

Lawrence Berkeley National Laboratory

LBL Publications

Title

Volumetric Density-Equalizing Reference Map with Applications.

Permalink

<https://escholarship.org/uc/item/7d4910jt>

Authors

Choi, Gary PT

Rycroft, Chris H

Publication Date

2021

DOI

10.1007/s10915-021-01411-4

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

Volumetric density-equalizing reference map with applications

Gary P. T. Choi · Chris H. Rycroft

Received: date / Accepted: date

Abstract The density-equalizing map, a technique developed for cartogram creation, has been widely applied to data visualization but only for 2D applications. In this work, we propose a novel method called the volumetric density-equalizing reference map (VDERM) for computing density-equalizing map for volumetric domains. Given a prescribed density distribution in a volumetric domain in \mathbb{R}^3 , the proposed method continuously deforms the domain, with different volume elements enlarged or shrunk according to the density distribution. With the aid of the proposed method, medical and sociological data can be visualized via deformations of 3D objects. The method can also be applied to adaptive remeshing and shape modeling. Furthermore, by exploiting the time-dependent nature of the proposed method, applications to shape morphing can be easily achieved. Experimental results are presented to demonstrate the effectiveness of the proposed method.

Keywords Density-equalizing map · reference map technique · volumetric deformation · data visualization · shape modeling

Mathematics Subject Classification (2000) 68U05 · 65D18 · 76R50


1 Introduction

In the notable work by Gastner and Newman [1], an algorithm for producing density-equalizing map projections was proposed based on the diffusion equation. Specifically, given a planar geographical map and a density distribution prescribed on every part of the

Gary P. T. Choi
John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA

E-mail: pchoi@g.harvard.edu
 <https://orcid.org/0000-0001-5407-9111>

Chris H. Rycroft
John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA, and

Mathematics Group, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
E-mail: chr@seas.harvard.edu
 <https://orcid.org/0000-0003-4677-6990>

map, the algorithm continuously deforms the map such that the difference in the density at different regions is transformed into a difference in the area of the regions. Regions with a larger prescribed density expand and those with a smaller density shrink. Ultimately, the density is equalized over the entire deformed map. The algorithm has been extensively applied to the visualization of biological and sociological data, such as the global amphibian species diversity [2], the global population and income [3], the world citation network [4], the national climate contributions to observed global warming [5], the global burden of drug-resistant tuberculosis in children [6], and the impact of austerity and the economic crisis in Europe [7].

In recent years, a few improvements and extensions of the above-mentioned work have been proposed. Choi and Rycroft [8] developed a method for computing density-equalizing maps for simply-connected open surfaces and explored the close connection between map projections and surface parameterization, which opens up a wide range of applications of density-equalizing maps to geometry processing. Gastner et al. [9] proposed a new approach for computing density-equalizing map projections by considering a linearization of the density diffusion process. Recently, Choi et al. [10] proposed a method for computing area-preserving density-equalizing maps for carotid artery flattening.

Note that all the above-mentioned works only focused on the formulation and application of density-equalizing map in 2D. With the advancement of technology, there is an increasing need for 3D data processing. In this work, we develop a novel method called the *volumetric density-equalizing reference map* (VDERM) that produces volumetric deformations based on a prescribed density distribution. Our method combines and extends the 2D density-equalizing map [1] and the reference map technique in solid mechanics [11, 12, 13]. The proposed method can be utilized for volumetric data visualization, shape modeling, adaptive remeshing etc. To the best of our knowledge, this is the first work on higher dimensional density-equalizing maps. Furthermore, all the previous works only focused on the use of density-equalizing map for producing a single final output. With the observation that density-equalizing map is a continuous deformation, we introduce the use of density-equalizing map for time-dependent applications such as shape morphing.

The rest of the paper is organized as follows. In Section 2, we highlight the contributions of our work. In Section 3, we review the background of diffusion-based map-making and the reference map technique in 2D. In Section 4, we describe our proposed method for volumetric density-equalizing maps and present numerical experiments to demonstrate its effectiveness. We then discuss several novel applications of volumetric density-equalizing reference map in Section 5. In Section 6, we conclude our work and discuss possible future works.

