CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.2020-0026

Vol. **1**, No. 4, pp. 766-801
December 2020

A Derivative-Free Geometric Algorithm for Optimization on a Sphere

Yannan Chen[1], Min Xi[2] and Hongchao Zhang[3,*]

[1] *School of Mathematical Sciences, South China Normal University, Guangzhou, China.*
[2] *School of Mathematics and Statistics, Guangdong University of Foreign Studies, Guangzhou China; School of Mathematical Sciences, Jiangsu Key Laboratory for NSLSCS, Nanjing Normal University, Nanjing, China.*
[3] *Department of Mathematics, Louisiana State University, Baton Rouge, LA 70803-4918, USA.*

Received 8 June 2020; Accepted 21 October 2020

Abstract. Optimization on a unit sphere finds crucial applications in science and engineering. However, derivatives of the objective function may be difficult to compute or corrupted by noises, or even not available in many applications. Hence, we propose a Derivative-Free Geometric Algorithm (DFGA) which, to the best of our knowledge, is the first derivative-free algorithm that takes trust region framework and explores the spherical geometry to solve the optimization problem with a spherical constraint. Nice geometry of the spherical surface allows us to pursue the optimization at each iteration in a local tangent space of the sphere. Particularly, by applying Householder and Cayley transformations, DFGA builds a quadratic trust region model on the local tangent space such that the local optimization can essentially be treated as an unconstrained optimization. Under mild assumptions, we show that there exists a subsequence of the iterates generated by DFGA converging to a stationary point of this spherical optimization. Furthermore, under the Łojasiewicz property, we show that all the iterates generated by DFGA will converge with at least a linear or sublinear convergence rate. Our numerical experiments on solving the spherical location problems, subspace clustering and image segmentation problems resulted from hypergraph partitioning, indicate DFGA is very robust and efficient for solving optimization on a sphere without using derivatives.

AMS subject classifications: 65K05, 90C30, 90C56

Key words: Derivative-free optimization, spherical optimization, geometry, trust region method, Łojasiewicz property, global convergence, convergence rate, hypergraph partitioning.

*Corresponding author. *Email addresses:* ynchen@scnu.edu.cn (Y. Chen), mxi@gdufs.edu.cn (M. Xi), hozhang@math.lsu.edu (H. Zhang)

1 Introduction

In this paper, we consider the following spherical optimization problem

$$\min f(\mathbf{x}) \qquad \text{s.t. } \mathbf{x}\in\mathcal{S}^{n-1}, \tag{1.1}$$

where $\mathcal{S}^{n-1} := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}$ is a unit sphere under the Euclidean norm $\|\cdot\|$ and $f : \mathcal{S}^{n-1} \to \mathbb{R}$ is continuously differentiable with Lipschitz continuous gradient. However, we assume that the derivatives of f are unavailable during algorithm designment. The spherical optimization problem (1.1) has extensive applications in science and engineering. For example, the classical Weber problem is to find the best location on a three dimensional sphere which minimizes the weighted sum of the distances to several destination points on the sphere [37,55]. In geophysics, climate modelling and global navigation, various nonlinear optimization problems on a sphere need to be solved for dealing with massive signals on the surface of the earth [13,18]. Finding the largest and smallest Z-eigenvalues of an even order symmetric tensor [26, 48] is equivalent to calculate the maximum and minimum values of a homogeneous polynomial associated with a tensor on a unit sphere, respectively. The best rank-one approximation of a symmetric tensor could be also formulated as a spherical optimization [58]. Other spherical optimization problems which have nonsmooth objectives include the robust subspace detection [29], the sparse principal component analysis (PCA) [2, 54], and the sparse blind deconvolution [17,34] etc. In addition, for some practical applications, the data may come from simulations or experiments, for which the analytic derivatives of the objective function are unavailable or prohibitively expensive to compute. For example, the objective functions proposed in [27, 39] depend on random variables whose distributions are unknown. In particular, for studying precision medicine, it is proposed to maximize the hypervolume under the manifold (HUM) [27], which can be interpreted as the probability of disease detection. Other examples include [28,49], where the evaluation of objective function needs solution of differential equations, and therefore, it is expensive or impossible to compute derivatives of the objective functions at each iteration. Hence, developing derivative-free algorithms for solving (1.1), which only uses the function values of f, has great importance in both theory and applications.

Recently, derivative-free optimization (DFO) has become an important research topic in nonlinear optimization since derivatives of the objective function may be difficult to compute or corrupted by noises, or even not available in many real applications. Hence, DFO methods need to be developed to solve these optimization problems without using derivatives. Currently, the DFO methods can be generally divided into three classes. The first class of methods approximate derivatives by finite-differences and then derivative-based methods can be applied using the approximate derivatives [12,14,32]. For instance, Nocedal et al. [14] combines the classical BFGS updating and an adaptive finite-difference technique for minimization without derivatives. The second class of methods are direct search methods [38], which for example include pattern search methods [33, 53], Nelder-Mead simplex method [43] and mesh adaptive direct search methods [7]. This class of

methods sample the objective functions according to a predetermined pattern strategy. Hence, the direct search methods are usually very robust, but may require a large amount of function evaluations in some applications. The third class of methods are model based methods [10, 41, 46, 59], which build local linear or quadratic models by function interpolations at each iteration. This class of methods intrinsically use the smoothness of the objective function. Hence, fast convergence can be often expected. One may refer to monographs [8, 24] for more general theory and literature review on DFO methods.

Our Derivative-Free Geometric Algorithm (DFGA) developed in this paper belongs to the model based methods, which in the literature include UOBYQA [46], NEWUOA [47], DFO [25], COBYLA [45], DFBOLS [57] and DFO-GN [16], etc. In particular, UOBYQA, NEWUOA and DFO are designed for solving unconstrained optimization. UOBYQA and DFO construct local quadratic approximation models, while NEWUOA uses more flexible models between linear and quadratic to approximate the objective function. Both DFBOLS and DFO-GN are derivative-free Gauss-Newton methods for solving nonlinear least squares optimization, while COBYLA uses local linear approximations of the objective and constraint functions for solving more general constrained optimization. However, to the best of our knowledge, DFGA is the first derivative-free method which is particularly designed to solve the spherical optimization (1.1) and explores the spherical geometry of the constraint. Note that although the spherical constraint is nonconvex, it is a smooth manifold from geometry point of view. At any given point on the sphere, there exists an $(n-1)$-dimensional tangent space of the sphere. Then, using Householder and Cayley transformations, we can establish a bijective map (called a chart) between the unit sphere (except one point) and $\mathbb{R}^{n-1}$ through the tangent space. We will see that the computational costs of the chart and its inverse are only $\mathcal{O}(n)$. Hence, this chart conveniently allows us to locally handle the spherical constraint as simply as $\mathbb{R}^{n-1}$. For solving the spherical optimization (1.1), our DFGA takes a trust region framework. At each iteration of DFGA, function values at $2n-1$ points on the unit sphere are used to construct a function interpolation model. In fact, through the chart, we can find the corresponding $2n-1$ points in $\mathbb{R}^{n-1}$ to build a quadratic model in $\mathbb{R}^{n-1}$ to locally approximate the objective function, which is then minimized inexactly in a trust region. Again through the chart, the approximate minimizer of the trust region model would provide a trial point on the sphere. Because of the bijection mapping by the chart, all the iterates generated by DFGA will be kept strictly feasible on the sphere, which is crucial in many applications since violation of the spherical constraint may lead to nonsense meanings in some applications. On the other hand, we do not resort to traditional techniques for handling nonlinear constraints, such as penalty method, augmented Lagrangian or filter methods. By exploring the spherical geometry of the sphere constraint, our derivative-free trust region approach can be locally simply treated as solving an unconstrained optimization. So, we call our algorithm a derivative-free geometric algorithm.

The following convergence results are established for DFGA. Under the boundness assumption on the Hessian of the local trust region model, we show that there at least exists a subsequence of the iterates generated by DFGA converging to a stationary point

of the spherical optimization (1.1). Furthermore, when the objective function satisfies the Łojasiewicz property, we show that the whole sequence of the iterates generated by DFGA will converge with at least a linear or sublinear convergence rate, which has not been discussed in the derivative-free optimization literature even for the unconstrained case. To verify the efficiency of DFGA, we compare different derivative-free optimization solvers using pattern search, finite difference and model based strategies to solve the classical Weber problem, the spherical location problem, the subspace clustering and image segmentation problems resulted from hypergraph partitioning. Our preliminary numerical results indicate that DFGA is quite robust, efficient and could be very useful for solving optimization on a sphere without using derivatives.

The remainder of the paper is organized as follows. In Section 2, by exploring the topological geometry, we introduce the Householder and Cayley transformations to construct a chart, which establishes a map between a unit sphere (except one point) and $\mathbb{R}^{n-1}$. Our derivative-free geometric algorithm based on a trust region framework is presented in Section 3. The global convergence and the convergence rate of DFGA are analyzed in Section 4. Some preliminary numerical experiments are reported in Section 5 to show the effectiveness of our algorithm. Finally, some conclusions are drawn in the last section.

2 Geometry of a unit sphere

In this section, we consider the geometry of the unit sphere embedded in $\mathbb{R}^n$ under the Euclidean norm (2-norm):

$$\mathcal{S}^{n-1}=\{\mathbf{x}\in\mathbb{R}^n:\|\mathbf{x}\|=1\},$$

where $\|\cdot\|$ denotes 2-norm throughout the paper. Though $\mathcal{S}^{n-1}$ is a nonconvex set, from the geometry point of view it is a smooth manifold [4, 9]. Before going to the concrete concept of manifold, let us recall the concept of a topological space.

Definition 2.1. *A topological space is a set $\mathbb{X}$ together with a family of subsets of $\mathbb{X}$, called the open sets, required to satisfy the following conditions:*

1. *The empty set and $\mathbb{X}$ itself are open;*
2. *If $\mathcal{U},\mathcal{V}\subseteq\mathbb{X}$ are open, so is $\mathcal{U}\cap\mathcal{V}$;*
3. *If the sets $\mathcal{U}_\alpha\subseteq\mathbb{X}$ are open, so is the union $\bigcup\mathcal{U}_\alpha$.*

A function $f:\mathbb{X}\to\mathbb{Y}$ from one topological space to another is defined to be *continuous* if, for any given open set $\mathcal{U}\subseteq\mathbb{Y}$, the inverse image $f^{-1}(\mathcal{U})\subseteq\mathbb{X}$ is open. Given a topological space $\mathbb{X}$ and an open set $\mathcal{U}\subseteq\mathbb{X}$, a *chart* is defined to be a continuous function $\varphi:\mathcal{U}\to\mathbb{R}^d$ with a continuous inverse (the inverse being defined on the set $\varphi(\mathcal{U})$).

Definition 2.2. *A d-dimensional manifold is a topological space $\mathcal{M}$ equipped with charts $\varphi_\alpha: U_\alpha \to \mathbb{R}^d$, where the collection of U_α are open sets covering $\mathcal{M}$, such that the transition function $\varphi_\alpha \circ \varphi_\beta^{-1}$ is smooth at where it is defined.*

Generally speaking, a manifold is a topological space that locally resembles the Euclidean space. The bridge locally connecting the manifold and the Euclidean space is a chart. In the following, we construct a useful chart for the manifold $\mathcal{S}^{n-1}$. For extensive and general discussion of optimization on manifold, one may refer to [4].

2.1 Tangent space

For an arbitrary point $\mathbf{x}$ on the unit sphere $\mathcal{S}^{n-1}$, the *normal space* and *tangent space* of $\mathcal{S}^{n-1}$ at $\mathbf{x}\in\mathcal{S}^{n-1}$ are defined as

$$\mathcal{N}_\mathbf{x}\mathcal{S}^{n-1}=\{\alpha\mathbf{x}\colon \alpha\in\mathbb{R}\} \quad \text{and} \quad \mathcal{T}_\mathbf{x}\mathcal{S}^{n-1}=\{\mathbf{y}\in\mathbb{R}^n\colon \mathbf{y}^T\mathbf{x}=0\}, \tag{2.1}$$

respectively. Hence, the normal space $\mathcal{N}_\mathbf{x}\mathcal{S}^{n-1}$ is a straight line with dimension 1 and the dimension of the tangent space $\mathcal{T}_\mathbf{x}\mathcal{S}^{n-1}$ is $n-1$.

To identify the tangent space, we study a bijection that maps $\mathcal{T}_\mathbf{x}\mathcal{S}^{n-1}$ to $\mathbb{R}^{n-1}$. Let $Q\in\mathbb{R}^{n\times(n-1)}$ be an orthonormal matrix such that

$$Q^TQ=I \quad \text{and} \quad Q^T\mathbf{x}=\mathbf{0}, \tag{2.2}$$

where $I\in\mathbb{R}^{(n-1)\times(n-1)}$ is the identity matrix. So, the columns of Q form a basis of the tangent space $\mathcal{T}_\mathbf{x}\mathcal{S}^{n-1}$. Then, for any $\mathbf{y}\in\mathcal{T}_\mathbf{x}\mathcal{S}^{n-1}$, there exists a unique $\mathbf{z}\in\mathbb{R}^{n-1}$ such that

$$\mathbf{y}=Q\mathbf{z} \quad \Longleftrightarrow \quad \mathbf{z}=Q^T\mathbf{y}. \tag{2.3}$$

While $\mathbf{y}$ is restricted in the tangent space (i.e., $\mathbf{y}^T\mathbf{x}=0$), the vector $\mathbf{z}$ is free in $\mathbb{R}^{n-1}$. Hence, it will be much convenient to construct local interpolation models based on the vector $\mathbf{z}$ for derivative-free optimization methods.

Note that the matrix Q in (2.2) is not unique. For computational efficiency, we can use the Householder transformation to generate Q [19]. Given $\mathbf{x}\in\mathcal{S}^{n-1}$ and $\mathbf{x}\neq\mathbf{e}_1$, where $\mathbf{e}_1=(1,0,\cdots,0)^T$, let

$$\mathbf{v}=\mathbf{x}-\mathbf{e}_1 \quad \text{and} \quad \beta=\frac{2}{\mathbf{v}^T\mathbf{v}}.$$

Then, the Householder matrix $\widetilde{Q}=I-\beta\mathbf{v}\mathbf{v}^T$ is an orthogonal matrix that maps $\mathbf{x}$ to $\mathbf{e}_1$, i.e., $\widetilde{Q}^T\widetilde{Q}=I$ and $\widetilde{Q}\mathbf{x}=\mathbf{e}_1$. Clearly, the second to the last columns of $\widetilde{Q}$ would make up a matrix Q satisfying (2.2). Hence, given $\mathbf{z}\in\mathbb{R}^{n-1}$, to calculate $\mathbf{y}=Q\mathbf{z}$, we can let

$$\widetilde{\mathbf{z}}=\begin{pmatrix}0\\ \mathbf{z}\end{pmatrix} \quad \text{and compute} \quad \mathbf{y}=Q\mathbf{z}=\widetilde{Q}\widetilde{\mathbf{z}}=(I-\beta\mathbf{v}\mathbf{v}^T)\widetilde{\mathbf{z}}=\widetilde{\mathbf{z}}-(\beta\mathbf{v}^T\widetilde{\mathbf{z}})\mathbf{v}.$$

On the other hand, given $\mathbf{y}\in\mathbb{R}^n$, to calculate $\mathbf{z}=Q^T\mathbf{y}$, we can compute

$$\widetilde{\mathbf{z}}=\widetilde{Q}^T\mathbf{y}=(I-\beta\mathbf{v}\mathbf{v}^T)\mathbf{y}=\mathbf{y}-(\beta\mathbf{v}^T\mathbf{y})\mathbf{v} \quad \text{and let} \quad \mathbf{z}=\widetilde{\mathbf{z}}(2\colon n).$$

So, once $\mathbf{x}\in\mathcal{S}^{n-1}$ is given, it only requires about 3 operations to calculate $\mathbf{v}$ and β. And the computation of $\mathbf{y}=Q\mathbf{z}$ or $\mathbf{z}=Q^T\mathbf{y}$ only needs about $4n$ operations.[†] Hence, the Householder transformation provides us a reliable and efficient way to compute the translations between $\mathbf{y}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$ and $\mathbf{z}\in\mathbb{R}^{n-1}$.

