

# A robust Hessian-based trust region algorithm for spherical conformal parameterizations

Zhong-Heng Tan<sup>1,2</sup>, Tiexiang Li<sup>1,2,3,\*</sup>, Wen-Wei Lin<sup>2,3,4</sup> & Shing-Tung Yau<sup>5</sup>

<sup>1</sup>School of Mathematics and Shing-Tung Yau Center, Southeast University, Nanjing 210096, China;

<sup>2</sup>Nanjing Center for Applied Mathematics, Nanjing 211135, China;

<sup>3</sup>Shanghai Institute for Mathematics and Interdisciplinary Sciences, Shanghai 200433, China;

<sup>4</sup>Department of Applied Mathematics, Yang Ming Chiao Tung University, Hsinchu 300, China;

<sup>5</sup>Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China

Email: zhtan95@seu.edu.cn, txli@seu.edu.cn, wwlin@math.nctu.edu.tw, styau@tsinghua.edu.cn

Received October 8, 2023; accepted June 26, 2024; published online August 6, 2024

**Abstract** Surface parameterizations are widely applied in computer graphics, medical imaging, and transformation optics. In this paper, we rigorously derive the gradient vector and Hessian matrix of the discrete conformal energy for spherical conformal parameterizations of simply connected closed surfaces of genus-zero. In addition, we give the sparsity structure of the Hessian matrix, which leads to a robust Hessian-based trust region algorithm for the computation of spherical conformal maps. Numerical experiments demonstrate the local quadratic convergence of the proposed algorithm with low conformal distortions. We subsequently propose an application of our method to surface registrations that still maintain local quadratic convergence.

**Keywords** spherical conformal parameterization, conformal energy minimization, Riemann surfaces of genus zero, Hessian matrix, local quadratic convergence

**MSC(2020)** 49Q10, 52C26, 65D18, 65F05, 68U05

**Citation:** Tan Z-H, Li T, Lin W-W, et al. A robust Hessian-based trust region algorithm for spherical conformal parameterizations. Sci China Math, 2025, 68: 1461–1486, <https://doi.org/10.1007/s11425-023-2316-3>

## 1 Introduction

The Poincaré-Klein-Koebe theorem is a fundamental theorem in Riemann geometry that states that a simply connected Riemann surface  $\mathcal{M}$  is conformally equivalent to either a unit sphere  $\mathbb{S}^2$ , a complex plane  $\mathbb{C}$  or a unit disk  $\mathbb{D}$ . From a numerical point of view, for a given discrete triangular mesh of  $\mathcal{M}$ , we require that the conformal parameterization between  $\mathcal{M}$  and the canonical shape is a bijective map from  $\mathcal{M}$  to  $\mathbb{S}^2$  (the Riemann surface of genus-zero) or  $\mathbb{D}$  (the Riemann surface with a single boundary) while minimizing the total angle distortion induced by the Dirichlet energy [6, 29, 41]. Conformal parameterizations, also known as angle-preserving parameterizations, preserve the intersection angle of two arbitrary intersecting curves on the surface up to the map. In other words, conformal parameterizations preserve the local shapes of the surfaces. The development of digital 3D object technologies has enabled the representation of smooth surfaces in the real world through high-resolution meshes on computers. High-resolution

\* Corresponding author

meshes have the ability to characterize the intricate geometrical structure of surfaces. Nevertheless, the high resolution also makes computations and processes more challenging on complicated meshes. Recently, surface parameterizations have received increasing attention due to their ability to transform intricate geometries into simply shaped regions. It is clear that the parameterization of a surface is not necessarily unique, even when the target region is given. Conformal parameterization is one of the most commonly used parameterization methods. As a result, it is used in several fields, such as medical imaging [17, 21, 30, 34], texture mapping [22, 29], and transformation optics [12, 37].

A detailed overview of surface conformal parameterizations can be found in classical surveys [1, 15, 20, 23, 33]. Numerous methods for conformal parameterizations have been proposed, including boundary first flattening [32], discrete conformal equivalence [16], harmonic energy minimization [28], and solving the Laplacian-Beltrami equation with a particular Dirac delta function as the right-hand term [22]. In earlier years, based on the time-flow technique, Jin et al. [26] and Yang et al. [38] proposed the discrete Ricci flow with the conformal circle packing metric, which is a negative gradient flow of some convex energy and can be accelerated by the Newton method. However, the evaluations of the related coefficients at each iterative step are somewhat complicated due to the circle packing metric. Huang et al. [24] proposed the quasi-implicit Euler method (QIEM) in view of the nonlinear heat diffusion process with normalization on  $\mathbb{S}^2$ . However, the convergence can be very slow, and there is thus far no theory to support its convergence. With the great development of graphics processing unit (GPU)-based computation in recent years, parallelizable algorithms were subsequently proposed [4].

In 2015, Choi et al. [5] applied the quasi-conformal approach to spherical parameterization and proposed the fast landmark aligned spherical harmonic (FLASH) algorithm, which produces the composition of the quasi-conformal maps with the same Beltrami coefficient to obtain the conformal map. In 2019, Yueh et al. [40] proposed a north-south hemisphere alternating iteration, called spherical conformal energy minimization (SCEM), which alternatively maps the north and south poles to infinity and fixes the corresponding hemisphere while updating the other hemisphere. From numerical experiments, both FLASH and SCEM are satisfactory with high accuracy and effectiveness compared with those of the other previously mentioned methods. FLASH is a direct method that solves two double-sized linear systems compared with SCEM, which utilizes stereographic projection to transform the unit sphere  $\mathbb{S}^2$  to the extended complex plane  $\overline{\mathbb{C}}$ . The resulting map, which is composed of the inverse stereographic projection and the computed conformal map, is the ideal conformal parameterization from the given Riemann surface of genus zero to  $\mathbb{S}^2$ . In addition, it simplifies the spherical constraint problem in  $\mathbb{R}^3$  into an unconstrained problem in  $\overline{\mathbb{C}}$ , making it highly efficient. However, the stereographic projection maps the north pole of  $\mathbb{S}^2$  to infinity and others to  $\mathbb{C}$ . Therefore, the computational error near the north pole becomes relatively large in practical applications. To mitigate this error, the north-south hemisphere alternatingly iterative SCEM reduces the conformal distortion near the poles and has numerically sublinear convergence and asymptotically  $R$ -linear convergence. However, larger computational errors in conformal quantities are transferred to the junction of the equator. From a theoretical point of view, both FLASH and SCEM consider minimizing the Dirichlet energy on the extended complex plane  $\overline{\mathbb{C}}$ . From a numerical point of view, both FLASH and SCEM may get larger angle distortions near the junction region of the computational conformal map because they are connected by two submaps via stereographic projection.

Different from FLASH and SCEM, in this paper, we solve the spherical conformal energy optimization problem to obtain the conformal map by directly employing the spherical coordinates for the representation of the conformal energy. Unlike the stereographic projection, the spherical coordinate representation is not a conformal map. We derive the related gradient vector and Hessian matrix and develop a robust Hessian-based trust region (HBTR) algorithm that has local quadratic convergence. The main contributions of this paper can be divided into three folds.

- We use the spherical coordinates to represent the conformal energy on  $\mathbb{S}^2$  and develop the HBTR algorithm to directly minimize the discrete conformal energy, which is described as the difference between the discrete Dirichlet energy on  $\mathbb{S}^2$  and the image area of the conformal map. In Subsection 5.3, numerical experiments demonstrate that the discrete conformal energy, mean, and standard deviation (SD) are

reduced to 1/4 and 1/2, respectively, when the mesh size is refined by a half.

- We give the explicit derivation of the gradient vector and Hessian matrix and present the significant sparsity of the Hessian matrix. This benefits the development of fast computations for calculating the Newton iterations. Combining the trust region technique, we thus propose the robust HBTR algorithm.

- Numerical experiments and comparisons with existing algorithms demonstrate the advantages of our method in terms of conformality and robustness. The computational cost of the HBTR algorithm is of the same order of magnitude as that of FLASH and SCEM but slightly more expensive for most models due to spending a high percentage of time selecting the convergence region. Furthermore, the bijectivity can almost be guaranteed because the conformal energy is expressed by the unity of spherical coordinates. For few meshes resulting in nonbijective maps, folding can be easily removed by performing a postprocessing algorithm [14].

The rest of this paper is organized as follows. Section 2 provides a brief review of the conformal map and conformal energy. Then, in Section 3, we present the theoretical derivation of the gradient vector and Hessian matrix. In Section 4, we describe the proposed algorithm for spherical conformal parameterization that uses the sparsity property of the Hessian matrix. The numerical performance and comparison with other methods are presented in Section 5. In Section 6, we demonstrate an application of the algorithm to the surface registration. A concluding remark is given in Section 7.

The frequently-used notations in this paper are listed here.  $\mathbf{a}$  and  $\mathbf{s}$  denote vectors.  $\mathbf{a}_i$  denotes the  $i$ -th entry of  $\mathbf{a}$ .  $\mathbf{1}_{m \times n}$  denotes the  $m \times n$  matrix of all ones. The notation without subscript  $\mathbf{1}$  denotes the vector of all ones with proper dimension.  $\mathbf{e}_i$  denotes the  $i$ -th column of the identity matrix with the proper dimension.  $[A]_{ij}$  denotes the  $(i, j)$ -th entry of the matrix  $A$ .  $\text{diag}(\mathbf{a})$  denotes the diagonal matrix with the  $(i, i)$ -th entry being  $\mathbf{a}_i$ .  $[v_i, v_j]$  denotes the edge formed by  $v_i$  and  $v_j$ .  $[v_i, v_j, v_k]$  denotes the triangle formed by  $v_i$ ,  $v_j$ , and  $v_k$ , and  $[[v_i, v_j, v_k]]$  denotes the area of that triangle. Furthermore,  $\langle A, B \rangle := \text{trace}(A^T B)$  denotes the inner product of two matrices  $A$  and  $B$ . The other notations are defined wherever they appear.

## 2 The conformal map and the conformal energy

In this section, we briefly review the continuous conformal map and the conformal energy. Readers can refer to [18, 19, 22, 25, 31] for more details.

Let  $\mathcal{M} = \mathbf{r}(u, v)$  be a surface and  $f$  be a continuous and bijective vector-valued map on  $\mathcal{M}$ , which maps  $\mathcal{M}$  to another surface  $\tilde{\mathcal{M}} = \tilde{\mathbf{r}}(u, v)$  with  $(u, v) \in \Omega \subset \mathbb{R}^2$ . Let  $\gamma_1, \gamma_2 \subset \mathcal{M}$  be two arbitrary curves intersecting at a point. Then, we call  $f$  conformal if the intersecting angle of  $\gamma_1$  and  $\gamma_2$  equals that of  $f(\gamma_1)$  and  $f(\gamma_2)$  in  $\tilde{\mathcal{M}}$ . Equivalently,  $f$  is conformal if and only if the first fundamental forms  $I(u, v)$  and  $\tilde{I}(u, v)$  of  $\mathcal{M}$  and  $\tilde{\mathcal{M}}$  with respect to  $(u, v)$ , i.e.,

$$I(u, v) = \begin{bmatrix} \langle \mathbf{r}_u, \mathbf{r}_u \rangle & \langle \mathbf{r}_u, \mathbf{r}_v \rangle \\ \langle \mathbf{r}_v, \mathbf{r}_u \rangle & \langle \mathbf{r}_v, \mathbf{r}_v \rangle \end{bmatrix}, \quad \tilde{I}(u, v) = \begin{bmatrix} \langle \tilde{\mathbf{r}}_u, \tilde{\mathbf{r}}_u \rangle & \langle \tilde{\mathbf{r}}_u, \tilde{\mathbf{r}}_v \rangle \\ \langle \tilde{\mathbf{r}}_v, \tilde{\mathbf{r}}_u \rangle & \langle \tilde{\mathbf{r}}_v, \tilde{\mathbf{r}}_v \rangle \end{bmatrix},$$

satisfy

$$\tilde{I}(u, v) = \eta(u, v)I(u, v),$$

where  $\mathbf{r}_u := \frac{\partial \mathbf{r}}{\partial u}$ ,  $\mathbf{r}_v := \frac{\partial \mathbf{r}}{\partial v}$ , and  $\eta(u, v)$  is a positive scalar function on  $\Omega$ . Then, the conformal energy functional [25] of  $f$  is defined as

$$E_C(f) = \frac{1}{2} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|_F^2 d\sigma - \mathcal{A}(f), \quad (2.1)$$

where  $\nabla_{\mathcal{M}}$  is the tangential gradient,  $d\sigma$  is the area element on  $\mathcal{M}$ , and  $\mathcal{A}(f)$  is the area of the image surface  $\tilde{\mathcal{M}} = f(\mathcal{M})$ . Let us note that the first term in (2.1) is the Dirichlet energy functional of  $f$ . It has been proved in [25, 31] that

- $E_C(f) \geq 0$ , and
- $E_C(f) = 0$  if and only if  $f$  is conformal.

From these properties, it is reasonable to adopt conformal energy minimization (CEM) to obtain the conformal map. In the continuous scheme, if the target surface  $\tilde{M}$  is given, its area remains constant. Thus, a method to solve (2.1) is to minimize the Dirichlet energy, which is a quadratic functional. In the discrete scheme, the area is actually not a constant. Introducing the discrete area term typically yields a better conformal parameterization. It is easy to verify that the optimization problem (2.1) has a trivial solution  $f = \text{constant}$ . In this case, all the vertices shrink into a point, and the area of the formed region becomes 0, which violates the requirement of  $f$  being bijective. We refer to the trivial solution as ‘degeneration’. Directly minimizing the Dirichlet energy without any additional constraints often leads to degeneration in practical computations. The area term is a natural penalty for the parameterization and hence can greatly weaken the degeneration during iteration [25]. On the other hand, introducing the area term can further decrease the conformal distortion. For disk parameterizations, Yueh et al. [41] proposed a disk conformal parameterization algorithm through Dirichlet energy minimization, while a novel algorithm minimizing the disk conformal energy was later proposed by Kuo et al. [27], in which they derived a particular and simple area expression of a discrete disk represented by polar coordinates to design a fast algorithm. As a consequence, conformal energy minimization [27] gives significantly less conformal energy and angle distortion compared with those of the Dirichlet energy minimization [41].

For the spherical parameterization, it is clearly feasible to introduce the area term. However, it is a significant challenge that the area of the discrete sphere, which is the sum of areas of all the triangles formed by vertices, cannot be simply expressed as in the disk case, even if both vertices are of unit length. Simultaneously, the gradient vector and Hessian matrix of the conformal energy also have complicated expressions. In the next section, we carefully derive the expression of the conformal energy for the spherical parameterization with spherical coordinates, along with its gradient vector and Hessian matrix. The Hessian matrix has a special sparsity structure as the Laplacian matrix, which guides the fast construction and computation associated with it.

### 3 Discrete conformal energy on closed surfaces

Let  $M$  be a discrete closed surface of genus-zero composed of triangles, called a triangulation. Given that  $M$  has  $n$  vertices, we denote the vertex set, edge set, and triangle face set of  $M$  by  $\mathcal{V}(M) = \{v_i, i = 1, 2, \dots, n\}$ ,  $\mathcal{E}(M) = \{[v_i, v_j]\}$ , and  $\mathcal{F}(M) = \{T_{ijk} := [v_i, v_j, v_k]\}$ , respectively.

In this paper, we aim to find a piecewise linear map  $f$  that maps the surface  $M$  to a discrete unit sphere  $\mathbb{S}^2$  conformally, in which all of the vertices are unit vectors. Every point in a triangle can be represented as the convex combination of its vertices via barycentric coordinates,

$$f(\hat{v}) = \lambda_i(\hat{v})f(v_i) + \lambda_j(\hat{v})f(v_j) + \lambda_k(\hat{v})f(v_k) \quad \text{for } \hat{v} \in T_{ijk}, \quad (3.1)$$

where

$$\lambda_i(\hat{v}) = \frac{|[\hat{v}, v_j, v_k]|}{|T_{ijk}|}, \quad \lambda_j(\hat{v}) = \frac{|[v_i, \hat{v}, v_k]|}{|T_{ijk}|}, \quad \lambda_k(\hat{v}) = \frac{|[v_i, v_j, \hat{v}]|}{|T_{ijk}|}.$$

After  $f(v_i), i = 1, 2, \dots, n$  is obtained, the discrete sphere is also formed. Hence, we compute  $\mathbf{f} = [\mathbf{f}_1^T, \mathbf{f}_2^T, \dots, \mathbf{f}_n^T]^T \in \mathbb{R}^{n \times 3}$  with  $\mathbf{f}_i = f(v_i) \in \mathbb{R}^{1 \times 3}$  for the given triangulation of  $M$ .

