\right)^n, \qquad n \in \mathbb{N}_0.$$

In spherical coordinates the surface element reads as $\mathrm{d}\mu_{\mathbb{S}^2}(\boldsymbol{x}) = \sin\theta \mathrm{d}\theta\mathrm{d}\varphi$ and the spherical harmonics obey the orthogonality relation

$$\int_{\mathbb{S}^2} Y_n^k(\boldsymbol{x})\overline{Y_m^l(\boldsymbol{x})}\mathrm{d}\mu_{\mathbb{S}^2}(\boldsymbol{x}) = \int_0^{2\pi}\int_0^{\pi} Y_n^k(\theta,\phi)\overline{Y_m^l(\theta,\phi)}\sin\theta\mathrm{d}\theta\mathrm{d}\varphi = \delta_{k,l}\delta_{n,m}.$$

Moreover, the spherical harmonics form an orthonormal basis of the space of all square integrable functions $L_2(\mathbb{S}^2) := \{f : \mathbb{S}^2 \to \mathbb{C} \ : \ \int_{\mathbb{S}^2} |f(\boldsymbol{x})|^2 \mathrm{d}\mu_{\mathbb{S}^2}(\boldsymbol{x}) < \infty\}$. Hence, every $f \in L_2(\mathbb{S}^2)$ has an unique expansion in spherical harmonics

$$f = \sum_{n=0}^{\infty}\sum_{k=-n}^{n} \hat{f}_n^k Y_n^k.$$

We say that f is a spherical polynomial of degree at most N if $\hat{f}_n^k = 0$, $n > N$, and we denote by $\Pi_N(\mathbb{S}^2)$ the space of all spherical polynomials of degree at most N. We remark that the dimension of $\Pi_N(\mathbb{S}^2)$ is $d_N := (N+1)^2$.

The evaluation of a spherical polynomial

$$f = \sum_{n=0}^{N}\sum_{k=-n}^{n} \hat{f}_n^k Y_n^k \in \Pi_N(\mathbb{S}^2)$$

on a sampling set $\mathcal{X}_M = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M\} \subset \mathbb{S}^2$ can be expressed by a matrix-vector multiplication

$$\boldsymbol{f} = \boldsymbol{Y}^N \hat{\boldsymbol{f}},$$

where $\boldsymbol{Y}^N$ is the nonequispaced spherical Fourier matrix

$$\boldsymbol{Y}^N := (Y_k^n(\boldsymbol{x}_i))_{i=1,\ldots,M;\ n=0,\ldots,N,|k|\le n} \in \mathbb{C}^{M\times d_N},$$

$\boldsymbol{f}$ is the vector of the sampling values

$$\boldsymbol{f} = (f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_M))^\top \in \mathbb{C}^M$$

and $\hat{\boldsymbol{f}}$ is the vector of spherical Fourier coefficients

$$\hat{\boldsymbol{f}} := (\hat{f}_n^k)_{n=0,\ldots,N,|k|\le n} \in \mathbb{C}^{d_N}.$$

Recently, fast approximate algorithms for the matrix times vector multiplication with the nonequispaced spherical Fourier matrix $\boldsymbol{Y}^N$ and its adjoint $\overline{\boldsymbol{Y}^N}^\top$ have been proposed in [20, 19]. The arithmetic complexity for the so called fast spherical Fourier transform and its adjoint is $\mathcal{O}(N^2\log^2 N + M\log^2(1/\epsilon))$, where $\epsilon > 0$ is a prescribed accuracy of the approximate algorithms. An implementation of these algorithms can be found in the Internet [16]. In the next section we use these fast algorithms for the evaluation of gradients and Hessians of spherical polynomials, as well.

3 Fast realization of the optimization methods

In the following we show, that we can realize each iteration step of the nonlinear CG method on the sphere $\mathbb{S}^2$ and the product manifold $\mathbb{S}^2_M$ with the nonequispaced spherical Fourier transform. More precisely we propose in Theorem 3.2 an efficient scheme for the computation of the spherical gradient and the Hessian of a spherical polynomial. Using this method we are able to compute in an efficient way numerical spherical t-designs for high polynomial degrees $t \in \mathbb{N}$.

3.1 Fast methods for evaluating the spherical gradient and the Hessian on $\mathbb{S}^2$

In the following we state that the components f_θ, f_φ of the spherical gradient

$$\nabla_{\mathbb{S}^2} f := f_\theta \boldsymbol{e}_\theta + f_\varphi \boldsymbol{e}_\varphi \tag{3.1}$$

and the components $f_{\theta,\theta}$, $f_{\varphi,\varphi}$, $f_{\varphi,\theta} = f_{\theta,\varphi}$ of the Hessian

$$\mathrm{H}_{\mathbb{S}^2} f := \begin{pmatrix} f_{\theta,\theta} & f_{\theta,\varphi} \\ f_{\varphi,\theta} & f_{\varphi,\varphi} \end{pmatrix}, \tag{3.2}$$

of a spherical polynomial f are spherical polynomials up to factors of $\sin\theta$ and $\cos\theta$. This allows us to utilize the nonequispaced fast spherical Fourier transforms for evaluating the gradient and the Hessian of a spherical polynomial on many points simultaneously.

Lemma 3.1. *Let $f \in \Pi_N(\mathbb{S}^2)$ be a spherical polynomial with spherical Fourier coefficient vector $\hat{\boldsymbol{f}} := (\hat{f}_n^k) \in \mathbb{C}^{d_N}$. Then the components of the spherical gradient $\nabla_{\mathbb{S}^2} f$, cf. (3.1), are expressed for $\boldsymbol{x} := \boldsymbol{x}(\theta,\varphi) \in \mathbb{S}^2 \setminus \{\pm \boldsymbol{e}_z\}$ by*

$$f_\theta(\boldsymbol{x}(\theta,\varphi)) := \frac{1}{\sin\theta} \sum_{n=0}^{N+1} \sum_{k=-n}^{n} (\hat{f}_\theta)_n^k \, Y_n^k(\theta,\varphi), \quad f_\varphi(\boldsymbol{x}(\theta,\varphi)) := \frac{1}{\sin\theta} \sum_{n=0}^{N} \sum_{k=-n}^{n} (\hat{f}_\varphi)_n^k \, Y_n^k(\theta,\varphi),$$

with spherical Fourier coefficients

$$(\hat{f}_\theta)_n^k := (n-1)\sqrt{\frac{n^2-k^2}{(2n-1)(2n+1)}}\hat{f}_{n-1}^k - (n+2)\sqrt{\frac{(n+1)^2-k^2}{(2n+3)(2n+1)}}\hat{f}_{n+1}^k, \tag{3.3}$$

where $\hat{f}_{N+2}^k = \hat{f}_{N+1}^k = \hat{f}_{-1}^k = 0$ and spherical Fourier coefficients

$$(\hat{f}_\varphi)_n^k := \mathrm{i}k\hat{f}_n^k. \tag{3.4}$$

Proof. The above assertion results from the representation (2.3) of the spherical gradient $\nabla_{\mathbb{S}^2}$ in spherical coordinates and the following relations for the partial derivatives of the spherical harmonics Y_n^k, cf. [29, pp. 146],

$$\begin{aligned} \frac{\partial}{\partial\varphi} Y_n^k(\theta,\varphi) =& \mathrm{i}k Y_n^k(\theta,\varphi), \\ \sin\theta \frac{\partial}{\partial\theta} Y_n^k(\theta,\varphi) =& n\sqrt{\frac{(n+1)^2-k^2}{(2n+1)(2n+3)}} Y_{n+1}^k(\theta,\varphi) - (n+1)\sqrt{\frac{n^2-k^2}{(2n+1)(2n-1)}} Y_{n-1}^k(\theta,\varphi), \end{aligned} \tag{3.5}$$

where for $|k| > n-1$ we have $Y_{n-1}^k \equiv 0$. ∎

Using Lemma 3.1 we define the 'bidiagonal'-like matrix $\boldsymbol{D}_\theta^N \in \mathbb{C}^{d_{N+1}\times d_N}$ as the matrix satisfying, cf. (3.3),

$$(\boldsymbol{D}_\theta^N \hat{\boldsymbol{f}})_n^k = (\hat{f}_\theta)_n^k, \qquad n = 0,\dots,N+1,\ k = -n,\dots,n, \tag{3.6}$$

and the diagonal matrix $\boldsymbol{D}_\varphi^N \in \mathbb{C}^{d_N\times d_N}$ with, cf. (3.4),

$$(\boldsymbol{D}_\varphi^N \hat{\boldsymbol{f}})_n^k = (\hat{f}_\varphi)_n^k, \qquad n = 0,\dots,N,\ k = -n,\dots,n. \tag{3.7}$$