2.2 Cayley transformation

The Householder transformation constructed in the previous subsection provides us a bijection between a tangent space and $\mathbb{R}^{n-1}$. Now, we build a bijection between the tangent space and a subset of the unit sphere using Cayley transformation. Let $W\in\mathbb{R}^{n\times n}$ be a skew-symmetric matrix and I be the identity matrix in $\mathbb{R}^{n\times n}$. Then $I+W$ is invertible. Cayley transformation produces an orthogonal matrix

$$O=(I+W)^{-1}(I-W), \tag{2.4}$$

whose eigenvalues do not contain -1. The converse is also true, i.e., $W=(I+O)^{-1}(I-O)$ is skew-symmetric if O is orthogonal and $I+O$ is invertible. Hence, Cayley transformation reveals a valuable relationship between orthogonal matrices and skew-symmetric matrices.

Given the current point $\mathbf{x}\in\mathcal{S}^{n-1}$ and the search direction $\mathbf{s}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$, let

$$W=\frac{1}{2}(\mathbf{x}\mathbf{s}^T-\mathbf{s}\mathbf{x}^T), \tag{2.5}$$

which is a skew-symmetric matrix. Then, the matrix O constructed in (2.4) is an orthogonal matrix and hence, we have

$$\mathbf{x}_+=O\mathbf{x}\in\mathcal{S}^{n-1}. \tag{2.6}$$

More specifically, we can explicitly obtain $\mathbf{x}_+$ by the following formula.

Lemma 2.1 ([36]). *Let $\mathbf{x}\in\mathcal{S}^{n-1}$ and $\mathbf{s}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$. Suppose that O,W and $\mathbf{x}_+$ are defined by* (2.4), (2.5) *and* (2.6), *respectively. Then, we have*

$$\mathbf{x}_+=\frac{(4-\|\mathbf{s}\|^2)\mathbf{x}+4\mathbf{s}}{4+\|\mathbf{s}\|^2}. \tag{2.7}$$

By (2.7), it is clear that the new point $\mathbf{x}_+$ is located in the plane spanned by the vectors $\mathbf{x}$ and $\mathbf{s}$. When the norm of $\mathbf{s}$ goes to zero, the new point $\mathbf{x}_+$ tends to $\mathbf{x}$. And when the norm of $\mathbf{s}$ becomes large, the $\mathbf{x}_+$ would tend to $-\mathbf{x}$, which is the opposite point of $\mathbf{x}$ on the unit sphere. Observe that there is no need to store the matrices O and W, and the

[†]Since $\mathbf{v}=\mathbf{x}-\mathbf{e}_1$, it only needs 1 subtraction for computing $\mathbf{v}$. Since $\mathbf{v}^T\mathbf{v}=\|\mathbf{x}\|^2-2x_1+1=2(1-x_1)$, we have $\beta=1/(1-x_1)$. Hence, it requires 2 operations to calculate β. Then, it costs $2n$ operations to compute $\beta\mathbf{v}^T\widetilde{\mathbf{z}}$ or $\beta\mathbf{v}^T\mathbf{y}$. Thereafter, we perform n multiplications and n subtractions to calculate $\widetilde{\mathbf{z}}-(\beta\mathbf{v}^T\widetilde{\mathbf{z}})\mathbf{v}$ or $\mathbf{y}-(\beta\mathbf{v}^T\mathbf{y})\mathbf{v}$.

new point $\mathbf{x}_+$ can be easily computed by (2.7) using about $5n$ operations. Indeed, we can define the following map:

$$\begin{aligned} \mathrm{Cay}_{\mathbf{x}}\colon\ \mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1} &\to \mathcal{S}^{n-1}\backslash\{-\mathbf{x}\} \\ \mathbf{s} &\mapsto \frac{(4-\|\mathbf{s}\|^2)\mathbf{x}+4\mathbf{s}}{4+\|\mathbf{s}\|^2}. \end{aligned} \tag{2.8}$$

The following lemma gives the inverse of $\mathrm{Cay}_{\mathbf{x}}$.

Lemma 2.2. *Let* $\mathbf{x}\in\mathcal{S}^{n-1}$. *The map* (2.8) *is a bijection and*

$$\begin{aligned} \mathrm{Cay}_{\mathbf{x}}^{-1}\colon\ \mathcal{S}^{n-1}\backslash\{-\mathbf{x}\} &\to \mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1} \\ \mathbf{x}_+ &\mapsto \frac{2(I-\mathbf{x}\mathbf{x}^T)\mathbf{x}_+}{1+\mathbf{x}_+^T\mathbf{x}}. \end{aligned} \tag{2.9}$$

Proof. For a given $\mathbf{x}_+\in\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}$, we will find a unique $\mathbf{s}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$ such that $\mathbf{x}_+=\mathrm{Cay}_{\mathbf{x}}(\mathbf{s})$. From (2.7), we know

$$4\mathbf{s}=(4+\|\mathbf{s}\|^2)\mathbf{x}_+-(4-\|\mathbf{s}\|^2)\mathbf{x}. \tag{2.10}$$

By taking norms of both sides of the above equation, it yields that

$$16\|\mathbf{s}\|^2=(4+\|\mathbf{s}\|^2)^2+(4-\|\mathbf{s}\|^2)^2-2(4+\|\mathbf{s}\|^2)(4-\|\mathbf{s}\|^2)\mathbf{x}_+^T\mathbf{x},$$

which gives

$$2(4-\|\mathbf{s}\|^2)\Big((1+\mathbf{x}_+^T\mathbf{x})\|\mathbf{s}\|^2-4(1-\mathbf{x}_+^T\mathbf{x})\Big)=0.$$

Note that $1+\mathbf{x}_+^T\mathbf{x}\neq 0$ since $\mathbf{x}_+\in\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}$. Hence, we have

$$\|\mathbf{s}\|^2=4 \quad \text{or} \quad \|\mathbf{s}\|^2=4(1-\mathbf{x}_+^T\mathbf{x})/(1+\mathbf{x}_+^T\mathbf{x}).$$

If $\|\mathbf{s}\|^2=4$, we have by (2.10) that $2\mathbf{x}_+=\mathbf{s}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$ and therefore, $\mathbf{x}_+^T\mathbf{x}=0$. Hence, we always have

$$\|\mathbf{s}\|^2=\frac{4(1-\mathbf{x}_+^T\mathbf{x})}{1+\mathbf{x}_+^T\mathbf{x}},$$

which together with (2.10) gives

$$\mathbf{s}=\frac{2(I-\mathbf{x}\mathbf{x}^T)\mathbf{x}_+}{1+\mathbf{x}_+^T\mathbf{x}}.$$

Clearly, $\mathbf{s}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$ since $I-\mathbf{x}\mathbf{x}^T$ is a projection matrix onto the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$. □

Now, we are ready to construct our required chart explicitly.

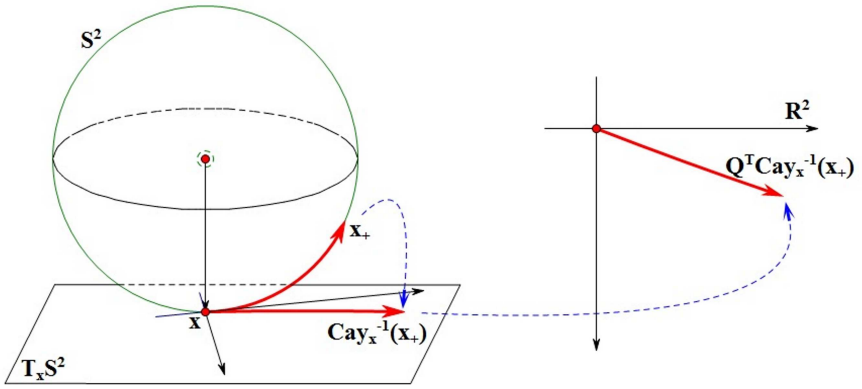


Figure 1: An illustration of the chart $\varphi_{\mathbf{x}}$.

Theorem 2.1. *Let* $\mathbf{x}\in\mathcal{S}^{n-1}$. *The map*

$$\begin{aligned}\varphi_{\mathbf{x}}\colon\ \mathcal{S}^{n-1}\backslash\{-\mathbf{x}\} &\to \mathbb{R}^{n-1}\\ \mathbf{x}_+ &\mapsto Q^T\mathrm{Cay}_{\mathbf{x}}^{-1}(\mathbf{x}_+)\end{aligned}\tag{2.11}$$

is a chart, whose inverse is

$$\begin{aligned}\varphi_{\mathbf{x}}^{-1}\colon\ \mathbb{R}^{n-1} &\to \mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}\\ \mathbf{z} &\mapsto \mathrm{Cay}_{\mathbf{x}}(Q\mathbf{z}),\end{aligned}\tag{2.12}$$

where $Q\in\mathbb{R}^{n\times(n-1)}$ *is any matrix satisfying* (2.2).

Proof. Consider the local geometry around $\mathbf{x}\in\mathcal{S}^{n-1}$ as illustrated in Fig. 1. For any point $\mathbf{x}_+\in\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}$, we have $\mathrm{Cay}_{\mathbf{x}}^{-1}(\mathbf{x}_+)\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1}$ by Lemma 2.2. Then, by (2.3), we obtain $Q^T\mathrm{Cay}_{\mathbf{x}}^{-1}(\mathbf{x}_+)\in\mathbb{R}^{n-1}$. In a word, we say that the chart $\varphi_{\mathbf{x}}$ acts

$$\mathbf{x}_+\in\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\} \quad\mapsto\quad \mathrm{Cay}_{\mathbf{x}}^{-1}(\mathbf{x}_+)\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1} \quad\mapsto\quad Q^T\mathrm{Cay}_{\mathbf{x}}^{-1}(\mathbf{x}_+)\in\mathbb{R}^{n-1}.$$

Conversely and similarly, the inverse $\varphi_{\mathbf{x}}^{-1}$ of the chart acts

$$\mathbf{z}\in\mathbb{R}^{n-1} \quad\mapsto\quad Q\mathbf{z}\in\mathcal{T}_{\mathbf{x}}\mathcal{S}^{n-1} \quad\mapsto\quad \mathrm{Cay}_{\mathbf{x}}(Q\mathbf{z})\in\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}.$$

Finally, by the definitions of $\mathrm{Cay}_{\mathbf{x}}$ and $\mathrm{Cay}_{\mathbf{x}}^{-1}$ in (2.8) and (2.9), it is clear that both $\varphi_{\mathbf{x}}$ and $\varphi_{\mathbf{x}}^{-1}$ are continuous map on their domain. The proof is complete. □

Note that the computational costs of $\mathrm{Cay}_{\mathbf{x}}$ and $\mathrm{Cay}_{\mathbf{x}}^{-1}$ are all about $5n$ operations. Hence, computing the chart or its inverse acting on a vector in Theorem 2.1 only takes about $9n$ operations.

3 A derivative-free geometric algorithm

The geometric algorithm proposed in this section is a feasible trust region method. Given an initial point on the unit sphere $\mathcal{S}^{n-1}$, we establish a local quadratic model of the objective function in a trust region by function value interpolations. Then, we find an approximate minimizer of the model in this trust region, which is a candidate of the next iterate on the sphere. This candidate will be either accepted or further improved in the framework of trust region methods. During this procedure, the chart introduced in the previous section enables us to locally handle sphere constrained optimization simply as an unconstrained optimization.

3.1 Interpolation

For unconstrained optimization without derivatives, it is well-studied to construct a linear or quadratic model by function value interpolations [21, 22]. In this subsection, we would generalize the minimum Frobenius norm model used in derivative-free methods for unconstrained optimization to our case where it has a sphere constraint.

Let $\mathbf{x}\in\mathcal{S}^{n-1}$ be the current iteration point. Since the local chart $\varphi_{\mathbf{x}}:\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}\to\mathbb{R}^{n-1}$ is bijective by Theorem 2.1, we can map every point $\mathbf{z}\in\mathbb{R}^{n-1}$ to $\varphi_{\mathbf{x}}^{-1}(\mathbf{z})$ on the sphere and then evaluate the function value $f(\varphi_{\mathbf{x}}^{-1}(\mathbf{z}))$. That is to say, we can define a local surrogate function

$$\begin{aligned}\widehat{f}_{\mathbf{x}}:\mathbb{R}^{n-1} &\to \mathbb{R}\\ \mathbf{z} &\mapsto (f\circ\varphi_{\mathbf{x}}^{-1})(\mathbf{z}),\end{aligned} \tag{3.1}$$

which would capture the contour profile of the objective function around $\mathbf{x}$. Note that $\widehat{f}_{\mathbf{x}}(\mathbf{0})=f(\mathbf{x})$. Now, suppose that we have a set of p ($n\le p\le\frac{1}{2}n(n+1)$) points on the unit sphere

$$\mathbf{x}^1,\mathbf{x}^2,\cdots,\mathbf{x}^p\in\mathcal{S}^{n-1}\backslash\{-\mathbf{x}\}$$

with known function values $f^i=f(\mathbf{x}^i)$ for $i=1,2,\cdots,p$. Using the chart $\varphi_{\mathbf{x}}$, we let

$$\mathbf{z}^i=\varphi_{\mathbf{x}}(\mathbf{x}^i)\in\mathbb{R}^{n-1},\quad i=1,2,\cdots,p.$$

Then, by the definition of local surrogate function (3.1), we know

$$\widehat{f}_{\mathbf{x}}(\mathbf{z}^i)=f^i,\quad i=1,2,\cdots,p.$$

Hence, we obtain a set of p points $\mathcal{Z}:=\{\mathbf{z}^1,\mathbf{z}^2,\cdots,\mathbf{z}^p\}\subset\mathbb{R}^{n-1}$ with their function values $\widehat{f}_{\mathbf{x}}(\mathbf{z}^i)$, $i=1,\cdots,p$.

Let $\mathbb{P}_{n-1}^d$ be the space of multivariate polynomials defined on $\mathbb{R}^{n-1}$ with degree less than or equal to d. Then, the set of $\frac{1}{2}n(n+1)$ monomials

$$\phi(\mathbf{z})=\{\phi_i,i=1,\cdots,\tfrac{1}{2}n(n+1)\}:=\{1,z_1,\cdots,z_{n-1},\tfrac{1}{2}z_1^2,z_1z_2,\cdots,\tfrac{1}{2}z_{n-1}^2\}$$

forms a natural basis of $\mathbb{P}^2_{n-1}$. We are going to construct a quadratic model

$$m(\mathbf{z}) := \frac{1}{2}\mathbf{z}^T H\mathbf{z} + \mathbf{g}^T\mathbf{z} + c = \alpha^T \phi(\mathbf{z}) \tag{3.2}$$

of $\widehat{f}_{\mathbf{x}}$ satisfying the interpolation linear system

$$M(\phi, \mathcal{Z})\alpha = \widehat{f}_{\mathbf{x}}(\mathcal{Z}), \tag{3.3}$$

where $H \in \mathbb{S}^{(n-1)\times(n-1)}$, $\mathbf{g} \in \mathbb{R}^{n-1}$, and $c \in \mathbb{R}$ are the model unknowns need to be determined. In addition, $\mathbb{S}^{(n-1)\times(n-1)}$ denotes the set of $(n-1)\times(n-1)$ symmetric matrices, $\alpha \in \mathbb{R}^{n(n+1)/2}$ assembles $\{c, \mathbf{g}, H\}$ according to the order of the monomial bases in $\phi(\mathbf{z})$,

$$M(\phi, \mathcal{Z}) = \begin{pmatrix} \phi_1(\mathbf{z}^1) & \phi_2(\mathbf{z}^1) & \cdots & \phi_{n(n+1)/2}(\mathbf{z}^1) \\ \phi_1(\mathbf{z}^2) & \phi_2(\mathbf{z}^2) & \cdots & \phi_{n(n+1)/2}(\mathbf{z}^2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{z}^p) & \phi_2(\mathbf{z}^p) & \cdots & \phi_{n(n+1)/2}(\mathbf{z}^p) \end{pmatrix} \quad \text{and} \quad \widehat{f}_{\mathbf{x}}(\mathcal{Z}) = \begin{pmatrix} \widehat{f}_{\mathbf{x}}(\mathbf{z}^1) \\ \widehat{f}_{\mathbf{x}}(\mathbf{z}^2) \\ \vdots \\ \widehat{f}_{\mathbf{x}}(\mathbf{z}^p) \end{pmatrix}.$$

When $n \le p < \frac{1}{2}n(n+1)$, there are more unknowns than the number of equations in the linear system (3.3). Thus, to construct model (3.2), we would like to require the minimum Frobenius norm on its Hessian matrix H (see [24]). Let

$$\phi_L(\mathbf{z}) := \{1, z_1, \cdots, z_{n-1}\} \quad \text{and} \quad \phi_Q(\mathbf{z}) := \left\{\tfrac{1}{2}z_1^2, z_1z_2, \cdots, \tfrac{1}{2}z_{n-1}^2\right\}$$

be the linear and quadratic parts of the basis $\phi(\mathbf{z})$, respectively, and α be also partitioned into α_L and α_Q accordingly. Then, the unknown α is determined by solving the following quadratic optimization problem

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^{n(n+1)/2}} \ & \frac{1}{2}\|\alpha_Q\|^2 \\ \text{s.t.} \ & M(\phi_Q, \mathcal{Z})\alpha_Q + M(\phi_L, \mathcal{Z})\alpha_L = \widehat{f}_{\mathbf{x}}(\mathcal{Z}). \end{aligned} \tag{3.4}$$

The optimization problem (3.4) has one unique solution if the following matrix is nonsingular

$$F(\phi, \mathcal{Z}) := \begin{pmatrix} M(\phi_Q, \mathcal{Z})M(\phi_Q, \mathcal{Z})^T & M(\phi_L, \mathcal{Z}) \\ M(\phi_L, \mathcal{Z})^T & 0 \end{pmatrix}. \tag{3.5}$$

If the matrix (3.5) is nonsingular, we say that the interpolation set $\mathcal{Z}$ is poised in the minimum Frobenius norm sense. Now, we give the definition of Λ-poisedness.