2 Contributions

The contributions of our work are three-fold:

- (i) Previous works of density-equalizing maps have only considered the formulation and its applications in 2D. Our work is the first work focusing on higher dimensional density-equalizing maps. Experimental results show that our method is capable of producing volumetric deformations accurately based on the prescribed density.
- (ii) Our proposed volumetric method leads to novel applications of density-equalizing map, such as 3D medical and sociological data visualization, adaptive volumetric remeshing, and deformation-based shape modeling.
- (iii) All the previous works on density-equalizing maps have only focused on the use of the final result produced by the density-equalizing map method. Our work is the first

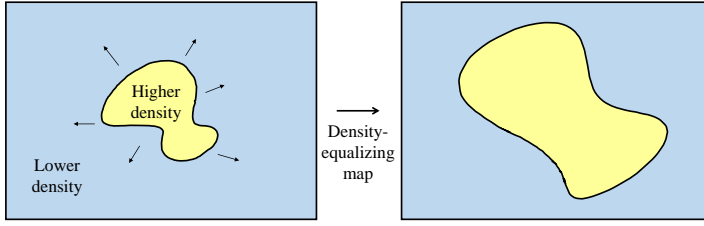


Fig. 1 An illustration of the diffusion-based cartogram. The density diffusion creates a velocity field that enlarges regions with higher density and shrinks regions with lower density.

work that considers the use of the entire density-equalization process, i.e. not only the final maps but also the intermediate states obtained throughout the process, for time-dependent applications such as object morphing.

3 Background

3.1 Diffusion-based cartogram

Gastner and Newman [1] proposed a method for creating cartograms based on density diffusion. Given a 2D domain, a positive quantity ρ called the *density* is first prescribed on every part of it. For instance, the 2D domain can be a geographical map of a country, and ρ can be the population density of different provinces. The goal of a diffusion-based cartogram is to deform the domain by enlarging the regions with higher ρ , and shrinking the regions with lower ρ . This is achieved by equalizing ρ , following the advection equation

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{j}, \quad (1)$$

where $\mathbf{j} = -\nabla \rho$ is the density flux. This yields the diffusion equation

$$\frac{\partial \rho}{\partial t} = \Delta \rho. \quad (2)$$

Any tracer particles on the domain will then move with velocity

$$\mathbf{v} = -\frac{\nabla \rho}{\rho}. \quad (3)$$

Therefore, the location of a tracer particle \mathbf{x} at time t can be given by

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{v}(\mathbf{x}, \tau) d\tau. \quad (4)$$

As $t \rightarrow \infty$, ρ is fully equalized on the 2D domain, and the resulting deformation produces an accurate density-based deformation according to the prescribed ρ . An illustration is given in Fig. 1.

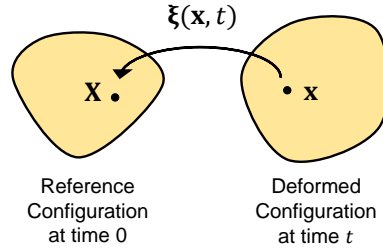


Fig. 2 An illustration of the reference map. $\xi(\mathbf{x}, t)$ is the inverse of the motion function, which indicates the reference location \mathbf{X} of the material occupying the position \mathbf{x} at time t .

3.2 The reference map technique

In solid mechanics, a standard mathematical approach is to consider a body in an undeformed reference configuration at time 0 that is mapped into a deformed configuration at time t [14]. Let $\mathbf{X} \in \mathbb{R}^d$ be a point in the reference configuration, and $\mathbf{x} \in \mathbb{R}^d$ be the corresponding point in the deformed configuration. Then the deformation can be described by the *motion function* $\chi(\mathbf{X}, t) = \mathbf{x}$. The *reference map* is then defined as the inverse motion function $\xi(\mathbf{x}, t) = \mathbf{X}$ (Fig. 2), which can be regarded as a vector field in the deformed configuration indicating the reference location of the material occupying the position \mathbf{x} at time t . It has been used in various applications such as inverse design [15, 16].

Kamrin, Rycroft, and coworkers developed the reference map technique (RMT) [11, 12, 13], which uses the reference map as the basis for a fully Eulerian approach to solid mechanics, since it provides a simple and effective method to describe an arbitrary deformation of a body. Since the initial configuration is undeformed, we have $\xi(\mathbf{x}, 0) = \mathbf{x}$. Now, for any tracer particle, the reference location of it is the same at all time t under the deformation. Therefore, we have

$$\dot{\xi} = \mathbf{0}, \quad (5)$$

yielding the advection equation

$$\frac{\partial \xi}{\partial t} + \mathbf{v} \cdot \nabla \xi = \mathbf{0}. \quad (6)$$

Eq. (6) can be solved on a discrete set of fixed Eulerian grid points using simple finite-difference schemes. In the RMT ξ is used to calculate mechanical quantities such as the Cauchy stress, but here we use it as a straightforward Eulerian-frame calculation of solid deformations.

