SIAM J. Imaging Sciences
Vol. 12, No. 2, pp. 1071–1098

A Novel Algorithm for Volume-Preserving Parameterizations of 3-Manifolds[*]

Mei-Heng Yueh[†], Tiexiang Li[‡], Wen-Wei Lin[§], and Shing-Tung Yau[¶]

Abstract. Manifold parameterizations have been applied to various fields of commercial industries. Several efficient algorithms for the computation of triangular surface mesh parameterizations have been proposed in recent years. However, the computation of tetrahedral volumetric mesh parameterizations is more challenging due to the fact that the number of mesh points would become enormously large when the higher-resolution mesh is considered and the bijectivity of parameterizations is more difficult to guarantee. In this paper, we develop a novel volumetric stretch energy minimization algorithm for volume-preserving parameterizations of simply connected 3-manifolds with a single boundary under the restriction that the boundary is a spherical area-preserving mapping. In addition, our algorithm can also be applied to compute spherical angle- and area-preserving parameterizations of genus-zero closed surfaces, respectively. Several numerical experiments indicate that the developed algorithms are more efficient and reliable compared to other existing algorithms. Numerical results on applications of the manifold partition and the mesh processing for three-dimensional printing are demonstrated thereafter to show the robustness of the proposed algorithm.

Key words. volumetric stretch energy minimization, volume-preserving parameterization, simply connected 3-manifold, manifold parameterization, genus-zero closed surface

AMS subject classifications. 15B48, 52C26, 65F05, 68U05, 65D18

DOI. 10.1137/18M1201184

1. Introduction.

A manifold parameterization refers to a bijective mapping between the manifold and the domain of a simple canonical shape. The mapping induces a canonical coordinate system on the manifold, which can be used to simplify the issues arising from geometry processing and computer graphics. In particular, surface (2-manifold) parameterizations have been widely studied and applied in various tasks of computer vision, such as surface registration, remeshing, morphing, and texture mapping. Several numerical algorithms for the computation of surface parameterizations have been developed by different research groups in recent years. Most of the parameterization algorithms for surfaces are based on

[*]Received by the editors July 18, 2018; accepted for publication (in revised form) April 10, 2019; published electronically June 13, 2019.

http://www.siam.org/journals/siims/12-2/M120118.html

Funding: The work of the second author was partially supported by National Natural Science Foundation of China grant 11471074. The work of the authors was partially supported by the Ministry of Science and Technology (MOST), the National Center for Theoretical Sciences (NCTS), the Taida Institute for Mathematical Sciences, the ST Yau Center in Taiwan, the Shing-Tung Yau Center at Southeast University, and the Center of Mathematical Sciences and Applications at Harvard University.

[†]Department of Mathematics, National Taiwan Normal University, Taipei, Taiwan (yue@ntnu.edu.tw).

[‡]School of Mathematics, Southeast University, Nanjing, China (txli@seu.edu.cn).

[§]Department of Applied Mathematics, National Chiao Tung University, Hsinchu, Taiwan (wwlin@math.nctu.edu.tw).

[¶]Department of Mathematics, Harvard University, Cambridge, MA 02138 (yau@math.harvard.edu).

minimizing the distortions of angle or area, or balancing between them. An angle-preserving parameterization is called a *conformal* parameterization, which has been applied in the texture mapping of surfaces [40, 50] and the image analysis of proximal femur surfaces [55]. On the other hand, an area-preserving parameterization is called an *authalic* or *equiareal* parameterization, which has been applied in the surface registration and the remeshing [17, 76]. More details for methods and applications of surface parameterizations can be found in the survey papers [30, 64, 42, 10, 33, 39].

In the past, most of the related works merely consider the surface data in three-dimensional (3D) space, which is mathematically a 2D object that can be bijectively mapped onto a planar domain. With the development of 3D imaging technology, such as *computed tomography scans* and *magnetic resonance imaging*, 3D images in the real world can be obtained easily. Such a 3D image is mathematically equivalent to a simply connected 3-manifold in $\mathbb{R}^3$ that can be bijectively mapped into a unit solid ball while preserving the local volume. To realize the volume-preserving parameterization, Su et al. [67] proposed a useful algorithm based on the discrete optimal mass transportation (OMT) [34]. However, it is usually very time-consuming to compute a required parameterization for 3-manifolds, especially when high-resolution mesh data are considered, e.g., it would cost more than 15 hours[1] using the OMT algorithm to compute a spherical volume-preserving mapping of a volumetric mesh of 290K tetrahedrons. In addition, the bijectivity of a volumetric mapping is more difficult to guarantee [29, 31].

1.1. Contributions.

In this paper, we develop a novel volumetric stretch energy minimization (VSEM) for the computation of a volume-preserving parameterization that maps a simply connected volumetric mesh with a single boundary into a unit solid ball. First, the boundary of the 3-manifold is mapped conformally into a unit sphere by minimizing the Dirichlet energy. Then, the boundary mapping is deformed to an equiareal mapping by minimizing the stretch energy. Finally, the volume-preserving parameterization is computed by minimizing the volumetric stretch energy. The contribution of this paper can be separated into three areas:

- Universality: The proposed energy minimization algorithm can be used to compute (i) angle-preserving (conformal) and area-preserving (equiareal) parameterizations, respectively, of genus-zero closed surfaces, and (ii) volume-preserving parameterizations of simply connected 3-manifolds with a single boundary.
- Improved effectiveness and accuracy: The effectiveness and the accuracy of parameterizations computed by the proposed volume-preserving parameterization algorithm are significantly improved compared to the OMT algorithm [67].
- Applications: Applications on the manifold partition for 3-manifolds and the mesh processing for 3D printing can be performed robustly by the proposed parameterization algorithm.

1.2. Notation and overview.

The following notation is used in this paper. Other notation will be clearly defined whenever they appear.

- Bold letters, e.g., $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, denote (complex) vectors.
- Capital letters, e.g., A, B, C, denote matrices.

[1] See the computational cost of the OMT algorithm for the *Bimba* mesh model in Table 6.3.

- Typewriter letters, e.g., $\mathtt{I}$, $\mathtt{J}$, $\mathtt{K}$, denote ordered sets of indices.
- $\mathbf{v}_i$ denotes the ith entry of the vector $\mathbf{v}$.
- $\mathbf{v}_{\mathtt{I}}$ denotes the subvector of $\mathbf{v}$ composed of $\mathbf{v}_i$ for $i \in \mathtt{I}$.
- $|\mathbf{v}|$ denotes the vector with the ith entry being $|\mathbf{v}_i|$.
- $\operatorname{diag}(\mathbf{v})$ denotes the diagonal matrix with the (i,i)th entry being $\mathbf{v}_i$.
- $A_{i,j}$ denotes the (i,j)th entry of the matrix A.
- $A_{\mathtt{I},\mathtt{J}}$ denotes the submatrix of A composed of $A_{i,j}$ for $i \in \mathtt{I}$ and $j \in \mathtt{J}$.
- $\mathbb{S}^n := \{\mathbf{x} \in \mathbb{R}^{n+1} \mid \|\mathbf{x}\| = 1\}$ denotes the n-sphere in $\mathbb{R}^{n+1}$.
- $\mathbb{B}^n := \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq 1\}$ denotes the solid n-ball in $\mathbb{R}^n$.
- $[v_0, \ldots, v_m]$ denotes the m-simplex determined by the points $v_0, \ldots, v_m$.
- $|[v_0, \ldots, v_m]|$ denotes the volume of the m-simplex $[v_0, \ldots, v_m]$.
- i denotes the imaginary unit $\sqrt{-1}$.
- I_n denotes the identity matrix of size $n \times n$.
- $\mathbf{0}$ denotes the zero vectors and matrices of appropriate sizes.

This paper is organized as follows. First, we introduce the related previous works and background in sections 2 and 3, respectively. Then, we propose a novel VSEM algorithm in section 4. The bijectivity of volume-preserving parameterizations is discussed in section 5. Numerical comparisons between our algorithm and the OMT algorithm are presented in section 6. Applications on the manifold partition and the mesh processing for 3D printing are demonstrated in section 7. Concluding remarks are given in section 8.

2. Previous works.

In this section, we briefly review the related previous works on computational algorithms for surface and volumetric parameterizations, respectively.

2.1. Surface parameterizations.

The major categories of surface parameterizations contain conformal parameterizations, equiareal parameterizations, and balancing parameterizations between angle and area distortions.

A conformal parameterization aims to minimize the angular distortion. Numerical methods for surface conformal parameterizations have been widely developed because the conformal parameterization preserves the local shape well. The existence of conformal mappings between surfaces is guaranteed by the Poincaré–Klein–Koebe uniformization theorem of 100 years ago. In particular, for a genus-zero closed surface, some feasible numerical methods including the linear Laplace–Beltrami equation [11], the nonlinear heat diffusion [35, 43], and the fast landmark aligned spherical harmonic (FLASH) algorithm [18] have been proposed to map it conformally to a unit sphere. In addition, for a simply connected open surface, some efficient numerical methods including the fast disk mapping [19], the linear disk mapping [16], and the conformal energy minimization (CEM) [75] have been recently developed to map it conformally to a unit disk. Furthermore, classical methods including discrete conformal parameterization [21], least-squares conformal mapping [50], angle-based flattening [62, 63, 77], discrete conformal equivalence [66], spectral conformal parameterization [53, 44], and orbifold Tutte embedding [8, 9, 7] have been proposed for the conformal parameterization of topological disks with free or mild shape constraints of boundaries. For conformal parameterizations of higher genus surfaces, some well-known methods have been developed, including the holomorphic one-form [36, 37, 46], the discrete Ricci flow [45, 78], and the discrete Calabi flow [79].

On the other hand, an equiareal parameterization aims to minimize the area distortion. Numerical methods for computing surface equiareal parameterizations have been gradually getting much more attention in recent years because equiareal mappings preserve the density of the vertices well. Some efficient numerical methods, including the stretch-minimizing method [60, 74], the Lie advection method [81], discrete OMT [80, 68], density-equalizing mapping [17], and the stretch energy minimization (SEM) [76] have been developed to achieve this purpose.

Additionally, for parameterizations of balancing distortions, some well-developed numerical methods, including the as-rigid-as-possible surface parameterization [51, 72], the most isometric parametrization [41, 20], the isometric distortion minimization [59], and boundary first flattening [61], have been proposed to reach a trade-off between minimizing the angle and area distortions.

2.2. Volumetric parameterizations. The computation of 3-manifold parameterizations is more challenging due to the fact that the number of vertices would become enormously large when the high-resolution volumetric mesh is considered. In addition, the bijectivity of the volumetric mapping is more difficult to guarantee because convex combination mappings in 3D space do not need to be bijective [31]. Note that conformal mappings between 3-manifolds, in general, do not exist. Still, volumetric mappings with small angular distortion are frequently adopted. Paillé and Poulin [56] and Chern, Pinkall, and Schröder [15] proposed conformal-based volumetric mapping methods by applying the Cauchy–Riemann equation to each canonical orthogonal plane in $\mathbb{R}^3$ and a low shear distortion on the decoupling of scaling and rotation, respectively. Kovalsky et al. [48, 49] developed methods to deform a given volumetric mapping into a bijective one with bounded distortion. Paillé et al. [57] proposed a spectral method based on the dihedral angle representation for computing the locally injective mapping of tetrahedral meshes. Jin et al. [47] proposed a method for computing the volumetric parameterization balancing between angle and volume distortions by minimizing the stretch-distortion energy. Naitsat, Saucan, and Zeevi [54] and Rabinovic et al. [59] proposed deformation methods for volumetric meshes based on the quasi-conformal homeomorphism and minimizing a linear combination of local isometric distortion measures [65], respectively. The above methods have mainly been developed for the computation of bijective volumetric mappings of 3-manifolds by minimizing angular or isometric distortions so that the qualities on angles and local shapes of tetrahedral meshes are well-preserved, but in general they are not volume-preserving.

On the other hand, it is worth noting that only a few of the existing algorithms consider the volumetric parameterizations for prescribed shapes that minimize the volume distortion. On the basis of OMT, Su et al. [67] developed a pioneering algorithm of volume-preserving parameterizations for tetrahedral meshes that maps a simply connected 3-manifold with a single boundary into a unit solid ball, which has advantages on manifold resampling and registration.

3. Discrete manifolds and parameterizations.

A 3-manifold refers to a 3D topological space in which each point of the manifold has a neighborhood homeomorphic to a subset of $\mathbb{R}^3$. In this paper, we consider simply connected discrete 3-manifolds with a single boundary that are embedded in $\mathbb{R}^3$. In practice, a discrete 3-manifold is represented as a tetrahedral mesh $\mathcal{M}$ composed of n vertices with coordinates in $\mathbb{R}^3$

$$\mathcal{V}(\mathcal{M}) = \left\{ v_s \equiv \left(v_s^1, v_s^2, v_s^3\right)^\top \in \mathbb{R}^3 \right\}_{s=1}^{n}$$

and tetrahedrons

$$\mathcal{T}(\mathcal{M}) = \left\{ [v_i, v_j, v_k, v_\ell] \subset \mathbb{R}^3 \text{ for some vertices } \{v_i, v_j, v_k, v_\ell\} \subset \mathcal{V}(\mathcal{M}) \right\}.$$

Here the bracket $[v_i, v_j, v_k, v_\ell]$ denotes the *convex hull* (3-simplex) of the affinely independent points $\{v_i, v_j, v_k, v_\ell\}$. Furthermore, we denote the set of triangular faces and edges of the mesh $\mathcal{M}$ by

$$\mathcal{F}(\mathcal{M}) = \{[v_i, v_j, v_k] \mid [v_i, v_j, v_k, v_\ell] \in \mathcal{T}(\mathcal{M}) \text{ for some } v_\ell \in \mathcal{V}(\mathcal{M})\}$$

and

$$\mathcal{E}(\mathcal{M}) = \{[v_i, v_j] \mid [v_i, v_j, v_k] \in \mathcal{F}(\mathcal{M}) \text{ for some } v_k \in \mathcal{V}(\mathcal{M})\},$$

respectively. The union of $\mathcal{T}(\mathcal{M})$, $\mathcal{F}(\mathcal{M})$, $\mathcal{E}(\mathcal{M})$, and $\mathcal{V}(\mathcal{M})$, forms a homogeneous simplicial 3-complex. Similarly, a 2-manifold is called a *surface*, which is represented as a triangular mesh composed of vertices and triangular faces.