Definition 3.1 ([24]). *Let $\Lambda > 0$ and $\mathcal{B}(\Delta) := \{\mathbf{z} \in \mathbb{R}^{n-1} : \|\mathbf{z}\| \le \Delta\} \subset \mathbb{R}^{n-1}$. Then, a poised set $\mathcal{Z} = \{\mathbf{z}^1, \cdots, \mathbf{z}^p\}$ is said to be Λ-poised in $\mathcal{B}(\Delta)$ (in the minimum Frobenius norm sense) if and only if, for any $\mathbf{z} \in \mathcal{B}(\Delta)$, there exists a solution $\lambda(\mathbf{z}) \in \mathbb{R}^p$ of*

$$\begin{aligned} \min \ & \|M(\phi_Q, \mathcal{Z})^T\lambda(\mathbf{z}) - \phi_Q(\mathbf{z})\|^2 \\ \text{s.t.} \ & M(\phi_L, \mathcal{Z})^T\lambda(\mathbf{z}) = \phi_L(\mathbf{z}) \end{aligned} \tag{3.6}$$

such that

$$\|\lambda(\mathbf{z})\|_\infty \leq \Lambda.$$

Note that the optimization problem (3.6) has a unique solution when the matrix $F(\phi,\mathcal{Z})$ in (3.5) is nonsingular, i.e., the set $\mathcal{Z}$ is poised (in the minimum Frobenius norm sense). In fact, the solution $\lambda(\mathbf{z})$ of (3.6) is just the Lagrange polynomial for the set $\mathcal{Z}$ with minimum Frobenius norm of the Hessian [24].

Suppose that we are given any set $\mathcal{Z}\subset\mathcal{B}(\Delta)$ with $n\leq|\mathcal{Z}|\leq\frac{1}{2}n(n+1)$. Here, $|\cdot|$ means the cardinality of a set. We can apply a finite number of substitutions of the points in $\mathcal{Z}$, in fact, at most $|\mathcal{Z}|-1$ points, such that the new resultant set is Λ-poised in $\mathcal{B}(\Delta)$ for a polynomial space $\mathbb{P}$, with dimension $|\mathcal{Z}|$ and $\mathbb{P}^1_{n-1}\subseteq\mathbb{P}\subseteq\mathbb{P}^2_{n-1}$ [51, 56, 57]. Since the selection of interpolation points and their poisedness are beyond the main scope of this paper, we refer the reader to [21,22,24]. Once the interpolation set $\mathcal{Z}$ is Λ-poised in $\mathcal{B}(\Delta)$, the interpolating polynomial obtained from (3.4) will be at least a fully linear model, as stated in the following lemma [57].

Lemma 3.1. *Given any $\Delta>0$ and Λ-poised set $\mathcal{Z}\subset\mathcal{B}(\Delta)$ with $n\leq|\mathcal{Z}|\leq\frac{1}{2}n(n+1)$. Let the interpolating model* (3.2) *be obtained from* (3.4). *If $\widehat{f}_{\mathbf{x}}:\mathbb{R}^{n-1}\to\mathbb{R}$ is continuously differentiable and $\nabla\widehat{f}_{\mathbf{x}}$ is Lipschitz continuous with Lipschitz constant L in an open set containing $\mathcal{B}(\Delta)$, then, for any $\mathbf{d}\in\mathcal{B}(\Delta)$, we have*

$$\begin{aligned}&\|\nabla\widehat{f}_{\mathbf{x}}(\mathbf{d})-\nabla m(\mathbf{d})\|\leq\widehat{\kappa}_{eg}(\|H\|+L)\Delta,\\&|\widehat{f}_{\mathbf{x}}(\mathbf{d})-m(\mathbf{d})|\leq\widehat{\kappa}_{ef}(\|H\|+L)\Delta^2,\end{aligned}$$

where $\widehat{\kappa}_{eg}$ and $\widehat{\kappa}_{ef}$ are constants depending only on n and Λ.

3.2 A trust region framework

Our trust region method for solving spherical optimization (1.1) is motivated and designed to have a similar spirit of the derivative-free trust region methods for unconstrained optimization [11, 23, 24, 31]. When derivatives are available, Absil et al. [1] designed the trust region method for Riemannian manifold that includes the spherical surface as a special case. At the current iteration point $\mathbf{x}_k\in\mathcal{S}^{n-1}$, we choose a set of interpolating points and construct a local quadratic model

$$m_k(\mathbf{d})=\frac{1}{2}\mathbf{d}^TH_k\mathbf{d}+\mathbf{g}_k^T\mathbf{d}+c_k,$$

using the method discussed in the previous subsection. Then, the following trust region subproblem

$$\min_{\mathbf{d}\in\mathbb{R}^{n-1}} m_k(\mathbf{d}) \qquad \text{s.t. } \|\mathbf{d}\|\leq\Delta_k, \tag{3.12}$$

Algorithm 3.1: A Derivative-Free Geometric Algorithm (DFGA) for spherical optimization

1: **Step 0: Initialization.**
Set positive parameters Λ, $0<\eta<1<\eta_1$, $0<\tau_0\ll\tau$, $\gamma_1<1<\gamma_2$, and $\varrho<\widetilde{\Delta}_0\leq\Delta_{\max}$. Sample a set of $p=2n-1$ points $\mathcal{X}_0=\{\mathbf{x}^1,\mathbf{x}^2,\cdots,\mathbf{x}^p\}\subset\mathcal{S}^{n-1}$ uniformly and evaluate function values therein $f(\mathcal{X}_0)$. Choose the best point $\mathbf{x}_0\in\mathcal{X}_0$ such that $f(\mathbf{x}_0)=\min_{1\leq i\leq p}f(\mathbf{x}^i)$. Set $k\leftarrow 0$.

2: **Step 1: Construct interpolation model.**
Compute $\mathcal{Z}_k=\{\mathbf{z}^i=\varphi_{\mathbf{x}_k}(\mathbf{x}^i):\mathbf{x}^i\in\mathcal{X}_k\}\subset\mathbb{R}^{n-1}$. Applying the minimum Frobenius norm model for $\mathcal{Z}_k$ and $\widehat{f}_{\mathbf{x}_k}(\mathcal{Z}_k)=f(\mathcal{X}_k)$, we have a quadratic model:

$$m_k(\mathbf{d})=\frac{1}{2}\mathbf{d}^T H_k\mathbf{d}+\mathbf{g}_k^T\mathbf{d}+c_k \tag{3.7}$$

and then set

$$\Delta_k=\min\{\widetilde{\Delta}_k,\tau\|\mathbf{g}_k\|\}. \tag{3.8}$$

If $\Delta_k\leq\varrho$, we ensure the Λ-poisedness of $\mathcal{Z}_k$. For this purpose, we possibly choose a new $\Delta_k\in(0,\widetilde{\Delta}_k]$, adjust the interpolation set $\mathcal{Z}_k$, and update H_k and $\mathbf{g}_k$ in the model m_k accordingly such that

$$\Delta_k=\min\{\max\{\tau_k\|\mathbf{g}_k\|,\widetilde{\Delta}_k\},\tau\|\mathbf{g}_k\|\}, \tag{3.9}$$

and $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$.

3: **Step 2: Compute a trial point.**
Solve the trust region subproblem (3.12) inexactly to obtain $\mathbf{d}_k$ satisfying (3.13), and then generate the feasible trial point

$$\mathbf{x}_k^+=\varphi_{\mathbf{x}_k}^{-1}(\mathbf{d}_k).$$

4: **Step 3: Update the iterate and the trust region radius.**
Evaluate $f(\mathbf{x}_k^+)$ and compute

$$\rho_k=\frac{f(\mathbf{x}_k^+)-f(\mathbf{x}_k)}{m_k(\mathbf{d}_k)-m_k(\mathbf{0})}. \tag{3.10}$$

If $\rho_k\geq\eta$, accept the trial point $\mathbf{x}_{k+1}=\mathbf{x}_k^+$, let $\tau_{k+1}=\tau_k$ and set

$$\widetilde{\Delta}_{k+1}\in[\Delta_k,\min\{\gamma_2\Delta_k,\Delta_{\max}\}];$$

Otherwise, set $\mathbf{x}_{k+1}=\mathbf{x}_k$, $\widetilde{\Delta}_{k+1}=\gamma_1\Delta_k$, and let

$$\tau_{k+1}=\begin{cases}\tau_k/\eta_1, & \text{if } \Delta_k>\widetilde{\Delta}_k,\\ \tau_k, & \text{otherwise.}\end{cases} \tag{3.11}$$

Let $\mathcal{X}_{k+1}=(\mathcal{X}_k\setminus\{\tilde{\mathbf{x}}_k\})\cup\{\mathbf{x}_k^+\}$, where $\tilde{\mathbf{x}}_k=\arg\max\{f(\mathbf{x}):\mathbf{x}\in\mathcal{X}_k\}$.

5: **Step 4: Set $k\leftarrow k+1$ and goto Step 1.**

is solved inexactly to obtain a trial step $\mathbf{d}_k$, where Δ_k is a proper trust region radius adaptively adjusted by DFGA. Next, we compute the trial point

$$\mathbf{x}_k^+ = \varphi_{\mathbf{x}_k}^{-1}(\mathbf{d}_k) \in \mathcal{S}^{n-1}$$

and evaluate its function value $f(\mathbf{x}_k^+)$. In fact, $f(\mathbf{x}_k^+) = \widehat{f}_{\mathbf{x}_k}(\mathbf{d}_k)$ and $f(\mathbf{x}_k) = \widehat{f}_{\mathbf{x}_k}(\mathbf{0})$. By comparing the actual function value reduction $f(\mathbf{x}_k) - f(\mathbf{x}_k^+)$ and the predicted function value reduction $m_k(\mathbf{0}) - m_k(\mathbf{d}_k)$, DFGA decides whether to accept the trial point as the next iteration point or not. The trust region radius may be enlarged if sufficient function value reduction is achieved, which also indicates the interpolation model is sufficiently accurate; otherwise, the trust region radius will be reduced. And when the trust region radius is sufficiently small, we would make sure the interpolation point set is Λ-poised such that the interpolation model will be at least fully linear. In addition, for both theoretical and practical efficiency reason, we prefer to keep the trust region radius be proportional to the norm of the model gradient in DFGA. This updating process is repeated until the sequence of iteration points converges or some stopping tolerances are satisfied. In practice, the trust region based derivative-free optimization algorithms often stop when the trust region radius is sufficiently small or the total number of function evaluations reaches a certain preset limit (the computational budget). The detail description of our derivative-free geometric algorithm is stated in Algorithm 3.1. Notice that usually only one point is replaced in the interpolation set $\mathcal{X}_k$ in an iteration of DFGA. This will only lead to a rank-2 change on the matrix $M(\phi_Q, \mathcal{Z})M(\phi_Q, \mathcal{Z})^T$ in (3.5). Taking advantage of this property will significantly reduce the computational cost of solving the minimum Frobenius model (3.4) at each iteration.

Now, let us establish some important properties of the trial step $\mathbf{d}_k$, which are crucial for our later convergence analysis. Let $\mathbf{d}_k^C$ be the Cauchy point of the trust-region subproblem (3.12) (see the definition in [44, 52].) As standard trust region method [20], we only need to solve the subproblem (3.12) inexactly, that is to find a trial step $\mathbf{d}_k$ satisfying

$$m_k(\mathbf{0}) - m_k(\mathbf{d}_k) \geq c_1 \left(m_k(\mathbf{0}) - m_k(\mathbf{d}_k^C) \right) > 0, \tag{3.13}$$

where $c_1 \in (0,1]$ is a constant. Then, the trial step $\mathbf{d}_k$ has the following properties.

Lemma 3.2. *Assume that $\mathbf{d}_k$ is an approximate solution of the trust-region subproblem* (3.12) *satisfying* (3.13) *and $\Delta_k \leq \tau \|\mathbf{g}_k\|$ for a constant $\tau > 0$. Then, we have*

$$m_k(\mathbf{0}) - m_k(\mathbf{d}_k) \geq \frac{c_1}{2} \|\mathbf{g}_k\| \min \left\{ \frac{\|\mathbf{g}_k\|}{\|H_k\|}, \Delta_k \right\}. \tag{3.14}$$

In addition, if $\|H_k\| \leq M$, where $M > 0$ is a constant, we have

$$m_k(\mathbf{0}) - m_k(\mathbf{d}_k) \geq c_2 \|\mathbf{g}_k\| \|\mathbf{d}_k\| \tag{3.15}$$

and

$$\|\mathbf{d}_k\| \geq c_3 \min\{\Delta_k, \|\mathbf{g}_k\|\}, \tag{3.16}$$

where c_2 and c_3 are two positive constants.

Proof. First, (3.14) is a well-known consequence of condition (3.13) [20]. Now, we establish (3.14) and (3.15). Since $\mathbf{d}_k^C$ is the Cauchy point of the trust region subproblem (3.12), we know

$$\mathbf{d}_k^C = \begin{cases} -\dfrac{\|\mathbf{g}_k\|^2}{\mathbf{g}_k^T H_k \mathbf{g}_k}\mathbf{g}_k, & \text{if } \Delta_k \mathbf{g}_k^T H_k \mathbf{g}_k \geq \|\mathbf{g}_k\|^3, \\ -\dfrac{\Delta_k}{\|\mathbf{g}_k\|}\mathbf{g}_k, & \text{otherwise.} \end{cases} \tag{3.17}$$

Then, it follows from (3.17) directly that

$$m_k(\mathbf{0}) - m_k(\mathbf{d}_k^C) \geq \frac{1}{2}\|\mathbf{g}_k\|\|\mathbf{d}_k^C\| \tag{3.18}$$

and

$$\|\mathbf{d}_k^C\| \geq \min\left\{\frac{\|\mathbf{g}_k\|^3}{\mathbf{g}_k^T H_k \mathbf{g}_k}, \Delta_k\right\} \geq \min\left\{\frac{\|\mathbf{g}_k\|}{\|H_k\|}, \Delta_k\right\}. \tag{3.19}$$

(In fact, we can see (3.14) follows from (3.13), (3.19) and (3.18).) By (3.19) and assumptions $\|H_k\| \leq M$ and $\Delta_k \leq \tau\|\mathbf{g}_k\|$ with $M > 0$ and $\tau > 0$, we have

$$\|\mathbf{d}_k^C\| \geq \min\left\{\frac{\Delta_k}{\tau M}, \Delta_k\right\} \geq \min\left\{\frac{1}{\tau M}, 1\right\}\Delta_k,$$

which together with (3.13) and (3.18) gives

$$\frac{m_k(\mathbf{0}) - m_k(\mathbf{d}_k)}{\|\mathbf{d}_k\|} \geq \frac{c_1\left(m_k(\mathbf{0}) - m_k(\mathbf{d}_k^C)\right)}{\Delta_k} \geq \frac{c_1}{2}\min\left\{\frac{1}{\tau M}, 1\right\}\|\mathbf{g}_k\|.$$

This gives (3.15) with $c_2 := \frac{c_1}{2}\min\left\{\frac{1}{\tau M}, 1\right\}$.