Furthermore, we introduce for the sampling points $\boldsymbol{x}_i := \boldsymbol{x}(\theta_i,\varphi_i)$, $i = 1,\dots,M$, the diagonal matrices $\boldsymbol{S} := \operatorname{diag}(\sin(\theta_1),\dots,\sin(\theta_M))$, $\boldsymbol{C} := \operatorname{diag}(\cos(\theta_1),\dots,\cos(\theta_M)) \in \mathbb{C}^{M\times M}$ and arrive at the following theorem.

Theorem 3.2. *For a given sampling set $\mathcal{X}_M := \{\boldsymbol{x}(\theta_1,\varphi_1),\dots,\boldsymbol{x}(\theta_M,\varphi_M)\} \subset \mathbb{S}^2 \setminus \{\pm\boldsymbol{e}_z\}$ with $\boldsymbol{x}_i := \boldsymbol{x}(\theta_i,\varphi_i)$ and a spherical polynomial $f \in \Pi_N(\mathbb{S}^2)$ with corresponding spherical Fourier coefficient vector $\hat{\boldsymbol{f}} \in \mathbb{C}^{d_N}$ we obtain the spherical gradient, cf.* (3.1),

$$\nabla_{\mathbb{S}^2} f(\boldsymbol{x}_i) = f_{\theta_i}\boldsymbol{e}_\theta(\boldsymbol{x}_i) + f_{\varphi_i}\boldsymbol{e}_\varphi(\boldsymbol{x}_i)$$

by the evaluation

$$\begin{aligned}\boldsymbol{f}_\theta &:= (f_{\theta_i})_{i=1,\dots,M} = \boldsymbol{S}^{-1}\boldsymbol{Y}^{N+1}\boldsymbol{D}_\theta^N\hat{\boldsymbol{f}},\\ \boldsymbol{f}_\varphi &:= (f_{\varphi_i})_{i=1,\dots,M} = \boldsymbol{S}^{-1}\boldsymbol{Y}^{N}\boldsymbol{D}_\varphi^N\hat{\boldsymbol{f}}\end{aligned}$$

and similar for the components of the Hessian, cf (3.2),

$$\mathrm{H}_{\mathbb{S}^2} f(\boldsymbol{x}_i) = \begin{pmatrix} f_{\theta_i,\theta_i} & f_{\theta_i,\varphi_i}\\ f_{\varphi_i,\theta_i} & f_{\varphi_i,\varphi_i}\end{pmatrix},$$

the representation

$$\begin{aligned}\boldsymbol{f}_{\theta,\theta} &= (f_{\theta_i,\theta_i})_{i=1,\dots,M} = \boldsymbol{S}^{-2}(\boldsymbol{Y}^{N+2}\boldsymbol{D}_\theta^{N+1}\boldsymbol{D}_\theta^N - \boldsymbol{C}\boldsymbol{Y}^{N+1}\boldsymbol{D}_\theta^N)\hat{\boldsymbol{f}},\\ \boldsymbol{f}_{\varphi,\varphi} &= (f_{\varphi_i,\varphi_i})_{i=1,\dots,M} = \boldsymbol{S}^{-2}(\boldsymbol{Y}^{N}\boldsymbol{D}_\varphi^{N}\boldsymbol{D}_\varphi^N + \boldsymbol{C}\boldsymbol{Y}^{N+1}\boldsymbol{D}_\theta^N)\hat{\boldsymbol{f}},\\ \boldsymbol{f}_{\varphi,\theta} &= (f_{\varphi_i,\theta_i})_{i=1,\dots,M} = \boldsymbol{S}^{-2}(\boldsymbol{Y}^{N+1}\boldsymbol{D}_\varphi^{N+1}\boldsymbol{D}_\theta^N - \boldsymbol{C}\boldsymbol{Y}^{N}\boldsymbol{D}_\varphi^N)\hat{\boldsymbol{f}}\end{aligned}$$

with $f_{\varphi_i,\theta_i} = f_{\theta_i,\varphi_i}$, $i = 1,\dots,M$. Furthermore all evaluations of the sampling vectors $\boldsymbol{f}_\theta$, $\boldsymbol{f}_\varphi$, $\boldsymbol{f}_{\theta,\theta}$, $\boldsymbol{f}_{\varphi,\varphi}$, $\boldsymbol{f}_{\varphi,\theta} \in \mathbb{C}^M$ are performed by means of the nonequispaced fast spherical Fourier transform in $\mathcal{O}(N^2\log^2 N + M\log^2(1/\epsilon))$ arithmetic operations.

Proof. The formulae for the components $\boldsymbol{f}_\theta$, $\boldsymbol{f}_\varphi$ follow immediately from Lemma 3.1 and the definitions (3.6), (3.7) of the matrices $\boldsymbol{D}_\theta^N$, $\boldsymbol{D}_\varphi^N$, $\boldsymbol{S}$ and $\boldsymbol{C}$. Furthermore, the representation of the Hessian cf. (2.4),

$$\sin^2\theta\,\mathrm{H}_{\mathbb{S}^2} = \left(\sin\theta\frac{\partial}{\partial\theta},\frac{\partial}{\partial\varphi}\right)^{\top}\left(\sin\theta\frac{\partial}{\partial\theta},\frac{\partial}{\partial\varphi}\right) + \cos\theta\begin{pmatrix}-\sin\theta\frac{\partial}{\partial\theta} & -\frac{\partial}{\partial\varphi}\\ -\frac{\partial}{\partial\varphi} & \sin\theta\frac{\partial}{\partial\theta}\end{pmatrix}.$$

yields together with (3.5) the remaining formulae. The complexity assertion follows from the observation that the matrix-vector multiplication of the matrices $\boldsymbol{D}_\theta^N$, $\boldsymbol{D}_\varphi^N$, $\boldsymbol{S}$ and $\boldsymbol{C}$ need $\mathcal{O}(N^2+M)$ and of the matrix $\boldsymbol{Y}^N$ needs $\mathcal{O}(N^2\log^2 N + M\log^2(1/\epsilon))$ arithmetic operations, respectively. ∎

Remark 3.3. We note that the problem of finding the global maximum of a real-valued spherical polynomial $f \in \Pi_N(\mathbb{S}^2)$ on the sphere $\mathbb{S}^2$ appears in a variety of applications. Usually one starts with one initial guess $\boldsymbol{x}_0 \in \mathbb{S}^2$ of an optimum $\boldsymbol{x}_* \in \mathbb{S}^2$ and uses an optimization algorithm like Newton's method. Unfortunately, $\boldsymbol{x}_0$ should be near to $\boldsymbol{x}_*$. With the proposed methods in Theorem 3.2 we are able to optimize simultaneously over many initial guesses with almost the same arithmetic complexity. Using a sufficiently dense and uniform distributed sampling set $\mathcal{X}_M := \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$ of such starting points increases notably the chance of finding a global maximum. Numerical results will presented elsewhere. □

3.2 Fast optimization for spherical t-designs

The concept of spherical t-designs was introduced by Delsarte, Goethals and Seidel [6] in 1977. There a spherical t-design on $\mathbb{S}^2$ is defined as a finite set $\mathcal{X}_M = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\} \subset \mathbb{S}^2$ satisfying

$$\int_{\mathbb{S}^2} f(\boldsymbol{x}) \mathrm{d}\mu_{\mathbb{S}^2}(\boldsymbol{x}) = \frac{4\pi}{M} \sum_{i=1}^{M} f(\boldsymbol{x}_i), \qquad \text{for all } f \in \Pi_t(\mathbb{S}^2).$$