A piecewise affine mapping $f : \mathcal{M} \to \mathbb{R}^3$ can be represented as a matrix

$$\mathbbm{f} = \left[f(v_1) \quad \cdots \quad f(v_n) \right]^\top \equiv \left[\mathbbm{f}_1 \quad \cdots \quad \mathbbm{f}_n \right]^\top \in \mathbb{R}^{n\times 3}. \tag{3.1}$$

Note that for the point $v \in \mathcal{M}$ which is not a vertex, v must belong to a tetrahedron $[v_i, v_j, v_k, v_\ell] \in \mathcal{T}(\mathcal{M})$. Then, the value $f(v)$ can be represented as a linear combination of $\mathbbm{f}_i$, $\mathbbm{f}_j$, $\mathbbm{f}_k$, and $\mathbbm{f}_\ell$ with coefficients being the *barycentric coordinates* of v in $[v_i, v_j, v_k, v_\ell]$, that is,

$$f|_{[v_i,v_j,v_k,v_\ell]}(v) = \lambda_i(v)\, \mathbbm{f}_i + \lambda_j(v)\, \mathbbm{f}_j + \lambda_k(v)\, \mathbbm{f}_k + \lambda_\ell(v)\, \mathbbm{f}_\ell,$$

where $\lambda_i(v) = \frac{|[v,v_j,v_k,v_\ell]|}{|[v_i,v_j,v_k,v_\ell]|}$, $\lambda_j(v) = \frac{|[v_i,v,v_k,v_\ell]|}{|[v_i,v_j,v_k,v_\ell]|}$, $\lambda_k(v) = \frac{|[v_i,v_j,v,v_\ell]|}{|[v_i,v_j,v_k,v_\ell]|}$, and $\lambda_\ell(v) = \frac{|[v_i,v_j,v_k,v]|}{|[v_i,v_j,v_k,v_\ell]|}$. Here the absolute value $|[v_0, \ldots, v_m]|$ denotes the volume of the m-simplex $[v_0, \ldots, v_m]$. In particular, $|[v_i, v_j, v_k, v_\ell]|$, $|[v_i, v_j, v_k]|$, and $|[v_i, v_j]|$ denote the volume, area and length of the tetrahedron $[v_i, v_j, v_k, v_\ell]$, triangle $[v_i, v_j, v_k]$, and interval $[v_i, v_j]$, respectively.

Remark. The matrix $\mathbbm{f}$ represents the unique piecewise affine mapping satisfying $f(v_s) = \mathbbm{f}_s$ for every $v_s \in \mathcal{V}(\mathcal{M})$.

In this paper, the considered *parameterization* of a 3-manifold $\mathcal{M} \subset \mathbb{R}^3$ is a bijective piecewise affine mapping f that maps $\mathcal{M}$ to the unit solid ball $\mathbb{B}^3$. A parameterization $f : \mathcal{M} \to \mathbb{B}^3$ is said to be *volume-preserving* if the Jacobian matrix $\mathrm{J}_{f^{-1}} = \left[\frac{\partial f^{-1}}{\partial u^1} \; \frac{\partial f^{-1}}{\partial u^2} \; \frac{\partial f^{-1}}{\partial u^3} \right]$ satisfies

$$\det\left(\mathrm{J}_{f^{-1}}(u^1, u^2, u^3)\right) = 1.$$

In other words, f preserves the local volume.

In addition, for a genus-zero closed triangular mesh $\mathcal{S}$, a parameterization $f : \mathcal{S} \to \mathbb{S}^2$ is said to be *conformal* or *angle-preserving* if the first fundamental form $\mathrm{I}_{f^{-1}}$ satisfies

$$\mathrm{I}_{f^{-1}}(u^1, u^2) = \lambda(u^1, u^2) I_2$$

for some positive scaling function λ, i.e., f preserves local angles. A parameterization $f : \mathcal{S} \to \mathbb{S}^2$ is said to be *equiareal* or *area-preserving* if the first fundamental form $\mathrm{I}_{f^{-1}}$ satisfies $\det\left(\mathrm{I}_{f^{-1}}(u^1, u^2)\right) = 1$, i.e., f preserves the local area.

4. The novel algorithm.

In this section, we develop a novel energy minimization algorithm for the computation of a volume-preserving parameterization that maps a simply connected 3-manifold with a single boundary to a unit solid ball. Given a simply connected tetrahedral mesh $\mathcal{M}$ with the boundary $\partial\mathcal{M} = \mathcal{S}$ being a genus-zero closed surface, our algorithm is composed of three stages as follows. First, the initial boundary mapping is chosen to be a spherical conformal parameterization $f : \mathcal{S} \to \mathbb{S}^2$, which is computed by minimizing the discrete Dirichlet energy [22]. Then, the boundary mapping is deformed into an equiareal parameterization, which is iteratively computed by minimizing the stretch energy [76]. Finally, the volume-preserving parameterization with the prescribed boundary mapping is computed by minimizing the volumetric stretch energy.

4.1. Initial conformal boundary parameterizations.

The initial spherical conformal parameterization is carried out by applying the CEM algorithm [75]. The original CEM algorithm aims to compute disk-shaped conformal parameterizations of simply connected open surfaces by minimizing the *discrete Dirichlet energy functional* [22, 38]

$$\mathcal{E}_D(f) = \frac{1}{2}\text{trace}\left(\mathbb{f}^\top L_D\, \mathbb{f}\right), \tag{4.1}$$

where L_D is the Laplacian matrix with

$$[L_D]_{i,j} = \begin{cases} -w_{i,j} \equiv -\frac{1}{2}(\cot\alpha_{i,j} + \cot\alpha_{j,i}) & \text{if } [v_i, v_j] \in \mathcal{E}(\mathcal{S}), \\ \sum_{k\neq i} w_{i,k} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

in which $\alpha_{i,j}$ and $\alpha_{j,i}$ are two angles opposite to the edge $[v_i, v_j]$. The modified CEM algorithm for genus-zero closed surfaces is stated as follows. First, an initial mapping $h^{(0)} : \mathcal{S} \to \overline{\mathbb{C}}$ is obtained by solving the Laplace–Beltrami equation

$$\triangle_{\mathcal{S}} h^{(0)} = \left(\frac{\partial}{\partial x} - \mathrm{i}\frac{\partial}{\partial y}\right)\delta_p, \tag{4.3}$$

where δ_p is the Dirac delta function at a selected point p on $\mathcal{S}$, and (x, y) are the local coordinates defined on a neighborhood of p. This method was originally proposed by Angenent et al. [11] and has a benefit that the resulting spherical parameterization is bijective. A straightforward proof for the bijectivity of the method [11] is given in Theorem 5.3. The algorithm for solving (4.3) is summarized in Algorithm 4.1.

Unsatisfactorily, the angular distortion of the mapping computed by Algorithm 4.1 would be relatively large at the neighborhood of p. To remedy this drawback, we improve the conformality of the mapping by iteratively minimizing the Dirichlet energy

$$\begin{cases} \triangle_{\mathcal{S}} h^{(k)}(v) = 0 & \text{if } \left|\text{Inv} \circ h^{(k-1)}(v)\right| < r, \\ h^{(k)}(v) = \text{Inv} \circ h^{(k-1)}(v) & \text{otherwise,} \end{cases} \tag{4.4}$$

where Inv denotes the inversion $\text{Inv}(z) = \frac{1}{\bar{z}}$. Once the iteration (4.4) converges, the desired spherical conformal parameterization is obtained by the inverse stereographic projection $\Pi_{\mathbb{S}^2}^{-1} : \overline{\mathbb{C}} \to \mathbb{S}^2$ defined by

$$\Pi_{\mathbb{S}^2}^{-1}(z) = \left(\frac{2\,\mathrm{Re}\, z}{|z|^2+1}, \frac{2\,\mathrm{Im}\, z}{|z|^2+1}, \frac{|z|^2-1}{|z|^2+1} \right). \tag{4.5}$$

The improved algorithm derived by (4.4) for the computation of spherical conformal parameterizations is summarized in Algorithm 4.2.

Remark. In fact, the FLASH algorithm [18] for spherical conformal parameterization is satisfactory with high accuracy and effectiveness. However, the accuracy and effectiveness of Algorithm 4.2 are slightly better than those of the FLASH algorithm [18]. Numerical comparisons between the FLASH algorithm [18] and Algorithm 4.2 are demonstrated in Appendix C.

Algorithm 4.1. Spherical conformal parameterizations [11].

```
Input: A genus-zero closed mesh $\mathcal{S}$.
Output: A spherical conformal parameterization $\mathbb{f}$.
1: Let $n$ be the number of vertices of $\mathcal{S}$.
2: Find the most regular triangular face by
   $$[v_a, v_b, v_c] = \mathop{\mathrm{argmin}}_{[v_i,v_j,v_k]\in\mathcal{F}(\mathcal{S})} \left\| \begin{bmatrix} |[v_i,v_j]| - \frac{1}{3}\left(|[v_i,v_j]| + |[v_j,v_k]| + |[v_k,v_i]|\right) \\ |[v_j,v_k]| - \frac{1}{3}\left(|[v_i,v_j]| + |[v_j,v_k]| + |[v_k,v_i]|\right) \\ |[v_k,v_i]| - \frac{1}{3}\left(|[v_i,v_j]| + |[v_j,v_k]| + |[v_k,v_i]|\right) \end{bmatrix} \right\|.$$
3: Set $\mathtt{B} = \{a, b, c\}$ and $\mathtt{I} = \{1, \dots, n\}\backslash\mathtt{B}$.
4: Set $\alpha = \frac{(v_c-v_a)^\top(v_b-v_a)}{\|v_b-v_a\|_2^2}$ and
   $$\mathbf{h}_\mathtt{B} = \begin{bmatrix} \frac{-1}{\|v_b-v_a\|_2} \\ \frac{1}{\|v_b-v_a\|_2} \\ 0 \end{bmatrix} + \mathrm{i} \begin{bmatrix} \frac{1-\alpha}{\|v_c-(v_a+\alpha(v_b-v_a))\|_2} \\ \frac{\alpha}{\|v_c-(v_a+\alpha(v_b-v_a))\|_2} \\ \frac{-1}{\|v_c-(v_a+\alpha(v_b-v_a))\|_2} \end{bmatrix}. \tag{4.6}$$
5: Compute $\mathbf{h}$ by solving the linear system
   $$[L_D]_{\mathtt{I},\mathtt{I}}\mathbf{h}_\mathtt{I} = -[L_D]_{\mathtt{I},\mathtt{B}}\mathbf{h}_\mathtt{B}, \tag{4.7}$$
   where $L_D$ is defined as in (4.2).
6: The spherical parameterization $\mathbb{f}$ is obtained by $\Pi_{\mathbb{S}^2}^{-1}(\mathbf{h}_\ell)$ in (4.5), $\ell = 1, \dots, n$.
```

4.2. Equiareal boundary parameterizations. The boundary spherical equiareal parameterization is carried out by applying the SEM algorithm [76]. The original SEM algorithm [76] aims to compute disk-shaped equiareal parameterizations of simply connected open surfaces by minimizing the *stretch energy functional*

$$\mathcal{E}_S(f) = \frac{1}{2}\mathrm{trace}\left(\mathbb{f}^\top L_S(f)\,\mathbb{f}\right), \tag{4.8}$$

Algorithm 4.2. CEM for spherical conformal parameterizations.

```
Input: A genus-zero closed mesh $\mathcal{S}$, a tolerance $\varepsilon$, a radius $r$ (e.g., $\varepsilon = 10^{-6}$, $r = 1.2$).
Output: A spherical conformal parameterization $\mathbb{f}$.
Let $n$ be the number of vertices of $\mathcal{S}$.
Compute a spherical conformal parameterization $\mathbb{g}$ using Algorithm 4.1.
Perform the stereographic projection $\mathbf{h}_\ell \leftarrow \frac{\mathbb{g}_{\ell,1}}{1-\mathbb{g}_{\ell,3}} + \mathrm{i}\frac{\mathbb{g}_{\ell,2}}{1-\mathbb{g}_{\ell,3}}$, $\ell = 1, \ldots, n$.
Set $\delta \leftarrow \infty$.
while $\delta > \varepsilon$ do
    Perform the inversion $\mathbf{h} \leftarrow \mathrm{diag}(|\mathbf{h}|)^{-2}\mathbf{h}$.
    Update the index sets $\mathtt{I} = \{i \mid |\mathbf{h}_i| < r\}$ and $\mathtt{B} = \{1, \ldots, n\}\backslash\mathtt{I}$.
    Update $\mathbf{h}$ by solving the linear system
        $$[L_D]_{\mathtt{I},\mathtt{I}}\mathbf{h}_{\mathtt{I}} = -[L_D]_{\mathtt{I},\mathtt{B}}\mathbf{h}_{\mathtt{B}},$$
    where $L_D$ is defined as in (4.2).
    Set $\mathbb{f}_\ell \leftarrow \Pi_{\mathbb{S}^2}^{-1}(\mathbf{h}_\ell)$ as in (4.5), $\ell = 1, \ldots, n$.
    Update $\delta \leftarrow \mathcal{E}_D(\mathbb{g}) - \mathcal{E}_D(\mathbb{f})$.
    Update $\mathbb{g} \leftarrow \mathbb{f}$.
end while
return The spherical parameterization $\mathbb{f}$.
```

where $L_S(f)$ is the stretch Laplacian matrix with

$$[L_S(f)]_{i,j} = \begin{cases} -w_{i,j}(f) \equiv -\frac{1}{2}\left(\frac{\cot(\alpha_{i,j}(f))}{\sigma_{f^{-1}}([v_i,v_j,v_k])} + \frac{\cot(\alpha_{j,i}(f))}{\sigma_{f^{-1}}([v_j,v_i,v_\ell])}\right) & \text{if } [v_i, v_j] \in \mathcal{E}(\mathcal{S}), \\ \sum_{\ell \neq i} w_{i,\ell}(f) & \text{if } j = i, \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

in which $\sigma_{f^{-1}}([v_i, v_j, v_k]) = \frac{|[v_i,v_j,v_k]|}{|f([v_i,v_j,v_k])|}$ is the stretch factor of f on the triangular face $[v_i, v_j, v_k]$. The modified SEM algorithm for genus-zero closed surfaces is stated as follows. First, the initial mapping $\mathbb{f}^{(0)}$ is a spherical conformal parameterization computed by Algorithm 4.2 and let $\mathbf{h}_\ell^{(0)} = \frac{\mathbb{f}_{\ell,1}^{(0)}}{1-\mathbb{f}_{\ell,3}^{(0)}} + \mathrm{i}\frac{\mathbb{f}_{\ell,2}^{(0)}}{1-\mathbb{f}_{\ell,3}^{(0)}}, \ell = 1, \ldots, n$. Then, the stretch energy (4.8) is iteratively minimized by

$$\left[L_S(f^{(k)})\right]_{\mathtt{I}^{(k)},\mathtt{I}^{(k)}} \mathbf{h}_{\mathtt{I}^{(k)}}^{(k+1)} = -\left[L_S(f^{(k)})\right]_{\mathtt{I}^{(k)},\mathtt{B}^{(k)}} \mathbf{h}_{\mathtt{B}^{(k)}}^{(k)}, \tag{4.10}$$

where L_S is defined in (4.9),

$$\mathtt{I}^{(k)} = \left\{\ell \;\middle|\; |(\mathrm{diag}(|\mathbf{h}^{(k)}|)^{-2}\mathbf{h}^{(k)})_\ell| < r\right\} \text{ and } \mathtt{B}^{(k)} = \{1, \ldots, n\}\backslash\mathtt{I}^{(k)}.$$

In practice, the radius r is chosen to be 1.2. Once $\mathbf{h}^{(k)}$ in the iteration (4.10) converges to $\mathbf{h}^{(*)}$, the desired spherical equiareal parameterization $\mathbb{f}$ is obtained by the inverse stereographic projection operates on $\mathbf{h}^{(*)}$.