On the other hand, by (3.13) and (3.18), we have

$$\frac{1}{2}\|H_k\|\|\mathbf{d}_k\|^2 + \|\mathbf{g}_k\|\|\mathbf{d}_k\| \geq m_k(\mathbf{0}) - m_k(\mathbf{d}_k) \geq c_1\left(m_k(\mathbf{0}) - m_k(\mathbf{d}_k^C)\right) \geq \frac{c_1}{2}\|\mathbf{g}_k\|\|\mathbf{d}_k^C\|,$$

which together with $\|\mathbf{d}_k\| > 0$ implies

$$\|\mathbf{d}_k\| \geq \frac{-\|\mathbf{g}_k\| + \sqrt{\|\mathbf{g}_k\|^2 + c_1\|H_k\|\|\mathbf{g}_k\|\|\mathbf{d}_k^C\|}}{\|H_k\|}.$$

Simplifying the above inequality, we have

$$\|\mathbf{d}_k\| \geq \frac{c_1\|\mathbf{g}_k\|\|\mathbf{d}_k^C\|}{\sqrt{\|\mathbf{g}_k\|^2 + c_1\|H_k\|\|\mathbf{g}_k\|\|\mathbf{d}_k^C\|} + \|\mathbf{g}_k\|} \geq \frac{c_1}{\sqrt{1 + c_1\|H_k\|\|\mathbf{d}_k^c\|/\|\mathbf{g}_k\|} + 1}\|\mathbf{d}_k^C\|.$$

This inequality together with $\|H_k\| \leq M$, $\|\mathbf{d}_k^c\| \leq \Delta_k$ and $\Delta_k \leq \tau\|\mathbf{g}_k\|$ gives

$$\|\mathbf{d}_k\| \geq \frac{c_1}{\sqrt{1 + c_1\|H_k\|\|\mathbf{d}_k^c\|/\|\mathbf{g}_k\|} + 1}\|\mathbf{d}_k^C\| \geq \frac{c_1}{\sqrt{1 + c_1\tau M} + 1}\|\mathbf{d}_k^C\|.$$

Recalling (3.19), we get the validity of (3.16) with $c_3 := \frac{c_1}{\sqrt{1+c_1\tau M}+1}\min\{\frac{1}{M}, 1\}$. □

4 Convergence analysis

In this section, we first study the global convergence of DFGA. In fact, according to the overall structure of Algorithm 3.1, global convergence of DFGA can be established following a similar approach of the trust region derivative-free algorithms for unconstrained optimization [23, 24]. However, due to the local Householder and Cayley mappings to handle the sphere constraint and a particular mechanism of maintaining the trust region radius be proportional to the norm of model gradient, proper considerations and adjustments are needed throughout the global convergence proof. Then, under the Łojasiewicz property, we further strengthen the global convergence result and establish the linear or sublinear convergence rate of DFGA for which, to the best of our knowledge, no similar results has been established in the derivative-free optimization literature. Overall, we need the following assumption.

Assumption 4.1. *There exists a constant M such that $\|H_k\| \leq M$ for all k.*

Note that the minimum Frobenius norm model obtained from (3.4) keeps $\|H_k\|$ as small as possible. Hence, the choice of minimum Frobenius norm model by DFGA also has practical convergence importance. Because the unit sphere $\mathcal{S}^{n-1}$ is compact and the function f is continuous on $\mathcal{S}^{n-1}$, there exists a lower bound $f_{\min}$ on the function values such that $f(\mathbf{x}) \geq f_{\min}$ for all $\mathbf{x} \in \mathcal{S}^{n-1}$. For any given point $\mathbf{x} \in \mathcal{S}^{n-1}$, we first establish the following lemma on the gradient of the surrogate function $\widehat{f}_{\mathbf{x}}$.

Lemma 4.1. *Let $\mathbf{x} \in \mathcal{S}^{n-1}$ and $\nabla f(\mathbf{x})$ be the gradient of f at $\mathbf{x}$. Then, the gradient of the function $\widehat{f}_{\mathbf{x}}$ defined in* (3.1) *reads as*

$$\nabla \widehat{f}_{\mathbf{x}}(\mathbf{z}) = \left(\frac{4Q^T}{4+\|\mathbf{z}\|^2} - \frac{16\mathbf{z}\mathbf{x}^T + 8\mathbf{z}\mathbf{z}^T Q^T}{(4+\|\mathbf{z}\|^2)^2} \right) \nabla f(\mathrm{Cay}_{\mathbf{x}}(Q\mathbf{z})). \tag{4.1}$$

Proof. Because of $(f \circ \varphi_{\mathbf{x}}^{-1})(\mathbf{z}) = f(\mathrm{Cay}_{\mathbf{x}}(Q\mathbf{z}))$, it yields that

$$\nabla \widehat{f}_{\mathbf{x}}(\mathbf{z}) = Q^T (\nabla \mathrm{Cay}_{\mathbf{x}}(Q\mathbf{z}))^T (\nabla f(\mathrm{Cay}_{\mathbf{x}}(Q\mathbf{z}))).$$

By the map (2.8) and $Q^T Q = I$, we get

$$\nabla \mathrm{Cay}_{\mathbf{x}}(\mathbf{s}) = \frac{4I}{4+\|\mathbf{s}\|^2} - \frac{16\mathbf{x}\mathbf{s}^T + 8\mathbf{s}\mathbf{s}^T}{(4+\|\mathbf{s}\|^2)^2}$$

and (4.1) then follows straightforwardly. □

Since f is Lipschitz continuously differentiable on $\mathcal{S}^{n-1}$, it follows from Lemma 4.1 that $\nabla \widehat{f}_{\mathbf{x}_k}$ is Lipschitz continuous in $\mathcal{B}(\Delta_{\max}+1)$ with Lipschitz constant, say $L > 0$, independent of $\mathbf{x}_k$. Hence, when the interpolation set $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$, by Lemma 3.1 we get

$$\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{d}) - \nabla m_k(\mathbf{d})\| \leq \kappa_{eg} \Delta_k, \tag{4.2}$$

$$|\widehat{f}_{\mathbf{x}_k}(\mathbf{d}) - m_k(\mathbf{d})| \leq \kappa_{ef} \Delta_k^2, \tag{4.3}$$

for all $\mathbf{d} \in \mathcal{B}(\Delta_k)$, where $\kappa_{eg} := \widehat{\kappa}_{eg}(M+L)$ and $\kappa_{ef} := \widehat{\kappa}_{ef}(M+L)$. In this sense, we say the interpolation model m_k is at least fully linear. The following lemma gives a relation between $\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})$ and $\nabla m_k(\mathbf{0})$ when the interpolation set $\mathcal{Z}_k$ is Λ-poised.

Lemma 4.2. *If the interpolation set $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$, then we have*

$$\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| \leq (1+\kappa_{eg}\tau)\|\mathbf{g}_k\|.$$

Proof. Since the interpolation set $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$, by (4.2) and $\mathbf{g}_k = \nabla m_k(\mathbf{0})$, we know

$$\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0}) - \mathbf{g}_k\| \leq \kappa_{eg}\Delta_k. \tag{4.4}$$

On the other hand, from Step 1 of DFGA, we see $\Delta_k \leq \tau\|\mathbf{g}_k\|$. Thus, it yields that

$$\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| \leq \|\mathbf{g}_k\| + \|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0}) - \mathbf{g}_k\| \leq \|\mathbf{g}_k\| + \kappa_{eg}\Delta_k \leq (1+\kappa_{eg}\tau)\|\mathbf{g}_k\|.$$

The proof is completed. □

Next, we show that when Δ_k is sufficiently small and the model is at least fully linear, the trial point $\mathbf{x}_k^+$ will be accepted and hence, the tentative trust region radius $\widetilde{\Delta}_{k+1}$ at next iteration can not be smaller than Δ_k. The following lemma plays a similar role as [23, Lemma 5.2] or [24, Lemma 10.6] for showing convergence of derivative-free algorithm for unconstrained optimization.

Lemma 4.3. *Suppose Assumption* 4.1 *holds and the interpolation set $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$. If*

$$\Delta_k \leq \frac{1}{\kappa_{eg}+c_4}\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| \quad \text{or} \quad \Delta_k \leq \frac{1}{c_4}\|\mathbf{g}_k\|, \tag{4.5}$$

where

$$c_4^{-1} := \min\left\{\frac{1}{M}, \frac{(1-\eta)c_1}{4\kappa_{ef}}, \tau\right\},$$

we have $\rho_k \geq \eta$ and therefore

$$\widetilde{\Delta}_{k+1} \geq \Delta_k.$$

Proof. Since the interpolation set $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$, we know (4.2) and (4.4) hold. Hence, by (4.4) and (4.5), we have

$$\|\mathbf{g}_k\| \geq \|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| - \|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0}) - \mathbf{g}_k\| \geq \|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| - \kappa_{eg}\Delta_k \geq c_4\Delta_k, \tag{4.6}$$

which gives

$$\Delta_k \leq c_4^{-1}\|\mathbf{g}_k\| = \min\left\{\frac{1}{M}, \frac{(1-\eta)c_1}{4\kappa_{ef}}, \tau\right\}\|\mathbf{g}_k\|. \tag{4.7}$$

By (3.14) and (4.7), we get

$$|m_k(\mathbf{0}) - m_k(\mathbf{d}_k)| \geq \frac{c_1}{2}\|\mathbf{g}_k\|\min\left\{\frac{\|\mathbf{g}_k\|}{M}, \Delta_k\right\} = \frac{c_1}{2}\Delta_k\|\mathbf{g}_k\|.$$

Then, by (4.3) and (4.7), we derive

$$\begin{aligned}|\rho_k-1| &= \frac{|\widehat{f}_{\mathbf{x}_k}(\mathbf{0})-\widehat{f}_{\mathbf{x}_k}(\mathbf{d}_k)-m_k(\mathbf{0})+m_k(\mathbf{d}_k)|}{|m_k(\mathbf{0})-m_k(\mathbf{d}_k)|}\\ &\leq \frac{2\kappa_{ef}\Delta_k^2}{(1/2)c_1\Delta_k\|\mathbf{g}_k\|}=\frac{4\kappa_{ef}}{c_1}\frac{\Delta_k}{\|\mathbf{g}_k\|}\\ &\leq 1-\eta.\end{aligned}$$

Hence, we obtain $\rho_k \geq \eta$. Finally, by Step 3 of DFGA, we verify this lemma. □

Now we show that the scalar τ_k in DFGA can only be reduced in a finite number of iterations, and hence is bounded below. This implies the trust region radius in DFGA will be proportional to the norm of model gradient $\|\mathbf{g}_k\|$, which in fact has both theoretical and practical importance in derivative-free optimization.

Lemma 4.4. *Under Assumption* 4.1, *for all k, we have*

$$\tau_k \geq \min\{\tau_0, 1/(c_4\eta_1)\} := c_5, \tag{4.8}$$

where the constant c_4 is given in Lemma 4.3.

Proof. By the rules of updating Δ_k in DFGA, i.e., (3.8) and (3.9), we claim that $\Delta_k > \widetilde{\Delta}_k$ only if the interpolation set $\mathcal{Z}_k$ is Λ-poised in $\mathcal{B}(\Delta_k)$ and $\Delta_k = \tau_k\|\mathbf{g}_k\|$. So, if $\Delta_k > \widetilde{\Delta}_k$ and $\tau_k < 1/c_4$, we have $\Delta_k = \tau_k\|\mathbf{g}_k\| < \frac{1}{c_4}\|\mathbf{g}_k\|$ and hence, we have $\rho_k \geq \eta$ by Lemma 4.3. In addition, it follows from (3.11) that τ_k is reduced by a factor $\eta_1 > 1$ only when $\rho_k < \eta$ and $\Delta_k > \widetilde{\Delta}_k$. By considering the above two facts, we deduce (4.8) holds. □

The following lemma reveals a fundamental property for establishing the global convergence of trust region methods. Similar techniques were first proposed in [42] and later were also used in [31].

Lemma 4.5. *Under Assumption* 4.1, *we have*

$$\sum_{k=0}^{\infty}\Delta_k^2 < +\infty. \tag{4.9}$$

Proof. We first consider the set of successful iterations, that is

$$\mathcal{K} := \{k : \rho_k \geq \eta\}.$$

Then, for all $k \in \mathcal{K}$, by Assumption 4.1 and (3.14), we have

$$\begin{aligned}&f(\mathbf{x}_k)-f(\mathbf{x}_{k+1}) = \widehat{f}_{\mathbf{x}_k}(\mathbf{0})-\widehat{f}_{\mathbf{x}_k}(\mathbf{d}_k)\\ \geq&\eta(m_k(\mathbf{0})-m_k(\mathbf{d}_k)) \geq \frac{\eta c_1}{2}\|\mathbf{g}_k\|\min\left\{\frac{\|\mathbf{g}_k\|}{\|H_k\|},\Delta_k\right\}\\ \geq&\frac{\eta c_1}{2\tau}\min\left\{\frac{1}{\tau M},1\right\}\Delta_k^2 := c_6\Delta_k^2.\end{aligned}$$

This inequality means

$$\Delta_k^2 \leq \frac{f(\mathbf{x}_k)-f(\mathbf{x}_{k+1})}{c_6}$$

for all $k\in\mathcal{K}$. Summarizing all $k\in\mathcal{K}$, we establish

$$\sum_{k\in\mathcal{K}}\Delta_k^2 \leq \sum_{k\in\mathcal{K}}\frac{f(\mathbf{x}_k)-f(\mathbf{x}_{k+1})}{c_6} \leq \frac{1}{c_6}(f(\mathbf{x}_0)-f_{\min}). \tag{4.10}$$

Second, we consider the set of unsuccessful iterations, that is

$$\bar{\mathcal{K}} := \mathcal{N}\setminus\mathcal{K} = \{k:\rho_k<\eta\},$$

where $\mathcal{N}=\{0,1,2,\cdots\}$ is the set of natural numbers. By Lemma 4.4, τ_k is bounded below. This together with the rule (3.11) for updating τ_k imply that the set

$$\widehat{\mathcal{K}} := \bar{\mathcal{K}}\cap\{k:\Delta_k>\widetilde{\Delta}_k\}$$

is a finite set. More precisely, we can deduce $|\widehat{\mathcal{K}}|\leq\max\{\log_{\eta_1}(\tau_0 c_4)+1,0\}$, where $|\widehat{\mathcal{K}}|$ means the cardinality of the set $\widehat{\mathcal{K}}$. Hence, there exists a $\bar{k}$ such that

$$\Delta_k\leq\widetilde{\Delta}_k, \quad \text{if } k\in\bar{\mathcal{K}} \text{ and } k\geq\bar{k}. \tag{4.11}$$

For convenience, we denote $\mathcal{K}=\{k_1,k_2,\cdots\}$ with $k_1<k_2<\cdots$. So, by (4.11), for any $k_i\in\mathcal{K}$ such that $k_{i+1}\geq k_i+2$ and $k_i\geq\bar{k}$, we have

$$\ell\in\bar{\mathcal{K}} \quad \text{and} \quad \Delta_\ell\leq\widetilde{\Delta}_\ell, \tag{4.12}$$

for all $\ell=k_i+1,\cdots,k_{i+1}-1$. By DFGA, we clearly have $\widetilde{\Delta}_{k_i+1}\leq\gamma_2\Delta_{k_i}$ and $\widetilde{\Delta}_{\ell+1}=\gamma_1\Delta_\ell$ for all $\ell=k_i+2,\cdots,k_{i+1}-1$. Hence, it follows from (4.12) that

$$\sum_{\ell=k_i+1}^{k_{i+1}-1}\Delta_\ell^2 \leq \frac{\Delta_{k_i+1}^2}{1-\gamma_1^2} \leq \frac{\gamma_2^2}{1-\gamma_1^2}\Delta_{k_i}^2. \tag{4.13}$$

Finally, by Lemmas 4.3 and 4.4, we have $|\mathcal{K}|=\infty$. Let

$$k_I = \min\{k\in\mathcal{K}:k\geq\bar{k}\}<\infty.$$

Then, summing the squares of Δ_k for $k \geq k_I$, it follows from (4.10) and (4.13) that

$$\begin{aligned}
\sum_{k=k_I}^{\infty} \Delta_k^2 &= \sum_{k\in\mathcal{K},k\geq k_I} \Delta_k^2 + \sum_{k\in\bar{\mathcal{K}},k\geq k_I} \Delta_k^2 \\
&= \sum_{k_i\in\mathcal{K},k_i\geq k_I} \left(\Delta_{k_i}^2 + \sum_{\ell=k_i+1}^{k_{i+1}-1} \Delta_\ell^2 \right) \\
&\leq \sum_{k_i\in\mathcal{K},k_i\geq k_I} \left(\Delta_{k_i}^2 + \frac{\gamma_2^2}{1-\gamma_1^2} \Delta_{k_i}^2 \right) \\
&= \frac{1+\gamma_2^2-\gamma_1^2}{1-\gamma_1^2} \sum_{k_i\in\mathcal{K},k_i\geq k_I} \Delta_{k_i}^2 \\
&\leq \frac{1+\gamma_2^2-\gamma_1^2}{(1-\gamma_1^2)c_6} (f(\mathbf{x}_0) - f_{\min}),
\end{aligned}$$

which implies (4.9) holds. The proof is completed. □

By Lemma 4.5, we can directly get the following corollary.