From the perspective of the conformal energy (2.1), the conformal map  $f$  is obtained by solving the optimization problem

$$\mathbf{f}^* = \underset{\mathbf{f}: \|\mathbf{f}_i\|=1, i=1,2,\dots,n}{\operatorname{argmin}} E_C(f), \quad (3.2)$$

where the discrete conformal energy  $E_C(f)$  of  $f$  on  $M$  is given by

$$E_C(f) = \frac{1}{2} \langle L\mathbf{f}, \mathbf{f} \rangle - A(f), \quad (3.3)$$

in which  $L$  is the Laplacian matrix defined as

$$[L]_{ij} = \begin{cases} -w_{ij}, & \text{if } i \neq j, \quad [v_i, v_j] \in \mathcal{E}(M), \\ \sum_{k \in \mathcal{N}(i)} w_{ik}, & \text{if } i = j, \\ 0, & \text{if } [v_i, v_j] \notin \mathcal{E}(M) \end{cases} \quad (3.4)$$

with cotangent weights  $w_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \alpha_{ji})$ . Here,  $\alpha_{ij}$  and  $\alpha_{ji}$  are the angles opposite to the edge  $[v_i, v_j]$ <sup>1)</sup> and  $\mathcal{N}(i)$  is the index set of the adjacent vertices  $v_i$ , as shown in Figure 1. Readers can refer to [19, Chapter 24] for a detailed derivation of the discrete Dirichlet energy on triangulation.  $A(f)$  in (3.3) is the area of the triangulation  $f(M)$ , which can be written as the sum of areas of all the triangles in  $f(M)$ , i.e.,  $A(f) = \sum_{T_{ijk} \in \mathcal{F}(M)} |f(T_{ijk})|$ .

Let  $v_{ij} = v_i - v_j$  and  $\mathbf{f}_{ij} = \mathbf{f}_i - \mathbf{f}_j$  for simplicity,  $\alpha_{ij}(f)$  and  $\alpha_{ki}(f)$  be the angle, respectively, opposite to the edges  $[\mathbf{f}_i, \mathbf{f}_j]$  and  $[\mathbf{f}_k, \mathbf{f}_i] \in \mathcal{E}(f(M))$ , and  $L(f)$  be a Laplacian matrix with respect to  $f$  as  $L$ , by replacing the cotangent weights as  $w_{ij}(f) = \frac{1}{2}(\cot \alpha_{ij}(f) + \cot \alpha_{ji}(f))$  in the target sphere  $\mathbb{S}^2$ .  $A(f)$  and its gradient with respect to  $\mathbf{f}$  can be represented as in the following lemma.

**Lemma 3.1.** *The area of the image of  $f$  and its gradient can be represented as*

$$\begin{aligned} A(f) &= \frac{1}{2} \langle L(f)\mathbf{f}, \mathbf{f} \rangle, \\ \nabla_{\mathbf{f}} A(f) &= L(f)\mathbf{f}. \end{aligned}$$

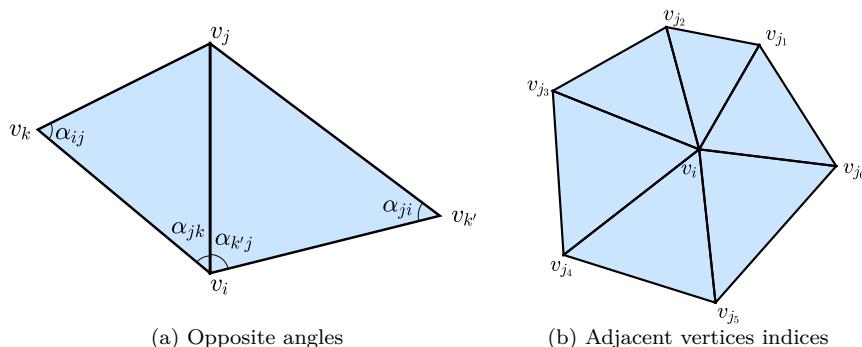
*Proof.* The area of a triangle  $f(T_{ijk}) \in \mathcal{F}(f(M))$  can be calculated by

$$\begin{aligned} |f(T_{ijk})| &= \frac{1}{2} \|\mathbf{f}_{ij} \times \mathbf{f}_{jk}\| = \frac{\|\mathbf{f}_{ij} \times \mathbf{f}_{jk}\|^2}{4|f(T_{ijk})|} \\ &= \frac{\langle \mathbf{f}_{jk} \times (\mathbf{f}_{ij} \times \mathbf{f}_{jk}), \mathbf{f}_{ij} \rangle}{4|f(T_{ijk})|} = \frac{\langle \mathbf{f}_{jk} \times (\mathbf{f}_{ki} \times \mathbf{f}_{ij}), \mathbf{f}_{ij} \rangle}{4|f(T_{ijk})|} \end{aligned} \quad (3.5a)$$

$$= \frac{\langle \langle \mathbf{f}_{ij}, \mathbf{f}_{jk} \rangle \mathbf{f}_{ki} - \langle \mathbf{f}_{jk}, \mathbf{f}_{ki} \rangle \mathbf{f}_{ij}, \mathbf{f}_{ij} \rangle}{4|f(T_{ijk})|} \quad (3.5b)$$

$$= -\frac{\langle \mathbf{f}_{ki}, \mathbf{f}_{ij} \rangle \langle \mathbf{f}_{jk}, \mathbf{f}_{jk} \rangle + \langle \mathbf{f}_{ij}, \mathbf{f}_{jk} \rangle \langle \mathbf{f}_{ki}, \mathbf{f}_{ki} \rangle + \langle \mathbf{f}_{jk}, \mathbf{f}_{ki} \rangle \langle \mathbf{f}_{ij}, \mathbf{f}_{ij} \rangle}{8|f(T_{ijk})|} \quad (3.5c)$$

$$= \frac{1}{4}(\cot \alpha_{jk}(f) \|\mathbf{f}_{jk}\|^2 + \cot \alpha_{ki}(f) \|\mathbf{f}_{ki}\|^2 + \cot \alpha_{ij}(f) \|\mathbf{f}_{ij}\|^2),$$



**Figure 1** (Color online) Illustrations of opposite angles and adjacent vertices indices sets. (a)  $\alpha_{ij}$  and  $\alpha_{ji}$  are the pair of opposite angles of the edge  $[v_i, v_j]$ ;  $\alpha_{jk}$  and  $\alpha_{kj}$  are the opposite angles of the edge  $[v_j, v_k]$  in  $T_{ijk}$  and  $[v_{k'}, v_j]$  in  $T_{k'ji}$ , respectively. (b)  $v_{j_1}, \dots, v_{j_6}$  are the adjacent vertices of  $v_i$ . Hence,  $\mathcal{N}(i) = \{j_1, j_2, \dots, j_6\}$

<sup>1)</sup> A closed surface has no boundary. Therefore, an edge must correspond to 2 opposite angles.

In the last equation (3.5c), we use the formula  $\cot \alpha_{ij}(f) = -\frac{\langle \mathbf{f}_{jk}, \mathbf{f}_{ki} \rangle}{2|T_{ijk}|}$ . Hence, the area  $A(f)$  can be represented as

$$\begin{aligned} A(f) &= \sum_{T_{ijk} \in \mathcal{F}(M)} |f(T_{ijk})| \\ &= \frac{1}{4} \sum_{T_{ijk} \in \mathcal{F}(M)} (\cot \alpha_{jk}(f) \|\mathbf{f}_{jk}\|^2 + \cot \alpha_{ki}(f) \|\mathbf{f}_{ki}\|^2 + \cot \alpha_{ij}(f) \|\mathbf{f}_{ij}\|^2) \\ &= \frac{1}{2} \sum_{ij:[v_i, v_j] \in \mathcal{E}(M)} w_{ij}(f) \|\mathbf{f}_{ij}\|^2 = \frac{1}{2} \langle L(f)\mathbf{f}, \mathbf{f} \rangle. \end{aligned}$$

Then, by using the derivation of (3.5a) and (3.5b), we have

$$\begin{aligned} \frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_i} &= \frac{2\mathbf{f}_{jk} \times (\mathbf{f}_{ij} \times \mathbf{f}_{jk})}{8|f(T_{ijk})|} = \frac{\langle \mathbf{f}_{ij}, \mathbf{f}_{jk} \rangle \mathbf{f}_{ki} - \langle \mathbf{f}_{jk}, \mathbf{f}_{ki} \rangle \mathbf{f}_{ij}}{4|f(T_{ijk})|} \\ &= \frac{1}{2} (\cot \alpha_{ij}(f) \mathbf{f}_{ij} - \cot \alpha_{ki}(f) \mathbf{f}_{ki}) \in \mathbb{R}^{1 \times 3}. \end{aligned} \quad (3.6)$$

Hence, the derivative of  $A(f)$  with respect to  $\mathbf{f}_i$  is

$$\frac{\partial A(f)}{\partial \mathbf{f}_i} = \sum_{T_{ijk} \in \mathcal{F}(M)} \frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_i} = \frac{1}{2} \sum_{T_{ijk} \in \mathcal{F}(M)} (\cot \alpha_{ij}(f) \mathbf{f}_{ij} - \cot \alpha_{ki}(f) \mathbf{f}_{ki}) = \mathbf{e}_i^T L(f) \mathbf{f}$$

for  $i = 1, 2, \dots, n$ . This completes the proof.  $\square$

From (3.3) and Lemma 3.1, we have

$$E_C(f) = \frac{1}{2} \langle D(f)\mathbf{f}, \mathbf{f} \rangle \quad \text{with } D(f) = L - L(f), \quad (3.7)$$

$$\nabla_{\mathbf{f}} E_C(f) = D(f)\mathbf{f}. \quad (3.8)$$

Clearly,  $D(f)$  has the same sparsity structure as  $L$  and  $L(f)$ , and the corresponding cotangent weight is  $[D(f)]_{ij} := \tilde{w}_{ij} = w_{ij} - w_{ij}(f)$ .

**Remark 3.2.** If  $f^*$  is a conformal map, the inner angles of the triangles in  $M$  are identical to those in  $f^*(M)$ . We have  $w_{ij}(f^*) = w_{ij}$  and  $D(f) = \mathbf{0}$ , which leads to  $E_C(f^*) = 0$ .

Let  $\mathbf{f} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]$  and  $\text{vec}(\mathbf{f}) = [\mathbf{x}^T, \mathbf{y}^T, \mathbf{z}^T]^T$  be its vectorization. Since the vertices of  $\mathbf{f}$  are on the sphere, we adopt the spherical coordinates to represent all the vertices by

$$\mathbf{x} = \cos \boldsymbol{\theta} \odot \sin \phi, \quad \mathbf{y} = \sin \boldsymbol{\theta} \odot \sin \phi, \quad \mathbf{z} = \cos \phi, \quad (3.9a)$$

and let

$$\mathbf{u} = \cos \boldsymbol{\theta} \odot \cos \phi, \quad \mathbf{v} = \sin \boldsymbol{\theta} \odot \cos \phi, \quad \mathbf{w} = \sin \phi, \quad (3.9b)$$

where  $\boldsymbol{\theta}, \phi \in \mathbb{R}^n$  are the azimuth and elevation angle vectors of the corresponding vertices of  $\mathbf{f}$ , respectively, and  $\odot$  is the Hadamard product. Similar to the stereographic projection in [5, 40], the spherical coordinate representation eliminates the unit-length constraint and transforms the Cartesian coordinate  $[\mathbf{x}, \mathbf{y}, \mathbf{z}]$  into the spherical coordinate  $[\boldsymbol{\theta}, \phi]$ , reducing the input variable scale by 1/3. Furthermore, the spherical coordinate projects the whole sphere into a bounded region  $[0, 2\pi] \times [0, \pi]$ , which avoids the computational error near the north pole [5] or the equator [40] by using the stereographic projection. In the further discussion, we define  $\nabla := [\nabla_{\boldsymbol{\theta}}^T, \nabla_{\phi}^T]^T$  with respect to  $(\boldsymbol{\theta}, \phi)$  unless a special illustration in the rest of the paper. The gradient of any scalar  $a$  and the Jacobi matrix of any vector  $\mathbf{a}$  with respect to  $(\boldsymbol{\theta}, \phi)$  are represented as

$$\nabla a = \left[ \frac{\partial a}{\partial \boldsymbol{\theta}_1}, \dots, \frac{\partial a}{\partial \boldsymbol{\theta}_n}, \frac{\partial a}{\partial \phi_1}, \dots, \frac{\partial a}{\partial \phi_n} \right]^T, \quad \nabla \mathbf{a} = \left[ \frac{\partial \mathbf{a}}{\partial \boldsymbol{\theta}_1}, \dots, \frac{\partial \mathbf{a}}{\partial \boldsymbol{\theta}_n}, \frac{\partial \mathbf{a}}{\partial \phi_1}, \dots, \frac{\partial \mathbf{a}}{\partial \phi_n} \right]^T,$$

respectively. Furthermore, the gradient of  $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}$ , and  $\mathbf{w}$  can be represented by themselves,

$$\begin{aligned}\nabla \mathbf{x} &= [-\text{diag}(\mathbf{y}), \text{diag}(\mathbf{u})], \quad \nabla \mathbf{y} = [\text{diag}(\mathbf{x}), \text{diag}(\mathbf{v})], \quad \nabla \mathbf{z} = [\mathbf{0}_{n \times n}, -\text{diag}(\mathbf{w})], \\ \nabla \mathbf{u} &= [-\text{diag}(\mathbf{v}), -\text{diag}(\mathbf{x})], \quad \nabla \mathbf{v} = [\text{diag}(\mathbf{u}), -\text{diag}(\mathbf{y})], \quad \nabla \mathbf{w} = [\mathbf{0}_{n \times n}, \text{diag}(\mathbf{z})].\end{aligned}\quad (3.10)$$

We now vectorize the gradient (3.8) and let

$$\mathbf{p} = D(f)\mathbf{x}, \quad \mathbf{q} = D(f)\mathbf{y}, \quad \mathbf{r} = D(f)\mathbf{z}. \quad (3.11)$$

From (3.7), we have  $E_C(f) = \frac{1}{2}(\mathbf{x}^T \mathbf{p} + \mathbf{y}^T \mathbf{q} + \mathbf{z}^T \mathbf{r})$ . Then, from (3.8) and (3.10), the gradient of  $E_C(f)$  with respect to  $(\boldsymbol{\theta}, \boldsymbol{\phi})$  is

$$\mathbf{g} := \nabla E_C = \nabla \text{vec}(\mathbf{f})^T \nabla_{\text{vec}(\mathbf{f})} E_C \quad (3.12a)$$

$$= \nabla \mathbf{x}^T \mathbf{p} + \nabla \mathbf{y}^T \mathbf{q} + \nabla \mathbf{z}^T \mathbf{r} = \begin{bmatrix} -\text{diag}(\mathbf{y})\mathbf{p} + \text{diag}(\mathbf{x})\mathbf{q} \\ \text{diag}(\mathbf{u})\mathbf{p} + \text{diag}(\mathbf{v})\mathbf{q} - \text{diag}(\mathbf{w})\mathbf{r} \end{bmatrix} \in \mathbb{R}^{2n \times 1}. \quad (3.12b)$$

Moreover, the Hessian matrix is also obtained

$$\begin{aligned}H &:= \nabla \nabla E_C \\ &= (\nabla \mathbf{x}^T \nabla \mathbf{p} + \nabla \mathbf{y}^T \nabla \mathbf{q} + \nabla \mathbf{z}^T \nabla \mathbf{r}) + \begin{bmatrix} -\text{diag}(\mathbf{p})\nabla \mathbf{y} + \text{diag}(\mathbf{q})\nabla \mathbf{x} \\ \text{diag}(\mathbf{p})\nabla \mathbf{u} + \text{diag}(\mathbf{q})\nabla \mathbf{v} - \text{diag}(\mathbf{r})\nabla \mathbf{w} \end{bmatrix} \in \mathbb{R}^{2n \times 2n},\end{aligned}\quad (3.13)$$

where

$$\begin{aligned}\nabla \mathbf{p} &= D(f)\nabla \mathbf{x} - \sum_{\ell=1}^n \mathbf{x}_\ell \nabla(L(f)\mathbf{e}_\ell), \\ \nabla \mathbf{q} &= D(f)\nabla \mathbf{y} - \sum_{\ell=1}^n \mathbf{y}_\ell \nabla(L(f)\mathbf{e}_\ell), \\ \nabla \mathbf{r} &= D(f)\nabla \mathbf{z} - \sum_{\ell=1}^n \mathbf{z}_\ell \nabla(L(f)\mathbf{e}_\ell).\end{aligned}$$