The algorithm for the computation of spherical equiareal parameterizations by (4.10) is summarized in Algorithm 4.3.

Algorithm 4.3. SEM for spherical equiareal parameterizations.

Input: A genus-zero closed mesh $\mathcal{S}$, a tolerance ε, a radius r (e.g., $\varepsilon = 10^{-6}$, $r = 1.2$).
Output: A spherical equiareal parameterization $\mathbb{f}$.

```
Let $n$ be the number of vertices of $\mathcal{S}$.
Compute a spherical conformal parameterization $\mathbb{g}$ using Algorithm 4.2.
Perform the stereographic projection $\mathbf{h}_\ell = \frac{\mathbb{g}_{\ell,1}}{1-\mathbb{g}_{\ell,3}} + \mathrm{i}\frac{\mathbb{g}_{\ell,2}}{1-\mathbb{g}_{\ell,3}}, \ell = 1, \ldots, n$.
Set $\delta \leftarrow \infty$.
while $\delta > \varepsilon$ do
    Update the matrix $A \leftarrow L_S(f)$, where $L_S(f)$ is defined as in (4.9).
    Perform the inversion $\mathbf{h} \leftarrow \operatorname{diag}(|\mathbf{h}|)^{-2}\mathbf{h}$.
    Update the index sets $\mathtt{I} = \{i \mid |\mathbf{h}_i| < r\}$ and $\mathtt{B} = \{1, \ldots, n\}\backslash\mathtt{I}$.
    Update $\mathbf{h}$ by solving the linear system $A_{\mathtt{I},\mathtt{I}}\mathbf{h}_{\mathtt{I}} = -A_{\mathtt{I},\mathtt{B}}\mathbf{h}_{\mathtt{B}}$.
    Set $\mathbb{f}_\ell \leftarrow \Pi_{\mathbb{S}^2}^{-1}(\mathbf{h}_\ell)$ as in (4.5), $\ell = 1, \ldots, n$.
    Update $\delta \leftarrow \mathcal{E}_S(\mathbb{g}) - \mathcal{E}_S(\mathbb{f})$.
    Update $\mathbb{g} \leftarrow \mathbb{f}$.
end while
return The spherical parameterization $\mathbb{f}$.
```

4.3. Volume-preserving parameterizations.

Let $\mathcal{M}$ be a tetrahedral mesh of n vertices in $\mathbb{R}^3$ and let $f : \mathcal{M} \to \mathbb{R}^3$ be a piecewise affine mapping on $\mathcal{M}$. The *volumetric Dirichlet energy functional* [69, 70, 71] is defined by

$$\mathcal{E}_{\mathbb{D}}(f) = \frac{1}{2}\operatorname{trace}\left(\mathbb{f}^\top L_{\mathbb{D}}\, \mathbb{f}\right), \tag{4.11}$$

where $L_{\mathbb{D}} \in \mathbb{R}^{n\times n}$ is the volumetric Laplacian matrix with

$$[L_{\mathbb{D}}]_{i,j} = \begin{cases} -w_{i,j} & \text{if } [v_i, v_j] \in \mathcal{E}(\mathcal{M}), \\ \sum_{k\neq i} w_{i,k} & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.12}$$

in which $w_{i,j}$ is the cotangent weight on the edge $[v_i, v_j]$ given by

$$w_{i,j} = \frac{1}{6} \sum_{\substack{\tau\in\mathcal{T}(\mathcal{M}) \\ [v_i,v_j]\cup[v_k,v_\ell]\subset\tau \\ [v_i,v_j]\cap[v_k,v_\ell]=\varnothing}} |[v_k, v_\ell]| \cot\theta_{i,j}^{k,\ell}, \tag{4.13}$$

where $\theta_{i,j}^{k,\ell}$ is the dihedral angle between $[v_i, v_k, v_\ell]$ and $[v_j, v_\ell, v_k]$ in the tetrahedron τ on the edge $[v_k, v_\ell]$, as illustrated in Figure 4.1. It is worth noting that the cotangent formula (4.13) contains the information of volumes of tetrahedrons, which is explicitly demonstrated in Proposition 4.1.

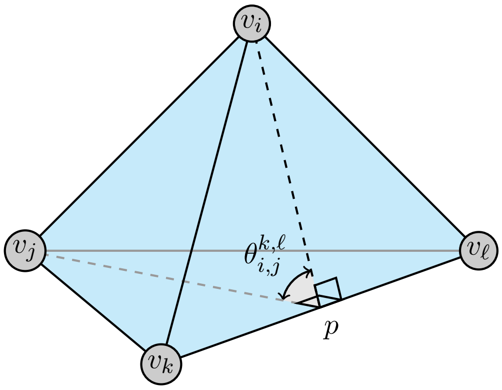


Figure 4.1. *An illustration for the dihedral angle between* $[v_i, v_k, v_\ell]$ *and* $[v_j, v_\ell, v_k]$ *in the tetrahedron* $[v_i, v_j, v_k, v_\ell]$.

Proposition 4.1. *The cotangent formula* (4.13) *can be written as*

$$w_{i,j} = \frac{1}{9} \sum_{\substack{\tau\in\mathcal{T}(\mathcal{M})\\ [v_i,v_j]\cup[v_k,v_\ell]\subset\tau\\ [v_i,v_j]\cap[v_k,v_\ell]=\varnothing}} \frac{|[v_i, v_k, v_\ell]||[v_j, v_\ell, v_k]| \cos\theta_{i,j}^{k,\ell}}{|\tau|}, \tag{4.14}$$

where $\theta_{i,j}^{k,\ell}$ *is the dihedral angle as in Figure* 4.1.

Proof. Let p be the orthogonal projection of v_i on $[v_k, v_\ell]$. The cotangent formula (4.13) can be written as

$$w_{i,j} = \frac{1}{6} \sum_{\substack{\tau\in\mathcal{T}(\mathcal{M})\\ [v_i,v_j]\cup[v_k,v_\ell]\subset\tau\\ [v_i,v_j]\cap[v_k,v_\ell]=\varnothing}} |[v_k, v_\ell]| \frac{(v_i - p)^\top (v_j - p)}{2|[v_i, v_j, p]|}.$$

By applying the identities $|[v_i, v_j, p]||[v_k, v_\ell]| = 3|[v_i, v_j, v_k, v_\ell]|$, $|[v_i, p]||[v_k, v_\ell]| = 2|[v_i, v_k, v_\ell]|$, and $|[v_j, p]||[v_k, v_\ell]| = 2|[v_j, v_\ell, v_k]|$, it follows that

$$w_{i,j} = \frac{1}{6} \sum_{\substack{\tau\in\mathcal{T}(\mathcal{M})\\ [v_i,v_j]\cup[v_k,v_\ell]\subset\tau\\ [v_i,v_j]\cap[v_k,v_\ell]=\varnothing}} \frac{4|[v_i, v_k, v_\ell]||[v_j, v_\ell, v_k]| \cos\theta_{i,j}^{k,\ell}}{6|\tau|},$$

which is as desired. ■

Similarly as [58, 73, 76] and based on Proposition 4.1, the cotangent formula (4.14) can be fairly changed according to the image of a mapping f by

$$w_{i,j}(f) = \frac{1}{9} \sum_{\substack{\tau\in\mathcal{T}(\mathcal{M})\\ [v_i,v_j]\cup[v_k,v_\ell]\subset\tau\\ [v_i,v_j]\cap[v_k,v_\ell]=\varnothing}} \frac{|f([v_i, v_k, v_\ell])||f([v_j, v_\ell, v_k])| \cos\theta_{i,j}^{k,\ell}(f)}{|f(\tau)|}, \tag{4.15}$$

where $\theta_{i,j}^{k,\ell}(f)$ is the dihedral angle between the triangular faces $f([v_i, v_k, v_\ell])$ and $f([v_j, v_\ell, v_k])$ in the tetrahedron $f(\tau)$. We now define the stretch factor of f on a tetrahedron $\tau \in \mathcal{T}(\mathcal{M})$ as

$$\sigma_{f^{-1}}(\tau) = \frac{|\tau|}{|f(\tau)|}. \tag{4.16}$$

In addition, for the purpose of the volume-preserving parameterization, the condition $|f(\tau)| = |\tau|$, for each $\tau \in \mathcal{T}(\mathcal{M})$, is required to be satisfied. We now impose the stretch factor $\sigma_{f^{-1}}(\tau)$ into $w_{i,j}(f)$ and derive a new formula for $w_{i,j}(f)$, which can be used to correct the stretch factor implicitly at each iteration step.

Since $|\tau| = \det(J_{f^{-1}}|_{f(\tau)})|f(\tau)| = \sigma_{f^{-1}}(\tau)|f(\tau)|$ (see Appendix A), from the condition of $|f(\tau)| = |\tau|$, the definition (4.15) is equivalent to

$$w_{i,j}(f) = \frac{1}{9} \sum_{\substack{\tau \in \mathcal{T}(\mathcal{M}) \\ [v_i,v_j]\cup[v_k,v_\ell]\subset\tau \\ [v_i,v_j]\cap[v_k,v_\ell]=\varnothing}} \frac{|f([v_i, v_k, v_\ell])||f([v_j, v_\ell, v_k])| \cos\theta_{i,j}^{k,\ell}(f)}{\sigma_{f^{-1}}(\tau)\,|f(\tau)|}, \tag{4.17}$$

where $\sigma_{f^{-1}}$ is the stretch factor as in (4.16).

The *volumetric stretch energy functional* on $\mathcal{M}$ is defined as

$$\mathcal{E}_{\mathbb{S}}(f) = \frac{1}{2}\text{trace}\left(\mathbb{f}^\top L_{\mathbb{S}}(f)\,\mathbb{f}\right), \tag{4.18}$$

where $L_{\mathbb{S}}(f)$ is the stretch volumetric Laplacian matrix with

$$[L_{\mathbb{S}}(f)]_{i,j} = \begin{cases} -w_{i,j}(f) & \text{if } [v_i, v_j] \in \mathcal{E}(\mathcal{M}), \\ \sum_{\ell\neq i} w_{i,\ell}(f) & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.19}$$

in which $w_{i,j}(f)$ is the modified weight given in (4.17).

Suppose an equiareal spherical boundary mapping $\mathbb{f}_{\mathtt{B}}^{(0)}$ is computed by Algorithm 4.3. Then, the volume-preserving parameterization is computed by minimizing the volumetric stretch energy (4.18) via the iteration

$$[L_{\mathbb{S}}(f^{(k)})]_{\mathtt{I},\mathtt{I}}\mathbb{f}_{\mathtt{I}}^{(k+1)} = -[L_{\mathbb{S}}(f^{(k)})]_{\mathtt{I},\mathtt{B}}\mathbb{f}_{\mathtt{B}}^{(0)} \tag{4.20}$$

for solving the sequential quadratic programmings, where the matrix $L_{\mathbb{S}}$ is defined in (4.19), $\mathtt{B} = \{s \,|\, v_s \in \partial\mathcal{M}\}$ and $\mathtt{I} = \{1, \ldots, n\}\backslash\mathtt{B}$.

The VSEM algorithm for the computation of volume-preserving parameterizations by (4.20) is summarized in Algorithm 4.4.

Remark. A mapping f is volume-preserving if and only if the stretch factor $\sigma_{f^{-1}}(\tau) = 1$ for every $\tau \in \mathcal{T}(\mathcal{M})$. As a result, f is a fix point of the iteration (4.20), i.e., a critical point of the energy functional $\mathcal{E}_{\mathbb{S}}$ in (4.18). However, it is not obvious the other way around since the energy functional (4.18) is highly nonlinear. Still, we expect the resulting mapping f to be volume-preserving because the condition $|\tau| = |f(\tau)|$ is imposed in (4.17) implicitly. So, it is reasonable to numerically check whether the resulting stretch factor $\sigma_{f^{-1}}(\tau)$ is very close to 1 for every $\tau \in \mathcal{T}(\mathcal{M})$. The distributions of $\sigma_{f^{-1}}$ for various benchmark mesh models are demonstrated later in Figure 6.4.