Corollary 4.1. *Under Assumption* 4.1, *we have* $\lim_{k\to\infty} \Delta_k = 0$.

Now we can establish the following lemma which immediately implies the global convergence of DFGA.

Lemma 4.6. *Under Assumption* 4.1, *we have*

$$\liminf_{k\to\infty} \|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| = 0.$$

Proof. We prove by contradiction. Assume there exists a constant $\varepsilon > 0$ such that

$$\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| \geq \varepsilon.$$

From Corollary 4.1, we see $\Delta_k \to 0$ as $k \to \infty$. So, $\Delta_k \leq \varrho$ for k sufficiently large. Hence, by Step 1 of DFGA, the model function m_k is at least fully linear when k is sufficiently large. In the following proof, we assume k is sufficiently large that $\Delta_k \leq \varrho$ and hence m_k is at least fully linear.

First, by Lemma 4.2, we get $\|\mathbf{g}_k\| \geq (1+\kappa_{eg}\tau)^{-1}\varepsilon$. So, when $\Delta_k \leq \frac{\tau\varepsilon}{\gamma_2(1+\kappa_{eg}\tau)}$, we know

$$\widetilde{\Delta}_{k+1} \leq \gamma_2 \Delta_k \leq \frac{\tau\varepsilon}{1+\kappa_{eg}\tau} \leq \tau \|\mathbf{g}_{k+1}\|$$

and hence $\Delta_{k+1} \geq \widetilde{\Delta}_{k+1}$ by Step 1 of DFGA. Furthermore, we have by Lemma 4.3 that $\widetilde{\Delta}_{k+1} \geq \Delta_k$ whenever $\Delta_k \leq (\kappa_{eg}+c_4)^{-1}\varepsilon$. Combining the above two observations, we have $\Delta_{k+1} \geq \Delta_k$ whenever

$$\Delta_k \leq \min\left\{ \varrho, \min\left\{ \frac{1}{\kappa_{eg}+c_4}, \frac{\tau}{\gamma_2(1+\kappa_{eg}\tau)} \right\} \varepsilon \right\}.$$

This, however, contradicts with $\Delta_k \to 0$ as $k \to \infty$. □

We recall that the projection of gradient ∇f onto $\mathcal{T}_{\mathbf{x}_k}\mathcal{S}^{n-1}$ is indeed

$$(I-\mathbf{x}_k\mathbf{x}_k^T)\nabla f(\mathbf{x}_k)=Q_kQ_k^T\nabla f(\mathbf{x}_k)=Q_k\nabla\widehat{f}_{\mathbf{x}_k}(\mathbf{0}),$$

where columns of $Q_k\in\mathbb{R}^{n\times(n-1)}$ form a basis of $\mathcal{T}_{\mathbf{x}_k}\mathcal{S}^{n-1}$ and the last equality holds by Lemma 4.1. By Lemma 4.6, we immediately get the following global convergence theorem on DFGA.

Theorem 4.2. *Under Assumption* 4.1*, we have*

$$\liminf_{k\to\infty}\|(I-\mathbf{x}_k\mathbf{x}_k^T)\nabla f(\mathbf{x}_k)\|=0.$$

We say $\mathbf{x}_*$ is a stationary point (also a KKT point) of the spherical optimization (1.1) if $\mathbf{x}_*\in\mathcal{S}^{n-1}$ and $(I-\mathbf{x}_*\mathbf{x}_*^T)\nabla f(\mathbf{x}_*)=\mathbf{0}$. Since $\mathcal{S}^{n-1}$ is a compact set and $\mathbf{x}_k\in\mathcal{S}^{n-1}$ for all k, Theorem 4.2 implies there exists at least a subsequence of the iterates $\{\mathbf{x}_k\}$ converging to a stationary point of the spherical optimization (1.1).

4.1 Convergence based on Łojasiewicz property

Łojasiewicz property is a kind of regularization property, which holds for a broad class of functions, such as polynomial, semi-algebraic and analytic functions [19,40]. Strong iterate convergence of trust region methods under analytic cost functions was first studied in [3] and [6]. In this subsection, we prove that the total sequence of the iterates generated by DFGA converges and establish its convergence rate under the Łojasiewicz property, respectively.

Definition 4.1 (Łojasiewicz property)**.** *Let $\mathbf{x}_*$ be a stationary point of the spherical optimization* (1.1)*. We say that the Łojasiewicz property holds at $\mathbf{x}_*$, if there exist $\theta\in[1/2,1)$, $\mu>0$, and a neighborhood $\mathcal{U}(\mathbf{x}_*)$ such that for all $\mathbf{x}\in\mathcal{U}(\mathbf{x}_*)\cap\mathcal{S}^{n-1}$,*

$$|f(\mathbf{x})-f(\mathbf{x}_*)|^\theta\le\mu\|(I-\mathbf{x}\mathbf{x}^T)\nabla f(\mathbf{x})\|=\mu\|\nabla\widehat{f}_{\mathbf{x}}(\mathbf{0})\|. \tag{4.14}$$

Based on the Łojasiewicz property, we have the following key lemma.

Lemma 4.7. *Suppose Assumption* 4.1 *holds and the Łojasiewicz property holds at a stationary point $\mathbf{x}_*$ of the spherical optimization* (1.1)*. Let $\mathbf{x}_0\in\mathcal{S}^{n-1}$ be the initial point of DFGA that is close to $\mathbf{x}_*$ in the sense that $\mathbf{x}_0\in\mathcal{B}(\mathbf{x}_*,r):=\{\mathbf{x}\in\mathbb{R}^n:\|\mathbf{x}-\mathbf{x}_*\|<r\}\subset\mathcal{U}(\mathbf{x}_*)$, where*

$$r>\|\mathbf{x}_0-\mathbf{x}_*\|+\frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)}|f(\mathbf{x}_0)-f(\mathbf{x}_*)|^{1-\theta}. \tag{4.15}$$

If $\Delta_k\le\varrho$ for all k, then we have

$$\mathbf{x}_k\in\mathcal{B}(\mathbf{x}_*,r) \tag{4.16}$$

for all k and

$$\sum_{k=0}^{\infty}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|\le\frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)}|f(\mathbf{x}_0)-f(\mathbf{x}_*)|^{1-\theta}. \tag{4.17}$$

Proof. We show (4.16) by induction. Obviously, we have $\mathbf{x}_0 \in \mathcal{B}(\mathbf{x}_*,r)$. Assume $\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_k \in \mathcal{B}(\mathbf{x}_*,r)$. We show in the following proof that $\mathbf{x}_{k+1} \in \mathcal{B}(\mathbf{x}_*,r)$. Clearly, if $\rho_k < \eta$, we have $\mathbf{x}_{k+1} = \mathbf{x}_k \in \mathcal{B}(\mathbf{x}_*,r)$. Hence, we only need to consider the case $\rho_k \geq \eta$.

Consider the following function:

$$\zeta(t) := \frac{\mu}{1-\theta}|t-f(\mathbf{x}_*)|^{1-\theta}, \tag{4.18}$$

which is nonnegative and concave for all $t > f(\mathbf{x}_*)$. Then, we have

$$\begin{aligned}\zeta(f(\mathbf{x}_k))-\zeta(f(\mathbf{x}_{k+1})) &\geq \zeta'(f(\mathbf{x}_k))\,(f(\mathbf{x}_k)-f(\mathbf{x}_{k+1}))\\ &= \frac{\mu}{|f(\mathbf{x}_k)-f(\mathbf{x}_*)|^{\theta}}\,(f(\mathbf{x}_k)-f(\mathbf{x}_{k+1}))\\ &\geq \frac{1}{\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\|}\,(f(\mathbf{x}_k)-f(\mathbf{x}_{k+1})),\end{aligned} \tag{4.19}$$

where the last inequality holds by Łojasiewicz property (4.14) at $\mathbf{x}_k$.

From (3.15) and $\rho_k \geq \eta$, we obtain

$$f(\mathbf{x}_k)-f(\mathbf{x}_{k+1}) \geq \eta\,(m_k(\mathbf{0})-m_k(\mathbf{d}_k)) \geq \eta c_2 \|\mathbf{g}_k\|\,\|\mathbf{d}_k\|.$$

In addition, since $\rho_k \geq \eta$, we know $\mathbf{x}_{k+1} = \mathbf{x}_k^+ = \varphi_{\mathbf{x}_k}^{-1}(\mathbf{d}_k) = \mathrm{Cay}_{\mathbf{x}_k}(Q_k\mathbf{d}_k)$ and

$$\mathbf{x}_{k+1}-\mathbf{x}_k = \frac{(4-\|Q_k\mathbf{d}_k\|^2)\mathbf{x}_k+4Q_k\mathbf{d}_k}{4+\|Q_k\mathbf{d}_k\|^2}-\mathbf{x}_k = \frac{-2\|Q_k\mathbf{d}_k\|^2\mathbf{x}_k+4Q_k\mathbf{d}_k}{4+\|Q_k\mathbf{d}_k\|^2}, \tag{4.20}$$

which gives

$$\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 = \frac{4\|Q_k\mathbf{d}_k\|^4+16\|Q_k\mathbf{d}_k\|^2}{(4+\|Q_k\mathbf{d}_k\|^2)^2} = \frac{4\|Q_k\mathbf{d}_k\|^2}{4+\|Q_k\mathbf{d}_k\|^2} \leq \|Q_k\mathbf{d}_k\|^2 = \|\mathbf{d}_k\|^2. \tag{4.21}$$

Hence, we have $\|\mathbf{d}_k\| \geq \|\mathbf{x}_{k+1}-\mathbf{x}_k\|$. By Lemma 4.2, we immediately get $\|\mathbf{g}_k\| \geq (1+\kappa_{eg}\tau)^{-1}\|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\|$. So, it follows from (4.19) that

$$\zeta(f(\mathbf{x}_k))-\zeta(f(\mathbf{x}_{k+1})) \geq \frac{c_2\eta}{1+\kappa_{eg}\tau}\|\mathbf{d}_k\| \geq \frac{c_2\eta}{1+\kappa_{eg}\tau}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|. \tag{4.22}$$

Therefore, by (4.15), we have

$$\begin{aligned}\|\mathbf{x}_{k+1}-\mathbf{x}_*\| &\leq \sum_{\ell=0}^{k}\|\mathbf{x}_{\ell+1}-\mathbf{x}_\ell\|+\|\mathbf{x}_0-\mathbf{x}_*\|\\ &\leq \frac{1+\kappa_{eg}\tau}{c_2\eta}\sum_{\ell=0}^{k}(\zeta(f(\mathbf{x}_\ell))-\zeta(f(\mathbf{x}_{\ell+1})))+\|\mathbf{x}_0-\mathbf{x}_*\|\\ &\leq \frac{1+\kappa_{eg}\tau}{c_2\eta}\zeta(f(\mathbf{x}_0))+\|\mathbf{x}_0-\mathbf{x}_*\| < r.\end{aligned}$$

Hence, $\mathbf{x}_{k+1}\in\mathcal{B}(\mathbf{x}_*,r)$. So, by induction the whole sequence $\{\mathbf{x}_k\}\subset\mathcal{B}(\mathbf{x}_*,r)$. Furthermore, by (4.22), we have

$$\begin{aligned}\sum_{\ell=0}^{\infty}\|\mathbf{x}_{\ell+1}-\mathbf{x}_\ell\|&\leq\frac{1+\kappa_{eg}\tau}{c_2\eta}\sum_{\ell=0}^{\infty}\zeta(f(\mathbf{x}_\ell))-\zeta(f(\mathbf{x}_{\ell+1}))\\&=\frac{1+\kappa_{eg}\tau}{c_2\eta}\zeta(f(\mathbf{x}_0)),\end{aligned}$$

which is just (4.17) by the definition of function $\zeta(\cdot)$ in (4.18). □

Under the Łojasiewicz property, we can in fact show the convergence of the whole sequence of iterates generated by DFGA.

Theorem 4.3. *Suppose that Assumption* 4.1 *holds and there exists a subsequence of the iterates* $\{\mathbf{x}_k\}$ *generated by DFGA converging to a stationary point* $\mathbf{x}_*$ *of the spherical optimization* (1.1)*, where the Łojasiewicz property holds. Then, we have*

$$\sum_{k=0}^{\infty}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|<+\infty, \tag{4.23}$$

which implies

$$\lim_{k\to\infty}\mathbf{x}_k=\mathbf{x}_*. \tag{4.24}$$

Proof. Suppose there exists a subsequence $\{\mathbf{x}_{k_i}\}$ converging to a stationary point $\mathbf{x}_*$ where the Łojasiewicz property holds. So, there exists an iterate $\mathbf{x}_{k_0}\in\mathcal{B}(\mathbf{x}_*,r)\subset\mathcal{U}(\mathbf{x}_*)$, where

$$r>\|\mathbf{x}_{k_0}-\mathbf{x}_*\|+\frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)}|f(\mathbf{x}_{k_0})-f(\mathbf{x}_*)|^{1-\theta},$$

and $\mathcal{U}(\mathbf{x}_*)$ is a neighborhood of $\mathbf{x}_*$ where the Łojasiewicz property holds. And also by Corollary 4.1, we can assume that k_0 is sufficiently large such that $\Delta_k\leq\varrho$ for all $k\geq k_0$. Then, it follows from Lemma 4.7 that $\sum_{\ell=k_0}^{\infty}\|\mathbf{x}_{\ell+1}-\mathbf{x}_\ell\|<+\infty$, which implies (4.23). By (4.23), the iterates $\{\mathbf{x}_k\}$ generated by DFGA form a convergent Cauchy sequence. So, (4.24) holds. □

By Theorem 4.2, there always exists a subsequence of iterates generated DFGA converging to a stationary point $\mathbf{x}_*$. Hence, if the Łojasiewicz property holds at all the stationary points of the spherical optimization (1.1), it will follow directly from Theorem 4.3 that the whole sequence of iterates generated by DFGA converges to a stationary point of the spherical optimization (1.1). Since ∇f is Lipschitz continuous on the unit sphere $\mathcal{S}^{n-1}$, under the conditions of Theorem 4.3, the conclusion of Theorem 4.2 can be strengthened to

$$\lim_{k\to\infty}\|(I-\mathbf{x}_k\mathbf{x}_k^T)\nabla f(\mathbf{x}_k)\|=0.$$

Now, we would like to discuss the convergence rate of $\{\mathbf{x}_k\}$ under the Łojasiewicz property. We start with the following lemma.