Via the chain rule

$$\nabla(L(f)\mathbf{e}_\ell) = \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell)\nabla \mathbf{x} + \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell)\nabla \mathbf{y} + \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell)\nabla \mathbf{z} \in \mathbb{R}^{n \times 2n},$$

we have

$$\begin{aligned}&\begin{bmatrix} \sum_{\ell=1}^n \mathbf{x}_\ell \nabla(L(f)\mathbf{e}_\ell) \\ \sum_{\ell=1}^n \mathbf{y}_\ell \nabla(L(f)\mathbf{e}_\ell) \\ \sum_{\ell=1}^n \mathbf{z}_\ell \nabla(L(f)\mathbf{e}_\ell) \end{bmatrix} \\ &= \begin{bmatrix} \sum_{\ell=1}^n \mathbf{x}_\ell \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{x}_\ell \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{x}_\ell \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell) \\ \sum_{\ell=1}^n \mathbf{y}_\ell \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{y}_\ell \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{y}_\ell \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell) \\ \sum_{\ell=1}^n \mathbf{z}_\ell \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{z}_\ell \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{z}_\ell \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell) \end{bmatrix} \begin{bmatrix} \nabla \mathbf{x} \\ \nabla \mathbf{y} \\ \nabla \mathbf{z} \end{bmatrix}.\end{aligned}\quad (3.14)$$

Plugging (3.14) into (3.13), we can rewrite the Hessian matrix as

$$H = \begin{bmatrix} \nabla \mathbf{x} \\ \nabla \mathbf{y} \\ \nabla \mathbf{z} \end{bmatrix}^T (L_1 - L_2) \begin{bmatrix} \nabla \mathbf{x} \\ \nabla \mathbf{y} \\ \nabla \mathbf{z} \end{bmatrix} - \begin{bmatrix} \text{diag}(\mathbf{p} \odot \mathbf{x} + \mathbf{q} \odot \mathbf{y}) & \text{diag}(\mathbf{p} \odot \mathbf{v} - \mathbf{q} \odot \mathbf{u}) \\ \text{diag}(\mathbf{p} \odot \mathbf{v} - \mathbf{q} \odot \mathbf{u}) & \text{diag}(\mathbf{p} \odot \mathbf{x} + \mathbf{q} \odot \mathbf{y} + \mathbf{r} \odot \mathbf{z}) \end{bmatrix} \quad (3.15a)$$

$$\equiv H_1 - H_2 - K, \quad (3.15b)$$

where

$$H_s = [\nabla \mathbf{x}^T, \nabla \mathbf{y}^T, \nabla \mathbf{z}^T] L_s \begin{bmatrix} \nabla \mathbf{x} \\ \nabla \mathbf{y} \\ \nabla \mathbf{z} \end{bmatrix}, \quad s = 1, 2,$$

$$K = \begin{bmatrix} \text{diag}(\mathbf{p} \odot \mathbf{x} + \mathbf{q} \odot \mathbf{y}) & \text{diag}(\mathbf{p} \odot \mathbf{v} - \mathbf{q} \odot \mathbf{u}) \\ \text{diag}(\mathbf{p} \odot \mathbf{v} - \mathbf{q} \odot \mathbf{u}) & \text{diag}(\mathbf{p} \odot \mathbf{x} + \mathbf{q} \odot \mathbf{y} + \mathbf{r} \odot \mathbf{z}) \end{bmatrix}$$

with

$$L_1 = \begin{bmatrix} D(f) & & \\ & D(f) & \\ & & D(f) \end{bmatrix}, \quad (3.16)$$

$$L_2 = \begin{bmatrix} \sum_{\ell=1}^n \mathbf{x}_\ell \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{x}_\ell \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{x}_\ell \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell) \\ \sum_{\ell=1}^n \mathbf{y}_\ell \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{y}_\ell \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{y}_\ell \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell) \\ \sum_{\ell=1}^n \mathbf{z}_\ell \nabla_{\mathbf{x}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{z}_\ell \nabla_{\mathbf{y}}(L(f)\mathbf{e}_\ell) & \sum_{\ell=1}^n \mathbf{z}_\ell \nabla_{\mathbf{z}}(L(f)\mathbf{e}_\ell) \end{bmatrix}.$$

In the matrix  $L_2$ , the Jacobian matrix of each column of  $L(f)$  should be considered. For the  $\ell$ -th column entries  $L(f)\mathbf{e}_\ell$ , it is clear that  $[L(f)]_{il} \neq 0$  if and only if  $i \in \mathcal{N}(\ell)$  or  $i = \ell$  by the definition of the Laplacian matrix in (3.4). The nondiagonal entries are negative cotangent weights, and the diagonal entries are the sums of the cotangent weights. Hence, the Jacobian matrix of  $L(f)\mathbf{e}_\ell$  is formed by the gradient of cotangent weights  $w_{ij}(f)$ , especially the gradient of cotangent functions  $c_{ij}(f) := \cot \alpha_{ij}(f)$ . Before discussing entries of  $\nabla L\mathbf{e}_\ell$ , we first give a lemma for the sparsity structure of the Jacobian block  $\sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell)$ ,  $\mathbf{a}, \mathbf{b} = \mathbf{x}, \mathbf{y}, \mathbf{z}$ .

**Lemma 3.3.** *Each subblock  $\sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell)$  of  $L_2$  in (3.16) with  $\mathbf{a}, \mathbf{b} = \mathbf{x}, \mathbf{y}, \mathbf{z}$  is of the identical sparsity structure to  $L$ .*

*Proof.* By the cotangent formula in (3.6), one can observe that the gradients with respect to  $\mathbf{x}, \mathbf{y}, \mathbf{z}$  are the 1st, 2nd, and 3rd entries of that with respect to  $\mathbf{f}$ , respectively. Hence,  $\sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell)$  has identical sparsity for all  $\mathbf{a}, \mathbf{b} = \mathbf{x}, \mathbf{y}, \mathbf{z}$ . Here, we denote the nonzero indices set of  $\mathbf{e}_i^T \nabla_{\mathbf{b}}(L(f)\mathbf{e}_j)$  by  $S(i, j) = \{k \mid [\nabla_{\mathbf{b}}(L(f)\mathbf{e}_j)]_{ik} \neq 0\}$ . Clearly, the nonzero indices set of  $\mathbf{e}_i^T \sum_{\ell=1}^n (L(f)\mathbf{e}_\ell)$  is  $\bigcup_{\ell=1}^n S(i, \ell)$ .

- For  $i \neq j$  and  $[v_i, v_j] \notin \mathcal{E}(M)$ , the entries are 0, and therefore, the gradients are also 0. Hence,

$$S(i, j) = \emptyset. \quad (3.17)$$

• For  $i \neq j$  and  $[v_i, v_j] \in \mathcal{E}(M)$ , by the definition of cotangent weight  $w_{ij}(f) = \frac{1}{2}(c_{ij}(f) + c_{ji}(f))$ , we have  $\mathbf{e}_i^T \nabla_{\mathbf{b}}(L(f)\mathbf{e}_j) = -\nabla_{\mathbf{b}} w_{ij}(f) = -\frac{1}{2}(\nabla_{\mathbf{b}} c_{ij}(f) + \nabla_{\mathbf{b}} c_{ji}(f))$ . As shown in Figure 1(a), one can see that  $w_{ij}(f)$  relates to only 4 vertices  $\mathbf{f}_i, \mathbf{f}_j, \mathbf{f}_k$ , and  $\mathbf{f}_{k'}$ . Therefore, the entries of  $\nabla_{\mathbf{b}} w_{ij}(f)$  are 0 except for the  $i, j, k, k'$ -th entries, i.e.,

$$S(i, j) = \{i, j, k, k'\} \subset \mathcal{N}(i) \cup \{i\}. \quad (3.18)$$

• For  $i = j$ , we have  $\mathbf{e}_i^T \nabla_{\mathbf{b}}(L(f)\mathbf{e}_i) = \nabla_{\mathbf{b}}[L(f)]_{ii} = \sum_{j \in \mathcal{N}(i)} \nabla_{\mathbf{b}} w_{ij}(f)$ . Hence, we can easily verify that  $\mathbf{e}_i^T \nabla_{\mathbf{b}}(L(f)\mathbf{e}_i)$  relates to the whole adjacent vertices of  $\mathbf{f}_i$  from Figure 1(b). It immediately follows that

$$S(i, i) = \mathcal{N}(i) \cup \{i\}. \quad (3.19)$$

Combining (3.17)–(3.19), we have  $S(i, \ell) \subset \mathcal{N}(i) \cup \{i\} = S(i, i)$  for  $\ell \neq i$ . It follows that  $\bigcup_{\ell=1}^n S(i, \ell) = S(i, i) = \mathcal{N}(i) \cup \{i\}$ . Since the index set of nonzero entries of  $\mathbf{e}_i^T L$  is  $\mathcal{N}(i) \cup \{i\}$ , the lemma is obtained immediately.  $\square$

Lemma 3.3 demonstrates that  $[\sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell)]_{ij} \neq 0$  if and only if  $[v_i, v_j] \in \mathcal{E}(M)$  or  $i = j$  for every  $i$  and  $j$ . As a result,  $L_2$  in (3.16) is stacked by 9 matrices with the same sparsity as  $L$  in the  $3 \times 3$  form. Additionally, one can further observe that  $H$  in (3.15a) is also stacked by 4 matrices with the same sparsity as  $L$  in the  $2 \times 2$  form. We summarize this idea as the following theorem.

**Theorem 3.4.**  *$H$  in (3.15a) is of identical sparsity to  $\mathbf{1}_{2 \times 2} \otimes L$ .*

*Proof.* By the Hessian matrix representation of  $H$  in (3.15a) and Lemma 3.3, the proof is obtained.  $\square$

Theorem 3.4 demonstrates the high sparsity of the Hessian matrix. This property confirms the feasibility of practically solving the large-scale linear system

$$H\mathbf{s} = -\mathbf{g}, \quad (3.20)$$

which inspires us to apply a Newton-type algorithm for solving the optimization problem (3.2). Furthermore, in the representation of  $H$  in (3.15a),  $L_1 - L_2 = \nabla_{\text{vec}(\mathbf{f})} \nabla_{\text{vec}(\mathbf{f})}^T E_C$  is the Hessian matrix of the conformal energy with respect to  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$ . The term  $[\nabla \mathbf{x}^T, \nabla \mathbf{y}^T, \nabla \mathbf{z}^T]^T$  in (3.15a) is the Jacobian matrix of  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  with respect to  $\boldsymbol{\theta}$  and  $\phi$ , which is stacked by  $3 \times 2$  diagonal matrices. The matrix  $K$  in (3.15a) is also stacked by  $2 \times 2$  diagonal matrices. These structures are invariant for arbitrary parameterization, and therefore, the sparsity structure of the Hessian matrix  $H$  is also invariant. Hence, this property is also available on other parameterizations of closed surfaces, including other expressions of spheres and other target regions.

We proceed further to analyze the entries of the Hessian matrix  $H$ . The entries of  $L_1$  are derived in (3.11) and (3.7), and the entries of  $K$  in (3.15a) are derived by (3.9a), (3.9b), and (3.11). We now give the specific derivation for entries of  $L_2$  in (3.15a). First, we give the gradients of

$$c_{ij}(f) = -\frac{\langle \mathbf{f}_{ki}, \mathbf{f}_{jk} \rangle}{2|f(T_{ijk})|}, \quad c_{jk}(f) = -\frac{\langle \mathbf{f}_{ij}, \mathbf{f}_{ki} \rangle}{2|f(T_{ijk})|}, \quad c_{ki}(f) = -\frac{\langle \mathbf{f}_{jk}, \mathbf{f}_{ij} \rangle}{2|f(T_{ijk})|}$$

in the triangle  $[\mathbf{f}_i, \mathbf{f}_j, \mathbf{f}_k]$ , where  $c_{ij}(f) := \cot \alpha_{ij}(f)$  as before. Since  $c_{ij}(f)$ ,  $c_{jk}(f)$ , and  $c_{ki}(f)$  only relates to vertices  $\mathbf{f}_i$ ,  $\mathbf{f}_j$ , and  $\mathbf{f}_k$ , their gradients with respect to other vertices are 0. The gradients with respect to  $\mathbf{f}_j$  are calculated by

$$\frac{\partial}{\partial \mathbf{f}_j} c_{ij}(f) = \frac{1}{|f(T_{ijk})|} \left[ -\frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_j} c_{ij}(f) - \frac{1}{2} \mathbf{f}_{ki} \right], \quad (3.21a)$$

$$\frac{\partial}{\partial \mathbf{f}_j} c_{jk}(f) = \frac{1}{|f(T_{ijk})|} \left[ -\frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_j} c_{jk}(f) + \frac{1}{2} \mathbf{f}_{ki} \right], \quad (3.21b)$$

$$\frac{\partial}{\partial \mathbf{f}_j} c_{ki}(f) = \frac{1}{|f(T_{ijk})|} \left[ -\frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_j} c_{ki}(f) + \frac{1}{2} (\mathbf{f}_{jk} - \mathbf{f}_{ij}) \right] \quad (3.21c)$$

with

$$\frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_i} = \frac{1}{2} (c_{ij}(f) \mathbf{f}_{ij} - c_{ki}(f) \mathbf{f}_{ki}), \quad (3.22a)$$

$$\frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_j} = \frac{1}{2} (c_{jk}(f) \mathbf{f}_{jk} - c_{ij}(f) \mathbf{f}_{ij}), \quad (3.22b)$$

$$\frac{\partial |f(T_{ijk})|}{\partial \mathbf{f}_k} = \frac{1}{2} (c_{ki}(f) \mathbf{f}_{ki} - c_{jk}(f) \mathbf{f}_{jk}). \quad (3.22c)$$

The others can be obtained by rotating the subscript  $i$ ,  $j$ , and  $k$  in turn.

Now, we derive the entries of matrix  $L_2$  in (3.15a). Let us consider the block  $\sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell)$  for  $\mathbf{a}, \mathbf{b} = \mathbf{x}, \mathbf{y}, \mathbf{z}$ , the  $(i, j)$ -th entry of which is

$$\begin{aligned} \left[ \sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell) \right]_{ij} &= \sum_{\ell=1}^n \mathbf{a}_\ell \frac{\partial}{\partial \mathbf{b}_j} [L(f)]_{i\ell} = \mathbf{a}_i \frac{\partial}{\partial \mathbf{b}_j} [L(f)]_{ii} + \sum_{\ell \in \mathcal{N}(i)} \mathbf{a}_\ell \frac{\partial}{\partial \mathbf{b}_j} [L(f)]_{i\ell} \\ &= \mathbf{a}_i \frac{\partial}{\partial \mathbf{b}_j} \sum_{\ell \in \mathcal{N}(i)} w_{i\ell}(f) - \sum_{\ell \in \mathcal{N}(i)} \mathbf{a}_\ell \frac{\partial}{\partial \mathbf{b}_j} w_{i\ell}(f) \end{aligned}$$

$$= \sum_{\ell \in \mathcal{N}(i)} \mathbf{a}_{i\ell} \frac{\partial}{\partial \mathbf{b}_j} w_{i\ell}(f).$$

(i) For  $j \neq i$ , there are only three vertices  $v_j, v_k$ , and  $v_{k'}$  related to  $v_j$  among all the adjacent vertices of  $v_i$ , as shown in Figure 1. Therefore,

$$\begin{aligned} & \left[ \sum_{\ell=1}^n \mathbf{a}_{\ell} \nabla_{\mathbf{b}}(L(f) \mathbf{e}_{\ell}) \right]_{ij} \\ &= \sum_{\ell \in \{j, k, k'\}} \mathbf{a}_{i\ell} \frac{\partial}{\partial \mathbf{b}_j} w_{i\ell}(f) \\ &= \mathbf{a}_{ik} \frac{\partial}{\partial \mathbf{b}_j} w_{ik}(f) + \mathbf{a}_{ik'} \frac{\partial}{\partial \mathbf{b}_j} w_{ik'}(f) + \mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_j} w_{ij}(f) \\ &= \frac{1}{2} \left[ \left( \mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_j} c_{ij}(f) - \mathbf{a}_{ki} \frac{\partial}{\partial \mathbf{b}_j} c_{ki}(f) \right) + \left( \mathbf{a}_{ik'} \frac{\partial}{\partial \mathbf{b}_j} c_{ik'}(f) - \mathbf{a}_{ji} \frac{\partial}{\partial \mathbf{b}_j} c_{ji}(f) \right) \right]. \end{aligned} \quad (3.23)$$

One can see that the terms in the first bracket relate only to triangle  $T_{ijk}$ , while the terms in the second bracket relate to triangle  $T_{k'ji}$ , which are of the same form. This characteristic is also similar to  $L$ , whose cotangent weight is  $w_{ij} = \frac{1}{2}(c_{ij} + c_{ji})$  with  $c_{ij}$  and  $c_{ji}$  also relating to  $T_{ijk}$  and  $T_{k'ji}$ , respectively. Hence, we only give the specific representation of the first bracket of (3.23). The second bracket is obtained similarly. By (3.21a)–(3.21c), we have

$$\mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_j} c_{ij}(f) + \mathbf{a}_{ik} \frac{\partial}{\partial \mathbf{b}_j} c_{ki}(f) = -\frac{1}{2|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{a}_i} \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{b}_j} + (2\mathbf{a}_{ki} \mathbf{b}_{jk} - \mathbf{a}_{jk} \mathbf{b}_{ki}) \right]. \quad (3.24)$$

(ii) For  $j = i$ , the related vertices are the whole adjacent vertices of  $v_i$  and itself, i.e.,  $\mathcal{N}(i) \cup \{i\}$ . Hence,

$$\left[ \sum_{\ell=1}^n \mathbf{b}_{\ell} \nabla_{\mathbf{a}}(L(f) \mathbf{e}_{\ell}) \right]_{ii} = \sum_{\ell \in \mathcal{N}(i)} \mathbf{a}_{i\ell} \frac{\partial}{\partial \mathbf{b}_i} w_{i\ell}(f) = \frac{1}{2} \sum_{\{j, k\} \in S_{\mathcal{E}}(i)} \left( \mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_i} c_{ij}(f) - \mathbf{a}_{ki} \frac{\partial}{\partial \mathbf{b}_i} c_{ki}(f) \right),$$

where  $S_{\mathcal{E}}(i) = \{\{j, k\} \mid j, k \in \mathcal{N}(i), [v_j, v_k] \in \mathcal{E}(M)\}$ . Similarly, via the gradient formulas (3.21a)–(3.21c), we have

$$\mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_i} c_{ij}(f) - \mathbf{a}_{ki} \frac{\partial}{\partial \mathbf{b}_i} c_{ki}(f) = -\frac{1}{2|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{a}_i} \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{b}_i} + \mathbf{a}_{jk} \mathbf{b}_{jk} \right]. \quad (3.25)$$

We now have derived the entries of the matrix  $L_1 - L_2$ , which is the Hessian matrix of the conformal energy with respect to Cartesian coordinates  $[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ . The following theorem shows that the nullity of  $L_1 - L_2$  is 3.