Algorithm 4.4. VSEM for volume-preserving parameterizations.

```
Input: A simply connected tetrahedral mesh $\mathcal{M}$, a tolerance $\varepsilon$ (e.g., $\varepsilon = 10^{-6}$).
Output: A volume-preserving parameterization $\mathbb{f}$.
Let $n$ be the number of vertices of $\mathcal{M}$.
Set $\mathtt{B} = \{s \,|\, v_s \in \partial\mathcal{M}\}$ and $\mathtt{I} = \{1, \ldots, n\}\backslash\mathtt{B}$.
Compute a spherical equiareal parameterization $\mathbb{g}_\mathtt{B}$ by Algorithm 4.3.
Compute $\mathbb{g}$ by solving the linear system

    $[L_\mathbb{D}]_{\mathtt{I},\mathtt{I}}\mathbb{g}_\mathtt{I} = -[L_\mathbb{D}]_{\mathtt{I},\mathtt{B}}\mathbb{g}_\mathtt{B}$,

  where $L_\mathbb{D}$ is defined as in (4.12).
Set $\delta \leftarrow \infty$.
Set $\mathbb{f}_\mathtt{B} \leftarrow \mathbb{g}_\mathtt{B}$.
while $\delta > \varepsilon$ do
  Update $A \leftarrow L_\mathbb{S}(\mathbb{g})$, where $L_\mathbb{S}(\mathbb{g})$ is defined as in (4.19).
  Update $\mathbb{f}$ by solving the linear system $A_{\mathtt{I},\mathtt{I}}\mathbb{f}_\mathtt{I} = -A_{\mathtt{I},\mathtt{B}}\mathbb{f}_\mathtt{B}$.
  Update $\delta \leftarrow \mathcal{E}_\mathbb{S}(\mathbb{g}) - \mathcal{E}_\mathbb{S}(\mathbb{f})$.
  Update $\mathbb{g} \leftarrow \mathbb{f}$.
end while
return The volumetric mapping $\mathbb{f}$.
```

5. Bijectivity of the parameterizations.

The bijectivity is one of the most important in the parameterization process. In the following, we first show that the spherical conformal parameterization of a genus-zero closed mesh produced by Algorithm 4.1 is bijective under the Delaunay assumption.

For convenience, we give the definition of an M-matrix [12] and a related lemma.

Definition 5.1. *A matrix $A \in \mathbb{R}^{m\times n}$ is nonnegative (positive) if all entries of A are nonnegative (positive). A squared matrix $A \in \mathbb{R}^{n\times n}$ is an M-matrix if $A = sI - B$ with B being nonnegative and $s \geq \rho(B)$, where $\rho(B)$ is the spectral radius of B.*

Lemma 5.2 (Theorems 1.4.7 and 1.4.10 in [52]). *Suppose $A \in \mathbb{R}^{n\times n}$ is a singular, irreducible M-matrix. Then each principal submatrix $\hat{A}$ of A other than A itself is a nonsingular M-matrix and $\hat{A}^{-1}$ is nonnegative.*

Theorem 5.3. *Given a genus-zero closed Delaunay triangular mesh $\mathcal{M}$ of n vertices, the spherical conformal parameterization of $\mathcal{M}$ produced by Algorithm 4.1 is bijective.*

Proof. The Delaunay property guarantees that the weights $w_{i,j}$ of L_D in (4.2) are positive. Let $D = \operatorname{diag}\left((\sum_{j=1}^n w_{1,j}, \ldots, \sum_{j=1}^n w_{n,j})^\top\right)$. Then

$$\left[D^{-1}L_D\mathbf{h}\right]_i = \sum_{v_j\in N(v_i)} \frac{w_{i,j}}{\sum_{j=1}^n w_{i,j}} \left(\mathbf{h}_j - \mathbf{h}_i\right),\ i = 1, \ldots, n,$$

where $N(v_i)$ is the set of neighboring vertices of v_i. Since $\sum_{v_j\in N(v_i)} \frac{w_{i,j}}{\sum_{j=1}^n w_{i,j}} = 1$, $D^{-1}L_D$ $\mathbf{h} = \mathbf{0}$ is equivalent to

$$\mathbf{h}_i = \sum_{v_j \in N(v_i)} \frac{w_{i,j}}{\sum_{j=1}^{n} w_{i,j}} \mathbf{h}_j, \ i = 1, \ldots, n. \tag{5.1}$$

It is known that L_D is a singular, irreducible M-matrix [75]. From Lemma 5.2, the matrix $[L_D]_{\mathtt{I},\mathtt{I}}$ in (4.7) is invertible and $-[L_D]_{\mathtt{I},\mathtt{I}}^{-1}[L_D]_{\mathtt{I},\mathtt{B}}$ is nonnegative. Therefore, from (5.1), the unique mapping $\mathbf{h}_{\mathtt{I}} = [L_D]_{\mathtt{I},\mathtt{I}}^{-1}[L_D]_{\mathtt{I},\mathtt{B}}\mathbf{h}_{\mathtt{B}}$ obtained by solving the linear system (4.7) is a convex combination mapping with the boundary $\mathbf{h}_{\mathtt{B}}$ in (4.6) being a triangle. From the result in [29], the interior mapping $\mathbf{h}_{\mathtt{I}}$ in (4.7) is bijective. Hence, the bijectivity of the spherical conformal parameterization $\mathbb{f}$ in step 6 of Algorithm 4.1 follows from the bijectivity of the inverse stereographic projection. ■

On the other hand, the bijectivity of the spherical conformal parameterization computed by the CEM algorithm, Algorithm 4.2, can be theoretically guaranteed if the boundary mapping $\mathbf{h}_{\mathtt{B}}$ in step 8 forms a convex polygon. Unfortunately, the polygon formed by $\mathbf{h}_{\mathtt{B}}$ is not necessarily convex, so the result in [29] cannot be applied. However, from the construction of the index set $\mathtt{B}$ in step 7 of Algorithm 4.2, the polygon formed by $\mathbf{h}_{\mathtt{B}}$ is *nearly convex*, i.e., there exists a circle $\gamma : \mathbb{S}^1 \to \mathbb{C}$ with radius r satisfying

$$\sup_{x \in \mathbb{S}^1, j \in \mathtt{B}} |\gamma(x) - \mathbf{h}_j| < \sup_{\substack{[v_i, v_j] \in \mathcal{E}(\mathcal{S}) \\ i \notin \mathtt{B}, j \in \mathtt{B}}} |\mathbf{h}_i - \mathbf{h}_j|. \tag{5.2}$$

It is reasonable to demonstrate the bijectivity of the spherical conformal parameterization via numerical checking of the resulting mapping computed by Algorithm 4.2. The method for checking the bijectivity of a discrete spherical mapping $f : \mathcal{S} \to \mathbb{S}^2$ can be found in Appendix B.1.

As we will see in Table C.2, the spherical conformal parameterization computed by the CEM algorithm, Algorithm 4.2, is bijective for each tested benchmark mesh model.

In addition, for the SEM algorithm, Algorithm 4.3, the resulting mapping seldom fails to be bijective, e.g., for the boundary mapping of the Bunny model,[2] the image of the resulting spherical mapping f contains three overlapped triangular faces. It is not surprising since the stretch cotangent weights in (4.9) are not necessarily positive so that the resulting mapping is not inevitably a convex combination mapping. To remedy this drawback, we unfold the overlapped triangular faces on the image of f by the procedures in Appendix B.2.

Furthermore, for the simply connected tetrahedral mesh with a single boundary, the convex combination property of the mapping follows similarly to Theorem 5.3.

Corollary 5.4. *Given a simply connected tetrahedral mesh $\mathcal{M}$ of n vertices with the boundary and interior indices being $\mathtt{B}$ and $\mathtt{I}$, respectively, suppose L is a Laplacian matrix as in (4.19) with positive weights and $f : \mathcal{M} \to \mathbb{R}^3$ is a piecewise affine mapping satisfying that the boundary mapping $\{\mathbb{f}_s := f(v_s) \mid s \in \mathtt{B}\}$ forms a convex polyhedron. Then $L_{\mathtt{I},\mathtt{I}}$ is invertible and the mapping*

$$\mathbb{f}_{\mathtt{I}} = -L_{\mathtt{I},\mathtt{I}}^{-1} L_{\mathtt{I},\mathtt{B}} \mathbb{f}_{\mathtt{B}} \tag{5.3}$$

is a convex combination mapping.

[2] Among all 17 tested surface mesh models, the Bunny model is the only one for which the resulting spherical mapping by the SEM algorithm, Algorithm 4.3, is not bijective.

Remark. The bijectivity of the volume-preserving parameterization of tetrahedral meshes, in general, is not guaranteed even if the mapping is a convex combination mapping with the boundary mapping being convex. An elegant counterexample has been given in [31].

6. Numerical experiments.

In this section, we demonstrate numerical experiments of the VSEM algorithm for volume-preserving parameterizations of simply connected tetrahedral meshes. All linear systems in our algorithms are solved by the backslash operator (\) in MATLAB. Some of surface and tetrahedral mesh models are obtained from TurboSquid [5], the AIM@SHAPE shape repository [2], the Stanford 3D scanning repository [4], a project page of ALICE [1], and Gu's website [32]. Some of tetrahedral mesh models are generated using *JIGSAW* mesh generators [23, 26, 24, 27, 25].

In order to better understand the shape of the mesh models and the image of the volume-preserving parameterizations computed by the VSEM algorithm, Algorithm 4.4, we first show the tetrahedral meshes of David Head and Human Brain and their volume-preserving parameterizations in Figures 6.1 and 6.2, respectively.

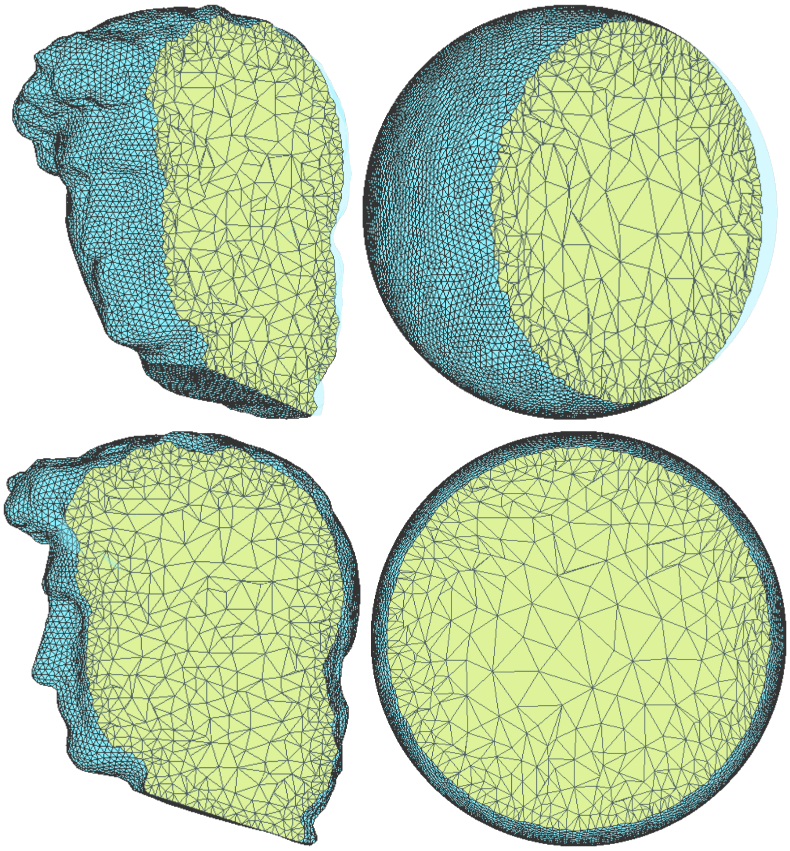


Figure 6.1. *The tetrahedral mesh of David Head (left) and its volume-preserving parameterization (right) by the VSEM algorithm, Algorithm 4.4.*

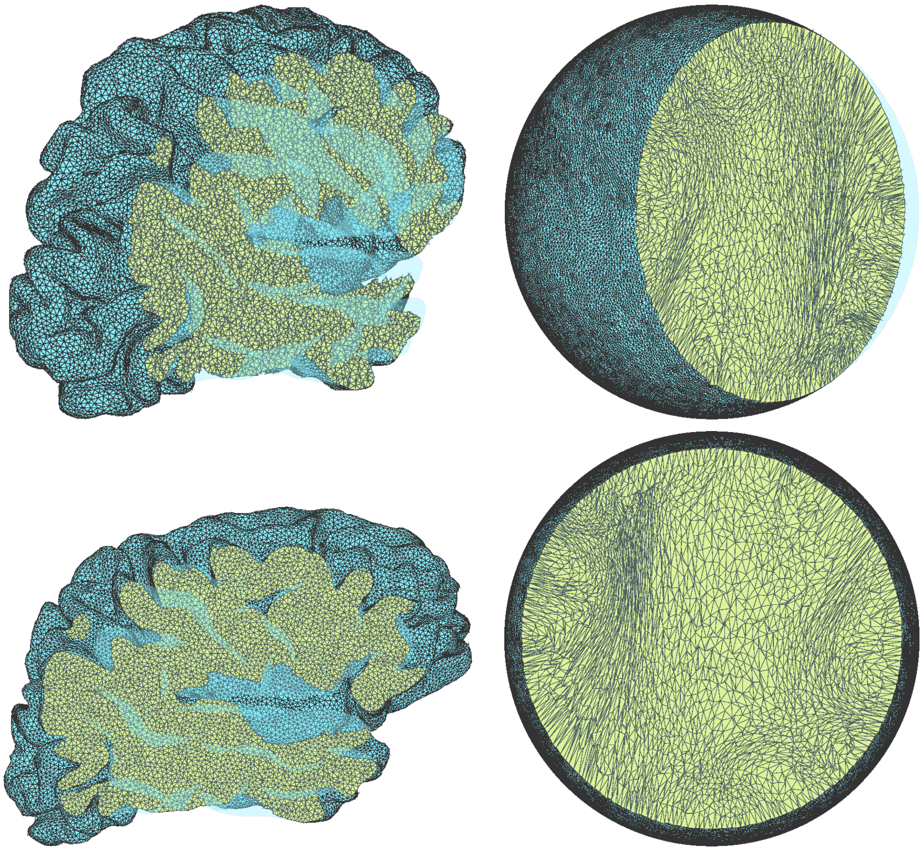


Figure 6.2. *The tetrahedral mesh of Human Brain (left) and its volume-preserving parameterization (right) by the VSEM algorithm, Algorithm 4.4.*

Then, we compare the effectiveness and accuracy of the VSEM algorithm, Algorithm 4.4 to the state-of-the-art OMT algorithm [67]. The executable program files of the OMT algorithm are obtained from Gu's website [32].