In the following we exploit the equivalent characterization used by Sloan and Womersley in [25],

$$A_t(\vec{\boldsymbol{x}}) := A_t(\boldsymbol{x}_1, \dots, \boldsymbol{x}_M) := \frac{1}{M^2} \sum_{n=1}^{t} \sum_{k=-n}^{n} \left| \sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right|^2 = 0. \tag{3.8}$$

This function A_t can be seen as the squared integration error for the set $\mathcal{X}_M$ to be a spherical t-design. Since $A_t \geq 0$, the problem of finding a t-design $\mathcal{X}_M$ reduces to finding a minimum of A_t. We aim to apply the Newton and CG methods on Riemannian manifolds proposed in Section 2.3 to the function $A_t : \mathbb{S}^2_M \to \mathbb{R}$. For that reason we present fast algorithms for the evaluation of A_t, the gradient $\nabla_{\mathbb{S}^2_M} A_t$ and the matrix-vector multiplication with the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t$.

The following Theorem provides us with a first taste for the use of fast nonequispaced spherical Fourier transforms.

Theorem 3.4. *For a given point $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1, \dots, \boldsymbol{x}_M) \in \mathbb{S}^2_M$ the evaluation of $A_t(\vec{\boldsymbol{x}})$, $t \in \mathbb{N}$, takes $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ arithmetic operations, where ϵ is a prescribed accuracy.*

Proof. By definition (3.8) we have to compute

$$A_t(\vec{\boldsymbol{x}}) = \frac{1}{M^2} (\overline{\boldsymbol{r}}^\top \boldsymbol{r} - |r_0^0|^2)$$

with the residual vector

$$\boldsymbol{r} := \left(r_n^k \right)_{n=0,\dots,t;\ |k| \leq n}, \qquad r_n^k := \left(\sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right).$$

We compute the conjugate vector $\overline{\boldsymbol{r}}$ in $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ arithmetic operations by a fast multiplication with the adjoint nonequispaced spherical Fourier matrix due to

$$\overline{\boldsymbol{r}} = \overline{\boldsymbol{Y}^t}^\top \boldsymbol{e},$$

where we use the vector $\boldsymbol{e} := (1, \dots, 1)^\top \in \mathbb{C}^M$, cf. Section 2.4. Since the vector $\boldsymbol{r}$ has $(t+1)^2$ components we compute its squared norm in $\mathcal{O}(t^2)$ arithmetic operations. ■

The next main Theorem 3.7, is derived from the following Lemmas 3.5 and 3.6.

Lemma 3.5. *For $f : \mathbb{S}^2 \to \mathbb{C}$ the spherical gradient and the Hessian of $|f|^2$ read as*

$$\nabla_{\mathbb{S}^2}|f(\boldsymbol{x})|^2 = 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\nabla_{\mathbb{S}^2}f(\boldsymbol{x})\right], \tag{3.9}$$

$$\mathrm{H}_{\mathbb{S}^2}|f(\boldsymbol{x})|^2 = 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\mathrm{H}_{\mathbb{S}^2}f(\boldsymbol{x}) + \nabla_{\mathbb{S}^2}f(\boldsymbol{x})\nabla_{\mathbb{S}^2}^\top\overline{f(\boldsymbol{x})}\right]. \tag{3.10}$$

Proof. Let $\boldsymbol{x} := \boldsymbol{x}(\theta,\varphi)$ be given in spherical coordinates, then we have by the product rule the relations

$$\begin{aligned}\frac{\partial}{\partial\theta}|f(\boldsymbol{x})|^2 &= \frac{\partial}{\partial\theta}\left(\overline{f(\boldsymbol{x})}f(\boldsymbol{x})\right) = \overline{f(\boldsymbol{x})}\frac{\partial}{\partial\theta}f(\boldsymbol{x}) + f(\boldsymbol{x})\frac{\partial}{\partial\theta}\overline{f(\boldsymbol{x})}\\ &= 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\frac{\partial}{\partial\theta}f(\boldsymbol{x})\right], \qquad (3.11)\\ \frac{\partial^2}{\partial\theta\partial\varphi}|f(\boldsymbol{x})|^2 &= \frac{\partial}{\partial\theta}\left(\overline{f(\boldsymbol{x})}\frac{\partial}{\partial\varphi}f(\boldsymbol{x}) + f(\boldsymbol{x})\frac{\partial}{\partial\varphi}\overline{f(\boldsymbol{x})}\right)\\ &= 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\frac{\partial^2}{\partial\theta\partial\varphi}f(\boldsymbol{x}) + \frac{\partial}{\partial\theta}\overline{f(\boldsymbol{x})}\cdot\frac{\partial}{\partial\varphi}f(\boldsymbol{x})\right], \qquad (3.12)\end{aligned}$$

and obtain similarly

$$\frac{\partial}{\partial\varphi}|f(\boldsymbol{x})|^2 = 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\frac{\partial}{\partial\varphi}f(\boldsymbol{x})\right], \tag{3.13}$$

$$\frac{\partial^2}{\partial\theta^2}|f(\boldsymbol{x})|^2 = 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\frac{\partial^2}{\partial\theta^2}f(\boldsymbol{x}) + \frac{\partial}{\partial\theta}\overline{f(\boldsymbol{x})}\cdot\frac{\partial}{\partial\theta}f(\boldsymbol{x})\right], \tag{3.14}$$

$$\frac{\partial^2}{\partial\varphi^2}|f(\boldsymbol{x})|^2 = 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\frac{\partial^2}{\partial\varphi^2}f(\boldsymbol{x}) + \frac{\partial}{\partial\varphi}\overline{f(\boldsymbol{x})}\cdot\frac{\partial}{\partial\varphi}f(\boldsymbol{x})\right]. \tag{3.15}$$

We get with the representation (2.3) of the spherical gradient and the relation (3.11) and (3.12) the assertion (3.9). Using (2.4) and (3.13) – (3.15) we infer

$$\mathrm{H}_{\mathbb{S}^2}|f(\boldsymbol{x})|^2 = 2\mathrm{Re}\left[\overline{f(\boldsymbol{x})}\mathrm{H}_{\mathbb{S}^2}f(\boldsymbol{x}) + \begin{pmatrix}\frac{\partial}{\partial\theta}\overline{f(\boldsymbol{x})}\cdot\frac{\partial}{\partial\theta}f(\boldsymbol{x}) & \frac{\partial}{\partial\theta}\overline{f(\boldsymbol{x})}\cdot\frac{1}{\sin\theta}\frac{\partial}{\partial\varphi}f(\boldsymbol{x})\\ \frac{1}{\sin\theta}\frac{\partial}{\partial\varphi}\overline{f(\boldsymbol{x})}\cdot\frac{\partial}{\partial\theta}f(\boldsymbol{x}) & \frac{1}{\sin\theta}\frac{\partial}{\partial\varphi}\overline{f(\boldsymbol{x})}\cdot\frac{1}{\sin\theta}\frac{\partial}{\partial\varphi}f(\boldsymbol{x})\end{pmatrix}\right]$$

and arrive finally at (3.10). ∎

Lemma 3.6. *For the point $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_M) \in \mathbb{S}^2_M$ the gradient $\nabla_{\mathbb{S}^2_M}$ is expressed by*

$$\nabla_{\mathbb{S}^2_M}A_t(\vec{\boldsymbol{x}}) = \frac{2}{M^2}\mathrm{Re}\left[\left(\nabla_{\mathbb{S}^2}^\top p(\boldsymbol{x}_1),\ldots,\nabla_{\mathbb{S}^2}^\top p(\boldsymbol{x}_M)\right)^\top\right], \tag{3.16}$$