Lemma 4.8. *Under Assumption* 4.1*, there exists a constant* $\varsigma>0$ *such that*

$$\|\mathbf{x}_k-\mathbf{x}_{k+1}\|\geq\varsigma\|\nabla\widehat{f}_{\mathbf{x}_k}(\mathbf{0})\|, \tag{4.25}$$

for all k sufficiently large such that $\Delta_k\leq\varrho$ *and* $\mathbf{x}_{k+1}\neq\mathbf{x}_k$.

Proof. By Corollary 4.1, we can assume that k is sufficiently large such that $\Delta_k\leq\varrho$ and $\mathbf{x}_{k+1}\neq\mathbf{x}_k$. Then, it follows from (4.21) and $\|\mathbf{d}_k\|\leq\Delta_k\leq\varrho$ that

$$\|\mathbf{x}_{k+1}-\mathbf{x}_k\|=\frac{2\|\mathbf{d}_k\|}{\sqrt{4+\|\mathbf{d}_k\|^2}}\geq\frac{2\|\mathbf{d}_k\|}{2+\|\mathbf{d}_k\|}\geq\frac{2\|\mathbf{d}_k\|}{2+\varrho}. \tag{4.26}$$

By Lemma 4.4, we have $\tau_k\geq c_5$, where c_5 is a positive constant. So, by (3.9), we get $\Delta_k\geq c_5\|\mathbf{g}_k\|$. Then, by Lemma 4.2, (3.16) and (4.26), we obtain

$$\begin{aligned}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|&\geq\frac{2\|\mathbf{d}_k\|}{2+\varrho}\geq\frac{2c_3}{2+\varrho}\min\{\Delta_k,\|\mathbf{g}_k\|\}\\&\geq\frac{2c_3}{2+\varrho}\min\{c_5,1\}\|\mathbf{g}_k\|\\&\geq\frac{2c_3\min\{c_5,1\}}{(2+\varrho)(1+\kappa_{eg}\tau)}\|\widehat{f}_{\mathbf{x}_k}(\mathbf{0})\|=:\varsigma\|\nabla\widehat{f}_{\mathbf{x}_k}(\mathbf{0})\|.\end{aligned}$$

The proof is completed. □

The following theorem shows the convergence rate of DFGA. We only need to consider the successful iterations where $\mathbf{x}_{k+1}\neq\mathbf{x}_k$.

Theorem 4.4. *Suppose that Assumption* 4.1 *holds and all the iterates* $\{\mathbf{x}_k\}$ *generated by DFGA are successful and converge to a stationary point* $\mathbf{x}_*$*, where the Łojasiewicz property holds. Then, we have the following convergence rate according to the parameter* θ *in* (4.14)*:*

- *If* $\theta=\frac{1}{2}$*, there exist* $\gamma>0$ *and* $\rho\in(0,1)$ *such that*

$$\|\mathbf{x}_k-\mathbf{x}_*\|\leq\gamma\rho^k. \tag{4.27}$$

  That is, the iterates $\{\mathbf{x}_k\}$ *converge to* $\mathbf{x}_*$ *with an R-linear rate.*

- *If* $\theta\in(\frac{1}{2},1)$*, there exists a* $\gamma>0$ *such that*

$$\|\mathbf{x}_k-\mathbf{x}_*\|\leq\gamma k^{-\frac{1-\theta}{2\theta-1}}. \tag{4.28}$$

  That is, the iterates $\{\mathbf{x}_k\}$ *converge to* $\mathbf{x}_*$ *with an R-sublinear rate.*

Proof. First, by Lemma 4.7 and Corollary 4.1, without loss of generality, we can simply assume $\mathbf{x}_0 \in \mathcal{B}(\mathbf{x}_*, r) \subset \mathcal{U}(\mathbf{x}_*)$, where $\mathbf{x}_0 \in \mathcal{S}^{n-1}$ satisfies (4.15) and $\Delta_k \leq \varrho$ for all k.

Now, denote $\delta_k = \sum_{i=k}^{\infty} \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \geq \|\mathbf{x}_k - \mathbf{x}_*\|$. Then, by Lemma 4.7, (4.14) and (4.25), we have

$$\begin{aligned}
\delta_k &= \sum_{i=k}^{\infty} \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \\
&\leq \frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)} |f(\mathbf{x}_k) - f(\mathbf{x}_*)|^{1-\theta} = \frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)} \left(|f(\mathbf{x}_k) - f(\mathbf{x}_*)|^{\theta} \right)^{\frac{1-\theta}{\theta}} \\
&\leq \frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)} \left(\mu \|\nabla \widehat{f}_{\mathbf{x}_k}(\mathbf{0})\| \right)^{\frac{1-\theta}{\theta}} \leq \frac{\mu(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)} \left(\mu\varsigma^{-1} \|\mathbf{x}_k - \mathbf{x}_{k+1}\| \right)^{\frac{1-\theta}{\theta}} \\
&= \frac{\mu^{\frac{1}{\theta}}(1+\kappa_{eg}\tau)}{c_2\eta(1-\theta)\varsigma^{\frac{1-\theta}{\theta}}} (\delta_k - \delta_{k+1})^{\frac{1-\theta}{\theta}} = c_7 (\delta_k - \delta_{k+1})^{\frac{1-\theta}{\theta}},
\end{aligned} \tag{4.29}$$

where $c_7 := (\mu^{\frac{1}{\theta}}(1+\kappa_{eg}\tau))/(c_2\eta(1-\theta)\varsigma^{\frac{1-\theta}{\theta}})$ is a positive constant.

First, consider the case that $\theta = \frac{1}{2}$. We have from the inequality (4.29) that

$$\delta_k \leq c_7(\delta_k - \delta_{k+1}),$$

which implies

$$\delta_{k+1} \leq \frac{c_7 - 1}{c_7} \delta_k.$$

Hence, noticing $\|\mathbf{x}_k - \mathbf{x}_*\| \leq \delta_k$, we have (4.27) holds with $\gamma = \delta_0$ and $\rho = \frac{c_7-1}{c_7}$.

Now, consider the case that $\theta \in (\frac{1}{2}, 1)$. Let $h(s) = s^{-\frac{\theta}{1-\theta}}$. Obviously, $h(s)$ is monotonely decreasing for $s > 0$. Then, the inequality (4.29) could be rewritten as

$$\begin{aligned}
c_7^{-\frac{\theta}{1-\theta}} &\leq h(\delta_k)(\delta_k - \delta_{k+1}) = \int_{\delta_{k+1}}^{\delta_k} h(\delta_k)\, \mathrm{d}s \\
&\leq \int_{\delta_{k+1}}^{\delta_k} h(s)\, \mathrm{d}s = -\frac{1-\theta}{2\theta-1} (\delta_k^{-\frac{2\theta-1}{1-\theta}} - \delta_{k+1}^{-\frac{2\theta-1}{1-\theta}}).
\end{aligned}$$

Let $\nu = -\frac{2\theta-1}{1-\theta} < 0$ since $\theta \in (\frac{1}{2}, 1)$. Then, we get

$$\delta_{k+1}^{\nu} - \delta_k^{\nu} \geq -\nu c_7^{-\frac{\theta}{1-\theta}} := c_8 > 0,$$

which gives

$$\delta_k \leq (\delta_0^{\nu} + c_8 k)^{\frac{1}{\nu}} \leq (c_8 k)^{\frac{1}{\nu}}.$$

Hence, (4.28) holds with $\gamma = c_8^{\frac{1}{\nu}}$. ∎

5 Numerical experiments

In this section, we apply DFGA to solve several well-known optimization problems with a sphere constraint. Our DFGA is implemented in MATLAB R2018b with parameters

$$\eta=0.05,\quad \eta_1=2,\quad \tau_0=0.0001,\quad \tau=10,\quad \gamma_1=0.25,\quad \gamma_2=2,\quad \varrho=0.1,\quad \widetilde{\Delta}_0=1,\quad \Delta_{\max}=10.$$

The algorithm terminates if Δ_k is sufficiently small and the function values do not decrease sufficiently after five successive iterates, that is

$$\Delta_k\leq 10^{-6}\cdot\sqrt{n}\quad \text{and}\quad \frac{|f(\mathbf{x}_k)-f(\mathbf{x}_{k-4})|}{1+|f(\mathbf{x}_k)|}\leq 10^{-10}\cdot n.$$

We will also stop the algorithm if the number of iterations exceeds 1000. In DFGA, the trust region subproblem (3.12) is solved inexactly by a truncated conjugate gradient method [44, 52], which guarantees the condition (3.13) holds. All codes are run on a Linux computer with 2.2GHz CPU and 64GB memory and we compare the following four numerical algorithms.

- PatternS: The pattern search method implemented as MATLAB built-in function "patternsearch";
- Fmincon: MATLAB built-in function "fmincon" with the choice of approximating gradients by finite difference method;
- COBYLA : a well-known model based derivative-free optimization software for solving optimization with general constraints [45, 50];
- DFGA: Algorithm 3.1 of this paper written in MATLAB.‡

5.1 The classical Weber problem

Given N destinations $\mathbf{a}^i\in\mathcal{S}^2$ and their associated positive weights w_i, $i=1,\cdots,N$, the classical Weber problem [37] on a unit sphere $\mathcal{S}^2$ is to find a source point $\mathbf{x}\in\mathcal{S}^2$ that minimizes

$$f(\mathbf{x})=\sum_{i=1}^{N}w_i d(\mathbf{x},\mathbf{a}^i),$$

where the metric $d(\cdot,\cdot)$ could be the Euclidean distance or geodesic distance, i.e.,

$$d_{Euc}(\mathbf{x},\mathbf{y})=\|\mathbf{x}-\mathbf{y}\|\quad \text{or}\quad d_{geo}(\mathbf{x},\mathbf{y})=2\arcsin\frac{\|\mathbf{x}-\mathbf{y}\|}{2},$$

‡Note that in later comparisons, CPU time of DFGA could be much less if it is written in C or Fortran, while both PatternS and Fmincon are built-in functions of MATLAB which are essentially written in C, and COBYLA is written in Fortran.

Table 1: The classical Weber problem with Euclidean distance.

Solver	Func	ConsE	#F	Time	Func	ConsE	#F	Time
		$\theta=30^\circ$				$\theta=40^\circ$		
PatternS	3.0000	4.3e-9	71808	8.59	2.5357	2.4e-7	262833	33.55
COBYLA	3.0000	6.9e-13	273	0.02	2.5357	4.0e-14	108	0.01
Fmincon	3.0000	8.9e-16	126	0.03	2.5357	2.9e-15	47	0.03
DFGA	3.0000	1.3e-15	51	0.01	2.5357	0.00	48	0.01
		$\theta=50^\circ$				$\theta=60^\circ$		
PatternS	2.0521	1.5e-7	114045	14.86	1.5529	1.1e-7	79082	10.05
COBYLA	2.0521	8.2e-13	87	0.01	1.5529	8.6e-13	91	0.01
Fmincon	2.0521	6.7e-16	42	0.01	1.5529	2.2e-16	52	0.01
DFGA	2.0521	4.4e-16	46	0.01	1.5529	0.00	30	0.00
		$\theta=70^\circ$				$\theta=80^\circ$		
PatternS	1.0419	1.9e-9	21372	2.85	0.5229	3.3e-9	16637	2.17
COBYLA	1.0419	6.9e-13	93	0.01	0.5229	2.5e-13	112	0.01
Fmincon	1.0419	8.9e-16	52	0.01	0.5229	2.2e-16	54	0.01
DFGA	1.0419	4.4e-16	45	0.01	0.5229	3.3e-16	34	0.01

respectively. Following the setting in [37], we use all weights $w_i=1$ and the following three destinations:

$$\mathbf{a}^1=(\cos\theta,0,\sin\theta)^T,\quad \mathbf{a}^2=\left(-\tfrac{1}{2}\cos\theta,\tfrac{\sqrt{3}}{2}\cos\theta,\sin\theta\right)^T,$$
$$\mathbf{a}^3=\left(-\tfrac{1}{2}\cos\theta,-\tfrac{\sqrt{3}}{2}\cos\theta,\sin\theta\right)^T.$$

We consider the Weber problem on a unit sphere using the Euclidean distance and six different latitudes $\theta\in\{30^\circ,40^\circ,50^\circ,60^\circ,70^\circ,80^\circ\}$. So, we have 6 test problems and all of them have the same solution $\mathbf{x}^*=(0,0,1)^T$, which is the north pole of $\mathcal{S}^2$. For each test, we run all the comparison algorithms using the same initial point $\mathbf{x}_0=\left(\frac{1}{2},\frac{1}{2},\frac{\sqrt{2}}{2}\right)^T$. The numerical results using Euclidean and geodesic distances are shown in Tables 1 and 2, respectively, where "Func" is the final function value, "ConsE" denotes the final constraint violation, "#F" is the total number of function evaluations, and "Time" gives the used CPU time in seconds. All solvers solve the test problems successfully. Clearly, COBYLA, Fmincon, and DFGA perform much better than the pattern search method PatternS. However, DFGA almost always uses the least number of function evaluations and CPU time. Furthermore, although both DFGA and COBYLA are model based methods, by particularly taking care of the spherical constraint, DFGA only takes about 50% number of function evaluations used by COBYLA. Since PatternS performs significantly worse than other methods, we only compare COBYLA, Fmincon and DFGA in later numerical experiments.

Table 2: The classical Weber problem with a geodesic distance.

Solver	Func	ConsE	#F	Time	Func	ConsE	#F	Time
		$\theta=30^\circ$				$\theta=40^\circ$		
PatternS	3.1416	9.7e-8	196752	25.24	2.6180	4.9e-7	131739	17.26
COBYLA	3.1416	1.4e-13	103	0.01	2.6180	4.5e-13	92	0.01
Fmincon	3.1416	6.7e-16	42	0.01	2.6180	3.6e-15	42	0.01
DFGA	3.1416	2.2e-16	33	0.01	2.6180	2.2e-16	38	0.01
		$\theta=50^\circ$				$\theta=60^\circ$		
PatternS	2.0944	9.9e-8	81794	10.54	1.5708	3.2e-7	49258	6.41
COBYLA	2.0944	8.7e-13	85	0.02	1.5708	4.7e-13	102	0.01
Fmincon	2.0944	1.3e-15	42	0.01	1.5708	2.2e-16	50	0.01
DFGA	2.0944	0.00	40	0.01	1.5708	2.2e-16	36	0.01
		$\theta=70^\circ$				$\theta=80^\circ$		
PatternS	1.0472	6.9e-8	41984	5.35	0.5236	3.5e-9	12448	1.70
COBYLA	1.0472	4.5e-13	90	0.01	0.5236	1.3e-12	74	0.01
Fmincon	1.0472	2.0e-15	52	0.01	0.5236	4.4e-16	53	0.01
DFGA	1.0472	3.3e-16	48	0.01	0.5236	0.00	52	0.01

5.2 Spherical location problem

In this numerical experiment, we consider solving the more general n-dimensional spherical location problem proposed in [30]. In this problem, the pole $\mathbf{x}_{pse} := (0,\cdots,0,1)^T \in \mathcal{S}^{n-1}$ is regarded as a pseudo-center. We then randomly generate N points in $\mathbb{R}^n$ under normal distributions $\mathcal{N}(\mathbf{x}_{pse}, I)$ and project these points onto $\mathcal{S}^{n-1}$ to obtain a set $\mathcal{A} := \{\mathbf{a}^1, \mathbf{a}^2, \cdots, \mathbf{a}^N\} \subset \mathcal{S}^{n-1}$. Our goal is to find a center of the set $\mathcal{A}$ on $\mathcal{S}^{n-1}$ by solving the following spherical optimization problem

$$\min\ f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}-\mathbf{a}^i\| \qquad \text{s.t. } \mathbf{x}\in\mathcal{S}^{n-1}.$$

We solve this problem with dimensions n varying from 10 to 100 and the number of points N in set $\mathcal{A}$ varying from 50 to 5000 as shown in Table 3. In total, we have 12 test problems. For each problem, we run COBYLA, Fmincom and DFGA using a same starting point on the sphere $\mathcal{S}^{n-1}$. The numerical results are shown in Table 3. Again all the software achieve about the same final function values. However, for some problems Fmincon or COBYLA could not maintain the constraint error as small as that given by DFGA, which may be critical in some real applications. Compared with COBYLA and Fmincon, we can see that DFGA saves about 90% and 70% function evaluations. In addition, we see that the CPU time of DFGA increases when either n and N increases. However, when the dimension n is fixed, DFGA only uses about the same amount of

Table 3: Numerical results on the location problems.