**Theorem 3.5.**  $L_1 - L_2$  is a symmetric Laplacian matrix with its null space having orthogonal basis  $[\mathbf{1}^T, \mathbf{0}^T, \mathbf{0}^T]^T$ ,  $[\mathbf{0}^T, \mathbf{1}^T, \mathbf{0}^T]^T$ , and  $[\mathbf{0}^T, \mathbf{0}^T, \mathbf{1}^T]^T$ .

*Proof.* Since  $L_1 - L_2$  is the Hessian matrix of the conformal energy with respect to Cartesian coordinates  $[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ , it is obviously symmetric. By the gradient formulas (3.21a)–(3.21c), it is easily seen that

$$\left( \frac{\partial}{\partial \mathbf{f}_i} + \frac{\partial}{\partial \mathbf{f}_j} + \frac{\partial}{\partial \mathbf{f}_k} \right) c_{ij}(f) = 0.$$

Therefore, by using the representation according to the adjacent vertices, we have

$$\begin{aligned} \sum_{j=1}^n \left[ \sum_{\ell=1}^n \mathbf{a}_{\ell} \nabla_{\mathbf{b}}(L(f) \mathbf{e}_{\ell}) \right]_{ij} &= \frac{1}{2} \sum_{j \in \mathcal{N}(i)} \mathbf{a}_{ij} \left( \frac{\partial}{\partial \mathbf{b}_i} + \frac{\partial}{\partial \mathbf{b}_j} + \frac{\partial}{\partial \mathbf{b}_k} \right) c_{ij}(f) \\ &\quad - \frac{1}{2} \sum_{j \in \mathcal{N}(i)} \mathbf{a}_{ji} \left( \frac{\partial}{\partial \mathbf{b}_i} + \frac{\partial}{\partial \mathbf{b}_j} + \frac{\partial}{\partial \mathbf{b}_{k'}} \right) c_{ji}(f) = 0, \quad i = 1, 2, \dots, n. \end{aligned}$$

Additionally, by the equality  $\mathbf{1}^T(\nabla_{\mathbf{b}} L(f)\mathbf{e}_\ell) = \nabla_{\mathbf{b}}(\mathbf{1}^T L(f)\mathbf{e}_\ell) = 0$ , we also have

$$\sum_{i=1}^n \left[ \sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell) \right]_{ij} = \left[ \sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(\mathbf{1}^T L(f)\mathbf{e}_\ell) \right] \mathbf{e}_j = 0, \quad j = 1, 2, \dots, n.$$

We conclude that the sum of each row and column of  $\sum_{\ell=1}^n \mathbf{a}_\ell \nabla_{\mathbf{b}}(L(f)\mathbf{e}_\ell)$  is zero, which guarantees the 3 orthogonal basis of the null space.  $\square$

Finally, we focus on

$$H_2 := [\nabla \mathbf{x}^T, \nabla \mathbf{y}^T, \nabla \mathbf{z}^T] L_2 \begin{bmatrix} \nabla \mathbf{x} \\ \nabla \mathbf{y} \\ \nabla \mathbf{z} \end{bmatrix} = \begin{bmatrix} H_{2,\theta\theta} & H_{2,\theta\phi} \\ H_{2,\phi\theta} & H_{2,\phi\phi} \end{bmatrix}$$

defined in (3.15a) and (3.15b). The formulas (3.24) and (3.25) show that the entries of  $L_2$  can be rewritten as the inner products of two terms related to  $\mathbf{a}$  and  $\mathbf{b}$ , respectively. Taking (3.25) as an example, we have

$$\mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_i} c_{ij}(f) + \mathbf{a}_{ik} \frac{\partial}{\partial \mathbf{b}_i} c_{ki}(f) = -\frac{1}{2|f(T_{ijk})|} \left\langle \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{a}_i}, \mathbf{a}_{jk} \right], \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{b}_i}, \mathbf{b}_{jk} \right] \right\rangle.$$

From gradient formulas (3.22a)–(3.22c), we can find that the first and second terms are only associated with  $\mathbf{a}$  and  $\mathbf{b}$ , respectively. Additionally, (3.24) can also be written as a similar inner product form. Then, since  $\mathbf{f}_i$  depends only on  $(\theta_i, \phi_i)$  for  $i = 1, 2, \dots, n$ , the Jacobian matrices of  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  with respect to  $\theta$  and  $\phi$  are diagonal. It is easy to express the products of Jacobian matrices and  $L_2$ . Without loss of generality, we discuss only entry  $[H_{2,\theta\phi}]_{ii}$ , i.e.,

$$\begin{aligned} [H_{2,\theta\phi}]_{ii} &= \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \sum_{\mathbf{a}, \mathbf{b}=\mathbf{x}, \mathbf{y}, \mathbf{z}} \left( \mathbf{a}_{ij} \frac{\partial}{\partial \mathbf{b}_i} c_{ij}(f) - \mathbf{a}_{ki} \frac{\partial}{\partial \mathbf{b}_i} c_{ki}(f) \right) \frac{\partial \mathbf{a}_i}{\partial \theta_i} \frac{\partial \mathbf{b}_i}{\partial \phi_i} \\ &= - \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{2|f(T_{ijk})|} \sum_{\mathbf{a}, \mathbf{b}=\mathbf{x}, \mathbf{y}, \mathbf{z}} \left\langle \frac{\partial \mathbf{a}_i}{\partial \theta_i} \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{a}_i}, \mathbf{a}_{jk} \right], \frac{\partial \mathbf{b}_i}{\partial \phi_i} \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{b}_i}, \mathbf{b}_{jk} \right] \right\rangle \\ &= - \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{2|f(T_{ijk})|} \left\langle \sum_{\mathbf{a}=\mathbf{x}, \mathbf{y}, \mathbf{z}} \frac{\partial \mathbf{a}_i}{\partial \theta_i} \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{a}_i}, \mathbf{a}_{jk} \right], \sum_{\mathbf{b}=\mathbf{x}, \mathbf{y}, \mathbf{z}} \frac{\partial \mathbf{b}_i}{\partial \phi_i} \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{b}_i}, \mathbf{b}_{jk} \right] \right\rangle \\ &= - \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{2|f(T_{ijk})|} \left\langle \frac{\partial \mathbf{f}_i}{\partial \theta_i}^T \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{f}_i}, \mathbf{f}_{jk} \right], \frac{\partial \mathbf{f}_i}{\partial \phi_i}^T \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{f}_i}, \mathbf{f}_{jk} \right] \right\rangle. \end{aligned}$$

The partial differential terms in (3.22a)–(3.22c) show that they are the linear combination of  $\mathbf{f}$ . Hence, only the sum with respect to  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  is necessary, such as

$$\begin{bmatrix} \mathbf{a}_{jk}^i \\ \mathbf{b}_{jk}^i \end{bmatrix} := \left( \frac{\partial \mathbf{f}_i}{\partial (\theta_i, \phi_i)} \right)^T \mathbf{f}_{jk} = \begin{bmatrix} -\mathbf{y}_i \mathbf{x}_{jk} + \mathbf{x}_i \mathbf{y}_{jk} \\ \mathbf{u}_i \mathbf{x}_{jk} + \mathbf{v}_i \mathbf{y}_{jk} - \mathbf{w}_i \mathbf{z}_{jk} \end{bmatrix}, \quad (3.26a)$$

$$\begin{bmatrix} \mathbf{a}_{ki}^i \\ \mathbf{b}_{ki}^i \end{bmatrix} := \left( \frac{\partial \mathbf{f}_i}{\partial (\theta_i, \phi_i)} \right)^T \mathbf{f}_{ki} = \begin{bmatrix} -\mathbf{y}_i \mathbf{x}_{ki} + \mathbf{x}_i \mathbf{y}_{ki} \\ \mathbf{u}_i \mathbf{x}_{ki} + \mathbf{v}_i \mathbf{y}_{ki} - \mathbf{w}_i \mathbf{z}_{ki} \end{bmatrix}, \quad (3.26b)$$

$$\begin{bmatrix} \mathbf{a}_{ij}^i \\ \mathbf{b}_{ij}^i \end{bmatrix} := \left( \frac{\partial \mathbf{f}_i}{\partial (\theta_i, \phi_i)} \right)^T \mathbf{f}_{ij} = \begin{bmatrix} -\mathbf{y}_i \mathbf{x}_{ij} + \mathbf{x}_i \mathbf{y}_{ij} \\ \mathbf{u}_i \mathbf{x}_{ij} + \mathbf{v}_i \mathbf{y}_{ij} - \mathbf{w}_i \mathbf{z}_{ij} \end{bmatrix}. \quad (3.26c)$$

The others are defined similarly by modifying the superscripts and subscripts. Here,  $\mathbf{a}_{..}^i$  denotes the  $\partial \theta_i$  term, and  $\mathbf{b}_{..}^i$  denotes the  $\partial \phi_i$  term. Thus, we have

$$\frac{\partial \mathbf{f}_i}{\partial \theta_i}^T \left[ \frac{2\partial|f(T_{ijk})|}{\partial \mathbf{f}_i}, \mathbf{f}_{jk} \right] = \frac{\partial \mathbf{f}_i}{\partial \theta_i}^T [c_{ij}(f)\mathbf{f}_{ij} - c_{ki}(f)\mathbf{f}_{ki}, \mathbf{f}_{jk}] = [c_{ij}(f)\mathbf{a}_{ij}^i - c_{ki}(f)\mathbf{a}_{ki}^i, \mathbf{a}_{jk}^i].$$

We can see that they differ only in  $\mathbf{f}_{..}$ ,  $\mathbf{a}_{..}^i$ , and  $\mathbf{b}_{..}^i$ . Hence, the entries of  $H_2$  differ from only those of  $L_2$  with the notations  $\mathbf{x}_{..}$ ,  $\mathbf{y}_{..}$ , and  $\mathbf{z}_{..}$  in (3.24) and (3.25) replaced by  $\mathbf{a}_{..}^i$  and  $\mathbf{b}_{..}^i$ . Taking  $H_{2,\theta\phi}$  as an example, we see that the diagonal and nondiagonal entries are

$$\begin{aligned}[H_{2,\theta\phi}]_{ii} &= -\sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\theta_i} \frac{2\partial|f(T_{ijk})|}{\partial\phi_i} + \mathbf{a}_{jk}^i \mathbf{b}_{jk}^i \right], \\ [H_{2,\theta\phi}]_{ij} &= -\frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\theta_i} \frac{2\partial|f(T_{ijk})|}{\partial\phi_j} + (2\mathbf{a}_{ki}^i \mathbf{b}_{jk}^j - \mathbf{a}_{jk}^i \mathbf{b}_{ki}^j) \right] \\ &\quad - \frac{1}{4|f(T_{k'ji})|} \left[ \frac{2\partial|f(T_{k'ji})|}{\partial\theta_i} \frac{2\partial|f(T_{k'ji})|}{\partial\phi_j} + (2\mathbf{a}_{ik'}^i \mathbf{b}_{k'j}^j - \mathbf{a}_{k'j}^i \mathbf{b}_{ik'}^j) \right],\end{aligned}$$

respectively, where

$$\frac{\partial|f(T_{ijk})|}{\partial\theta_i} = \frac{1}{2}(c_{ij}(f)\mathbf{a}_{ij}^i - c_{ki}(f)\mathbf{a}_{ki}^i), \quad (3.27a)$$

$$\frac{\partial|f(T_{ijk})|}{\partial\phi_i} = \frac{1}{2}(c_{ij}(f)\mathbf{b}_{ij}^i - c_{ki}(f)\mathbf{b}_{ki}^i). \quad (3.27b)$$

The other entries are obtained similarly by replacing  $\partial\mathbf{x}_i$ ,  $\partial\mathbf{y}_i$ ,  $\partial\mathbf{z}_i$  and  $\mathbf{x}_{..}$ ,  $\mathbf{y}_{..}$ ,  $\mathbf{z}_{..}$  by  $\partial\theta_i$ ,  $\partial\phi_i$  and  $\mathbf{a}_{..}^i$ ,  $\mathbf{b}_{..}^i$  as in (3.26a)–(3.26c), respectively. Now, we have obtained the entries of the Hessian matrix  $H$  in (3.15a).