and the Hessian $\mathrm{H}_{\mathbb{S}^2_M}$ of $A_t : \mathbb{S}^2_M \to \mathbb{R}$ is represented by

$$\begin{aligned}\mathrm{H}_{\mathbb{S}^2_M}A_t(\vec{\boldsymbol{x}}) = \frac{2}{M^2}\mathrm{Re}\Bigg[&\begin{pmatrix}\mathrm{H}_{\mathbb{S}^2}p(\boldsymbol{x}_1) & & \boldsymbol{0}\\ & \ddots & \\ \boldsymbol{0} & & \mathrm{H}_{\mathbb{S}^2}p(\boldsymbol{x}_M)\end{pmatrix}\\ &+\sum_{n=1}^{t}\sum_{k=-n}^{n}\begin{pmatrix}\nabla_{\mathbb{S}^2}Y_n^k(\boldsymbol{x}_1)\\ \vdots\\ \nabla_{\mathbb{S}^2}Y_n^k(\boldsymbol{x}_M)\end{pmatrix}\left(\nabla_{\mathbb{S}^2}^\top\overline{Y_n^k(\boldsymbol{x}_1)},\ldots,\nabla_{\mathbb{S}^2}^\top\overline{Y_n^k(\boldsymbol{x}_M)}\right)\Bigg],\end{aligned} \tag{3.17}$$

where the spherical polynomial

$$p(\boldsymbol{y}) := \sum_{n=1}^{t} \sum_{k=-n}^{n} \hat{p}_n^k Y_n^k(\boldsymbol{y}) \in \Pi_t(\mathbb{S}^2)$$

is defined by its spherical Fourier coefficients

$$\hat{p}_n^k := \sum_{i=1}^{M} \overline{Y_n^k(\boldsymbol{x}_i)}, \qquad n = 1, \dots, t,\ k = -n, \dots, n. \tag{3.18}$$

Proof. Form equation (2.5) we know that the gradient of A_t is

$$\nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) = \left(\nabla^1_{\mathbb{S}^2} A_t(\vec{\boldsymbol{x}}), \dots, \nabla^M_{\mathbb{S}^2} A_t(\boldsymbol{x})\right). \tag{3.19}$$

With (3.8) and the linearity of $\nabla^l_{\mathbb{S}^2}$ we infer

$$\nabla^l_{\mathbb{S}^2} A_t(\vec{\boldsymbol{x}}) = \frac{1}{M^2} \sum_{n=1}^{t} \sum_{k=-n}^{n} \nabla^l_{\mathbb{S}^2} \left| \sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right|^2.$$

Hence, using (3.9) from Lemma 3.5 we obtain for $n = 1, \dots, t$, $k = -n, \dots, n$, the relation

$$\nabla^l_{\mathbb{S}^2} \left| \sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right|^2 = 2\mathrm{Re} \left[\left(\sum_{i=1}^{M} \overline{Y_n^k(\boldsymbol{x}_i)} \right) \nabla_{\mathbb{S}^2} Y_n^k(\boldsymbol{x}_l) \right] \tag{3.20}$$

and the first assertion (3.16) follows by definition of p and (3.19). For building up the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t$ we need the expressions for $\mathrm{H}^l_{\mathbb{S}^2} A_t$ and $\nabla^l_{\mathbb{S}^2} \nabla^{m\,\top}_{\mathbb{S}^2} A_t$, $l, m = 1, \dots, M$, $l \neq m$, cf. (2.6). From (3.20) we arrive at

$$\begin{aligned} \nabla^l_{\mathbb{S}^2} \nabla^{m\,\top}_{\mathbb{S}^2} \left| \sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right|^2 &= 2\mathrm{Re} \left[\nabla^l_{\mathbb{S}^2} \left(\sum_{i=1}^{M} \overline{Y_n^k(\boldsymbol{x}_i)} \right) \nabla_{\mathbb{S}^2}{}^{\top} Y_n^k(\boldsymbol{x}_m) \right] \\ &= 2\mathrm{Re} \left[\nabla_{\mathbb{S}^2} \overline{Y_n^k(\boldsymbol{x}_l)} \nabla_{\mathbb{S}^2}{}^{\top} Y_n^k(\boldsymbol{x}_m) \right], \end{aligned}$$

which yields by definition (3.8) of A_t the equation

$$\nabla^l_{\mathbb{S}^2} \nabla^{m\,\top}_{\mathbb{S}^2} A_t(\vec{\boldsymbol{x}}) = \frac{2}{M^2} \mathrm{Re} \left[\sum_{n=1}^{t} \sum_{k=-n}^{n} \nabla_{\mathbb{S}^2} \overline{Y_n^k(\boldsymbol{x}_l)} \nabla_{\mathbb{S}^2}{}^{\top} Y_n^k(\boldsymbol{x}_m) \right].$$

From (3.10) of Lemma 3.5 we obtain

$$\mathrm{H}^l_{\mathbb{S}^2} \left| \sum_{i=1}^{M} Y_n^k(\boldsymbol{x}_i) \right|^2 = 2\mathrm{Re} \left[\left(\sum_{i=1}^{M} \overline{Y_n^k(\boldsymbol{x}_i)} \right) \mathrm{H}_{\mathbb{S}^2} Y_n^k(\boldsymbol{x}_l) + \nabla_{\mathbb{S}^2} \overline{Y_n^k(\boldsymbol{x}_l)} \nabla_{\mathbb{S}^2}{}^{\top} Y_n^k(\boldsymbol{x}_l) \right]$$

and after summing up we conclude by definition of p again

$$\mathrm{H}^l_{\mathbb{S}^2} A_t(\vec{\boldsymbol{x}}) = \frac{2}{M^2} \mathrm{Re} \left[\mathrm{H}_{\mathbb{S}^2} p(\boldsymbol{x}_l) + \sum_{n=1}^{t} \sum_{k=-n}^{n} \nabla_{\mathbb{S}^2} \overline{Y_n^k(\boldsymbol{x}_l)} \nabla_{\mathbb{S}^2}{}^{\top} Y_n^k(\boldsymbol{x}_l) \right].$$

Thus, the Hessian of $A_t(\vec{\boldsymbol{x}})$ reads as stated in (3.17). ∎

Theorem 3.7. *For a given point* $\vec{\boldsymbol{x}} := (\boldsymbol{x}(\theta_1, \varphi_1), \dots, \boldsymbol{x}_M(\theta_M, \varphi_M)) \in \mathbb{S}^2_M$ *and a polynomial degree* $t \in \mathbb{N}$ *the calculation of the gradient* $\nabla_{\mathbb{S}^2_M} A_t(\boldsymbol{x})$ *and multiplication of the Hessian* $\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}})$ *with a tangent vector* $\boldsymbol{v} := (\boldsymbol{v}_1^\top, \dots, \boldsymbol{v}_M^\top)^\top \in \mathbb{R}^{2M}$, *cf.* (2.7), *takes* $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ *arithmetic operations, where* ϵ *is a prescribed accuracy.*

Proof. In order to compute the components of the spherical gradient $\nabla_{\mathbb{S}^2_M} A_t(\boldsymbol{x}_1, \dots, \boldsymbol{x}_M)$ we evaluate the spherical Fourier coefficients $\hat{p}_n^k$, cf. (3.18), $n = 1, \dots, t$, $k = -n, \dots, n$, by a realization of the adjoint matrix-vector multiplication with the matrix $\boldsymbol{Y}^t$ in $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ arithmetic operations. Due to the representation, cf. Lemma 3.6

$$\nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) = \frac{2}{M^2} \mathrm{Re} \left[\left(\nabla_{\mathbb{S}^2}^\top p(\boldsymbol{x}_1), \dots, \nabla_{\mathbb{S}^2}^\top p(\boldsymbol{x}_M) \right)^\top \right]$$

we evaluate by Theorem 3.2 the components of the spherical gradient $\nabla_{\mathbb{S}^2} p$ on the points $\boldsymbol{x}_i$, $i = 1, \dots, M$, where we set $p_0^0 := 0$. This is also performed in $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ arithmetic operations.