We now introduce the volume distortion to measure the accuracy of a volume-preserving parameterization by the *total volume distortion* as well as the mean and standard deviation (SD) of *local volume ratios*. The total volume distortion of a mapping f on a mesh $\mathcal{M}$ is defined as

$$\mathcal{D}_{\mathcal{M}}(f) = \frac{1}{4} \sum_{v \in \mathcal{V}(\mathcal{M})} \left| \frac{\sum_{\tau \in \mathcal{N}(v)} |\tau|}{|\mathcal{M}|} - \frac{\sum_{\tau \in \mathcal{N}(v)} |f(\tau)|}{|f(\mathcal{M})|} \right|, \tag{6.1}$$

where $\mathcal{N}(v) = \{\tau \in \mathcal{T}(\mathcal{M}) \,|\, v \subset \tau\}$ is the set of neighboring tetrahedrons of the vertex v, and $|\mathcal{M}|$ and $|f(\mathcal{M})|$ denote volumes of $\mathcal{M}$ and its image, respectively. A mapping f is volume-preserving if $\mathcal{D}_{\mathcal{M}}(f) = 0$. The local volume ratio $\mathcal{R}_f$ on a vertex v is defined as

$$\mathcal{R}_f(v) = \frac{\sum_{\tau \in \mathcal{N}(v)} |\tau| / |\mathcal{M}|}{\sum_{\tau \in \mathcal{N}(v)} |f(\tau)| / |f(\mathcal{M})|}. \tag{6.2}$$

Table 6.1

The total volume distortion (6.1) of volume-preserving parameterizations by OMT and the VSEM algorithm, Algorithm 4.4. Here "—" means the executable program file of OMT does not work.

Model	# Tetrahedrons	OMT [67]	VSEM
Bunny	84,787	0.0627	0.0322
Fandisk	88,374	0.0645	0.0506
David Head	90,575	0.0627	0.0142
Lion	97,131	0.0666	0.0262
Max Planck	115,767	0.0400	0.0086
Venus	130,429	0.0346	0.0052
Apple	169,888	0.0300	0.0030
Bimba	290,353	0.0500	0.0184
Human Brain	852,565	—	0.0790

Table 6.2

The mean and SD of local volume ratios of volume-preserving parameterizations by OMT and the VSEM algorithm, Algorithm 4.4. Here "—" means the executable program file of OMT does not work.

Model	# Tetrahedrons	OMT [67]		VSEM	
		Mean	SD	Mean	SD
Bunny	84,787	1.5015	42.5488	1.1044	1.4497
Fandisk	88,374	1.3520	6.3902	1.2468	2.8231
David Head	90,575	1.0621	0.6465	1.0323	0.6634
Lion	97,131	1.0417	3.1432	1.0246	0.1215
Max Planck	115,767	1.0568	1.6567	1.0235	0.7186
Venus	130,429	1.0183	0.4585	1.0066	0.3662
Apple	169,888	1.0048	0.0564	1.0002	0.0086
Bimba	290,353	1.0342	1.6458	1.0066	0.4764
Human Brain	852,565	—	—	1.0266	0.8850

The mean and SD of $\mathcal{R}_f(v)$, for all $v \in \mathcal{V}(\mathcal{M})$, are used to measure the local volume distortion. A mapping f is volume-preserving if the mean is 1 and the SD is 0.

In Tables 6.1 and 6.2 and Figure 6.3, we show the total volume distortion as well as the mean and SD of local volume ratios of volume-preserving parameterizations of various tested benchmark mesh models obtained by OMT and the VSEM algorithm, Algorithm 4.4, respectively. We see that VSEM has much smaller total volume distortions and local volume ratios than those of OMT. In addition, we show the execution time of OMT and VSEM in Table 6.3, which indicates that the effectiveness of VSEM outperforms the OMT by several hundreds of times.

Furthermore, in Figure 6.4, we show histograms of the stretch factors $\sigma_{f^{-1}}(\tau)$, for $\tau \in \mathcal{T}(\mathcal{M})$, of volume-preserving parameterizations f computed by the VSEM algorithm, Algorithm 4.4. It indicates that the stretch factors of most tetrahedrons are closed to 1, as we desired.

Recall that the bijectivity of convex combination mappings on 3-manifolds is, in general, not guaranteed [31]. It is reasonable to numerically check the bijectivity of volume-preserving parameterizations computed by the VSEM algorithm, Algorithm 4.4. The method for checking the bijectivity of a volumetric mapping can be found in Appendix B.3. In Table 6.4, we show the percentages of bijective tetrahedrons of volume-preserving parameterizations by OMT and

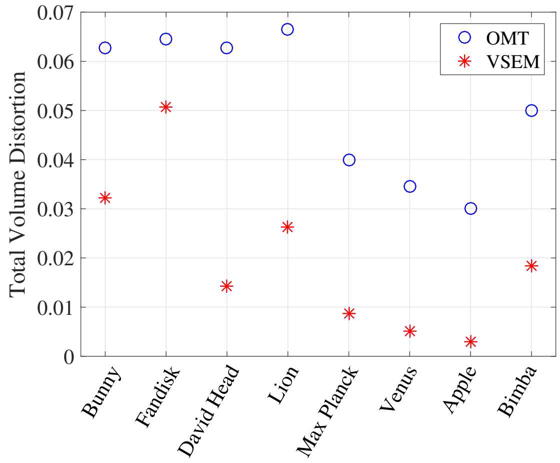


Figure 6.3. *The total volume distortion* (6.1) *of volume-preserving parameterizations by OMT and the VSEM algorithm,* Algorithm 4.4.

Table 6.3

The computational cost (second) of volume-preserving parameterizations by OMT and the VSEM algorithm, Algorithm 4.4. *Here "—" means the executable program file of OMT does not work.*

Model	# Tetrahedrons	OMT [67]		VSEM	
		Time	#Iter.	Time	#Iter.
Bunny	84,787	24438.40	231	6.26	10
Fandisk	88,374	8542.10	61	6.35	10
David Head	90,575	6911.32	24	6.13	10
Lion	97,131	13716.90	50	6.95	10
Max Planck	115,767	6679.72	47	9.81	10
Venus	130,429	5558.35	40	13.35	10
Apple	169,888	5599.84	37	20.08	10
Bimba	290,353	54856.90	193	33.09	10
Human Brain	852,565	—	—	157.79	10

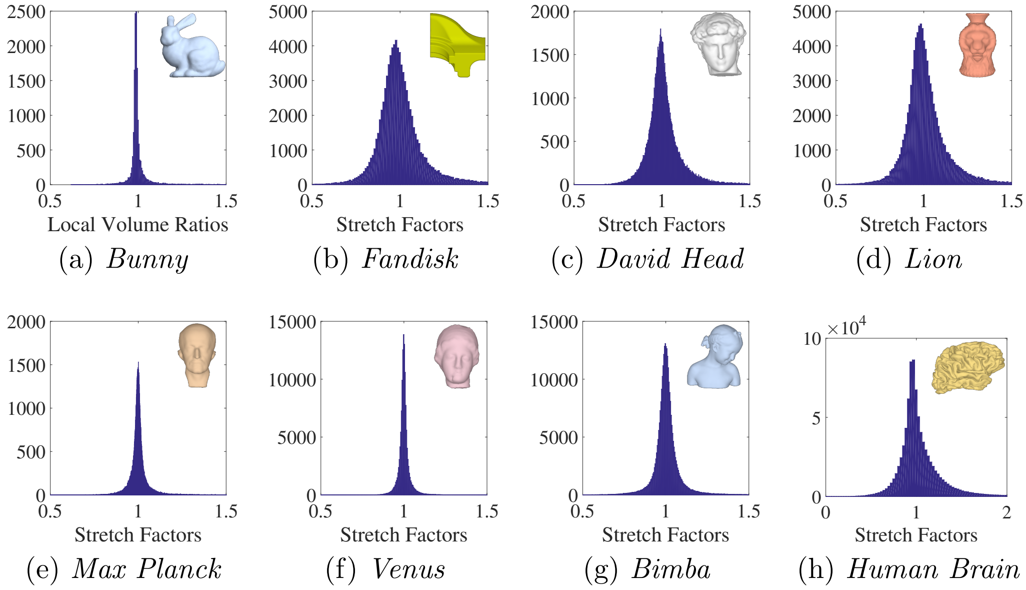
VSEM, respectively. We observe that most of tetrahedrons are mapped bijectively by OMT and VSEM, and the bijectivity of VSEM is much better than that of OMT.

Remark. Thanks to the *large-scale bounded distortion mapping* [3, 49], for each tested tetrahedral volumetric mesh model listed in Table 6.4, the overlapped tetrahedrons can be unfolded into a 100% bijective volume-preserving parameterization, slightly sacrificing the total volume distortion and the spherical boundary.

In summary, the proposed VSEM algorithm, Algorithm 4.4 has better accuracy and effectiveness than OMT that would cost less than 10 seconds for computing volume-preserving parameterizations of meshes of 100,000 tetrahedrons. It should be quite satisfactory in practical applications.

7. Applications.

In this section, we present sample applications of volume-preserving parameterizations for 3-manifolds, namely, the manifold partition and the mesh processing for 3D printing.



(a) *Bunny* (b) *Fandisk* (c) *David Head* (d) *Lion*

(e) *Max Planck* (f) *Venus* (g) *Bimba* (h) *Human Brain*

Figure 6.4. *Histograms of stretch factors $\sigma_{f^{-1}}(\tau)$ of volume-preserving parameterizations obtained by VSEM.*

Table 6.4

The percentages of bijective tetrahedrons for volume-preserving parameterizations by OMT and VSEM. Here "—" means the executable program file of OMT does not work.

Model	# Tetrahedrons	OMT [67]	VSEM
Bunny	84,787	99.19%	99.90%
Fandisk	88,374	98.79%	99.94%
David Head	90,575	99.90%	99.99%
Lion	97,131	99.58%	99.97%
Max Planck	115,767	99.91%	99.99%
Venus	130,429	99.96%	99.99%
Apple	169,888	99.96%	100.00%
Bimba	290,353	99.29%	99.96%
Human Brain	852,565	—	99.81%

7.1. Manifold partitions. The aim of the manifold partition problem is to find an optimal partition of a manifold in terms of minimizing a certain energy functional [14, 13]. In the previous works, the considered manifolds are usually of simple shapes, e.g., a disk, a sphere, a solid ball, and a cube. It is worth noting that the optimal partition usually has the property that each part has a similar volume. With aid of the VSEM algorithm, Algorithm 4.4, for volume-preserving parameterizations, a manifold can be easily partitioned into several simply connected submanifolds with a similar volume that could be good initially for solving the partition problem [14, 13].

First, a 3-manifold $\mathcal{M}$ is mapped into a unit solid ball $\mathbb{B}^3$ by f using VSEM. Then, the partition of $\mathcal{M}$ can be achieved by slicing $\mathbb{B}^3$ with suitable cutting planes. The uniform sampling of $\mathbb{B}^3$ can be controlled by using the spherical coordinate system, and the tetrahedral

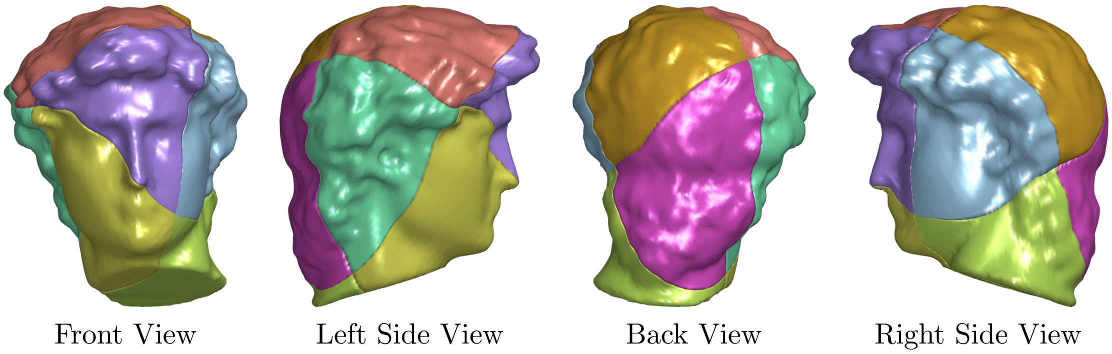
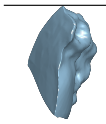
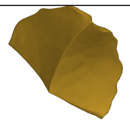
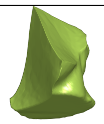
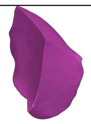
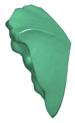
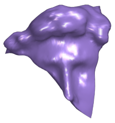
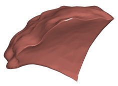


Figure 7.1. *The partition of the David Head model in four views.*

Table 7.1
The partition of the David Head model and their volumes. The total volume is normalized to be 1.