4 Volumetric density-equalizing reference map (VDERM)

Based on the 2D diffusion-based cartogram and the reference map technique introduced above, we propose a method for computing the volumetric density-equalizing reference map of 3D domains.

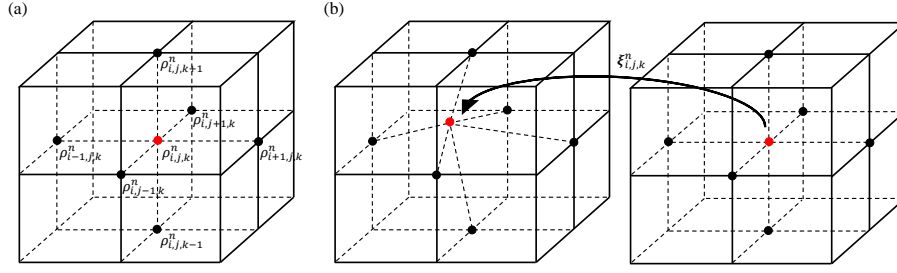


Fig. 3 A illustration of density diffusion and reference map update in our proposed volumetric density-equalizing reference map method. (a) The diffusion of ρ is done iteratively by solving the diffusion equation (8), based on the six neighboring nodes. (b) The density gradient is then used for updating the reference map field ξ via the advection equation (13).

4.1 Formulation

Suppose we are given a rectangular solid domain $D \subset \mathbb{R}^3$, discretized as a $L \times M \times N$ 3D grid with grid spacing h in all the x -, y - and z -directions. Let the coordinates of the grid points be (ih, jh, kh) , where $0 \leq i \leq L-1$, $0 \leq j \leq M-1$, $0 \leq k \leq N-1$.

Consider a prescribed density $\rho^0 = \rho^0(\mathbf{x})$ defined in D , discretized as $\rho_{i,j,k}^0 = \rho^0(ih, jh, kh)$ for all i, j, k . Let $\rho(\mathbf{x}, t)$ be the density at time t , with $\rho(\mathbf{x}, 0) = \rho^0(\mathbf{x})$. The diffusion of ρ follows from the diffusion equation

$$\frac{\partial \rho}{\partial t}(\mathbf{x}, t) = \Delta \rho(\mathbf{x}, t), \quad (7)$$

which can be discretized using the following backward Euler scheme (Fig. 3):

$$\frac{\rho_{i,j,k}^n - \rho_{i,j,k}^{n-1}}{\delta t} = \frac{1}{h^2} \left(\rho_{i+1,j,k}^n + \rho_{i-1,j,k}^n + \rho_{i,j+1,k}^n + \rho_{i,j-1,k}^n + \rho_{i,j,k+1}^n + \rho_{i,j,k-1}^n - 6\rho_{i,j,k}^n \right), \quad (8)$$

where δt is the timestep to be determined, and $n = t/\delta t$ is the iteration number. Eq. (8) can be further simplified as

$$\rho^n = (I - \delta t \Delta)^{-1} \rho^{n-1}, \quad (9)$$

where Δ is an $LMN \times LMN$ matrix representing the seven-point Laplacian stencil. Note that $I - \delta t \Delta$ is a sparse symmetric positive definite matrix and is independent of n . Therefore, after setting up the matrix $A = I - \delta t \Delta$ based on the backward Euler scheme (8) and preconditioning the system once, we can efficiently solve Eq. (9) at each iteration by reusing both the matrix A and the preconditioner.

Throughout the diffusion process, the density gradient of ρ induces a velocity field

$$\mathbf{v}(\mathbf{x}, t) = -\frac{\nabla \rho(\mathbf{x}, t)}{\rho(\mathbf{x}, t)}, \quad (10)$$

which can be discretized using the central difference scheme:

$$\begin{cases} (\mathbf{v}_x)_{i,j,k}^n = -\frac{\rho_{i+1,j,k}^n - \rho_{i-1,j,k}^n}{2h\rho_{i,j,k}^n}, \\ (\mathbf{v}_y)_{i,j,k}^n = -\frac{\rho_{i,j+1,k}^n - \rho_{i,j-1,k}^n}{2h\rho_{i,j,k}^n}, \\ (\mathbf{v}_z)_{i,j,k}^n = -\frac{\rho_{i,j,k+1}^n - \rho_{i,j,k-1}^n}{2h\rho_{i,j,k}^n}. \end{cases} \quad (11)$$