For the matrix-vector multiplication of the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}})$ with the vector $\boldsymbol{v} \in \mathbb{R}^{2M}$ we proceed as follows. At first we evaluate the Hessian $\mathrm{H}_{\mathbb{S}^2} p$ at the points $\boldsymbol{x}_i$, $i = 1, \dots, M$, as in the case of the gradient by means of Theorem 3.2. After that we simply multiply the obtained 2×2 matrices $\mathrm{H}_{\mathbb{S}^2} p(\boldsymbol{x}_i)$ with the corresponding components $\boldsymbol{v}_i$, $i = 1, \dots, M$ of the vector $\boldsymbol{v}$. This yields the multiplication of the vector $\boldsymbol{v}$ with the first summand of the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}})$, cf. (3.17). For the second summand we define the spherical Fourier coefficients

$$\hat{v}_n^k := \sum_{i=1}^M \nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_i)} \boldsymbol{v}_i = \left(\nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_1)}, \dots, \nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_M)} \right) \boldsymbol{v}, \qquad n = 1, \dots, t,\ k = -n, \dots, n,$$

which is performed by the adjoint transform for evaluating M points of the spherical gradient of a spherical polynomial of degree t, cf. Theorem 3.2. After this intermediate step we define the spherical polynomial

$$V := \sum_{n=1}^t \sum_{n=-k}^k \hat{v}_n^k Y_n^k$$

and compute the spherical gradients $\nabla_{\mathbb{S}^2} V$ at the points $\boldsymbol{x}_i$, $i = 1, \dots, M$. Thus, the i-th component of the vector

$$\sum_{n=1}^t \sum_{k=-n}^n \begin{pmatrix} \nabla_{\mathbb{S}^2} Y_n^k(\boldsymbol{x}_1) \\ \vdots \\ \nabla_{\mathbb{S}^2} Y_n^k(\boldsymbol{x}_M) \end{pmatrix} \left(\nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_1)}, \dots, \nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_M)} \right) \boldsymbol{v},$$

is computed by a nonequispaced spherical Fourier transform, cf. Theorem 3.2,

$$\nabla_{\mathbb{S}^2} V(\boldsymbol{x}_i) = \sum_{n=1}^t \sum_{k=-n}^n \hat{v}_n^k \nabla_{S^2} Y_n^k(\boldsymbol{x}_i).$$

All in all the multiplication $\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) \boldsymbol{v}$ is done in $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ arithmetic operations. ∎

Remark 3.8. In the numerical Section 4 we also consider the approximate Hessian

$$\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) := \frac{2}{M^2}\mathrm{Re}\left[\sum_{n=1}^{t}\sum_{k=-n}^{n}\begin{pmatrix}\nabla_{\mathbb{S}^2} Y_n^k(\boldsymbol{x}_1)\\ \vdots \\ \nabla_{\mathbb{S}^2} Y_n^k(\boldsymbol{x}_M)\end{pmatrix}\left(\nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_1)}, \dots, \nabla_{\mathbb{S}^2}^\top \overline{Y_n^k(\boldsymbol{x}_M)}\right)\right], \quad (3.21)$$

where we dropped the diagonal part in equation (3.17). This approximation is motivated as follows. We consider the function $A_t(\vec{\boldsymbol{x}})$ as residual of the vector-valued function $\boldsymbol{A}_t : \mathbb{S}^2_M \to \mathbb{C}^{(t+1)^2-1}$ with components $(\boldsymbol{A}_t)_n^k := \sum_{i=1}^M Y_n^k(\boldsymbol{x}_i)$, $n = 1, \dots, t$, $k = -n, \dots, n$ and denote by

$$\boldsymbol{J}(\vec{\boldsymbol{x}}) := \left(\nabla_{\mathbb{S}^2}^\top Y_n^k(\boldsymbol{x}_1) \qquad \dots \qquad \nabla_{\mathbb{S}^2}^\top Y_n^k(\boldsymbol{x}_M)\right)_{n=1,\dots,t;\, k=-n,\dots,n}$$

its Jacobian. Then we have the following expressions

$$\begin{aligned}\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) &= \frac{2}{M^2}\mathrm{Re}\left[\overline{\boldsymbol{J}(\vec{\boldsymbol{x}})}^\top \boldsymbol{J}(\vec{\boldsymbol{x}})\right],\\ \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) &= \frac{2}{M^2}\mathrm{Re}\left[\overline{\boldsymbol{J}(\vec{\boldsymbol{x}})}^\top \boldsymbol{A}_t\right].\end{aligned}$$

Hence, the Newton step with the approximate Hessian $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$ solves a normal equation as known from the Gauss-Newton algorithm.

Note that the evaluation of this approximation is more stable than for the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t$, see Figure 4.1. We further remark that from this simplification one also gains an improvement in speed, since less nonequispaced spherical Fourier transforms are necessary, cf. Theorem 3.7. □

It is well known that Newton's method is very sensitive to initial distributions, which might cause some stability problems. Hence, we also consider a stabilized version, like a variant of the Levenberg-Marquardt algorithm, see Algorithm 2. There we replace the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t(\boldsymbol{x}_l)$ by the matrix $\mathrm{H}_{\mathbb{S}^2_M} A_t(\boldsymbol{x}_l) + \|\boldsymbol{g}_l\|_2 \boldsymbol{I}$. In addition we determine in each iteration step l the step length α_l. To this end, we solve (2.8) by a one dimensional Newton method and obtain Algorithm 1. In order to avoid the inversion of the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t(\boldsymbol{x}_l)$ and $\mathrm{H}_{\mathbb{S}^2_M} A_t(\boldsymbol{x}_l) + \|\boldsymbol{g}_l\|_2 \boldsymbol{I}$, respectively, we consider also the nonlinear CG method for computing spherical t-designs. This method is described in Algorithm 3, where we use the line search Algorithm 1, as well. From Theorem 3.7 we conclude that every iteration step of the CG method, cf. Algorithm 3, requires only $\mathcal{O}(t^2 \log^2 t + M \log^2(1/\epsilon))$ arithmetic operations.

4 Numerical results

In this section, we present some numerical examples which show the suitability of the proposed optimization algorithms for computing numerically spherical t-designs. At first we compare the Algorithms 2 and 3 in Example 4.1. Besides the performance of the algorithms we stress the issue of stability. The numerical results indicate that evaluating the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t$ as suggested in Lemma 3.6 and Lemma 3.1 is relatively unstable. The second Example 4.2 is based on the fast evaluation of the matrix times vector multiplication with the more stable evaluation of the matrix $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$. There we show the performance of the nonlinear CG method for high polynomial degrees t.

Algorithm 1 Linesearch$_{A_t}$: Line search for A_t on $\mathbb{S}^2_M$

Input: starting point $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1, \dots, \boldsymbol{x}_M) \in \mathbb{S}^2_M$, descent direction $\boldsymbol{d} \in \mathrm{T}_{\vec{\boldsymbol{x}}}\mathbb{S}^2_M$, accuracy $\varepsilon > 0$, limit of iterations $L_{\max} \in \mathbb{N}$

initialize $l := 0$, $\boldsymbol{g}_0 := \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}) \in \mathrm{T}_{\vec{\boldsymbol{x}}}\mathbb{S}^2_M$, $\alpha_0 := -\frac{\langle \boldsymbol{g}_0, \boldsymbol{d}\rangle}{\langle \boldsymbol{d}, \mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}})\boldsymbol{d}\rangle}$
while $l < L_{\max}$ **do**
  $\vec{\boldsymbol{x}}_{l+1} := \exp_{\vec{\boldsymbol{x}}}(\alpha_l \boldsymbol{d}) \quad \in \mathbb{S}^2_M$
  if $A_t(\vec{\boldsymbol{x}}_{l+1}) > A_t(\vec{\boldsymbol{x}})$ **then**
    $\alpha_{l+1} := \alpha_l/2$ (back-tracking)
  else
    $\boldsymbol{d}_{l+1} := \boldsymbol{P}_{\exp_{\vec{\boldsymbol{x}}}(\alpha_l \boldsymbol{d})}(\boldsymbol{d}) \quad \in \mathrm{T}_{\vec{\boldsymbol{x}}_{l+1}}\mathbb{S}^2_M$
    $\boldsymbol{g}_{l+1} := \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_{l+1}) \quad \in \mathrm{T}_{\vec{\boldsymbol{x}}_{l+1}}\mathbb{S}^2_M$
    if $\varepsilon > \frac{|\langle \boldsymbol{g}_{l+1}, \boldsymbol{d}_{l+1}\rangle|}{\|\boldsymbol{d}_{l+1}\|_2 \|\boldsymbol{g}_{l+1}\|_2}$ **then**
      break
    end if
    $\alpha_{l+1} := \alpha_l - \frac{\langle \boldsymbol{g}_{l+1}, \boldsymbol{d}_{l+1}\rangle}{\langle \mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_{l+1})\boldsymbol{d}_{l+1}, \boldsymbol{d}_{l+1}\rangle}$
  end if
  $l := l + 1$
end while