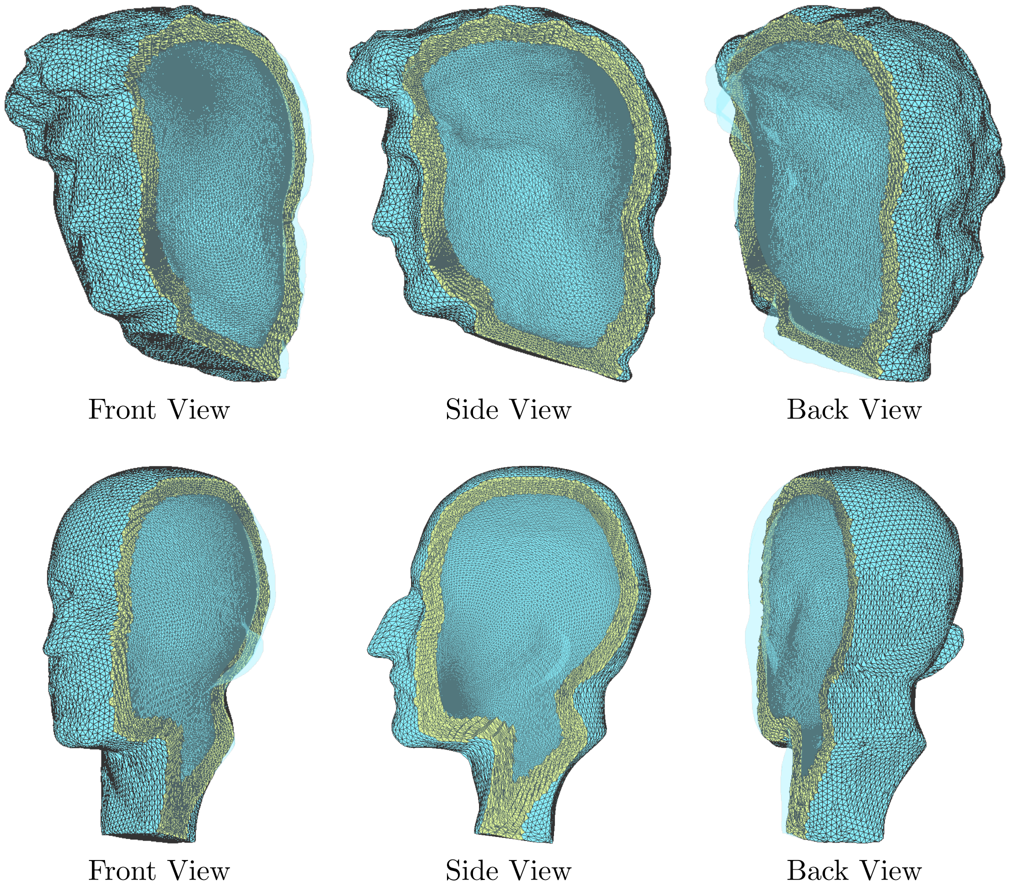
Part	Volume	Part	Volume	Part	Volume	Part	Volume
	0.1249		0.1247		0.1257		0.1247
	0.1248		0.1250		0.1255		0.1247
SD of volumes: 2.7227×10^{-4}							

mesh can be constructed by the Delaunay triangulation algorithm (`delaunayTriangulation`) in MATLAB due to the fact that $\mathbb{B}^3$ is convex.

Figure 7.1 shows the equal-volume partition of the David Head model $\mathcal{M}$. The cutting planes are chosen to be $x = 0$, $y = 0$, and $z = 0$ so that the uniform tetrahedral mesh of a ball $\mathbb{B}^3$ is partitioned into eight equal-volume partitions. Then, each partition of $\mathbb{B}^3$ is mapped to $\mathcal{M}$ via f^{-1} and the corresponding volume is shown in Table 7.1. Note that, here, the volume of $\mathcal{M}$ is normalized to be 1. From Table 7.1, we see that each partition of the manifold has a volume around $1/8$ with SD less than 2.73×10^{-4}, which is quite satisfactory.

7.2. Mesh processing for 3D printing. Nowadays, 3D printing is a popular topic. Issues arising from data processing for 3D printing are getting much attention. In order to save printing material, it is common to hollow out the interior of the model when a solid object is printed. With aid of volume-preserving parameterizations, the thickness of the model can be easily controlled. First, a 3-manifold $\mathcal{M}$ is mapped into the ball $\mathbb{B}^3$ with VSEM. Then, the uniform tetrahedral mesh of $\mathbb{B}^3$ is constructed by the spherical coordinate system as before.

In Figure 7.2, we show the hollowed David Head and Max Planck models in three views. The capacity of the interior is roughly 63% of the volume of the original solid manifold. From Figure 7.2, we see that the thickness of each hollowed model is almost equally distributed with a smooth interior surface.



Front View Side View Back View

Front View Side View Back View

Figure 7.2. *The hollowed David Head and Max Planck models in three views.*

8. Concluding remarks.

In this paper, we develop a novel algorithm by minimizing the volumetric stretch energy, which can be used to compute volume-preserving parameterizations for simply connected tetrahedral meshes with a single boundary. In addition, we generalize the CEM [75] and SEM [76] algorithms to compute the spherical conformal and equiareal parameterizations of genus-zero closed triangular meshes, respectively. Numerical experiments indicate that the proposed VSEM algorithm for volume-preserving parameterizations outperforms the state-of-the-art OMT algorithm [67] with better effectiveness and accuracy. Applications of the manifold partition and the mesh processing for 3D printing are demonstrated to show the practicality of our algorithm.

Appendix A. Equivalence of the volumetric stretch factor.

Theorem A.1. *Let $f : \mathcal{M} \to \mathbb{R}^3$ be a piecewise affine mapping on the tetrahedral mesh $\mathcal{M}$ and $J_{f^{-1}} = [\, \frac{\partial f^{-1}}{\partial u^1} \; \frac{\partial f^{-1}}{\partial u^2} \; \frac{\partial f^{-1}}{\partial u^3} \,]$. Suppose $\sigma_{f^{-1}}$ is the volumetric stretch factor defined as in* (4.16). *Then*

$$\sigma_{f^{-1}}(\tau) = \det\left(J_{f^{-1}}|_{f(\tau)}\right)$$

for every $\tau \in \mathcal{T}(\mathcal{M})$.

Proof. Let $\mathbb{f}_s = f(v_s) \equiv (\mathbb{f}_s^1, \mathbb{f}_s^2, \mathbb{f}_s^3)^\top$ as in (3.1). Recalling the tetrahedron $\tau = [v_i, v_j, v_k, v_\ell]$, we have

$$f|_\tau(v) = \frac{|[v, v_j, v_k, v_\ell]|\mathbb{f}_i + |[v_i, v, v_k, v_\ell]|\mathbb{f}_j + |[v_i, v_j, v, v_\ell]|\mathbb{f}_k + |[v_i, v_j, v_k, v]|\mathbb{f}_\ell}{|[v_i, v_j, v_k, v_\ell]|}.$$

The inverse mapping via the barycentric coordinate is written as

$$f^{-1}|_{f(\tau)}(u) = \frac{|[u, \mathbb{f}_j, \mathbb{f}_k, \mathbb{f}_\ell]|v_i + |[\mathbb{f}_i, u, \mathbb{f}_k, \mathbb{f}_\ell]|v_j + |[\mathbb{f}_i, \mathbb{f}_j, u, \mathbb{f}_\ell]|v_k + |[\mathbb{f}_i, \mathbb{f}_j, \mathbb{f}_k, u]|v_\ell}{|[\mathbb{f}_i, \mathbb{f}_j, \mathbb{f}_k, \mathbb{f}_\ell]|}.$$

Note that the derivatives are translation invariant. Without loss of generality, we assume that $v_\ell = 0$ and $\mathbb{f}_\ell = 0$. Then the partial derivatives of f^{-1} are written as

$$\left.\frac{\partial f^{-1}}{\partial u^\alpha}\right|_{f(\tau)} = \frac{(\mathbb{f}_j^{\alpha+1}\mathbb{f}_k^{\alpha+2} - \mathbb{f}_j^{\alpha+2}\mathbb{f}_k^{t+1})v_i + (\mathbb{f}_k^{\alpha+1}\mathbb{f}_i^{\alpha+2} - \mathbb{f}_k^{\alpha+2}\mathbb{f}_i^{t+1})v_j + (\mathbb{f}_i^{\alpha+1}\mathbb{f}_j^{\alpha+2} - \mathbb{f}_j^{\alpha+2}\mathbb{f}_i^{t+1})v_k}{6|[\mathbb{f}_i, \mathbb{f}_j, \mathbb{f}_k, \mathbb{f}_\ell]|}$$

for $\alpha = 1, 2, 3$ and $\alpha + 1, \alpha + 2 \in \{1, 2, 3\}$ are of module 3. A direct computation yields that

$$\det(J_{f^{-1}}|_{f(\tau)}) = \frac{v_i^3(v_j^1 v_k^2 - v_j^2 v_k^1) - v_i^2(v_j^1 v_k^3 - v_j^3 v_k^1) + v_i^1(v_j^2 v_k^3 - v_j^3 v_k^2)}{\mathbb{f}_i^3(\mathbb{f}_j^1 \mathbb{f}_k^2 - \mathbb{f}_j^2 \mathbb{f}_k^1) - \mathbb{f}_i^2(\mathbb{f}_j^1 \mathbb{f}_k^3 - \mathbb{f}_j^3 \mathbb{f}_k^1) + \mathbb{f}_i^1(\mathbb{f}_j^2 \mathbb{f}_k^3 - \mathbb{f}_j^3 \mathbb{f}_k^2)} = \sigma_{f^{-1}}(\tau). \qquad \blacksquare$$

Appendix B. Bijectivity checking and correction.

B.1. Bijectivity checking for spherical parameterizations. The bijectivity of a discrete spherical mapping $f : \mathcal{S} \to \mathbb{S}^2$ can be checked as follows:

1. Compute the face normal $\mathbb{n} : \mathcal{F}(\mathcal{S}) \to \mathbb{R}^3$ of each triangle $\tau = [v_1, v_2, v_3]$ on the image of f by

$$\mathbb{n}(\tau) = \frac{(f(v_2) - f(v_1)) \times (f(v_3) - f(v_1))}{\|(f(v_2) - f(v_1)) \times (f(v_3) - f(v_1))\|}. \tag{B.1}$$

2. Compute the face center $\mathbb{c} : \mathcal{F}(\mathcal{S}) \to \mathbb{R}^3$ of each triangle $\tau = [v_1, v_2, v_3]$ on the image of f by $\mathbb{c}(\tau) = \frac{1}{3}\sum_{j=1}^3 f(v_j)$.
3. Compute the inner product $\mathbb{n}(\tau) \cdot \mathbb{c}(\tau)$ for every $\tau \in \mathcal{F}(\mathcal{S})$.
4. Compute the number of overlapped faces $|\{\tau \in \mathcal{F}(\mathcal{S}) \mid \mathbb{n}(\tau) \cdot \mathbb{c}(\tau) \leq 0\}|$.

When the number of overlapped faces is zero, the discrete spherical mapping f is bijective.

B.2. Bijectivity correction for spherical parameterizations. The overlapped triangular faces on the image of a spherical mapping f can be unfolded by the following procedures:

1. Construct the Laplacian matrix L_M with the mean value weight [28]

$$w_{i,j} = \frac{\tan\frac{\varphi_{i,j}^1(f)}{2} + \tan\frac{\varphi_{i,j}^2(f)}{2}}{\|v_j - v_i\|}, \tag{B.2}$$

where $\varphi_{i,j}^1(f)$ and $\varphi_{i,j}^2(f)$ are the angles $\angle\mathbb{f}_\ell\mathbb{f}_i\mathbb{f}_j$ and $\angle\mathbb{f}_j\mathbb{f}_i\mathbb{f}_k$, respectively, as illustrated in Figure B.1.
2. For each overlapped triangular face $f(\tau)$, we update the coordinates of the vertices in $f(\tau)$ by solving the linear system $[L_M]_{\mathtt{I},\mathtt{I}}\mathbb{f}_{\mathtt{I}} = -[L_M]_{\mathtt{I},\mathtt{B}}\mathbb{f}_{\mathtt{B}}$, where $\mathtt{I}$ is the index set of the three vertices in $f(\tau)$ and $\mathtt{B}$ is the index set of the remaining vertices.

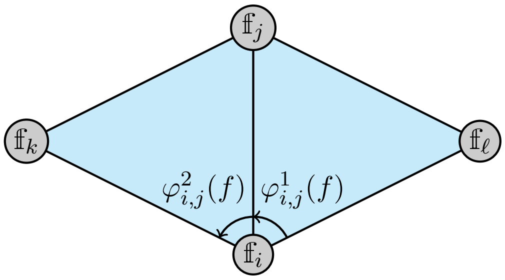


Figure B.1. *An illustration for the mean value weight.*

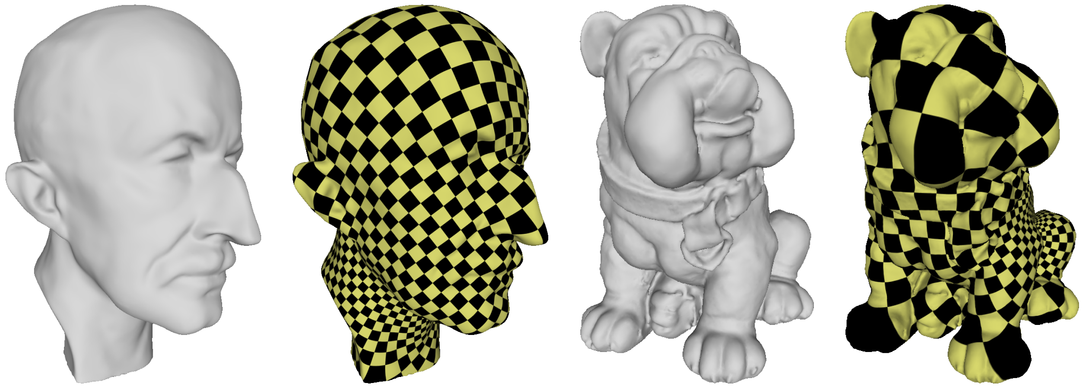
3. The vertices in $f(\tau)$ are projected to the unit sphere so that the resulting mapping is still a spherical mapping.

The above procedures are consecutively performed until there is no overlapped triangular face. It is worth noting that the mean value weight (B.2) is always positive so that the updated mapping would be a convex combination mapping.