**Theorem 3.6.** Let  $H = [H_{\theta\theta} \ H_{\theta\phi} \ H_{\phi\theta} \ H_{\phi\phi}]$  with  $H_{\theta\phi} = H_{\phi\theta}^T$ . The diagonal and nondiagonal entries of the blocks are

$$\begin{cases} [H_{\theta\theta}]_{ii} = \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{4|f(T_{ijk})|} \left[ \left( \frac{2\partial|f(T_{ijk})|}{\partial\theta_i} \right)^2 + (\mathbf{a}_{jk}^i)^2 \right] + \sum_{j \in \mathcal{N}(i)} \tilde{w}_{ij}(\mathbf{x}_i \mathbf{x}_j + \mathbf{y}_i \mathbf{y}_j), \\ [H_{\theta\theta}]_{ij} = \frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\theta_i} \frac{2\partial|f(T_{ijk})|}{\partial\theta_j} + (2\mathbf{a}_{ki}^i \mathbf{a}_{jk}^j - \mathbf{a}_{jk}^i \mathbf{a}_{ki}^j) \right] \\ \quad + \frac{1}{4|f(T_{k'ji})|} \left[ \frac{2\partial|f(T_{k'ji})|}{\partial\theta_i} \frac{2\partial|f(T_{k'ji})|}{\partial\theta_j} + (2\mathbf{a}_{ik'}^i \mathbf{a}_{k'j}^j - \mathbf{a}_{k'j}^i \mathbf{a}_{ik'}^j) \right] - \tilde{w}_{ij}(\mathbf{x}_i \mathbf{x}_j + \mathbf{y}_i \mathbf{y}_j), \\ [H_{\phi\phi}]_{ii} = \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{4|f(T_{ijk})|} \left[ \left( \frac{2\partial|f(T_{ijk})|}{\partial\phi_i} \right)^2 + (\mathbf{b}_{jk}^i)^2 \right] + \sum_{j \in \mathcal{N}(i)} \tilde{w}_{ij}(\mathbf{x}_i \mathbf{x}_j + \mathbf{y}_i \mathbf{y}_j + \mathbf{z}_i \mathbf{z}_j), \\ [H_{\phi\phi}]_{ij} = \frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\phi_i} \frac{2\partial|f(T_{ijk})|}{\partial\phi_j} + (2\mathbf{b}_{ki}^i \mathbf{b}_{jk}^j - \mathbf{b}_{jk}^i \mathbf{b}_{ki}^j) \right] \\ \quad + \frac{1}{4|f(T_{k'ji})|} \left[ \frac{2\partial|f(T_{k'ji})|}{\partial\phi_i} \frac{2\partial|f(T_{k'ji})|}{\partial\phi_j} + (2\mathbf{b}_{ik'}^i \mathbf{b}_{k'j}^j - \mathbf{b}_{k'j}^i \mathbf{b}_{ik'}^j) \right] - \tilde{w}_{ij}(\mathbf{u}_i \mathbf{u}_j + \mathbf{v}_i \mathbf{v}_j + \mathbf{w}_i \mathbf{w}_j), \end{cases}$$

and

$$\begin{cases} [H_{\phi\theta}]_{ii} = \sum_{\{j,k\} \in S_{\mathcal{E}}(i)} \frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\phi_i} \frac{2\partial|f(T_{ijk})|}{\partial\theta_i} + \mathbf{b}_{jk}^i \mathbf{a}_{jk}^i \right] + \sum_{j \in \mathcal{N}(i)} \tilde{w}_{ij}(\mathbf{v}_i \mathbf{x}_j - \mathbf{u}_i \mathbf{y}_j), \\ [H_{\phi\theta}]_{ij} = \frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\phi_i} \frac{2\partial|f(T_{ijk})|}{\partial\theta_j} + (2\mathbf{b}_{ki}^i \mathbf{a}_{jk}^j - \mathbf{b}_{jk}^i \mathbf{a}_{ki}^j) \right] \\ \quad + \frac{1}{4|f(T_{k'ji})|} \left[ \frac{2\partial|f(T_{k'ji})|}{\partial\phi_i} \frac{2\partial|f(T_{k'ji})|}{\partial\theta_j} + (2\mathbf{b}_{ik'}^i \mathbf{a}_{k'j}^j - \mathbf{b}_{k'j}^i \mathbf{a}_{ik'}^j) \right] \\ \quad - \tilde{w}_{ij}(\mathbf{v}_i \mathbf{x}_j - \mathbf{u}_i \mathbf{y}_j), \end{cases} \quad (3.28)$$

respectively.

Finally, we present a theorem for the proposed algorithm in Section 4, which demonstrates singularity and the corresponding eigenpair of  $H$ .

**Theorem 3.7.** The null space of  $H$  defined in (3.15a) has a basis  $[\mathbf{1}^T, \mathbf{0}^T]^T$ , where  $\dim(\mathbf{1}) = \dim(\mathbf{0}) = n$ .

*Proof.* To prove the assertion, we need to indicate only that

$$\mathbf{e}_i^T H_{\theta\theta} \mathbf{1} = 0 \quad \text{and} \quad \mathbf{e}_i^T H_{\phi\theta} \mathbf{1} = 0, \quad \text{for } i = 1, 2, \dots, n.$$

We first consider  $H_{\phi\theta}$ . By (3.28), we have

$$\begin{aligned} \mathbf{e}_i^T H_{\phi\theta} \mathbf{1} &= \sum_{\{j,k\} \in S_{\varepsilon}(i)} \frac{1}{4|f(T_{ijk})|} \left[ \frac{2\partial|f(T_{ijk})|}{\partial\phi_i} \cdot 2 \left( \frac{\partial}{\partial\theta_i} + \frac{\partial}{\partial\theta_j} + \frac{\partial}{\partial\theta_k} \right) |f(T_{ijk})| \right. \\ &\quad \left. + \mathbf{b}_{jk}^i \mathbf{a}_{jk}^i + 2\mathbf{b}_{ki}^i \mathbf{a}_{jk}^j - 2\mathbf{b}_{ij}^i \mathbf{a}_{jk}^k - \mathbf{b}_{jk}^i \mathbf{a}_{ij}^k \mathbf{b}_{jk}^i \mathbf{a}_{ij}^k \right]. \end{aligned} \quad (3.29)$$

It is easy to verify that

$$\mathbf{a}_{ij}^i = \mathbf{a}_{ij}^j, \quad \mathbf{a}_{jk}^j = \mathbf{a}_{jk}^k, \quad \mathbf{a}_{ki}^k = \mathbf{a}_{ki}^i. \quad (3.30)$$

Combining (3.30) with (3.27a), we have

$$\left( \frac{\partial}{\partial\theta_i} + \frac{\partial}{\partial\theta_j} + \frac{\partial}{\partial\theta_k} \right) |f(T_{ijk})| = 0.$$

Hence, by using (3.30) again, we see that (3.29) becomes

$$\begin{aligned} \mathbf{e}_i^T H_{\phi\theta} \mathbf{1} &= \sum_{\{j,k\} \in S_{\varepsilon}(i)} \frac{1}{4|f(T_{ijk})|} [\mathbf{b}_{jk}^i \mathbf{a}_{jk}^i + 2\mathbf{b}_{ki}^i \mathbf{a}_{jk}^j - 2\mathbf{b}_{ij}^i \mathbf{a}_{jk}^k - \mathbf{b}_{jk}^i \mathbf{a}_{ij}^k \mathbf{b}_{jk}^i \mathbf{a}_{ij}^k] \\ &= \sum_{\{j,k\} \in S_{\varepsilon}(i)} \frac{1}{4|f(T_{ijk})|} \mathbf{b}_{jk}^i (\mathbf{a}_{ij}^i + \mathbf{a}_{jk}^i + \mathbf{a}_{ki}^i) = 0. \end{aligned}$$

Similarly, we also have  $\mathbf{e}_i^T H_{\theta\theta} \mathbf{1} = 0$ . Notably,  $\mathbf{e}_i^T H_{\phi\phi} \mathbf{1} \neq 0$ . Therefore, the theorem is proved.  $\square$

**Remark 3.8.** Geometrically, the one-dimensional null space of  $H$  reveals that the conformal energy is invariant up to a rotation along the latitude, while that along the longitude is not characterized, since it is not linearly related to  $\phi$ .

## 4 The Hessian-based trust region algorithm

In this section, we develop an HBTR algorithm to minimize the conformal energy for the computation of the conformal map from a closed surface of genus-zero to a unit sphere. Here, we review the optimization problem,

$$\min E_C(\boldsymbol{\theta}, \boldsymbol{\phi}) := \frac{1}{2} \langle D(f) \mathbf{f}, \mathbf{f} \rangle, \quad \mathbf{f} = [\cos \boldsymbol{\theta} \odot \sin \boldsymbol{\phi}, \sin \boldsymbol{\theta} \odot \sin \boldsymbol{\phi}, \cos \boldsymbol{\phi}]. \quad (4.1)$$

We loosen the box constraint  $(\boldsymbol{\theta}, \boldsymbol{\phi}) \in [0, 2\pi]^n \times [0, \pi]^n$  and consider (4.1) as an unconstrained problem. In Section 3, the gradient vector and Hessian matrix of  $E_C(\boldsymbol{\theta}, \boldsymbol{\phi})$  are derived in (3.12a) and Theorem 3.6, respectively. The sparsity of the Hessian matrix shown in Theorem 3.4 guarantees the feasibility of fast computation associated with  $H$ . Specifically, we solve the large-scale sparse linear system to obtain the Newton direction:  $H\mathbf{s} = -\mathbf{g}$  as in (3.20). As Theorem 3.7 demonstrated,  $H$  is singular, and the general solution can be expressed as a spherical solution with arbitrary rotation along latitude according to Remark 3.8. Therefore, we fix the first entry of  $\mathbf{s}$  and let

$$H = \begin{bmatrix} h_{11} & \mathbf{h}_1^T \\ \mathbf{h}_1 & \hat{H} \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ \hat{\mathbf{s}} \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} g_1 \\ \hat{\mathbf{g}} \end{bmatrix}.$$

Then, we solve the linear system

$$\hat{H}\hat{\mathbf{s}} = -\hat{\mathbf{g}}. \quad (4.2)$$

It is easy to verify that  $H\mathbf{s} = -\mathbf{g}$  holds for  $\mathbf{s} = [0, \hat{\mathbf{s}}^T]^T$ . Geometrically, this approach means that the longitude of the first vertex always remains invariant and that meaningless rotation of the sphere is avoided during the iteration.

**Remark 4.1.**  $H$  can be rearranged from  $L \otimes \mathbf{1}_{2 \times 2}$  to  $\mathbf{1}_{2 \times 2} \otimes L$ . The rearranged matrix is constructed by the  $2 \times 2$  block in the  $L$  form such that the block lower-upper (LU) decomposition is appropriate for the fast computation of the linear system (4.2).

Therefore, it is natural to use the Newton-type method to solve the optimization problem (4.1). However, the pure Newton method is insufficient for solving the optimization problem directly because the Newton method is well known to have local quadratic convergence. Unfortunately, it is almost impractical to directly seek an initial guess sufficiently close to the ideal solution for (4.1), especially when facing surfaces with high-curvature regions or complicated shapes. Additionally, the conformal energy of (4.1) with respect to  $(\theta, \phi)$  is nonconvex and nonlinear. Hence, the quadratic convergence generally disappears at the beginning of iterations, and it might take much time to reach the neighborhood with quadratic convergence via the Newton step. Moreover, importantly, the Hessian matrix  $H$  is not uniformly positive semidefinite, i.e.,  $H$  may be indefinite at some points  $(\theta, \phi)$ . Indeed, the negative curvature occurs frequently during iteration in practical experiments. Consequently, it is possible that the Newton direction may not be a descent direction.

To overcome this drawback, we introduce the negative gradient direction, which is a descent direction. More specifically, we search the trial step  $\mathbf{d}$  from the 2D subspace spanned by the Newton direction  $\mathbf{s}$  and the gradient direction  $\mathbf{g}$ . If  $\mathbf{s}$  is not a descent direction, it is a negative curvature direction, which is still beneficial information for the choice of descent direction. Moreover, there must be a descent direction in this 2D subspace since  $-\mathbf{g} \in \text{span}(\mathbf{s}, \mathbf{g})$ . When the iterative point is in the convergence neighborhood, the Newton direction guarantees the quadratic convergence of the algorithm. We utilize the trust region method to search the trial step from  $\text{span}(\mathbf{s}, \mathbf{g})$ . In other words, during each iterative step, we consider the trust region subproblem proposed by Shultz et al. [35]

$$\begin{aligned} & \min \mathbf{d}^T H \mathbf{d} + \mathbf{g}^T \mathbf{d} \\ & \text{s.t. } \|\mathbf{d}\| \leq \Delta, \quad \mathbf{d} \in \text{span}(\mathbf{s}, \mathbf{g}), \end{aligned} \tag{4.3}$$

where  $\Delta$  is the trust region radius. The 2D optimization problem (4.3) is easy to solve and costs less time. The global and local convergence of the trust region method with the subproblem (4.3) has been proven by [35], and the practical experiment has been verified later in [3].

For the error measurement, it is worth noting that  $E_C(\theta, \phi)$  is a periodic function. Since we loosen the box constraint  $(\theta, \phi) \in [0, 2\pi]^n \times [0, \pi]^n$ ,  $(\theta, \phi)$  might move larger than  $2\pi$ , while the vertices on the sphere move much less. Hence, it is inappropriate to check the length of step  $\mathbf{d}$ . Inspired by [39], we adopt the optimal rotation and measure the error with

$$\delta^{(k)} = \min_{R \in SO(3)} \|\mathbf{f}^{(k+1)} - \mathbf{f}^{(k)} R\|^2, \tag{4.4}$$

where

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = I, \det(R) = 1\}$$

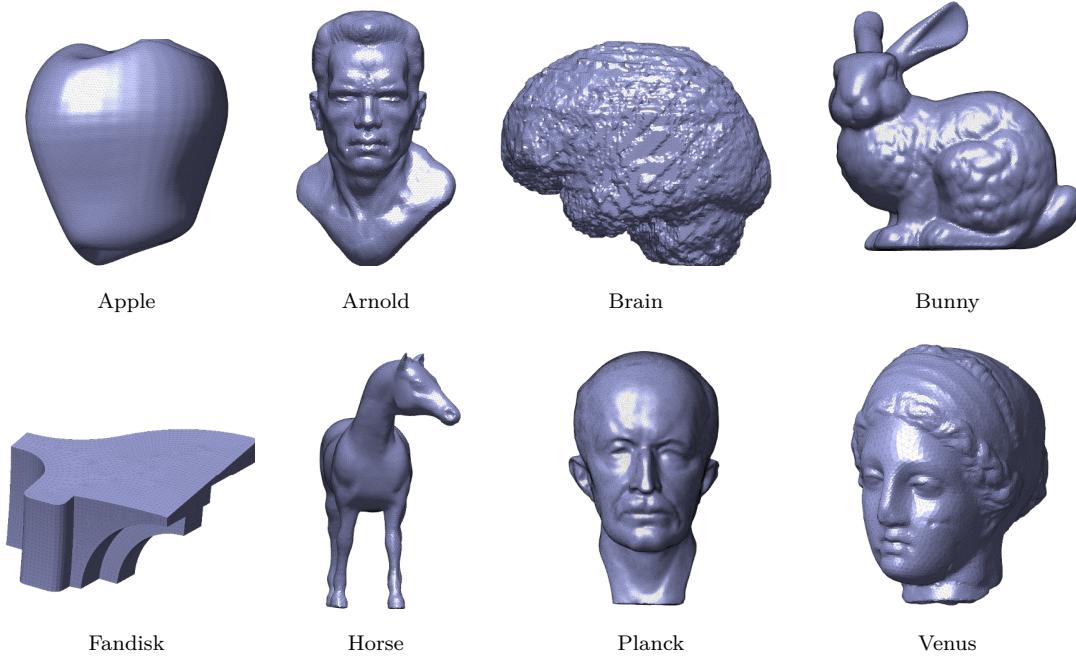
and  $\mathbf{f}^{(k)}$  are the vertices on the sphere at the  $k$ -th iteration. The error measurement eliminates the error from rotation. In addition, it avoids the miscalculation caused by the periodicity of  $E_C(\theta, \phi)$ .

We summarize the proposed HBTR algorithm for the optimization problem (4.1).

**Remark 4.2.** Similar to the disk parameterization in [27], the bijectivity of the resulting map can almost be guaranteed under the spherical coordinate representation. If not, the folding triangles can be removed by the mean value coordinate [14].

## 5 Numerical experiments

In this section, we describe the numerical performance of our proposed HBTR method for spherical conformal parameterization on several triangulation models. All the experimental programs are executed in MATLAB R2021a on a personal computer with a 2.50 GHz CPU and 64 GB RAM. Most of the triangulation models are taken from AIM@SHAPE shape repository (<http://visionair.ge.imati.cnr.it/ontologies/shapes/>), ALICE (<https://www.loria.fr/en/research/teams/alice/>), Gu's personal website (<https://www3.cs.stonybrook.edu/~gu/>), the Stanford 3D scanning repository (<http://graphics.stanford.edu/data/3Dscanrep>), Human Connectome Project (<http://www.humanconnectome.org>), and TurboSquid (<https://www.turbosquid.com/>). The triangulations of brain cortical surfaces are generated from BraTS datasets [2] via library JIGSAW [7–11] and toolbox Iso2Mesh (<http://iso2mesh.sourceforge.net>) [13, 36]. From the above-mentioned benchmarks, we take the triangulation models for experiments as shown in Figure 2 and present their basic information for numbers of vertices and faces in Table 1. Notably, no folding occurs on the 8 models by HBTR. We apply the SCEM algorithm proposed in [40] for the initial guess in the experiments. Among the vast experiments, the SCEM can stably provide a great initial guess in a very short time, which is appropriate for the HBTR algorithm.