Output: step length α_{l-1}

In the following examples, we consider two different initial distributions for the proposed methods. The first one is a realization of a random uniform distribution on the sphere $\mathbb{S}^2$, whereas the second one is a relatively uniform distribution given by the Fibonacci spiral on the sphere with M points given by $\boldsymbol{x}(\theta_n, \varphi_n)$, $n := 1, \dots, M$, with

$$\theta_n := \arccos\left(\frac{2n - (M+1)}{M}\right), \qquad \varphi_n := \pi(2n - (M+1))\phi^{-1},$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio, cf. [27]. We rotate these spiral points by a random rotation in order to avoid points on the poles. Other good candidates for relatively uniform distributed points are for example proposed in [24, 8, 10], which behave similarly as initial distribution for computing spherical t-designs.

The Algorithms 1, 2 and 3 are implemented in Matlab R2010a. We used the FFTW 3.2.2 [9] and the NFFT 3.1.3 [16] libraries written in C. The mex-interface of the nfft-library [18] to Matlab was used for performing the nonequispaced fast spherical Fourier transforms. The methods where tested on an Intel Core i7 CPU 920 processor with 12 GB memory and a standard 64 Bit Linux. Throughout our experiments we applied the NFFT routines with precomputed Kaiser–Bessel functions, an oversampling factor of two, and a cutoff parameter $m = 9$. For the NFSFT routines we used the threshold $\kappa = 1000$ for the stabilization. In the Algorithms 1 – 3 we set the accuracy to $\varepsilon = 1\mathrm{e}-13$. We denote the Algorithm 2 using the matrix $\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l)$ and $\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l) + \|\boldsymbol{g}_l\|_2 \boldsymbol{I}$ by 'Newton' and 'Levenberg-Marquardt', respectively. The 'Gauss-Newton' algorithm with the approximate Hessian $\tilde{\mathrm{H}}_{S^2_M}$ behaves similar, thus we omit the numerical results for this method. The occurring matrices are computed

Algorithm 2 Newton-like methods on $\mathbb{S}^2_M$ for computing spherical t-designs

Input: initial distribution $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathbb{S}^2_M$, accuracy $\varepsilon > 0$, limit of iterations $L_{\max} \in \mathbb{N}$

initialize $l := 0$, $\vec{\boldsymbol{x}}_0 := \vec{\boldsymbol{x}}$, $\boldsymbol{g}_0 := \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_0) \in \mathrm{T}_{\vec{\boldsymbol{x}}_0}\mathbb{S}^2_M$
while $\varepsilon < \|\boldsymbol{g}_l\|_2$ and $l < L_{\max}$ **do**

$$\boldsymbol{d}_l := \begin{cases} -\left[\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l)\right]^{-1} \boldsymbol{g}_l & \in \mathrm{T}_{\vec{\boldsymbol{x}}_l}\mathbb{S}^2_M \quad \text{(Newton)} \\ -\left[\mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l) + \|\boldsymbol{g}_l\|_2 \boldsymbol{I}\right]^{-1} \boldsymbol{g}_l & \in \mathrm{T}_{\vec{\boldsymbol{x}}_l}\mathbb{S}^2_M \quad \text{(Levenberg-Marquardt)} \\ -\left[\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l)\right]^{-1} \boldsymbol{g}_l & \in \mathrm{T}_{\vec{\boldsymbol{x}}_l}\mathbb{S}^2_M \quad \text{(Gauss-Newton)} \end{cases}$$

  if $\langle \boldsymbol{d}_l, \boldsymbol{g}_l \rangle \geq 0$ **then**
    $\boldsymbol{d}_l := -\boldsymbol{g}_l$ (enforce descent direction)
  end if
  compute step length $\alpha_l := \mathrm{Linesearch}_{A_t}(\vec{\boldsymbol{x}}_l, \boldsymbol{d}_l)$, cf. Algorithm 1
  $\vec{\boldsymbol{x}}_{l+1} := \exp_{\vec{\boldsymbol{x}}_l}(\alpha_l \boldsymbol{d}_l) \quad \in \mathbb{S}^2_M$
  $\boldsymbol{g}_{l+1} := \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l) \quad \in \mathrm{T}_{\vec{\boldsymbol{x}}_l}\mathbb{S}^2_M$
  $l := l + 1$
end while

Output: numerical spherical t-design $\vec{\boldsymbol{x}}_l \in \mathbb{S}^2_M$

Algorithm 3 Method of conjugate gradients on $\mathbb{S}^2_M$ for computing spherical t-designs

Input: initial distribution $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathbb{S}^2_M$, accuracy $\varepsilon > 0$, limit of iterations $L_{\max} \in \mathbb{N}$, restart interval $r \in N$

initialize $l := 0$, $\vec{\boldsymbol{x}}_0 := \vec{\boldsymbol{x}}$, $\boldsymbol{g}_0 := \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_0) \in \mathrm{T}_{\vec{\boldsymbol{x}}_0}\mathbb{S}^2_M$, $\boldsymbol{d}_0 := -\boldsymbol{g}_0$
while $\varepsilon < \|\boldsymbol{g}_l\|_2$ and $l < L_{\max}$ **do**
  compute step length $\alpha_l := \mathrm{Linesearch}_{A_t}(\vec{\boldsymbol{x}}_l, \boldsymbol{d}_l)$, cf. Algorithm 1
  $\vec{\boldsymbol{x}}_{l+1} := \exp_{\vec{\boldsymbol{x}}_l}(\alpha_l \boldsymbol{d}_l) \quad \in \mathbb{S}^2_M$
  if $l + 1 \equiv 0 \mod r$ **then**
    $\boldsymbol{d}_{l+1} := -\boldsymbol{g}_{l+1}$
  else
    $\tilde{\boldsymbol{d}}_l := \boldsymbol{P}_{\exp_{\vec{\boldsymbol{x}}_l}(\alpha_l \boldsymbol{d}_l)}(\boldsymbol{d}_l) \in \mathrm{T}_{\vec{\boldsymbol{x}}_{l+1}}\mathbb{S}^2_M$
    $\boldsymbol{g}_{l+1} := \nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l) \quad \in \mathrm{T}_{\vec{\boldsymbol{x}}_l}\mathbb{S}^2_M$
    $\beta_l := \max\left\{0, \frac{\langle \boldsymbol{g}_{l+1}, \mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_{l+1})\tilde{\boldsymbol{d}}_l \rangle}{\langle \mathrm{H}_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_{l+1})\tilde{\boldsymbol{d}}_l, \tilde{\boldsymbol{d}}_l \rangle}\right\}$
    $\boldsymbol{d}_{l+1} := -\boldsymbol{g}_{l+1} + \beta_l \tilde{\boldsymbol{d}}_l$
  end if
  $l := l + 1$
end while

Output: numerical spherical t-design $\vec{\boldsymbol{x}}_l \in \mathbb{S}^2_M$

by the fast spherical Fourier transforms, see Lemma 3.6, in $\mathcal{O}(Mt^2\log^2 t + M^2\log^2(1/\epsilon))$ arithmetic operations and the corresponding linear systems are solved by Matlab's standard solver. Further we denote by 'CG with $\mathrm{H}_{\mathbb{S}^2_M} A_t$' the Algorithm 3 using the matrix $\mathrm{H}_{\mathbb{S}^2_M} A_t$ and by 'CG with $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$' the Algorithm 3 using the matrix $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$ in Algorithm 3 and in Algorithm 1. The maximum number of iterations $L_{\max}$ for the line search, Algorithm 1, is 20 with an exception for the conjugate gradient method with the approximated Hessian $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$ of (3.21) where only one iteration is performed.