To track the deformation of D under the velocity field, consider the reference map $\boldsymbol{\xi}(\mathbf{x}, t)$, discretized as $\boldsymbol{\xi}_{i,j,k}^n = \boldsymbol{\xi}((ih, jh, kh), n\delta t)$. $\boldsymbol{\xi}$ can be obtained by solving the advection equation

$$\frac{\partial \boldsymbol{\xi}}{\partial t}(\mathbf{x}, t) - \frac{\nabla \rho(\mathbf{x}, t)}{\rho(\mathbf{x}, t)} \cdot \nabla \boldsymbol{\xi}(\mathbf{x}, t) = \mathbf{0}, \quad (12)$$

which can be discretized using the second-order upwind scheme:

$$\frac{\boldsymbol{\xi}_{i,j,k}^n - \boldsymbol{\xi}_{i,j,k}^{n-1}}{\delta t} = d_x + d_y + d_z, \quad (13)$$

where

$$d_x = \begin{cases} (\mathbf{v}_x)_{i,j,k}^n d_x^- & \text{if } (\mathbf{v}_x)_{i,j,k}^n > 0, \\ (\mathbf{v}_x)_{i,j,k}^n d_x^+ & \text{if } (\mathbf{v}_x)_{i,j,k}^n \leq 0, \end{cases} \quad (14)$$

$$d_y = \begin{cases} (\mathbf{v}_y)_{i,j,k}^n d_y^- & \text{if } (\mathbf{v}_y)_{i,j,k}^n > 0, \\ (\mathbf{v}_y)_{i,j,k}^n d_y^+ & \text{if } (\mathbf{v}_y)_{i,j,k}^n \leq 0, \end{cases} \quad (15)$$

$$d_z = \begin{cases} (\mathbf{v}_z)_{i,j,k}^n d_z^- & \text{if } (\mathbf{v}_z)_{i,j,k}^n > 0, \\ (\mathbf{v}_z)_{i,j,k}^n d_z^+ & \text{if } (\mathbf{v}_z)_{i,j,k}^n \leq 0, \end{cases} \quad (16)$$

with $d_x^-, d_x^+, d_y^-, d_y^+, d_z^-, d_z^+$ being the three-point finite-difference discretization of the spatial derivative $\nabla \boldsymbol{\xi}$:

$$\begin{cases} d_x^- = \frac{3\xi_{i,j,k}^{n-1} - 4\xi_{i-1,j,k}^{n-1} + \xi_{i-2,j,k}^{n-1}}{2h}, \\ d_x^+ = \frac{-\xi_{i+2,j,k}^{n-1} + 4\xi_{i+1,j,k}^{n-1} - 3\xi_{i,j,k}^{n-1}}{2h}, \\ d_y^- = \frac{3\xi_{i,j,k}^{n-1} - 4\xi_{i,j-1,k}^{n-1} + \xi_{i,j-2,k}^{n-1}}{2h}, \\ d_y^+ = \frac{-\xi_{i,j+2,k}^{n-1} + 4\xi_{i,j+1,k}^{n-1} - 3\xi_{i,j,k}^{n-1}}{2h}, \\ d_z^- = \frac{3\xi_{i,j,k}^{n-1} - 4\xi_{i,j,k-1}^{n-1} + \xi_{i,j,k-2}^{n-1}}{2h}, \\ d_z^+ = \frac{-\xi_{i,j,k+2}^{n-1} + 4\xi_{i,j,k+1}^{n-1} - 3\xi_{i,j,k}^{n-1}}{2h}. \end{cases} \quad (17)$$

For $i = 1, L-2$, we replace d_x^- and d_x^+ with the two-point difference

$$\begin{cases} d_x^- = \frac{\xi_{i,j,k}^{n-1} - \xi_{i-1,j,k}^{n-1}}{h}, \\ d_x^+ = \frac{\xi_{i+1,j,k}^{n-1} - \xi_{i,j,k}^{n-1}}{h}. \end{cases} \quad (18)$$