**Figure 2** (Color online) The triangulation models for the experiments

**Table 1** The features of triangulation models with  $\#V$  and  $\#F$  being the number of vertices and triangle faces, respectively

Mesh	$\#V$	$\#F$	Mesh	$\#V$	$\#F$
Apple	17,839	35,674	Fandisk	6,475	12,946
Arnold	14,530	29,056	Horse	21,013	42,022
Brain	32,160	64,316	Planck	51,108	102,212
Bunny	55,684	111,364	Venus	14,303	28,602

### 5.1 Convergence behavior and conformal distortion

We first present the convergence behavior of the proposed Algorithm 1. Figure 3 shows the relationship between the number of iterations  $k$  and conformal energy  $E_C^{(k)}$ , the infinity norm of the gradient  $\|\mathbf{g}^{(k)}\|_\infty$  and the error  $\delta^{(k)}$  by Algorithm 1 for models in Figure 2. As shown, the conformal energy  $E_C^{(k)}$  with the scale on the right decreases linearly first. Meanwhile,  $\|\mathbf{g}^{(k)}\|_\infty$  and  $\delta^{(k)}$  with the scale on the left remain stable within a range. At this moment, the gradient and Newton directions are utilized for the trial step. Then,  $E_C^{(k)}$  tends to level off, while  $\|\mathbf{g}^{(k)}\|_\infty$  and  $\delta^{(k)}$  descend sharply. More specifically,  $\|\mathbf{g}^{(k)}\|_\infty$  becomes  $10^{-10}$  on most of the models when the iteration loops terminate, implying that the iteration stops at a critical point and the algorithm converges. Moreover,  $\delta^{(k)}$  descends in quadratic order. Taking *Brain* as an example, we have

$$\delta^{(25)} = 7.4 \times 10^{-4}, \quad \delta^{(26)} = 9.3 \times 10^{-6}, \quad \delta^{(27)} = 5.9 \times 10^{-10}.$$

Obviously, it shows the quadratic convergence of the HBTR algorithm.

---

**Algorithm 1** HBTR for the spherical CEM problem

---

**Require:** Triangulation  $M$  with vertices  $\{v_i, i = 1, 2, \dots, n\}$ , tolerance  $\varepsilon$ .

**Ensure:**  $\mathbf{f} \in \mathbb{R}^{n \times 3}$  inducing the conformal map  $f$  as in (3.1).

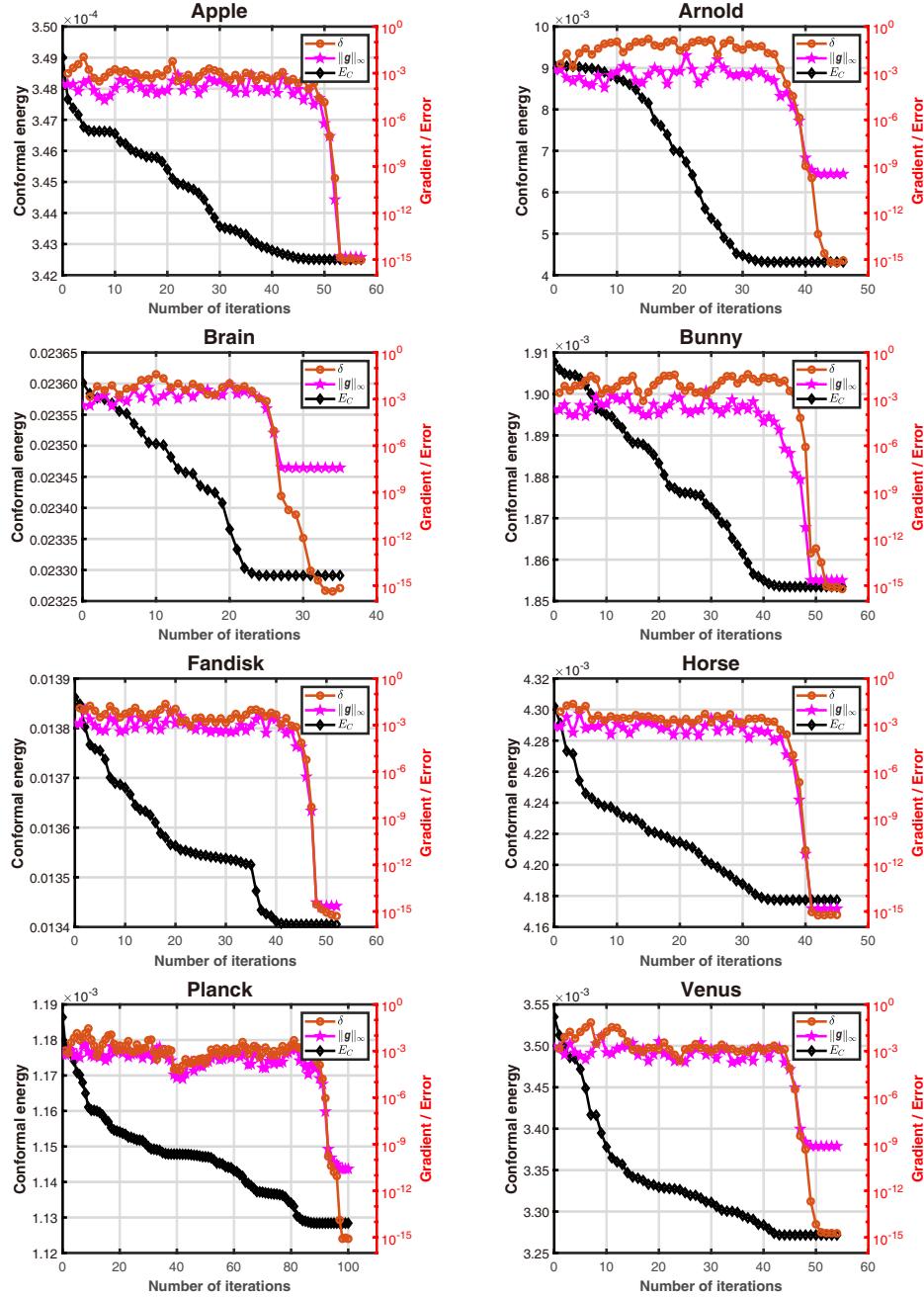
- 1: Set  $k = 0$  and  $\delta^{(0)} = +\infty$ .
  - 2: Compute the initial guess  $\mathbf{f}$  and the corresponding conformal energy  $E_C^{(0)}$  in (3.3).
  - 3: **while**  $\delta > \varepsilon$  **do**
  - 4:   Compute the gradient vector  $\mathbf{g}$  and the Hessian matrix  $H$  by (3.12a) and Theorem 3.6.
  - 5:   Solve the linear system  $\hat{H}\hat{\mathbf{s}} = -\hat{\mathbf{g}}$  via block LU decomposition to get the Newton direction  $\mathbf{s}$ .
  - 6:   Solve the trust region subproblem (4.3) to get the trial step  $\mathbf{d}$ .
  - 7:   Let  $E \leftarrow E_C((\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathbf{d})$ . If  $E_C^{(k)} > E$ , set  $k \leftarrow k + 1$  and update
$$(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathbf{d},$$

$$E_C^{(k)} \leftarrow E.$$
  - 8:   Compute the error  $\delta^{(k)}$  by (4.4) and tune the trust region radius  $\Delta$ .
  - 9: **end while**
- 

Figures 4 and 5 present the histograms of absolute angle distortion (degree) of each angle, denoted by  $|\alpha_{jk} - \alpha_{jk}(f)|$  for  $v_i$  in  $T_{ijk}$ , and Beltrami coefficients  $\mu$  [5] of triangular faces, respectively. If  $|\mu| = 0$ , the map  $f$  is conformal. The subplots in the upper right of each histogram are the front and back of the angle distortion distributions/Beltrami coefficient distributions on the resulting unit spheres, respectively. One can see that most of angle distortions are less than 5 degrees and most of Beltrami coefficients are less than 0.1 as well, guaranteeing the conformal performance of the HBTR algorithm. Angle distortion has similar performance to the Beltrami coefficient. Furthermore, Figure 6 shows the absolute value of the discrete Gauss curvature  $|\kappa|$ , the average angle distortion  $\epsilon_\alpha$ , and the average of the norm of the Beltrami coefficient  $|\overline{\mu}|$  at each vertex for the models *Arnold*, *Brain*, *Fandisk*, and *Horse*, denoted by

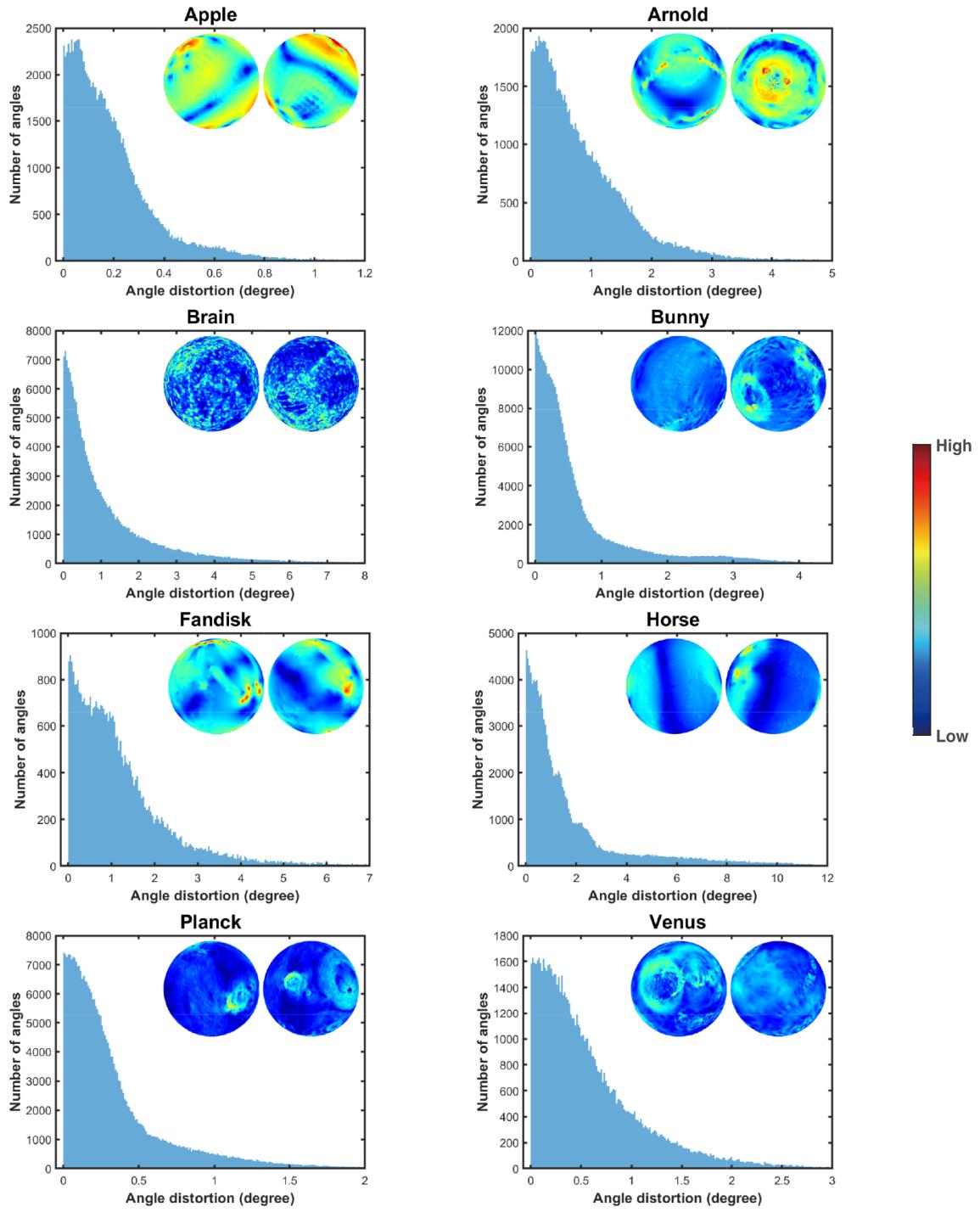
$$\begin{aligned} \kappa(v_i) &= 2\pi - \sum_{\{j,k\} \in S_\varepsilon(i)} \alpha_{jk}, \\ \epsilon_\alpha(v_i) &= \frac{1}{|\mathcal{N}(i)|} \sum_{\{j,k\} \in S_\varepsilon(i)} |\alpha_{jk} - \alpha_{jk}(f)|, \\ |\overline{\mu}|(v_i) &= \frac{1}{|\mathcal{N}(i)|} \sum_{T_{ijk} \ni i} |\mu(T_{ijk})|, \end{aligned}$$

where  $|\mathcal{N}(i)|$  is the number of adjacent vertices of  $v_i$  and  $\mu(T_{ijk})$  is the Beltrami coefficient on triangle  $T_{ijk}$ . We approximately find  $|\overline{\mu}| \propto \epsilon_\alpha \propto \sqrt{|\kappa|}$  in these 4 models except *Brain*, which approximately satisfies  $|\overline{\mu}| \propto \epsilon_\alpha \propto |\kappa|$ . The figures are plotted according to the relationships. The textures illustrate the high similarity of high curvatures, large angle distortions, and large Beltrami coefficient distributions.



**Figure 3** (Color online) The relationship between the number of iterations  $k$  and the conformal energy  $E_C^{(k)}$ , the infinity norm of gradient  $\|\mathbf{g}^{(k)}\|_\infty$  and the error  $\delta^{(k)}$  by Algorithm 1. The  $x$ -axes represent the number of iterations. The left  $y$ -axes represent the conformal energies, and the right  $y$ -axes represent infinity norms of gradient and errors, respectively

In other words, the large angle distortion and the large Beltrami coefficient regions are mainly at those with high curvatures, such as the ears, eyes, and nose of *Arnold*, corners of *Fandisk*, and ears and legs of *Horse*. The small angle distortion and small Beltrami coefficient regions are roughly at those with low curvatures. Therefore, the HBTR performs relatively poorly at vertices with high curvatures, which is an issue in our future work. Additionally, the angle distortion and the Beltrami coefficient have diffusion trends. Taking *Fandisk* as an example, we see that high-curvature regions are on corners and edges, and the curvatures in other regions are mostly 0. Furthermore, the angle distortions and the Beltrami coefficients diffuse from the corners to the adjacent regions gradually. These phenomena also occur in the other models, which are not shown.