Example 4.1. Here we consider the computation of spherical 10-designs. For this problem size all proposed algorithms are applicable, and we obtain the results of Table 4.1. We observe a relative high number of iterations for the Newton-like methods, since quadratically convergence is only achieved in very small neighborhood of a stationary point. For this examples the Newton method degenerates to a steepest descent algorithm. Furthermore, the computed spherical t-designs of all methods leads to integration errors $\sqrt{A_t}$ of comparable magnitude. We remark that Hardin and Sloane found a spherical 10-design with $M = 60$ points, see [14, 15]. After several attempts with random initial guesses we where also able to compute various spherical t-designs $\vec{\boldsymbol{x}}_* \in \mathbb{S}^2_{60}$ with integration error $\sqrt{A_{10}(\vec{\boldsymbol{x}}_*)} < 1.0\mathrm{e}-14$.

Furthermore, we want to address the issue of stability for the proposed methods. Therefor we use the fact that the function A_t, its gradient $\nabla_{\mathbb{S}^2_M} A_t$ and its Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t$ are rotational invariant. For instance we know

$$A_t(\boldsymbol{R}\boldsymbol{x}_1, \dots, \boldsymbol{R}\boldsymbol{x}_M) = A_t(\boldsymbol{x}_1, \dots, \boldsymbol{x}_M),$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is an arbitrary rotation matrix, i.e., $\det \boldsymbol{R} = 1$, $\boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}$. For simplicity we use the abbreviation

$$\boldsymbol{R}\vec{\boldsymbol{x}} := (\boldsymbol{R}\boldsymbol{x}_1, \dots, \boldsymbol{R}\boldsymbol{x}_M).$$

Using the rotational invariance property we compare the stability of the CG method with the Hessian matrix $\mathrm{H}_{\mathbb{S}^2_M} A_t$ and its approximation $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$, cf. (3.21), as follows. Let the initial distribution $\vec{\boldsymbol{x}} := (\boldsymbol{x}_1, \dots, \boldsymbol{x}_M)$ and a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ be given. If we denote with the subscript $l \in \mathbb{N}$ the l-th iterated in Algorithm 3 we obtain a simple measure for the stability by the error

$$S(\boldsymbol{R}, l) := \mathrm{d}_{\mathbb{S}^2_M}(\boldsymbol{R}\vec{\boldsymbol{x}}_l, (\boldsymbol{R}\vec{\boldsymbol{x}})_l),$$

since in exact arithmetic this distance is zero. For a random rotation matrix $\boldsymbol{R}$ and the random initial distribution $\vec{\boldsymbol{x}}$ with $M = 60$ points of Table 4.1 we obtain the results shown in Figure 4.1. One recognizes that the conjugate gradient method with the approximated Hessian $\tilde{\mathrm{H}}_{\mathbb{S}^2_M}$ is more stable than the same algorithm with the Hessian $\mathrm{H}_{\mathbb{S}^2_M}$, whereas the results in Table 4.1 are comparable. Similar results are obtained for the Newton-like methods, if we replace the Hessian $\mathrm{H}_{\mathbb{S}^2_M}$ by the approximated Hessian $\tilde{\mathrm{H}}_{\mathbb{S}^2_M}$. □

Example 4.2. From Example 4.1 we conclude that the most efficient and stable algorithm seams to be the CG method with the approximated Hessian $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$. Hence, it is used for the following examples. There, we consider the performance of this algorithm with respect to M, the number of points we spend for achieving a spherical t-design. For comparison we introduce the oversampling factor

$$\sigma(\mathcal{X}_M, t) := \frac{2M-3}{(t+1)^2-1} \approx \frac{2M}{t^2}$$

M	method	$\sqrt{A_{10}(\vec{\boldsymbol{x}}_l)}$	$\|\nabla_{\mathbb{S}^2_M} A_{10}(\vec{\boldsymbol{x}}_l)\|_2$	iteration l	time
60 (random)	Newton	2.8e-4	3.6e-14	458	4min
	Levenberg-Marquardt	9.4e-4	2.0e-14	56	31s
	CG with $\mathrm{H}_{\mathbb{S}^2_M} A_t$	4.9e-4	2.1e-10	2000	107s
	CG with $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$	4.5e-4	1.0e-13	1808	5s
60 (spiral)	Newton	1.6e-3	1.2e-5	2000	18min
	Levenberg-Marquardt	4.4e-4	1.8e-15	19	10s
	CG with $\mathrm{H}_{\mathbb{S}^2_M} A_t$	5.5e-5	1.9e-12	2000	100s
	CG with $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$	5.5e-5	1.0e-13	1710	5s
62 (random)	Newton	3.9e-3	1.7e-5	2000	18min
	Levenberg-Marquardt	2.1e-15	1.3e-15	69	40s
	CG with $\mathrm{H}_{\mathbb{S}^2_M} A_t$	1.1e-8	9.3e-10	2000	86s
	CG with $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$	6.4e-8	3.4e-9	2000	6s
62 (spiral)	Newton	1.7e-3	1.3e-5	2000	18min
	Levenberg-Marquardt	2.2e-15	1.3e-15	54	30s
	CG with $\mathrm{H}_{\mathbb{S}^2_M} A_t$	1.5e-12	8.2e-14	1033	82s
	CG with $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$	1.1e-12	9.4e-14	1170	3s

Table 4.1: Numerical results for computing spherical 10-designs from random and spiral initial distributions $\vec{\boldsymbol{x}}$ with M points. The maximum number of iterations $L_{\max}$ is set to 2000.

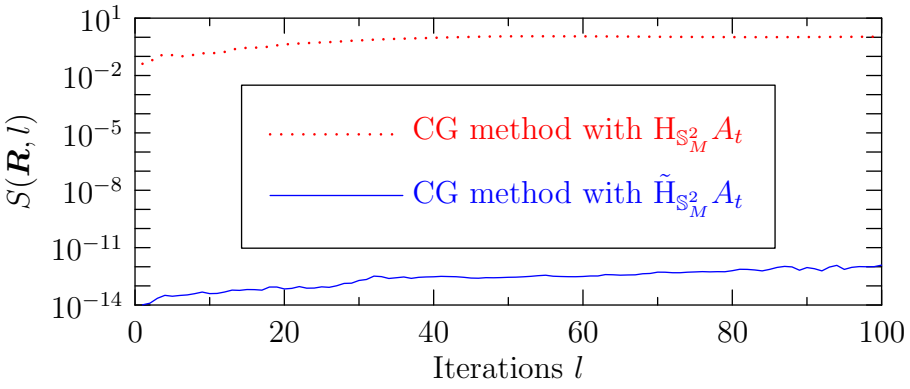

Figure 4.1: Comparison of the error $S(\boldsymbol{R}, l)$ for CG methods with the Hessian $\mathrm{H}_{\mathbb{S}^2_M} A_t$ and its approximation $\tilde{\mathrm{H}}_{\mathbb{S}^2_M} A_t$ respectively.

of a spherical t-design $\mathcal{X}_M \subset \mathbb{S}^2$. This factor is determined by the ratio of the degrees of freedom for choosing the M points on the sphere $\mathbb{S}^2$ up to rotational symmetry, and the number of spherical harmonics Y_n^k we want to integrate exactly by the average over the sampling values. Hence it can be seen as a measure for how far the given spherical t-design is a way from a putatively minimal spherical t-design, with $M \approx \frac{t^2}{2}$ points, i.e., $\sigma(\mathcal{X}_M, t) \approx 1$. This quantity is similar to the efficiency of arbitrary quadrature rules on the

sphere introduced by McLaren, cf. [21]. The numerical results indicate that it is much easier to find numerical spherical t-designs with a little bit more oversampling, say $\sigma(\mathcal{X}_M) \geq 1.05$. We present the results in Table 4.2 and observe that a random distribution seams to be a better initial distribution than the relatively uniform distributed points from the Fibonacci spiral for higher polynomial degrees t. In Figure 4.2 the computed 100-designs for a random and the spiral distribution are illustrated.