B.3. Bijectivity checking for volumetric parameterizations. The bijectivity of a volumetric tetrahedral mesh mapping $f : \mathcal{M} \to \mathbb{B}^3$ can be checked as follows:

1. Compute the face normals $\mathbb{n}_k : \mathcal{T}(\mathcal{M}) \to \mathbb{R}^3$, $k = 1, \ldots, 4$, for each face of the tetrahedron $\tau \equiv [v_1, v_2, v_3, v_4]$ on the image of f by
$$\begin{aligned} \mathbb{n}_1(\tau) &= \mathbb{n}([v_2, v_3, v_4]), & \mathbb{n}_2(\tau) &= \mathbb{n}([v_1, v_4, v_3]), \\ \mathbb{n}_3(\tau) &= \mathbb{n}([v_1, v_2, v_4]), & \mathbb{n}_4(\tau) &= \mathbb{n}([v_1, v_3, v_2]), \end{aligned}$$
where $\mathbb{n}$ is the face normal defined as (B.1).
2. Compute the face centers $\mathbb{c}_k : \mathcal{T}(\mathcal{M}) \to \mathbb{R}^3$, for each face of the tetrahedron $\tau \equiv [v_1, v_2, v_3, v_4]$ on the image of f by
$$\mathbb{c}_k(\tau) = \frac{1}{3} \sum_{\substack{j=1 \\ j \neq k}}^{4} f(v_j)$$
for $k = 1, \ldots, 4$.
3. Compute the tetrahedron center $\mathbb{t} : \mathcal{T}(\mathcal{M}) \to \mathbb{R}^3$ of each tetrahedron $\tau \equiv [v_1, v_2, v_3, v_4]$ on the image of f by $\mathbb{t}(\tau) = \frac{1}{4} \sum_{j=1}^{4} f(v_j)$.
4. Compute the inner products $\mathbb{n}_k(\tau) \cdot (\mathbb{c}_k(\tau) - \mathbb{t}(\tau))$, $k = 1, \ldots, 4$, for every $\tau \in \mathcal{T}(\mathcal{M})$.
5. Compute the number of overlapped tetrahedrons
$$\left| \{ \tau \in \mathcal{T}(\mathcal{M}) \,|\, \mathbb{n}_k(\tau) \cdot (\mathbb{c}_k(\tau) - \mathbb{t}(\tau)) \leq 0 \text{ for some } k \in \{1, \ldots, 4\} \} \right| .$$

Appendix C. Numerical comparison for spherical conformal parameterizations. In this section, we demonstrate numerical results of spherical conformal parameterizations computed by the modified CEM algorithm, Algorithm 4.2. Also, we compare the effectiveness and accuracy of Algorithm 4.2 to other state-of-the-art algorithms, namely, the quasi-implicit Euler method (QIEM) for heat diffusion [43] and the FLASH algorithm [18].



(a) *Max Planck* (b) *Bulldog*

Figure C.1. *The mesh models of* (a) *Max Planck and* (b) *Bulldog as well as their spherical conformal parameterizations by the CEM Algorithm* 4.2*.*

Table C.1

The angular distortion (degree) of the spherical conformal parameterizations by QIEM, FLASH, and the CEM algorithm, Algorithm 4.2*, with maximal number of iterations being* 5*.*

		QIEM [43]		FLASH [18]		CEM	
Model	# Faces	Mean	SD	Mean	SD	Mean	SD
Heart	15,358	0.9713	1.6089	0.8646	1.5991	0.7988	1.6044
Venus	28,602	1.0045	0.8582	0.7650	0.7282	0.6947	0.6990
Arnold	29,056	1.2167	1.1614	0.9338	0.9428	0.9163	0.9359
Apple	35,674	0.2792	0.2464	0.2231	0.2306	0.2186	0.2277
Bulldog	99,590	1.0822	1.7948	1.0654	1.7860	1.0298	1.7906
Lion Statue	100,000	1.4776	1.4933	1.3638	1.2240	1.2897	1.1908
Max Planck	102,212	0.4114	0.5271	0.4272	0.5157	0.4079	0.5117
Bunny	111,364	2.5941	8.6253	0.6618	0.8314	0.6621	0.8306

The MATLAB codes of QIEM are reproduced by the authors. The MATLAB codes of FLASH (`spherical_conformal_map`) are obtained from the MATLAB File Exchange website [6].

In Figure C.1, we show the genus-zero mesh models of Max Planck and Bulldog and their spherical conformal parameterizations by the CEM algorithm. In Table C.1 and Figure C.2, we show the mean and SD of angular distortions of parameterizations produced by QIEM, FLASH, and Algorithm 4.2 for spherical conformal parameterizations, respectively. From Figure C.2, we observe that the mean and SD of the CEM algorithm for most of the tested models are slightly smaller than those of QIEM and FLASH. For the model of Bunny, QIEM would produce relatively larger mean and SD of angular distortions.

On the other hand, Table C.2 and Figure C.3 show the execution time of QIEM, FLASH, and the CEM algorithm, Algorithm 4.2, respectively. From Table C.2, we observe that for the mesh models with around 100,000 faces, Algorithm 4.2 would cost roughly 1 second for the computation of a spherical conformal parameterization, while QIEM would cost more than 20 seconds.

In summary, Algorithm 4.2 has a similar effectiveness and accuracy to FLASH but outperforms the QIEM.

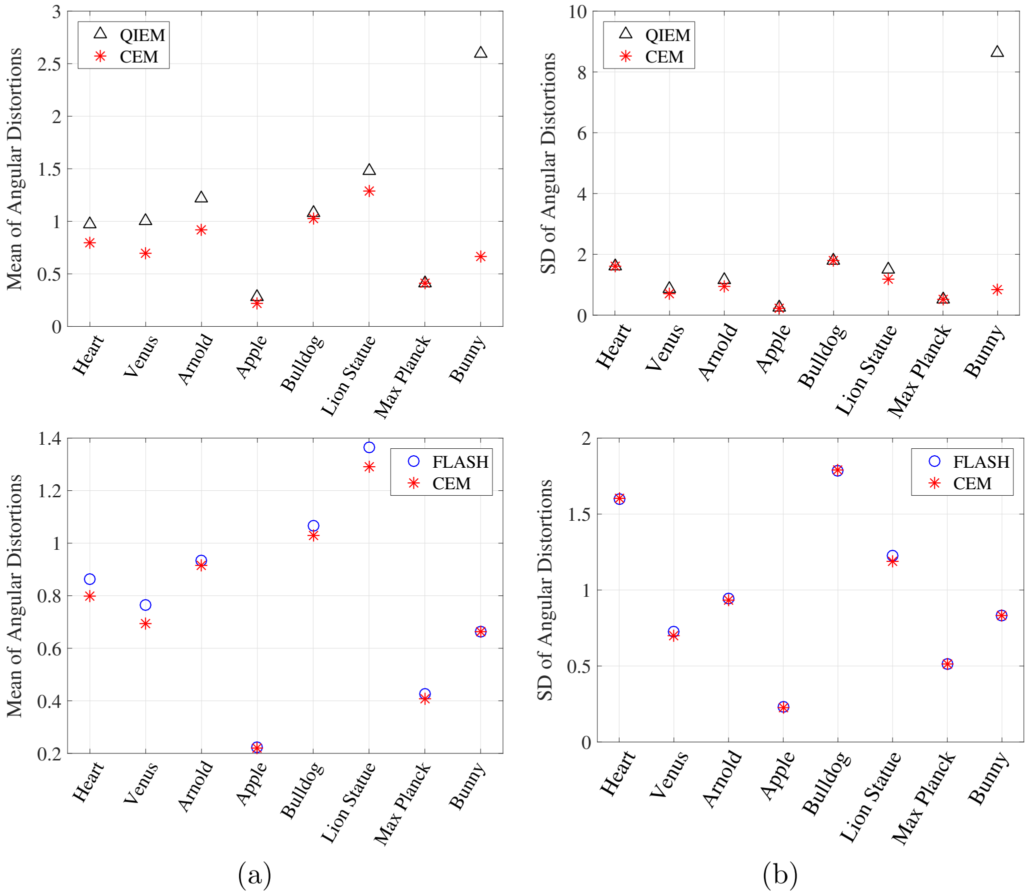


Figure C.2. *The* (a) *mean and* (b) *SD of angular distortions of the conformal parameterizations by QIEM, FLASH, and the CEM algorithm, Algorithm* 4.2, *with maximal number of iterations being* 5.

Table C.2

The computational cost (second) of the spherical conformal parameterizations by QIEM, FLASH, and the CEM algorithm, Algorithm 4.2, *with maximal number of iterations being* 5.

Model	# Faces	QIEM [43]	FLASH [18]	CEM		
		Time	Time	Time	#Iter.	#Overlap
Heart	15,358	0.45	0.21	0.20	5	0
Venus	28,602	5.97	0.31	0.25	5	0
Arnold	29,056	8.26	0.31	0.22	5	0
Apple	35,674	1.34	0.40	0.34	5	0
Bulldog	99,590	29.28	1.15	0.94	5	0
Lion Statue	100,000	34.11	1.22	1.04	5	0
Max Planck	102,212	31.78	1.21	0.93	5	0
Bunny	111,364	24.27	1.54	1.24	5	0

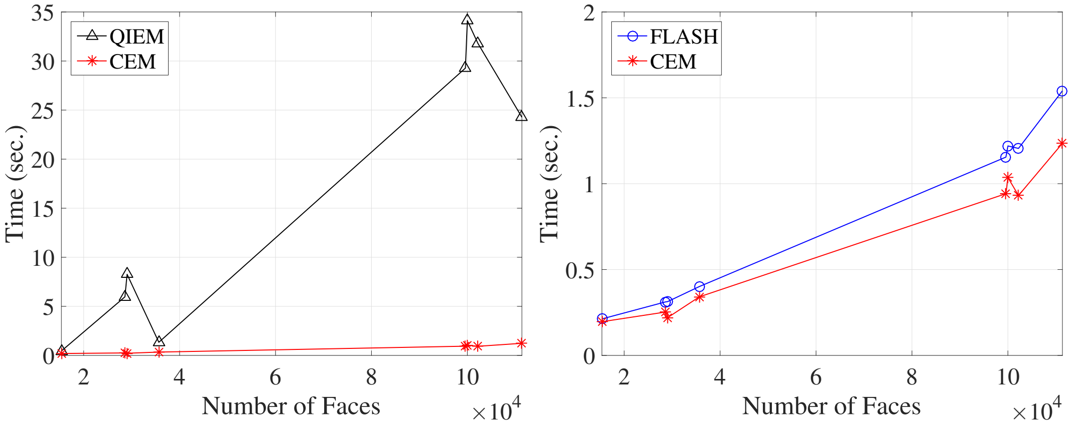


Figure C.3. *Computational cost (second) vs number of faces by QIEM, FLASH, and the CEM algorithm, Algorithm* 4.2*, with maximal number of iterations being* 5*.*

Acknowledgments. The authors want to thank Prof. Xianfeng David Gu, Prof. Lok Ming Ronald Lui, and Gary Pui-Tung Choi for useful discussions and the executable program files of the OMT and the FLASH algorithms, respectively.

REFERENCES

[1] *ALICE*, http://alice.loria.fr (2016).

[2] *Digital Shape Workbench Shape Repository*, http://visionair.ge.imati.cnr.it/ontologies/shapes (2016).

[3] *Large-Scale Bounded Distortion Mappings*, https://shaharkov.github.io/LargeScaleBD.html (2018).

[4] *The Stanford* 3*D Scanning Repository*, http://graphics.stanford.edu/data/3Dscanrep (2016).

[5] *TurboSquid*, http://www.turbosquid.com (2016).

[6] *Spherical Conformal Map, File Exchange, MATLAB Central*, https://www.mathworks.com/matlabcentral/fileexchange/65551-spherical-conformal-map (2018).

[7] N. Aigerman, S. Z. Kovalsky, and Y. Lipman, *Spherical orbifold Tutte embeddings*, ACM Trans. Graphics, 36 (2017), 90, https://doi.org/10.1145/3072959.3073615.

[8] N. Aigerman and Y. Lipman, *Orbifold Tutte embeddings*, ACM Trans. Graphics, 34 (2015), 190, https://doi.org/10.1145/2816795.2818099.

[9] N. Aigerman and Y. Lipman, *Hyperbolic orbifold Tutte embeddings*, ACM Trans. Graphics, 35 (2016), 217, https://doi.org/10.1145/2980179.2982412.

[10] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene, *Recent advances in remeshing of surfaces*, in Shape Analysis and Structuring, Math. Vis., Springer, Berlin, 2008, pp. 53–82, https://doi.org/10.1007/978-3-540-33265-7_2.

[11] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis, *On the Laplace-Beltrami operator and brain surface flattening*, IEEE Trans. Med. Imaging, 18 (1999), pp. 700–711, https://doi.org/10.1109/42.796283.

[12] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Appl. Math. 9, SIAM, Philadelphia, 1994, https://doi.org/10.1137/1.9781611971262.

[13] B. Bogosel, *Efficient algorithm for optimizing spectral partitions*, Appl. Math. Comput., 333 (2018), pp. 61–75, https://doi.org/10.1016/j.amc.2018.03.087.

[14] B. Bourdin, D. Bucur, and E. Oudet, *Optimal partitions for eigenvalues*, SIAM J. Sci. Comput., 31 (2010), pp. 4100–4114, https://doi.org/10.1137/090747087.

[15] A. Chern, U. Pinkall, and P. Schröder, *Close-to-conformal deformations of volumes*, ACM Trans. Graphics, 34 (2015), 56, https://doi.org/10.1145/2766916.

[16] G. P.-T. Choi and L. M. Lui, *A linear formulation for disk conformal parameterization of simply-connected open surfaces*, Adv. Comput. Math., 44 (2018), pp. 87–114, https://doi.org/10.1007/s10444-017-9536-x.

[17] G. P. T. Choi and C. H. Rycroft, *Density-equalizing maps for simply connected open surfaces*, SIAM J. Imaging Sci., 11 (2018), pp. 1134–1178, https://doi.org/10.1137/17M1124796.

[18] P. T. Choi, K. C. Lam, and L. M. Lui, *FLASH: Fast landmark aligned spherical harmonic parameterization for genus-*0 *closed brain surfaces*, SIAM J. Imaging Sci., 8 (2015), pp. 67–94, https://doi.org/10.1137/130950008.

[19] P. T. Choi and L. M. Lui, *Fast disk conformal parameterization of simply-connected open surfaces*, J. Sci. Comput., 65 (2015), pp. 1065–1090, https://doi.org/10.1007/s10915-015-9998-2.