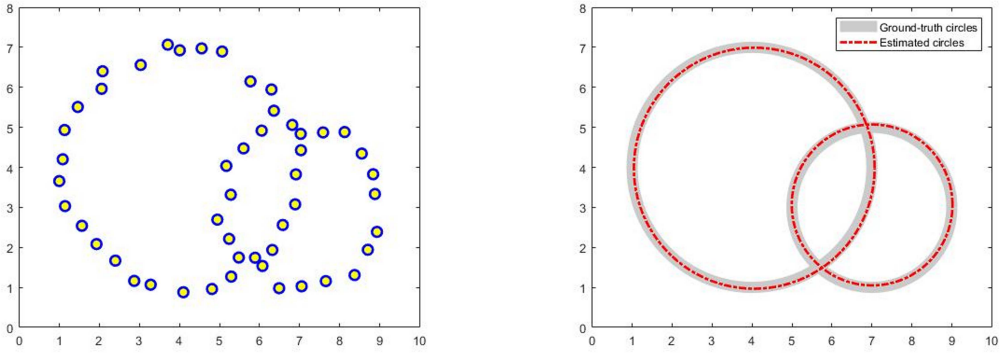
N	Solver	Func	ConsE	#F	Time	Func	ConsE	#F	Time
			$n=10$				$n=40$		
50	COBYLA	1.1136	1.9e-13	461	0.05	1.2593	4.2e-13	4452	1.69
	Fmincon	1.1136	0.00	334	0.04	1.2593	0.00	2044	0.26
	DFGA	1.1136	5.5e-16	133	0.05	1.2593	8.9e-16	438	0.51
500	COBYLA	1.1399	1.4e-12	477	0.14	1.2922	1.1e-13	5070	3.01
	Fmincon	1.1399	4.4e-15	381	0.12	1.2922	0.00	2145	0.73
	DFGA	1.1399	2.7e-15	166	0.12	1.2922	1.9e-15	470	0.65
5000	COBYLA	1.1604	1.7e-13	459	1.01	1.2943	7.0e-13	5287	13.96
	Fmincon	1.1604	8.9e-16	442	0.99	1.2943	0.00	2235	5.35
	DFGA	1.1604	2.4e-15	178	0.53	1.2943	7.1e-15	495	1.93
			$n=70$				$n=100$		
50	COBYLA	1.2594	1.2e-13	8693	12.02	1.2819	5.1e-14	14085	51.84
	Fmincon	1.2594	0.00	3023	0.32	1.2819	8.9e-15	3007	0.29
	DFGA	1.2594	1.0e-14	666	1.46	1.2819	6.0e-15	836	4.12
500	COBYLA	1.3157	5.0e-13	12075	19.41	1.3273	6.5e-13	19084	75.16
	Fmincon	1.3157	0.00	3031	1.03	1.3274	1.9e-10	3028	1.04
	DFGA	1.3157	8.9e-16	869	2.34	1.3274	4.0e-15	1015	4.99
5000	COBYLA	1.3240	4.8e-13	11804	44.66	1.3379	2.0e-13	22328	139.80
	Fmincon	1.3240	1.9e-13	3057	7.77	1.3380	6.5e-10	3027	8.18
	DFGA	1.3240	1.3e-15	802	3.20	1.3379	8.0e-15	1352	7.92

function evaluations as the number of fixed points N increases. This is a very desirable property for efficient derivative-free optimization algorithm.

5.3 Subspace clustering

Subspace clustering is a crucial problem in pattern analysis and machine learning [15,19]. For instance, there are 50 points in a plane roughly located on two crossing circles as shown in Fig. 2(a), where the larger circle has center $(4,4)^T$ with radius 3 and the smaller one has center $(7,3)^T$ with radius 2. Suppose both the centers and the radii of these two circles are unknown. The subspace clustering problem is to estimate them from positions of these 50 points. One approach for solving this subspace clustering problem is to first partition the 50 points into two sets: one set of points are estimated on the larger circle and the other set of points are estimated on the smaller circle. Then, we fit each set of points by a circle to obtain the center and radius we want to find. Hence, the partition step is crucial in the overall approach.

The spectral method based on a weighted hypergraph $\mathbb{G}$ [19] is quite effective for this



(a) Points for clustering two circles (b) Estimated vs. ground truth

Figure 2: Fifty points for clustering two circles (a) and estimated circles vs. the ground truth (b).

partition. In this method, the 50 points first form a vertex set $\mathbb{V}=\{1,2,\cdots,50\}$. Since positions of the 50 points are known, we can fit a circle by every four points (say $i,j,k,\ell\in\mathbb{V}$) using the linear least squares method. Suppose (x^i,y^i), (x^j,y^j), (x^k,y^k), (x^ℓ,y^ℓ) are the coordinates of these four points i,j,k,ℓ, respectively. Then, the center (x,y) and radius R of this circle can be estimated by solving the least squares model with fitting error

$$r=\min_{\alpha}\|A\alpha-\mathbf{b}\|,$$

where

$$A=\begin{pmatrix}2x^i & 2y^i & 1\\ 2x^j & 2y^j & 1\\ 2x^k & 2y^k & 1\\ 2x^\ell & 2y^\ell & 1\end{pmatrix},\quad \alpha=\begin{pmatrix}x\\ y\\ R^2-x^2-y^2\end{pmatrix},\quad \mathbf{b}=\begin{pmatrix}(x^i)^2+(y^i)^2\\ (x^j)^2+(y^j)^2\\ (x^k)^2+(y^k)^2\\ (x^\ell)^2+(y^\ell)^2\end{pmatrix}.$$

By this way, we can connect an edge $E:=\{i,j,k,\ell\}$ with weight $w=\exp(-r)$. Then, a medium scale random hypergraph can be generated by the following way. First, we generate a complete graph which has $\binom{50}{2}=1225$ edges and each edge contains 2 vertices. Second, for each edge of the graph, randomly choose other two different vertices from $\mathbb{V}$ and add these two vertices to the graph edge. Hence, we get 1225 edges and each of them contains 4 vertices. Repeating the above process in total 120 times, we can obtain 147,000 edges and each of them contains 4 vertices. So, we have constructed a random weighted hypergraph $\mathbb{G}=(\mathbb{V},\mathbb{E},\mathbf{w})$, where $\mathbb{E}=\{E_p:p=1,2,\cdots,147000\}$ is the set of edges and $\mathbf{w}=(w_p)\in\mathbb{R}_+^{147000}$ is the weight vector with component being the weight of each edge E_p. Moreover, $\mathbb{G}$ is a 4-uniform connected hypergraph.

Next, we turn to construct the Laplacian tensor of the weighted hypergraph $\mathbb{G}=(\mathbb{V},\mathbb{E},\mathbf{w})$. For each $i\in\mathbb{V}$, the degree of the vertex i is defined as

$$d_i=\sum_{E_p\in\mathbb{E},i\in E_p}w_p.$$

Table 4: Numerical results on subspace clustering.

Solver	Func	ConsE	#F	Time
COBYLA	0.06228	3.22e-13	4283	9.49
Fmincon	0.06275	3.55e-7	3036	4.55
DFGA	0.05618	1.55e-15	672	2.10

Let $\mathbf{e}_i \in \mathbb{R}^{50}$ be the i-th column of the identity matrix. For the vertex i in the E_p, we define

$$\mathbf{u}_i^p := \frac{3}{4\sqrt[4]{d_i}}\mathbf{e}_i - \sum_{j\in E_p, j\neq i} \frac{1}{4\sqrt[4]{d_j}}\mathbf{e}_j.$$

Then, the Laplacian tensor of $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbf{w})$ is represented as

$$\mathcal{L}(\mathbb{G}) := \sum_{E_p\in\mathbb{E}} \left(w_p \sum_{i\in E_p} \mathbf{u}_i^p \circ \mathbf{u}_i^p \circ \mathbf{u}_i^p \circ \mathbf{u}_i^p \right), \tag{5.1}$$

where "$\circ$" stands for the outer product of vectors and $\mathbf{u}\circ\mathbf{u}\circ\mathbf{u}\circ\mathbf{u}$ is indeed a fourth order rank-one tensor. The smallest Z-eigenvalue of $\mathcal{L}(\mathbb{G})$ is 0 and the associated Z-eigenvector is $\mathbf{z}_0 = \widetilde{\mathbf{d}}/\|\widetilde{\mathbf{d}}\|$ where $\widetilde{\mathbf{d}} \in \mathbb{R}^{50}_+$ with $\widetilde{d}_i = \sqrt[4]{d_i}$ [35]. The Z-eigenvector $\mathbf{z}_1$ corresponding to the second smallest Z-eigenvalue of $\mathcal{L}(\mathbb{G})$ is called the Fiedler vector, which is very useful for clustering. In fact, the Fiedler vector satisfies $\mathbf{z}_1^T\mathbf{z}_1 = 1$ and $\mathbf{z}_1^T\mathbf{z}_0 = 0$. Consider the subspace $\mathbf{z}_0^\perp$ that is perpendicular to $\mathbf{z}_0$. Let $Q \in \mathbb{R}^{50\times 49}$ be an orthonormal basis of $\mathbf{z}_0^\perp$. We can represent the Fiedler vector as $\mathbf{z}_1 = Q\mathbf{x}$ with $\mathbf{x}^T\mathbf{x} = 1$. Hence, to find the Fiedler vector, we can apply DFGA to solve the following spherical optimization problem

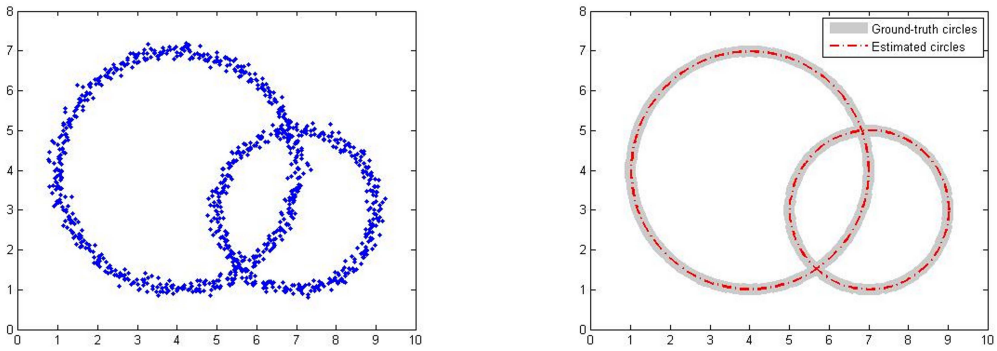
$$\min\ f(\mathbf{x}) = \langle \mathcal{L}(\mathbb{G}), (Q\mathbf{x})\circ(Q\mathbf{x})\circ(Q\mathbf{x})\circ(Q\mathbf{x})\rangle \qquad \text{s.t. } \mathbf{x}\in\mathcal{S}^{49}, \tag{5.2}$$

where $\langle \mathcal{L}, \mathbf{z}\circ\mathbf{z}\circ\mathbf{z}\circ\mathbf{z}\rangle = \sum_{i,j,k,\ell} L_{ijk\ell} z_i z_j z_k z_\ell$ is the inner product of tensors. In fact, the objective function can be explicitly written as

$$f(\mathbf{x}) = \sum_{E_p\in\mathbb{E}} \left[w_p \sum_{i\in E_p} \left(\frac{3\mathbf{e}_i^T Q\mathbf{x}}{4\sqrt[4]{d_i}} - \sum_{j\in E_p, j\neq i} \frac{\mathbf{e}_j^T Q\mathbf{x}}{4\sqrt[4]{d_j}} \right)^4 \right].$$

Now, we employ DFGA, COBYLA, and Fmincon to solve the spherical optimization problem (5.2) using a same starting point. We can see from the numerical results given in Table 4 that DFGA uses much less number of function evaluations than both COBYLA and Fmincon, but achieves the minimum final function value and the smallest constraint evaluation.

With the solution $\mathbf{x}^*$ of (5.2) returned by DFGA, we can compute the Fiedler vector $\mathbf{z}_1 = Q\mathbf{x}^*$. Then, $\mathbf{z}_1$ will naturally partition vertices $\mathbb{V}$ into two sets as $\{i \in \mathbb{V} : (\mathbf{z}_1)_i \geq 0\}$



(a) Points for clustering two circles (b) Estimated vs. ground truth

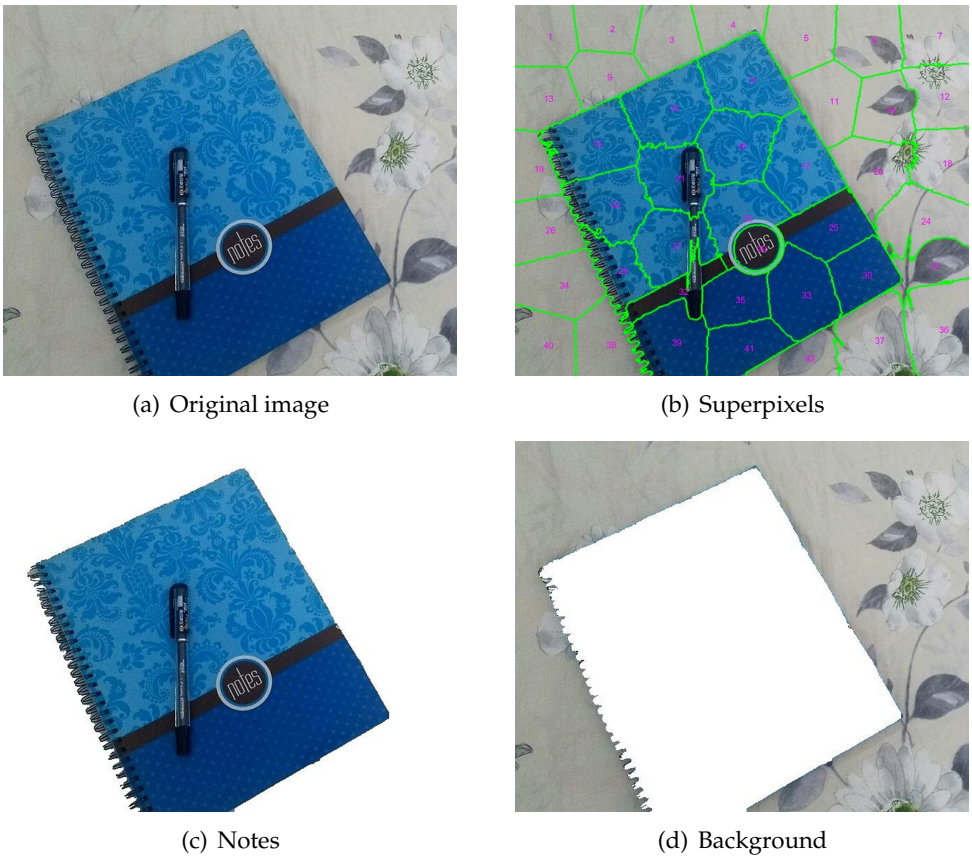
Figure 3: One thousand points for clustering two circles (a) and estimated circles vs. the ground truth (b).

and $\{i\in\mathbb{V}:(\mathbf{z}_1)_i<0\}$. After fitting the positions of vertices in each set, we get two circles shown in Fig. 2(b). For comparison, the ground truth circles are also shown in Fig. 2(b).