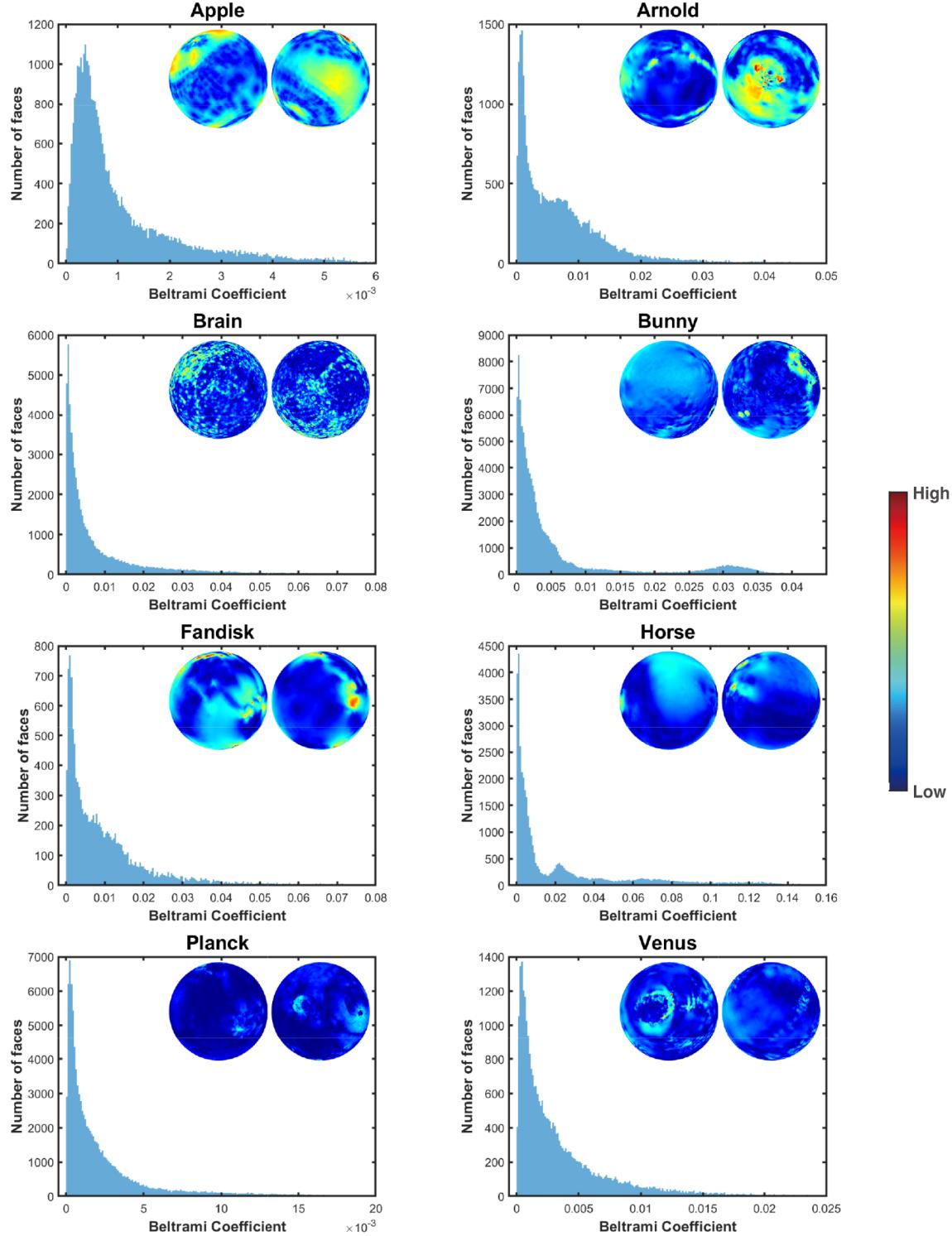


**Figure 4** (Color online) Histograms of angle distortions on triangulation models

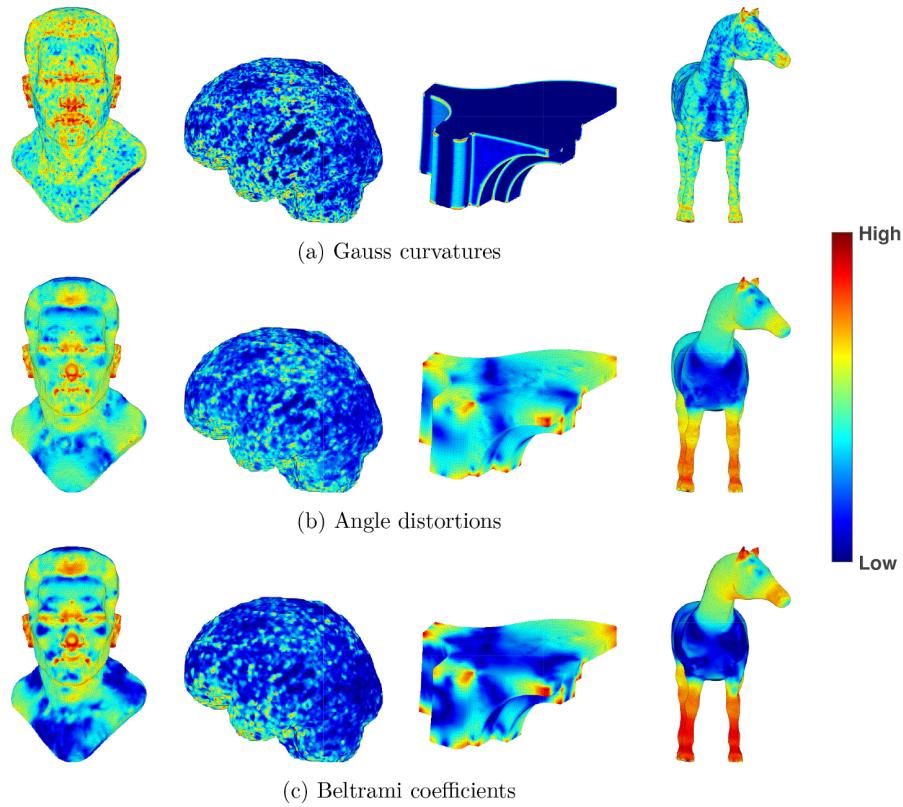
## 5.2 Comparison with state-of-the-art algorithms

In this subsection, we compare the accuracy of the proposed HBTR algorithm with that of two state-of-the-art algorithms for spherical conformal parameterization, namely, FLASH [5] and the SCEM [40]. The algorithm FLASH, simply put, via the stereographic projection, applies the composition of two quasi-conformal maps to construct the ideal conformal map, which is not an iterative algorithm. The MATLAB

program of FLASH is obtained from Choi's website (<https://www.math.cuhk.edu.hk/~ptchoi/>). The SCEM algorithm, as mentioned in Section 2, adopts the north-south hemisphere alternating iteration to compute the conformal map. The maximum number of iterations of the HBTR and SCEM is 500.



**Figure 5** (Color online) Histograms of the Beltrami coefficients in the triangulation models



**Figure 6** (Color online) (a) Gauss curvature distributions, (b) angle distortion distributions, and (c) Beltrami coefficient distributions for models *Arnold*, *Brain*, *Fandisk*, and *Horse*

The loop termination condition of SCEM is that the difference between the conformal energies of two consecutive iteration steps is less than  $10^{-9}$ . For the HBTR algorithm, we set  $\varepsilon = 10^{-9}$ , as in Figure 3, which guarantees convergence.

Table 2 shows the comparison of conformal energies between FLASH, SCEM, and HBTR. We can see that the conformal energies by HBTR are smallest among all the models. In the view of angle distortions, it is observed that the HBTR algorithm has well performance for 50-th percentile and 75-th percentile as FLASH and SCEM. In addition, among the 8 testing examples, FLASH, SCEM, and HBTR did not produce foldings for spherical conformal maps.

**Table 2** Comparison of conformal energies and angle distortions among FLASH, SCEM, and HBTR

Mesh	Conformal energy			Angle distortion					
				50-th percentile			75-th percentile		
	FLASH [5]	SCEM [40]	HBTR	FLASH [5]	SCEM [40]	HBTR	FLASH [5]	SCEM [40]	HBTR
Apple	3.74E-04	3.49E-04	3.43E-04	0.162	0.156	0.153	0.280	0.268	0.264
Arnold	7.82E-03	9.07E-03	4.32E-03	0.676	0.675	0.659	1.232	1.173	1.233
Brain	2.69E-02	2.36E-02	2.33E-02	0.821	0.729	0.729	1.849	1.694	1.694
Bunny	1.93E-03	1.90E-03	1.85E-03	0.343	0.339	0.363	0.759	0.746	0.735
Fandisk	3.50E-02	1.39E-02	1.34E-02	1.071	0.936	0.936	1.974	1.646	1.640
Horse	4.58E-03	4.30E-03	4.18E-03	1.027	1.020	1.010	2.292	2.283	2.283
Planck	1.71E-03	1.19E-03	1.13E-03	0.311	0.245	0.231	0.578	0.457	0.446
Venus	6.66E-03	3.54E-03	3.27E-03	0.666	0.453	0.442	1.186	0.831	0.821

### 5.3 Convergence behavior of the discrete scheme

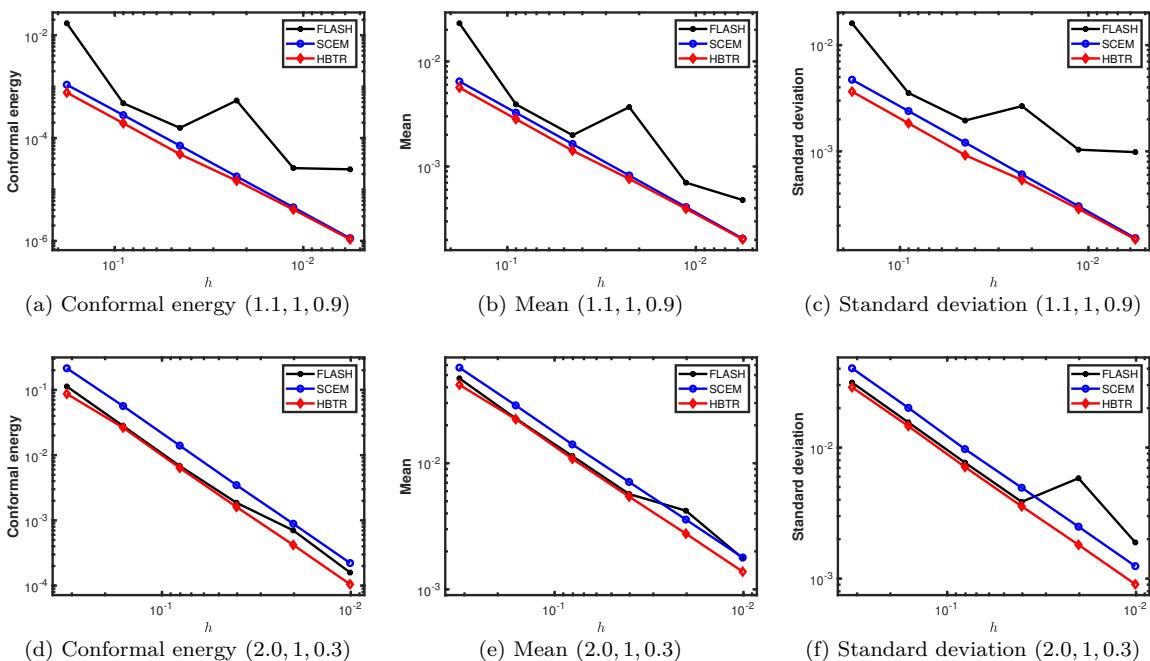
In this subsection, we check the numerical convergence of the discrete conformal energy of the resulting map to the continuous energy in (2.1). We consider two ellipsoids with semiaxis lengths of  $(1.1, 1, 0.9)$  and  $(2.0, 1, 0.3)$  and generate triangular meshes with different resolutions. The basic information of the meshes is in Table 3, where  $h$  represents the maximum diameter of triangles in the mesh. Then, we use FLASH, SCEM, and HBTR to compute conformal parameterizations. The conformal energies, means and SDs of angle distortion are used to measure the conformal distortion of the algorithms. Figure 7 shows the relationship between the measurements and  $h$ . The  $x$ -axis represents  $h$  and the  $y$ -axis represents the conformal energy, mean, and SD of angle distortion, respectively. We can see that the conformal energy, mean, and SD do not stably decrease as  $h$  decreases for FLASH, while  $E_C = O(h^2)$  and angle distortions are linearly related to  $h$  for SCEM and HBTR. Specifically, as  $h \rightarrow \frac{1}{2}h$  and  $E_C \rightarrow \frac{1}{4}E_C$ , the means and SDs are reduced by half. Therefore, SCEM and HBTR are robust in the respective of convergence of the discrete scheme. The conformal energies, means and SDs are lowest for HBTR compared with those of FLASH and SCEM, demonstrating the robustness and accuracy of HBTR.

### 5.4 Removal of folding triangles

HBTR does not necessarily guarantee the bijectivity of the resulting map, i.e., folding triangles may occur in the image region  $\mathbb{S}^2$ . In this subsection, we apply a postprocessing method, named mean value

**Table 3** The ellipsoids meshes for checking the convergence of the discrete scheme

	#V	642	2,562	10,242	40,962	163,842	655,362
	#F	1,280	5,120	20,480	81,920	327,680	1,310,720
$h$	$(1.1, 1, 0.9)$	0.1796	0.0901	0.0451	0.0226	0.0113	0.0056
	$(2.0, 1, 0.3)$	0.3192	0.1607	0.0806	0.0403	0.0202	0.0101



**Figure 7** (Color online) The relationship between conformal energy, mean, and standard deviation of angle distortion and  $h$  of algorithms FLASH, SCEM, and HBTR for 2 ellipsoids. The figures from the top row to the bottom row are for ellipsoids  $(1.1, 1, 0.9)$  and  $(2.0, 1, 0.3)$ , respectively

coordinates [14], to remove the folding triangles. Let  $L_{MV}$  be a Laplacian matrix defined as

$$[L_{MV}]_{ij} = \begin{cases} -w_{MV,ij}, & \text{if } i \neq j, [v_i, v_j] \in \mathcal{E}(M), \\ \sum_{k \in \mathcal{N}(i)} w_{MV,ik}, & \text{if } i = j, \\ 0, & \text{if } [v_i, v_j] \notin \mathcal{E}(M) \end{cases} \quad (5.1)$$

with

$$w_{MV,ij} = \frac{\tan(\alpha_{jk}/2) + \tan(\alpha_{k'j}/2)}{\|v_{ij}\|},$$

where  $\alpha_{jk}$  and  $\alpha_{k'j}$  are the angles opposite to vertex  $v_i$  in triangles  $T_{ijk}$  and  $T_{k'ji}$ , respectively, as shown in Figure 1(a). Unlike  $L$  in conformal energy in (3.3),  $L_{MV}$  is not symmetric. However, its weights  $w_{MV}$  must be positive. In the discrete case, the stereographic projection  $\Pi$  transforms  $\mathbb{S}^2$  to a domain in  $\mathbb{R}^2$ , and subsequently, mean value coordinates are applied to the projected domain to eliminate the folding triangles. By the way, the numerically conformal distortion of the stereographic projection  $\Pi$  is proportional to the distance between the vertex and the south pole. Selecting a point on a slender triangle as the north pole can easily cause additional folding triangles to occur. It is suggested to select a point in an unfolding triangle that is as regular as possible and positioned away from the folding triangles as the north pole of the stereographic projection. The postprocessing method is concluded in Algorithm 2.

---

**Algorithm 2** Mean value coordinates for removing folding triangles

---

**Require:** Triangulation  $M$  with vertices  $\{v_i, i = 1, 2, \dots, n\}$ , and  $\mathbf{f} \in \mathbb{R}^{n \times 3}$  obtained by Algorithm 1.  
**Ensure:**  $\mathbf{f} \in \mathbb{R}^{n \times 3}$  inducing the conformal map  $f$  as in (3.1) with postprocessing of bijectivity correction by the mean value coordinates [14].

- 1: Generate Laplacian matrix  $L_{MV}$  defined in (5.1).
- 2: Search the folding triangles in  $f(M)$ . Let  $\mathbb{I} = \{i \mid T_{ijk} \text{ is a folding triangle}\}$  be the index set of vertices contained in the folding triangles and  $\mathbb{0} = \{1, 2, \dots, n\} \setminus \mathbb{I}$ .
- 3: **while**  $\mathbb{I} \neq \emptyset$  **do**
- 4:   Select the face center of an unfolding triangle as regular as possible and positioned away from the folding triangles as the north pole. Perform the stereographic projection  $\Pi(\mathbf{f}) \rightarrow \mathbf{h}$ .
- 5:   Update the vertices  $\mathbf{h}_{\mathbb{I}}$  by solving the linear system,

$$[L_{MV}]_{\mathbb{I}\mathbb{I}} \mathbf{h}_{\mathbb{I}} = -[L_{MV}]_{\mathbb{I}\mathbb{0}} \mathbf{h}_{\mathbb{0}}. \quad (5.2)$$

- 6:   Perform the inverse stereographic projection  $\Pi^{-1}(\mathbf{h}) \rightarrow \mathbf{f}$  and update  $\mathbb{I}$  and  $\mathbb{0}$ .
- 7: **end while**

---

We present 2 examples, the resulting maps of which by HBTR are not bijective. Then, we use Algorithm 2 to remove the folding triangles. Table 4 shows the conformal energies, angle distortions, and the number of folding triangles before and after the removal. Although Algorithm 2 may generally lead to a slight conformality distortion, we can see that the conformal energies decrease slightly, and the angle distortions are almost unchanged, while the folding triangles disappear. This can be attributed to the fact that the folding triangles often possess a relatively large conformality distortion, and the total distortion can be finally balanced after removing the folding triangles.

**Table 4** The result of postprocessing for removing the folding triangles. The left and right of ‘/’ are the values before and after the postprocessing, respectively

Mesh	#V	#F	Conformal energy	Mean of angle distortion	#folding
Bimba	502,575	1,005,146	1.132E-3 / 1.113E-3	5.328E-3 / 5.328E-3	24 / 0
Right brain	163,842	327,680	2.806E-2 / 2.799E-2	2.252E-2 / 2.252E-2	34 / 0

## 6 Applications to surface registrations

Given a fixed surface  $M_0$  and a series of moving surfaces  $M_t, t = 1, 2, \dots$ , the surface registration aims to find bijective maps from the moving surfaces to the fixed surface. It is broadly applied in computer vision and medical imaging. The goal of the surface registration is to transform surfaces from different sources into one coordinate system. Therefore, the registration should ensure that the predominant features in the fixed surface correspond to those in the target surfaces, which are often expressed as landmarks in practical applications. It is generally not easy to manage the registration because of the complicated structure of surfaces. With the help of parameterization, we can transform the closed fixed surface  $M_0$  into a unit sphere  $\mathbb{S}^2$  via the conformal map  $f_0$  and then register the moving surfaces to the obtained unit sphere  $\mathbb{S}^2$  via registration map  $f_{\text{reg},t}$ . As a result, the map  $\tilde{f}_{\text{reg},t} = f_0^{-1} \circ f_{\text{reg},t}$  is the registration map from  $M_t$  to  $M_0$ . To obtain  $f_{\text{reg},t}$ , we consider the optimization problem

$$\min E_B(\boldsymbol{\theta}, \boldsymbol{\phi}) := E_C(\boldsymbol{\theta}, \boldsymbol{\phi}) + \lambda E_{\text{reg}}(\boldsymbol{\theta}, \boldsymbol{\phi}), \quad (6.1)$$

where  $E_{\text{reg}}$  is the registration loss and  $\lambda$  is its parameter. The representation of  $E_{\text{reg}}$  depends on the expression of predominant features. The most common representation is the landmark-based registration

$$E_{\text{reg}} = \frac{1}{2|S_L|} \sum_{i \in S_L} \|\mathbf{f}_{t,i} - \mathbf{f}_{0,i}\|_F^2, \quad (6.2)$$

where  $\mathbf{f}_{t,i}$  are the feature vertices on the surface  $M_t$ , and  $S_L$  and  $|S_L|$  are the indices set and the number of the landmark vertices, respectively. The landmark-based registration aims to align the landmark vertices such that the features of the surfaces are also aligned.