[20] P. Degener, J. Meseth, and R. Klein, *An adaptable surface parameterization method*, in Proceedings of the 12th International Meshing Roundtable, 2003, pp. 201–213.

[21] M. Desbrun, M. Meyer, and P. Alliez, *Intrinsic parameterizations of surface meshes*, Comput. Graph. Forum, 21 (2002), pp. 209–218.

[22] R. J. Duffin, *Distributed and lumped networks*, J. Math. Mech., 8 (1959), pp. 793–826, http://www.jstor.org/stable/24900689.

[23] D. Engwirda, *Locally Optimal Delaunay-Refinement and Optimisation-Based Mesh Generation*, Ph.D. thesis, University of Sydney, 2014, http://hdl.handle.net/2123/13148.

[24] D. Engwirda, *Voronoi-based point-placement for three-dimensional delaunay-refinement*, Procedia Engineering, 124 (2015), pp. 330–342, https://doi.org/10.1016/j.proeng.2015.10.143.

[25] D. Engwirda, *Conforming restricted Delaunay mesh generation for piecewise smooth complexes*, Procedia Engineering, 163 (2016), pp. 84–96, https://doi.org/10.1016/j.proeng.2016.11.024.

[26] D. Engwirda and D. Ivers, *Face-centred Voronoi refinement for surface mesh generation*, Procedia Engineering, 82 (2014), pp. 8–20, https://doi.org/10.1016/j.proeng.2014.10.364.

[27] D. Engwirda and D. Ivers, *Off-centre Steiner points for Delaunay-refinement on curved surfaces*, Comput.-Aided Design, 72 (2016), pp. 157–171, https://doi.org/10.1016/j.cad.2015.10.007.

[28] M. S. Floater, *Mean value coordinates*, Comput. Aided Geom. Design, 20 (2003), pp. 19–27, https://doi.org/10.1016/S0167-8396(03)00002-5.

[29] M. S. Floater, *One-to-one piecewise linear mappings over triangulations*, Math. Comput., 72 (2003), pp. 685–696, http://www.jstor.org/stable/4099926.

[30] M. S. Floater and K. Hormann, *Surface parameterization: A tutorial and survey*, in Advances in Multiresolution for Geometric Modelling, Math. Vis., Springer, Berlin, 2005, pp. 157–186, https://doi.org/10.1007/3-540-26808-1_9.

[31] M. S. Floater and V. Pham-Trong, *Convex combination maps over triangulations, tilings, and tetrahedral meshes*, Adv. Comput. Math., 25 (2006), pp. 347–356, https://doi.org/10.1007/s10444-004-7620-5.

[32] D. X. Gu, *Home Page*, http://www3.cs.stonybrook.edu/~gu (2017).

[33] D. X. Gu, F. Luo, and S.-T. Yau, *Fundamentals of computational conformal geometry*, Math. Comput. Sci., 4 (2010), pp. 389–429, https://doi.org/10.1007/s11786-011-0065-6.

[34] X. Gu, F. Luo, J. Sun, and S.-T. Yau, *Variational principles for Minkowski type problems, discrete optimal transport, and discrete Monge-Ampère equations*, Asian J. Math., 20 (2016), pp. 383–398, https://doi.org/10.4310/AJM.2016.v20.n2.a7.

[35] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, *Genus zero surface conformal mapping and its application to brain surface mapping*, IEEE Trans. Med. Imaging, 8 (2004), pp. 949–958, https://doi.org/10.1109/TMI.2004.831226.

[36] X. Gu and S.-T. Yau, *Computing conformal structures of surfaces*, Commun. Inf. Syst., 2 (2002), pp. 121–146, https://doi.org/10.4310/CIS.2002.v2.n2.a2.

[37] X. Gu and S.-T. Yau, *Global conformal surface parameterization*, in Proceedings of Eurographics Symposium on Geometry Processing, L. Kobbelt, P. Schroeder, and H. Hoppe, eds., 2003, pp. 127–137, http://dl.acm.org/citation.cfm?id=882370.882388.

[38] X. Gu and S.-T. Yau, *Computational Conformal Geometry*, Higher Education Press, Beijing, 2008.

[39] X. D. Gu, W. Zeng, F. Luo, and S.-T. Yau, *Numerical computation of surface conformal mappings*, Comput. Methods Funct. Theory, 11 (2011), pp. 747–787, https://doi.org/10.1007/BF03321885.

[40] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle, *Conformal surface parameterization for texture mapping*, IEEE Trans. Vis. Comput. Graph., 6 (2000), pp. 181–189, https://doi.org/10.1109/2945.856998.

[41] K. Hormann, *MIPS: An Efficient Global Parametrization Method*, in ACM Press/Addison-Wesley, Reading, MA, 2000.

[42] K. Hormann, B. Lévy, and A. Sheffer, *Mesh parameterization: Theory and practice*, in Proceedings of SIGGRAPH'07, ACM, 2007, https://doi.org/10.1145/1281500.1281510.

[43] W.-Q. Huang, X. D. Gu, T.-M. Huang, S.-S. Lin, W.-W. Lin, and S.-T. Yau, *High performance computing for spherical conformal and Riemann mappings*, Geom. Imag. Comput., 1 (2014), pp. 223–258, https://doi.org/10.4310/GIC.2014.v1.n2.a2.

[44] W.-Q. Huang, X. D. Gu, W.-W. Lin, and S.-T. Yau, *A novel symmetric skew-Hamiltonian isotropic Lanczos algorithm for spectral conformal parameterizations*, J. Sci. Comput., 61 (2014), pp. 558–583, https://doi.org/10.1007/s10915-014-9840-2.

[45] M. Jin, J. Kim, F. Luo, and X. Gu, *Discrete surface Ricci flow*, IEEE Trans. Vis. Comput. Graphics, 14 (2008), pp. 1030–1043, https://doi.org/10.1109/TVCG.2008.57.

[46] M. Jin, Y. Wang, S.-T. Yau, and X. Gu, *Optimal global conformal surface parameterization*, in Proceedings of IEEE Visualization, 2004, pp. 267–274, https://doi.org/10.1109/VISUAL.2004.75.

[47] Y. Jin, G.-P. Qian, J.-Y. Zhao, J. Chang, R.-F. Tong, and J. Zhang, *Stretch-minimizing volumetric parameterization*, J. Comput. Sci. Tech., 30 (2015), pp. 553–564, https://doi.org/10.1007/s11390-015-1545-y.

[48] S. Z. Kovalsky, N. Aigerman, R. Basri, and Y. Lipman, *Controlling singular values with semidefinite programming*, ACM Trans. Graphics, 33 (2014), 68, https://doi.org/10.1145/2601097.2601142.

[49] S. Z. Kovalsky, N. Aigerman, R. Basri, and Y. Lipman, *Large-scale bounded distortion mappings*, ACM Trans. Graphics, 34 (2015), 191, https://doi.org/10.1145/2816795.2818098.

[50] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, *Least squares conformal maps for automatic texture atlas generation*, ACM Trans. Graphics, 21 (2002), pp. 362–371, https://doi.org/10.1145/566654.566590.

[51] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, *A local/global approach to mesh parameterization*, Comput. Graphics Forum, 27 (2008), pp. 1495–1504, https://doi.org/10.1111/j.1467-8659.2008.01290.x.

[52] J. J. Molitierno, *Applications of Combinatorial Matrix Theory to Laplacian Matrices of Graphs*, CRC Press, Boca Raton, FL, 2012.

[53] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun, *Spectral conformal parameterization*, Comput. Graphics Forum, 27 (2008), pp. 1487–1494.

[54] A. Naitsat, E. Saucan, and Y. Y. Zeevi, *Volumetric quasi-conformal mappings—quasi-conformal mappings for volume deformation with applications to geometric modeling*, in Proceedings of the 10th International Conference on Computer Graphics Theory and Applications, Vol. 1, INSTICC, SciTePress, 2015, pp. 46–57, https://doi.org/10.5220/0005298900460057.

[55] N. Narra, S. Abe, V. Dimitrov, R. Nikander, R. Kouhia, H. Sievänen, and J. Hyttinen, *Ricci-flow based conformal mapping of the proximal femur to identify exercise loading effects*, Sci. Rep., 8 (2018), https://doi.org/10.1038/s41598-018-23248-y.

[56] G.-P. Paillé and P. Poulin, *As-conformal-as-possible discrete volumetric mapping*, Comput. Graphics, 36 (2012), pp. 427–433, https://doi.org/10.1016/j.cag.2012.03.014.

[57] G.-P. Paillé, N. Ray, P. Poulin, A. Sheffer, and B. Lévy, *Dihedral angle-based maps of tetrahedral meshes*, ACM Trans. Graphics, 34 (2015), 54, https://doi.org/10.1145/2766900.

[58] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Exp. Math., 2 (1993), pp. 15–36.

[59] M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung, *Scalable locally injective mappings*, ACM Trans. Graphics, 36 (2017), 16, https://doi.org/10.1145/2983621.

[60] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe, *Texture mapping progressive meshes*, in Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, 2001, pp. 409–416, https://doi.org/10.1145/383259.383307.

[61] R. Sawhney and K. Crane, *Boundary first flattening*, ACM Trans. Graphics, 37 (2017), 5, https://doi.org/10.1145/3132705.

[62] A. Sheffer and E. de Sturler, *Parameterization of faceted surfaces for meshing using angle-based flattening*, Eng. Comput., 17 (2001), pp. 326–337.

[63] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, *ABF++: Fast and robust angle based flattening*, ACM Trans. Graphics, 24 (2005), pp. 311–330.

[64] A. Sheffer, E. Praun, and K. Rose, *Mesh parameterization methods and their applications*, Found. Trends. Comput. Graphics Vision, 2 (2006), pp. 105–171.

[65] J. Smith and S. Schaefer, *Bijective parameterization with free boundaries*, ACM Trans. Graphics, 34 (2015), 70, https://doi.org/10.1145/2766947.

[66] B. Springborn, P. Schröder, and U. Pinkall, *Conformal equivalence of triangle meshes*, ACM Trans. Graphics, 27 (2008), 77, https://doi.org/10.1145/1360612.1360676.

[67] K. Su, W. Chen, N. Lei, J. Zhang, K. Qian, and X. Gu, *Volume preserving mesh parameterization based on optimal mass transportation*, Comput.-Aided Des., 82 (2017), pp. 42–56, https://doi.org/10.1016/j.cad.2016.05.020.

[68] K. Su, L. Cui, K. Qian, N. Lei, J. Zhang, M. Zhang, and X. D. Gu, *Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation*, Comput. Aided Geom. Design, 46 (2016), pp. 76–91, https://doi.org/10.1016/j.cagd.2016.05.005.

[69] Y. Tong, S. Lombeyda, A. N. Hirani, and M. Desbrun, *Discrete multiscale vector field decomposition*, ACM Trans. Graphics, 22 (2003), pp. 445–452, https://doi.org/10.1145/882262.882290.

[70] Y. Wang, X. Gu, T. F. Chan, P. M. Thompson, and S.-T. Yau, *Volumetric harmonic brain mapping*, in 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro, Vol. 2, 2004, pp. 1275–1278, https://doi.org/10.1109/ISBI.2004.1398778.

[71] Y. Wang, X. Gu, and S.-T. Yau, *Volumetric harmonic map*, Commun. Inf. Syst., 3 (2004), pp. 191–202, https://projecteuclid.org:443/euclid.cis/1088692283.

[72] Z. Wang, Z.-x. Luo, J.-l. Zhang, and E. Saucan, *ARAP++: An extension of the local/global approach to mesh parameterization*, Frontiers Inform. Technol. Electron. Eng., 17 (2016), pp. 501–515, https://doi.org/10.1631/FITEE.1500184.

[73] Y. Xu, R. Chen, C. Gotsman, and L. Liu, *Embedding a triangular graph within a given boundary*, Comput. Aided Geom. Design, 28 (2011), pp. 349–356, https://doi.org/10.1016/j.cagd.2011.07.001.

[74] S. Yoshizawa, A. Belyaev, and H. P. Seidel, *A fast and simple stretch-minimizing mesh parameterization*, in Proceedings of Shape Modeling Applications, 2004, pp. 200–208, https://doi.org/10.1109/SMI.2004.1314507.

[75] M.-H. Yueh, W.-W. Lin, C.-T. Wu, and S.-T. Yau, *An efficient energy minimization for conformal parameterizations*, J. Sci. Comput., 73 (2017), pp. 203–227, https://doi.org/10.1007/s10915-017-0414-y.

[76] M.-H. Yueh, W.-W. Lin, C.-T. Wu, and S.-T. Yau, *A novel stretch energy minimization algorithm for equiareal parameterizations*, J. Sci. Comput., 78 (2019), pp. 1353–1386, https://doi.org/10.1007/s10915-018-0822-7.

[77] R. Zayer, B. Lévy, and H.-P. Seidel, *Linear angle based parameterization*, in Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Aire-la-Ville, Switzerland, Eurographics Association, 2007, pp. 135–141, http://dl.acm.org/citation.cfm?id=1281991.1282010.

[78] M. Zhang, R. Guo, W. Zeng, F. Luo, S.-T. Yau, and X. Gu, *The unified discrete surface Ricci flow*, Graphical Models, 76 (2014), pp. 321–339, https://doi.org/10.1016/j.gmod.2014.04.008.

[79] H. Zhao, X. Li, H. Ge, N. Lei, M. Zhang, X. Wang, and X. Gu, *Conformal mesh parameterization using discrete Calabi flow*, Comput. Aided Geom. Design, 63 (2018), pp. 96–108, https://doi.org/10.1016/j.cagd.2018.03.001.

[80] X. Zhao, Z. Su, X. D. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo, *Area-preservation mapping using optimal mass transport*, IEEE Trans. Visual Comput. Graphics, 19 (2013), pp. 2838–2847, https://doi.org/10.1109/TVCG.2013.135.

[81] G. Zou, J. Hu, X. Gu, and J. Hua, *Authalic parameterization of general surfaces using Lie advection*, IEEE Trans. Visual. Comput. Graphics, 17 (2011), pp. 2005–2014, https://doi.org/10.1109/TVCG.2011.171.