□

t	M	$\sigma(\mathcal{X}_M, t)$	$\sqrt{A_t(\vec{\boldsymbol{x}}_l)}$	$\|\nabla_{\mathbb{S}^2_M} A_t(\vec{\boldsymbol{x}}_l)\|_2$	iteration l	time
49	1300 (random)	1.04	5.2e-12	9.2e-14	2211	3min
49	1300 (spiral)	1.04	1.7e-11	9.3e-14	7469	10min
50	1300 (random)	1.00	1.9e-5	8.8e-14	50212	1h
50	1300 (spiral)	1.00	6.8e-6	9.1e-14	96444	2h
100	5200 (random)	1.02	9.9e-12	9.8e-14	4211	27min
100	5200 (spiral)	1.02	1.6e-10	9.7e-14	57235	7.5h
200	21000 (random)	1.04	4.1e-12	9.9e-14	2597	1h
200	21000 (spiral)	1.04	1.0e-9	9.4e-14	173675	3d
500	130000 (random)	1.04	1.0e-11	9.9e-14	5394	21h
1000	520000 (random)	1.04	3.1e-11	1.8e-13	10600	10d
1000	1002000 (random)	2.00	9.7e-12	9.8e-14	4286	5d
1000	1002000 (spiral)	2.00	3.2e-11	9.0e-14	7500	7.5d

Table 4.2: Computing of spherical t-designs by Algorithm 3.

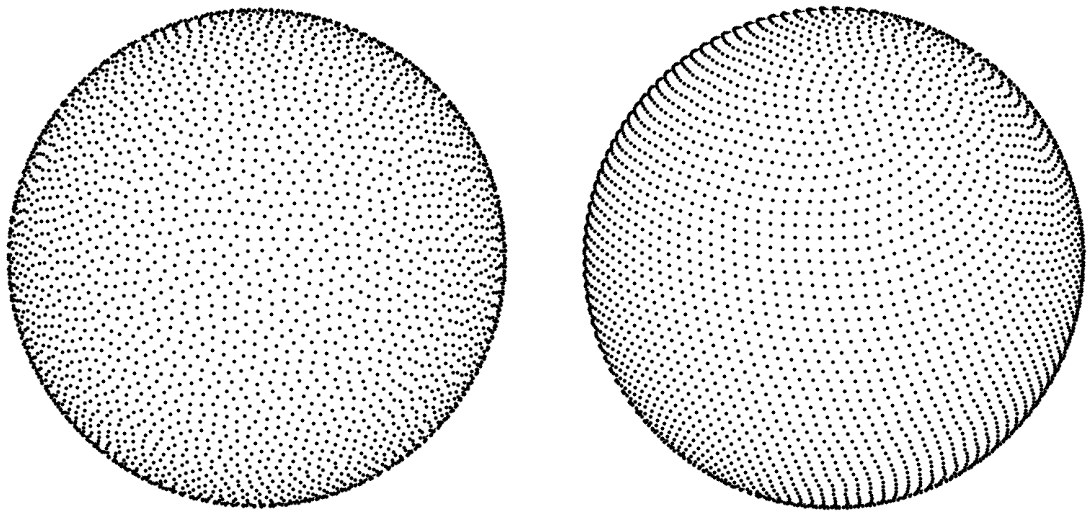

Figure 4.2: Illustration of the computed spherical 100-designs with 5200 random points and sprial points respectively.

In summary we are able to compute numerical spherical t-designs for high polynomial degrees t. The computed spherical t-designs are available from [11].

Acknowledgment

The authors gratefully acknowledge support by German Research Foundation within the project PO 711/9-2.

References

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

[2] A. Bondarenko, D. Radchenko, and M. Viazovska. Optimal asymptotic bounds for spherical designs. *arXiv:1009.4407v3 [math.MG]*, 2011.

[3] X. Chen, A. Frommer, and B. Lang. Computational existence proof for spherical t-designs. *Numer. Math.*, 117, 2010.

[4] L. Conlon. *Differentiable manifolds: a first course*. Birkhäuser Advanced Texts: Basler Lehrbücher. [Birkhäuser Advanced Texts: Basel Textbooks]. Birkhäuser Boston Inc., Boston, MA, 1993.

[5] J. W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM J. Numer. Anal.*, 4:10 – 26, 1967.

[6] P. Delsarte, J. M. Goethals, and J. J. Seidel. Spherical codes and designs. *Geom. Dedicata*, 6:363 – 388, 1977.

[7] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20:303 – 353, 1999.

[8] W. Freeden, T. Gervens, and M. Schreiner. *Constructive Approximation on the Sphere*. Oxford University Press, Oxford, 1998.

[9] M. Frigo and S. G. Johnson. FFTW, C subroutine library. `http://www.fftw.org`, 2009.

[10] K. M. Górski, 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.*, pages 759 – 771, 2005.

[11] M. Gräf. Numerical spherical designs on S^2. `http://www.tu-chemnitz.de/~grman/computations/pointsS2.php`, 2010.

[12] M. Gräf, S. Kunis, and D. Potts. On the computation of nonnegative quadrature weights on the sphere. *Appl. Comput. Harmon. Anal.*, 27:124 – 132, 2009.

[13] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.*, 2:35 – 58, 2006.

[14] R. H. Hardin and N. J. A. Sloane. Mclarens improved snub cube and other new spherical designs in three dimensions. *Discrete and Comput. Geom.*, 15:429 – 441, 1996.

[15] R. H. Hardin and N. J. A. Sloane. Spherical designs, a library of putatively optimal spherical t-designs. `http://www.research.att.com/~njas/sphdesigns`, 2002.

[16] J. Keiner, S. Kunis, and D. Potts. NFFT 3.0, C subroutine library. `http://www.tu-chemnitz.de/~potts/nfft`.

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

[18] J. Keiner, S. Kunis, and D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Software*, 36:Article 19, 1 – 30, 2009.

[19] J. Keiner and D. Potts. Fast evaluation of quadrature formulae on the sphere. *Math. Comput.*, 77:397 – 419, 2008.

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

[21] A. D. McLaren. Optimal numerical integration on a sphere. *Math. Comput.*, 17:361 – 383, 1963.

[22] C. Müller. *Spherical Harmonics*. Springer, Aachen, 1966.

[23] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems Volume II: Standard Information for Functionals*. Eur. Math. Society, EMS Tracts in Mathematics Vol 12, 2010.

[24] E. B. Saff and A. B. J. Kuijlaars. Distributing many points on a sphere. *Math. Intelligencer*, 19:5 – 11, 1997.

[25] I. H. Sloan and R. S. Womersley. A variational characterisation of spherical designs. *J. Approx. Theory*, 159:308 – 318, 2009.

[26] S. T. Smith. Optimization techniques on Riemannian manifolds. In *Hamiltonian and gradient flows, algorithms and control*, volume 3 of *Fields Inst. Commun.*, pages 113 – 136. Amer. Math. Soc., Providence, RI, 1994.

[27] R. Swinbank and R. J. Purser. Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society*, 132:1769 – 1793, 2006.

[28] C. Udrişte. *Convex functions and optimization methods on Riemannian manifolds*, volume 297 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1994.

[29] D. Varshalovich, A. Moskalev, and V. Khersonskii. *Quantum Theory of Angular Momentum*. World Scientific Publishing, Singapore, 1988.