The conformal energy term guarantees the conformality of the map, while the registration loss term aligns the features of the surfaces. Therefore, the combination of the conformal energy and the registration loss results in a conformal (as possible) registration map. The conformal registration map is an elastic registration and preserves the local shape of the surface, which is widely used in the field of medical imaging. For the optimization problem (6.1), it is easy to derive the gradient vector and Hessian matrix of the registration loss generally. Benefiting from their simple representations, we can also utilize HBTR to solve it. Moreover, the conformal energy is invariant up to arbitrary rotation on  $\mathbb{S}^2$ . Therefore, we introduce an optimal rotation to further decrease the registration loss. Based on Algorithm 1, we present the following spherical conformal registration algorithm (see Algorithm 3).

---

**Algorithm 3** HBTR for the spherical conformal registration

---

**Require:** Fixed surface  $M_0$  and moving surface  $M_t$  with the landmarks, registration parameter  $\lambda$ , and tolerance  $\varepsilon$ .

**Ensure:** The conformal registration map  $\tilde{f}_{\text{reg},t}$ .

- 1: Compute the spherical conformal map of  $M_0$  by Algorithm 1, denoted by  $f_0$ .
- 2: Set  $k = 0$  and  $\delta^{(0)} = +\infty$ .
- 3: Compute the initial guess  $\mathbf{f}$  of  $M_t$ .
- 4: Compute the optimal rotation  $R$  according to the landmarks, update  $\mathbf{f} \leftarrow \mathbf{f}R$ , and compute the corresponding term  $E_B^{(0)}$  in (6.1).
- 5: **while**  $\delta > \varepsilon$  **do**
- 6:   Compute the gradient vector  $\mathbf{g}$  and the Hessian matrix  $H$  of (6.1).
- 7:   Solve the linear system  $H\mathbf{s} = -\mathbf{g}$  via block LU decomposition.
- 8:   Solve the trust region subproblem (4.3) to get the trial step  $\mathbf{d}$ .
- 9:   Compute the optimal rotation  $R$  according to the landmarks to update  $\mathbf{d}$ .
- 10:   Let  $E \leftarrow E_B((\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathbf{d})$ . If  $E_B^{(k)} > E$ , set  $k \leftarrow k + 1$  and update

$$(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathbf{d}, \\ E_B^{(k)} \leftarrow E.$$

- 11:   Compute the error  $\delta^{(k)}$  by (4.4) and tune the trust region radius  $\Delta$ .

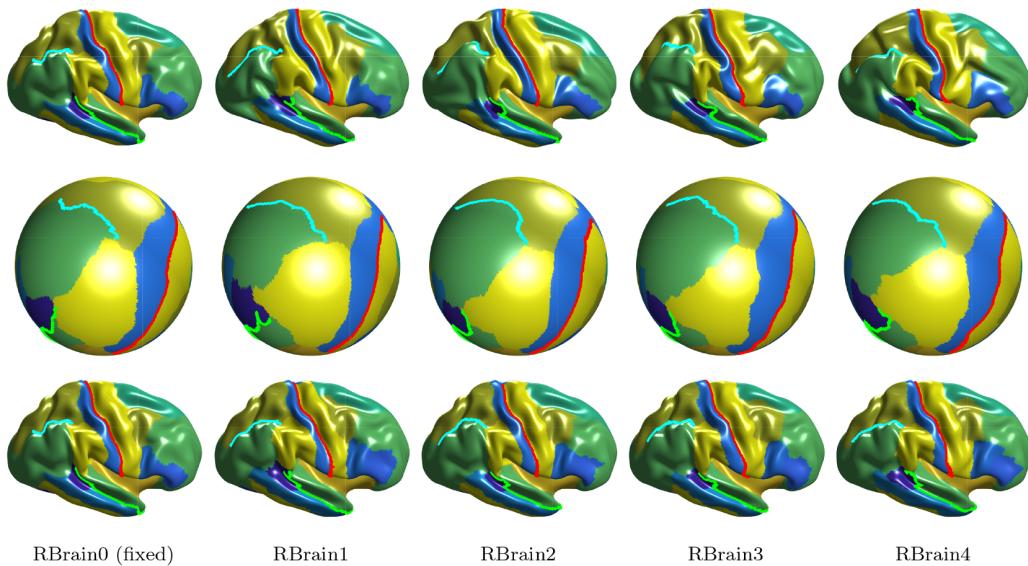
12: **end while**

- 13: Let  $\mathbf{f}_{\text{reg},t} = [\cos \boldsymbol{\theta} \odot \sin \boldsymbol{\phi}, \sin \boldsymbol{\theta} \odot \sin \boldsymbol{\phi}, \cos \boldsymbol{\phi}]$ . Compute  $\tilde{f}_{\text{reg},t} = f_0^{-1} \circ f_{\text{reg},t}$ , which is the registration map of  $M_t$ .
-

To present the registration performance of our method, we take 5 right brain cortex meshes  $R\text{Brain}0-R\text{Brain}4$  from the Human Connectome Project as an example, which is shown in the top row of Figure 8, in which the regions in different colors represent different parts of the brain. We select 3 landmark curves for each brain, which are in red, green, and cyan, respectively, as shown in the top row of Figure 8. We select  $R\text{Brain}0$  as the fixed surface and register  $R\text{Brain}1-R\text{Brain}4$  to it. The parameter  $\lambda$  in (6.1) is chosen as 1, 5, and 10 successively. The middle row shows the resulting spheres by solving the registration problem (6.1) with the landmark curves, respectively, with  $\lambda = 5$ . The bottom row shows the corresponding registered brains. The high similarity of the landmark curves between the fixed brain and registered brains illustrates the well performance of our method. Notably, all the registration maps are bijective. Table 5 presents their conformal energies, angle distortions, and registration losses. As the parameter  $\lambda$  increases, the conformal energies and angle distortions remain low. It is worth noting that the conformal energy  $E_C$  and the registration loss  $E_{\text{reg}}$  are highly nonlinear with respect to the spherical coordinates  $\theta$  and  $\phi$ . Consequently, it is hard to accurately obtain the global minimizer of  $E_B$  in (6.1) by HBTR. Thus, as the parameter  $\lambda$  increases, the resulting local minimizer  $f$  by HBTR may also lead to a slight increase in the registration loss, as shown in  $R\text{Brain}4$  for  $\lambda = 1$  to  $\lambda = 5$ , while in the majority of cases, the registration loss tends to decrease.

## 7 Conclusions

In this paper, we employ spherical coordinates and directly solve the spherical CEM problem for the computation of the surface conformal parameterization. Then, we give the explicit derivations of the gradient vector and Hessian matrix of the discrete conformal energy, which preserves the sparsity as the Laplacian matrix. Due to the sparsity of the Hessian matrix, the robust algorithm, called HBTR, is developed to solve the spherical CEM problem. HBTR sufficiently combines the local quadratic convergence and continuing descent advantages of the gradient and the Newton directions. The numerical experiments actually show the conformality, the robustness, and the local quadratic convergence of the HBTR. For the discrete scheme, we also present the quadratic convergence of the discrete conformal energy to the continuous scheme. Since the gradient vector and Hessian matrix of the registration loss have simple representations, we utilize the HBTR to propose a modified version of HBTR for the application



**Figure 8** (Color online) The right brains for the registration, registered spheres, and registered brains from top to bottom with  $\lambda = 5$ . The curves in red, green, and cyan are the landmark curves

**Table 5** The conformal energies, angle distortions, and registration losses of registered brain surfaces with  $\lambda = 1, 5, 10$ , respectively

Mesh	$\lambda$	Conformal energy	Angle distortion		Registration loss
			Mean	SD	
RBrain1	1	1.31E-03	3.56E-01	3.13E-01	4.95E-03
	5	3.41E-03	5.53E-01	5.27E-01	3.55E-03
	10	4.73E-03	6.28E-01	6.21E-01	1.39E-03
RBrain2	1	1.18E-03	3.50E-01	3.20E-01	4.35E-03
	5	3.00E-03	5.31E-01	5.43E-01	3.90E-03
	10	8.31E-03	7.55E-01	8.92E-01	3.00E-03
RBrain3	1	9.98E-04	3.28E-01	2.89E-01	3.87E-03
	5	2.34E-03	4.69E-01	4.42E-01	2.66E-03
	10	5.28E-03	5.86E-01	6.55E-01	1.83E-03
RBrain4	1	1.05E-03	3.35E-01	3.02E-01	2.67E-03
	5	2.63E-03	4.93E-01	4.82E-01	3.04E-03
	10	6.65E-03	7.38E-01	7.41E-01	2.40E-03

to surface registrations. Significantly, the modified algorithm 3 also has quadratic convergence, suggesting its potential for extension to other applications of our method.

**Acknowledgements** Tiexiang Li was supported by National Natural Science Foundation of China (Grant No. 12371377) and the Jiangsu Provincial Scientific Research Center of Applied Mathematics (Grant No. BK20233002). This work was supported by Shanghai Institute for Mathematics and Interdisciplinary Sciences (Grant No. SIMIS-ID-2024-LG). The authors thank Tianhe-2 and the Big Data Computing Center in Southeast University, China, for the use of their computing resources.

## References

- Alliez P, Ucelli G, Gotsman C, et al. Recent advances in remeshing of surfaces. In: De Floriani L, Spagnuolo M, eds. Shape Analysis and Structuring. Mathematics and Visualization. Berlin-Heidelberg: Springer, 2008, 53–82
- Bakas S, Baid U, Farahani K, et al. The international brain tumor segmentation (BraTS) cluster of challenges. Zenodo, <https://zenodo.org/records/7837974>, 2023
- Byrd R H, Schnabel R B, Shultz G A. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Math Program*, 1988, 40: 247–263
- Choi G P T, Leung-Liu Y, Gu X, et al. Parallelizable global conformal parameterization of simply-connected surfaces via partial welding. *SIAM J Imaging Sci*, 2020, 13: 1049–1083
- Choi P T, Lam K C, Lui L M. FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces. *SIAM J Imaging Sci*, 2015, 8: 67–94
- Desbrun M, Meyer M, Alliez P. Intrinsic parameterizations of surface meshes. *Comput Graph Forum*, 2002, 21: 209–218
- Engwirda D. Locally optimal Delaunay-refinement and optimisation-based mesh generation. PhD Thesis. Sydney: The University of Sydney, 2014
- Engwirda D. Voronoi-based point-placement for three-dimensional Delaunay-refinement. *Procedia Engrg*, 2015, 124: 330–342
- Engwirda D. Conforming restricted Delaunay mesh generation for piecewise smooth complexes. *Procedia Engrg*, 2016, 163: 84–96
- Engwirda D. Generalised primal-dual grids for unstructured co-volume schemes. *J Comput Phys*, 2018, 375: 155–176
- Engwirda D, Ivers D. Off-centre Steiner points for Delaunay-refinement on curved surfaces. *Comput-Aided Des*, 2016, 72: 157–171
- Eskandari H. Strictly conformal transformation optics for directivity enhancement and unidirectional cloaking of a cylindrical wire antenna. *Sci Rep*, 2022, 12: 16278
- Fang Q, Boas D A. Tetrahedral mesh generation from volumetric binary and grayscale images. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. Boston: IEEE, 2009, 1142–1145
- Floater M S. Mean value coordinates. *Comput Aided Geom Design*, 2003, 20: 19–27
- Floater M S, Hormann K. Surface parameterization: A tutorial and survey. In: Dodgson N A, Floater M S, Sabin M A, eds. Surface Parameterization. Mathematics and Visualization. Berlin-Heidelberg: Springer, 2005, 1–51

- M A, eds. *Advances in Multiresolution for Geometric Modelling. Mathematics and Visualization*. Berlin-Heidelberg: Springer, 2005, 157–186
- 16 Gillespie M, Springborn B, Crane K. Discrete conformal equivalence of polyhedral surfaces. *ACM Trans Graph*, 2021, 40: 103
- 17 Gu X, Wang Y, Chan T F, et al. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans Med Imag*, 2004, 23: 949–958
- 18 Gu X, Yau S-T. *Computational Conformal Geometry*. Advanced Lectures in Mathematics, vol. 3. Boston-Beijing: Int Press, Higher Ed Press, 2007
- 19 Gu X, Yau S-T. *Computational Conformal Geometry*. Boston-Beijing: Int Press, Higher Ed Press, 2020
- 20 Gu X, Zeng W, Luo F, et al. Numerical computation of surface conformal mappings. *Comput Methods Funct Theory*, 2012, 11: 747–787
- 21 Gurijala K C, Shi R, Zeng W, et al. Colon flattening using heat diffusion Riemannian metric. *IEEE Trans Vis Comput Graph*, 2013, 19: 2848–2857
- 22 Haker S, Angenent S, Tannenbaum A, et al. Conformal surface parameterization for texture mapping. *IEEE Trans Vis Comput Graph*, 2000, 6: 181–189
- 23 Hormann K, Lévy B, Sheffer A. *Mesh Parameterization: Theory and Practice*. New York: ACM, 2007
- 24 Huang W Q, Gu X D, Huang T M, et al. High performance computing for spherical conformal and Riemann mappings. *Geom Imaging Comput*, 2014, 1: 223–258
- 25 Hutchinson J E. Computing conformal maps and minimal surfaces. In: Dziuk G, Huisken G, Hutchinson J, eds. *Theoretical and Numerical Aspects of Geometric Variational Problems. Proceedings of the Centre for Mathematics and Its Applications*, vol. 26. Canberra: Austral Nat Univ, 1991, 140–161
- 26 Jin M, Kim J, Luo F, et al. Discrete surface Ricci flow. *IEEE Trans Vis Comput Graph*, 2008, 14: 1030–1043
- 27 Kuo Y C, Lin W W, Yueh M H, et al. Convergent conformal energy minimization for the computation of disk parameterizations. *SIAM J Imaging Sci*, 2021, 14: 1790–1815
- 28 Lai R, Wen Z, Yin W, et al. Folding-free global conformal mapping for genus-0 surfaces by harmonic energy minimization. *J Sci Comput*, 2014, 58: 705–725
- 29 Lévy B, Petitjean S, Ray N, et al. Least squares conformal maps for automatic texture atlas generation. *ACM Trans Graph*, 2002, 21: 362–371
- 30 Nie J, Liu T, Li G, et al. Least-square conformal brain mapping with spring energy. *Comput Med Imaging Graph*, 2007, 31: 656–664
- 31 Pinkall U, Polthier K. Computing discrete minimal surfaces and their conjugates. *Exp Math*, 1993, 2: 15–36
- 32 Sawhney R, Crane K. Boundary first flattening. *ACM Trans Graph*, 2017, 37: 5
- 33 Sheffer A, Praun E, Rose K. Mesh parameterization methods and their applications. *Found Trends Comput Graph Vis*, 2006, 2: 105–171
- 34 Shi R, Zeng W, Su Z, et al. Hyperbolic harmonic mapping for surface registration. *IEEE Trans Pattern Anal Mach Intell*, 2017, 39: 965–980
- 35 Shultz G A, Schnabel R B, Byrd R H. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J Numer Anal*, 1985, 22: 47–67
- 36 Tran A P, Yan S, Fang Q. Improving model-based functional near-infrared spectroscopy analysis using mesh-based anatomical and light-transport models. *Neurophotonics*, 2020, 7: 015008
- 37 Xu L, Chen H. Conformal transformation optics. *Nat Photonics*, 2015, 9: 15–23
- 38 Yang Y-L, Guo R, Luo F, et al. Generalized discrete Ricci flow. *Comput Graph Forum*, 2009, 28: 2005–2014
- 39 Yueh M-H, Huang T-M, Li T, et al. Projected gradient method combined with homotopy techniques for volume-measure-preserving optimal mass transportation problems. *J Sci Comput*, 2021, 88: 64
- 40 Yueh M-H, Li T, Lin W W, et al. A novel algorithm for volume-preserving parameterizations of 3-manifolds. *SIAM J Imaging Sci*, 2019, 12: 1071–1098
- 41 Yueh M-H, Lin W W, Wu C T, et al. An efficient energy minimization for conformal parameterizations. *J Sci Comput*, 2017, 73: 203–227