Whereafter, to examine the performance of DFGA for solving a large dimensional problem, we increase the number of points roughly around the two circles from 50 to 1000. Fig. 3(a) depicts the positions of these points. By a similar approach as before, we construct a 4-uniform hypergraph with 1,000 vertices and 2,997,000 edges. The associated Laplacian tensor $\mathcal{L}(\mathbb{G})$ and the corresponding spherical optimization problem would still have format (5.1) and (5.2), respectively. But the constraint in (5.2) turns to be $\mathbf{x}\in\mathcal{S}^{999}$. To solve this larger dimensional problem, DFGA costs 5,164 function evaluations to find an approximate solution $\mathbf{x}^*$, while both FMINCON and COBYLA can not solve the problem. The estimated circles and the ground truth circles are illustrated in Fig. 3(b).

5.4 Image segmentation

The spectral hypergraph method described for the subspace clustering problem in the previous subsection could also be applied for image segmentation. Suppose we want to separate the main object (the note book) and background in a given image in Fig. 4(a). We can first employ the SLIC superpixel approach [5] to produce a set of 42 superpixels; See Fig. 4(b). Using these superpixels and a similar approach in the last subsection, we can construct a weighted hypergraph $\mathbb{G}=(\mathbb{V},\mathbb{E},\mathbf{w})$, where these superpixels constitute the vertex set $\mathbb{V}=\{1,2,\cdots,42\}$ and the set $\mathbb{E}$ has 77,490 edges. For each edge $E_p\in\mathbb{E}$, the weight w_p is proportional to the similarity of color distributions of superpixels color_p and is inversely proportional to the distance among superpixels dist_p. Here, we only briefly discuss on how to compute color_p and dist_p. One can refer to [19] for the details on how to compute the weight w_p. Consider the image in the HSV color space, where HSV stands for hue, saturation and value, respectively. Hue is divided into twelve intervals. Saturation and value are each divided into four intervals. Hence, the whole HSV color



(a) Original image (b) Superpixels

(c) Notes (d) Background

Figure 4: Segment an image of a notes and a pen.

space is divided into 192 areas. Then, we count the HSV color distribution $\text{hsv}_i \in \mathbb{R}_+^{192}$ in these areas for $i = 1,\cdots,42$. The similarity of color distributions of the superpixels in an edge $E_p = \{i,j,k,\ell\}$ is defined as

$$\text{color}_p = \frac{\text{hsv}_i^T(\text{hsv}_j * \text{hsv}_k * \text{hsv}_\ell)}{\|\text{hsv}_i\|_4 \|\text{hsv}_j\|_4 \|\text{hsv}_k\|_4 \|\text{hsv}_\ell\|_4},$$

where $*$ is the component-wise Hadamard product.[§] For calculating dist_p, we first find the center cent_i of each superpixel, $i = 1,\cdots,42$, and then the star distance among superpixels in an edge E_p is set by

$$\text{dist}_p = \sum_{i \in E_p} (\text{cent}_i - \overline{\text{cent}}_p)^4,$$

[§]We have $\text{hsv}_i^T(\text{hsv}_j * \text{hsv}_k * \text{hsv}_\ell) = \sum_{p=1}^{192} (\text{hsv}_i)_p (\text{hsv}_j)_p (\text{hsv}_k)_p (\text{hsv}_\ell)_p$.

Table 5: Numerical results on image segmentation.

Solver	Func	ConsE	#F	time
COBYLA	0.000552	3.67e-13	3124	3.93
Fmincon	0.000552	9.16e-12	3003	2.43
DFGA	0.000553	3.44e-15	507	0.97

where $\overline{\text{cent}}_p = \frac{1}{4}(\text{cent}_i + \text{cent}_j + \text{cent}_k + \text{cent}_\ell)$. With these color_p and dist_p, we can compute the weight w_p, and therefore, construct the 4-uniform hypergraph $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbf{w})$.

According to this hypergraph $\mathbb{G}$, we can again generate its Laplacian tensor $\mathcal{L}(\mathbb{G})$ given in (5.1) and establish the following optimization model

$$\min f(\mathbf{x}) = \langle \mathcal{L}(\mathbb{G}), (Q\mathbf{x}) \circ (Q\mathbf{x}) \circ (Q\mathbf{x}) \circ (Q\mathbf{x}) \rangle \qquad \text{s.t. } \mathbf{x} \in \mathcal{S}^{41},$$

using the same approach introduced in the last subsection. We can see from the numerical results given in Table 5 that DFGA again takes much less number of function evaluations than COBYLA and Fmincon to solve this resulted optimization. Finally, the signs of the resulting Fiedler vector will again provide a segmentation: one is the note book image shown in Fig. 4(c) and the other is the background shown in Fig. 4(d).

6 Conclusions

In this paper, we propose a derivative-free geometric algorithm to solve the spherically constrained optimization problem (1.1). This DFGA combines the function interpolation techniques used in derivative-free optimization and the local spherical geometry on a sphere in a trust region framework. Using the chart, a map defined from the sphere to $\mathbb{R}^{n-1}$, we are able to keep all the iterates being strictly feasible on the sphere, which is crucial in many applications, and locally minimize the objective function as an unconstrained optimization. We have shown that there at least exists a subsequence generated by DFGA converging to a stationary point of the spherical optimization problem (1.1). Furthermore, under the Łojasiewicz property, we have shown the convergence of all the iterates generated by DFGA with at least a linear or sublinear convergence rate. Our numerical experiments on comparing different derivative-free optimization solvers indicate DFGA is quite robust, efficient and could be very useful for solving spherically constrained optimization arising from practical problems, for which the explicit calculations of the derivatives of the objective function are difficult or even impossible.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under grants 11771405, 11901118, and 11571178, and by the USA National Science Foundation under grants 1522654 and 1819161.

The authors are grateful to the associate editor and two anonymous referees for their comments which helped us to improve our manuscript essentially.

References

[1] P. A. ABSIL, C. G. BAKER, AND K. A. GALLIVAN, *Trust-region methods on Riemannian manifolds*, Found. Comput. Math., 7 (2007), pp. 303–330.
[2] P. A. ABSIL AND S. HOSSEINI, *A collection of nonsmooth Riemannian optimization problems*, In: Hosseini S., Mordukhovich B., Uschmajew A. (eds) Nonsmooth Optimization and Its Applications. International Series of Numerical Mathematics, vol 170. Birkhäuser, Cham. (2019), pp. 1–15.
[3] P. A. ABSIL, R. MAHONY, AND B. ANDREWS, *Convergence of the iterates of descent methods for analytic cost functions*, SIAM J. Optim., 16 (2005), pp. 531–547.
[4] P. A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, 2008.
[5] R. ACHANTA, A. SHAJI, K. SMITH, A. LUCCHI, P. FUA, AND S. SÜSSTRUNK, *SLIC superpixels compared to state-of-the-art superpixel methods*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2012), pp. 2274–2282.
[6] H. ATTOUCH AND J. BOLTE, *On the convergence of the proximal algorithm for nonsmooth functions involving analytic features*, Math. Program., 116 (2009), pp. 5–16.
[7] C. AUDET AND J. E. DENNIS JR., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., 17 (2006), pp. 188–217.
[8] C. AUDET AND W. HARE, *Derivative-Free and Blackbox Optimization*, Springer, Berlin, 2017.
[9] J. BAEZ AND J. P. MUNIAIN, *Gauge Fields, Knots and Gravity*, World Scientific, Singapore, 1994.
[10] A. S. BANDEIRA, K. SCHEINBERG, AND L. N. VICENTE, *Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization*, Math. Program., 134 (2012), pp. 223–257.
[11] A. S. BANDEIRA, K. SCHEINBERG, AND L. N. VICENTE, *Convergence of trust-region methods based on probabilistic models*, SIAM J. Optim., 24 (2014), pp. 1238–1264.
[12] R. R. BARTON, *Computing forward difference derivatives in engineering optimization*, Eng. Optimiz., 20 (1992), pp. 205–224.
[13] T. BENDORY, S. DEKEL, AND A. FEUER, *Super-resolution on the sphere using convex optimization*, IEEE Trans. Signal Process., 63 (2015), pp. 2253–2262.
[14] A. S. BERAHAS, R. H. BYRD, AND J. NOCEDAL, *Derivative-free optimization of noisy functions via quasi-Newton methods*, SIAM J. Optim., 29 (2019), pp. 965–993.
[15] S. R. BULÒ AND M. PELILLO, *A game-theoretic approach to hypergraph clustering*, IEEE Trans. Pattern Anal. Mach. Intell., 35 (2013), pp. 1312–1327.
[16] C. CARTIS AND L. ROBERTS, *A derivative-free Gauss-Newton method*, Math. Program. Comput., 11 (2019), pp. 631–674.
[17] S. CHEN, S. MA, A. SO, AND T. ZHANG, *Proximal gradient method for nonsmooth optimization over the Stiefel manifold*, SIAM J. Optim., 30 (2020), pp. 210–239.
[18] X. CHEN AND R. WOMERSLEY, *Spherical designs and nonconvex minimization for recovery of sparse signals on the sphere*, SIAM J. Imaging Sci., 11 (2018), pp. 1390–1415.
[19] Y. CHEN, L. QI, AND X. ZHANG, *The Fiedler vector of a Laplacian tensor for hypergraph partitioning*, SIAM J. Sci. Comput., 39 (2017), pp. A2508–A2537.

[20] A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *Trust-Region Methods*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
[21] A. R. CONN, K. SCHEINBERG, AND L. N. VICENTE, *Geometry of interpolation sets in derivative free optimization*, Math. Program., 111 (2008), pp. 141–172.
[22] A. R. CONN, K. SCHEINBERG, AND L. N. VICENTE, *Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation*, IMA J. Numer. Anal., 28 (2008), pp. 721–748.
[23] A. R. CONN, K. SCHEINBERG, AND L. N. VICENTE, *Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points*, SIAM J. Optim., 20 (2009), pp. 387–415.
[24] A. R. CONN, K. SCHEINBERG, AND L. N. VICENTE, *Introduction to Derivative-Free Optimization* SIAM, Philadelphia, 2009.
[25] A. R. CONN AND PH. L. TOINT, *An algorithm using quadratic interpolation for unconstrained derivative-free optimization*, in Nonlinear Optimization and Application, G. D. Pillo and F. Giannessi (eds.), Plenium Publishing, New York, 1996, pp. 27–47.
[26] C. F. CUI, Y. H. DAI, AND J. NIE, *All real eigenvalues of symmetric tensors*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1582–1601.
[27] P. DAS, D. DE, R. MAITI, B. CHAKRABORTY, C. B. PETERSON, *Estimating the optimal linear combination of biomarkers using spherically constrained optimization*, arXiv:1909.04024, (2019).
[28] M. FARSI, M. ASEMANI, AND M. R. RAHIMPOUR, *Mathematical modeling and optimization of multi-stage spherical reactor configurations for large scale dimethyl ether production*, Fuel Process. Technol., 126 (2014), pp. 207–214.
[29] M. FORNASIER, H. HUANG, L. PARESCHI, AND P. SÜNNEN, *Consensus-based optimization on the sphere II: convergence to global minimizers and machine learning*, arXiv:2001.11988v3, (2020).
[30] S. GÖRNER AND C. KANZOW, *On Newton's method for the Fermat–Weber location problem*, J. Optim. Theory Appl., 170 (2016), pp. 107–118.
[31] S. GRATTON, C. W. ROYER, L. N. VICENTE, AND Z. ZHANG, *Complexity and global rates of trust-region methods based on probabilistic models*, IMA J. Numer. Anal., 38 (2018), pp 1579–1597.
[32] A. GRIEWANK, *Computational differentiation and optimization*, in Mathematical Programming: State of the Art, J. R. Birge and K. G. Murty (eds.), The University of Michigan, Ann Arbor, MI, 1994, pp. 102–131.
[33] R. HOOKE AND T. A. JEEVES, *Direct search solution of numerical and statistical problems*, J. ACM, 8 (1961), pp. 212–229.
[34] J. HU, X. LIU, Z. WEN, AND Y. YUAN, *A brief introduction to manifold optimization*, Journal of the Operations Research Society of China, 8 (2020), pp. 199–248.
[35] S. HU AND L. QI, *Algebraic connectivity of an even uniform hypergraph*, J. Comb. Optim., 24 (2012), pp. 564–579.
[36] B. JIANG AND Y. DAI, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Math. Program. A, 153 (2015), pp. 535–575.
[37] I. N. KATZ AND L. COOPER, *Optimal location on a sphere*, Comput. Math. Appl., 6 (1980), pp. 175–196.
[38] T. G. KOLDA, R. M. LEWIS, AND V. TORCZON, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.
[39] C. LIU, A. LIU, AND S. HALABI, *A min–max combination of biomarkers to improve diagnostic accuracy*, Statist. Med., 30 (2011), pp. 2005–2014.
[40] S. ŁOJASIEWICZ, *Une propriété topologique des sous-ensembles analytiques réels*, in Les Équations

aux Dérivées Partielles, Éditions du centre National de la Recherche Scientifique, Paris, 1963, pp. 87–89.

[41] M. MARAZZI AND J. NOCEDAL, *Wedge trust region methods for derivative free optimization*, Math. Program. A, 91 (2002), pp. 289–305.

[42] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, In: Bachem A., Korte B., Grötschel M. (eds) Mathematical Programming The State of the Art. Springer, Berlin, Heidelberg. (1983), pp. 258–287.

[43] J. A. NELDER AND R. MEAD, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[44] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, Science & Business Media, 2006.

[45] M. J. D. POWELL, *A direct search optimization method that models the objective and constraint functions by linear interpolation*, in Advances in Optimization and Numerical Analysis, S. Gomez and J. P. Hennart (eds.), Kluwer, Dordrecht, 1994, pp. 51–67.

[46] M. J. D. POWELL, *UOBYQA: unconstrained optimization by quadratic approximation*, Math. Program., 92 (2002), pp. 555–582.

[47] M. J. D. POWELL, *The NEWUOA software for unconstrained optimization without derivatives*, in Large-Scale Nonlinear Optimization, P. G. Di, M. Roma (eds), Springer, Boston, MA, 2006.

[48] L. QI, *Eigenvalues of a real supersymmetric tensor*, J. Symb. Comput., 40 (2005), pp. 1302–1324.

[49] M. R. RAHIMPOUR, D. IRANSHAHI, AND A. M. BAHMANPOUR, *Dynamic optimization of a multi-stage spherical, radial flow reactor for the naphtha reforming process in the presence of catalyst deactivation using differential evolution (DE) method*, Int. J. Hydrogen Energy, 35 (2010), pp. 7498–7511.

[50] T. M. RAGONNEAU AND Z. ZHANG, *PDFO: Powell's derivative-free optimization solvers*, Available at http://zhangzk.net/software.html, (2020).

[51] P. R. SAMPAIO AND PH. L. TOINT, *A derivative-free trust-funnel method for equality-constrained nonlinear optimization*, Comput. Optim. Appl., 61 (2015), pp. 25–49.

[52] W. SUN AND Y. X. YUAN, *Optimization Theory and Methods: Nonlinear Programming*, Springer, Science & Business Media, 2006.

[53] V. TORCZON, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

[54] N. XIAO, X. LIU, AND Y. YUAN, *Exact penalty function for $L_{2,1}$ norm minimization over the Stiefel manifold*, SIAM J. Optim., (2020) (under review).

[55] K. YAMAGUCHI, *Borda winner in facility location problems on sphere*, Soc. Choice Welf., 46 (2016), pp. 893–898.

[56] H. ZHANG AND A. R. CONN, *On the local convergence of a derivative-free algorithm for least-squares minimization*, Comput. Optim. Appl., 51 (2012), pp. 481–507.

[57] H. ZHANG, A. R. CONN, AND K. SCHEINBERG, *A derivative-free algorithm for least-squares minimization*, SIAM J. Optim., 20 (2010), pp. 3555–3576.

[58] X. ZHANG, C. LING, AND L. QI, *The best rank-1 approximation of a symmetric tensor and related spherical optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 806–821.

[59] Z. ZHANG, *Sobolev seminorm of quadratic functions with applications to derivative-free optimization*, Math. Program., 146 (2014), pp. 77–96.