Similarly, for $j = 1, M-2$, we use

$$\begin{cases} d_y^- = \frac{\xi_{i,j,k}^{n-1} - \xi_{i,j-1,k}^{n-1}}{h}, \\ d_y^+ = \frac{\xi_{i,j+1,k}^{n-1} - \xi_{i,j,k}^{n-1}}{h}, \end{cases} \quad (19)$$

and for $k = 1, N - 2$ we use

$$\begin{cases} d_z^- = \frac{\xi_{i,j,k}^{n-1} - \xi_{i,j,k-1}^{n-1}}{h}, \\ d_z^+ = \frac{\xi_{i,j,k+1}^{n-1} - \xi_{i,j,k}^{n-1}}{h}. \end{cases} \quad (20)$$

As for the choice of the timestep δt , note that the backward Euler scheme (8) for the diffusion equation is unconditionally stable. A necessary condition for the convergence of the second-order upwind scheme (13) for the advection equation is given by the Courant-Friedrichs-Lewy (CFL) condition [17], which says that the numerical domain of dependence must include the physical domain of dependence. More specifically, consider multiplying δt on both sides in Eq. (13) and making $\xi_{i,j,k}^n$ as the subject. Then, the remaining terms involving $\xi_{i,j,k}^{n-1}$ will be

$$\begin{aligned} & \xi_{i,j,k}^{n-1} - \delta t \left(\frac{3\xi_{i,j,k}^{n-1} |(\mathbf{v}_x)_{i,j,k}^n|}{2h} + \frac{3\xi_{i,j,k}^{n-1} |(\mathbf{v}_y)_{i,j,k}^n|}{2h} + \frac{3\xi_{i,j,k}^{n-1} |(\mathbf{v}_z)_{i,j,k}^n|}{2h} \right) \\ &= \xi_{i,j,k}^{n-1} \left(1 - \delta t \left(\frac{3|(\mathbf{v}_x)_{i,j,k}^n|}{2h} + \frac{3|(\mathbf{v}_y)_{i,j,k}^n|}{2h} + \frac{3|(\mathbf{v}_z)_{i,j,k}^n|}{2h} \right) \right). \end{aligned} \quad (21)$$

Note that we have the absolute signs in the above expression because for a negative $(\mathbf{v}_x)_{i,j,k}^n$, $(\mathbf{v}_y)_{i,j,k}^n$, or $(\mathbf{v}_z)_{i,j,k}^n$, the corresponding d_x^+ , d_y^+ , or d_z^+ involves a negative sign in the coefficient of $\xi_{i,j,k}^{n-1}$. Now, since the full numerical domain of dependence of the scheme must contain the physical domain of dependence, for all i, j, k, n , the coefficient of $\xi_{i,j,k}^{n-1}$ must satisfy

$$0 \leq 1 - \delta t \left(\frac{3|(\mathbf{v}_x)_{i,j,k}^n|}{2h} + \frac{3|(\mathbf{v}_y)_{i,j,k}^n|}{2h} + \frac{3|(\mathbf{v}_z)_{i,j,k}^n|}{2h} \right) \leq 1, \quad (22)$$

which implies that

$$\delta t \left(\frac{3|(\mathbf{v}_x)_{i,j,k}^n|}{2h} + \frac{3|(\mathbf{v}_y)_{i,j,k}^n|}{2h} + \frac{3|(\mathbf{v}_z)_{i,j,k}^n|}{2h} \right) \leq 1. \quad (23)$$

In our problem, the magnitude of the velocity field decreases as n increases because diffusion smooths out sharp gradients in density. Therefore, we should have

$$\delta t \leq \frac{2h}{3 \max_{i,j,k} (|(\mathbf{v}_x)_{i,j,k}^0| + |(\mathbf{v}_y)_{i,j,k}^0| + |(\mathbf{v}_z)_{i,j,k}^0|)}. \quad (24)$$

In practice, we take the upper bound and set the timestep as

$$\delta t = \frac{2h}{3 \max_{i,j,k} (|(\mathbf{v}_x)_{i,j,k}^0| + |(\mathbf{v}_y)_{i,j,k}^0| + |(\mathbf{v}_z)_{i,j,k}^0|)}. \quad (25)$$

In the continuous formulation, as $t \rightarrow \infty$, the density $\rho(\mathbf{x}, t)$ is equalized over D , and the associated reference map $\xi_{\text{final}}(\mathbf{x}) = \xi(\mathbf{x}, \infty)$ becomes a density-equalizing reference map. In the discrete case, the result converges to a density-equalizing reference map as the change of the density becomes very small. Therefore, we use the following convergence criterion:

$$\frac{\|\rho^n - \rho^{n-1}\|_2}{\text{mean}(\rho^{n-1})} \leq \varepsilon, \quad (26)$$

Algorithm 1: Volumetric density-equalizing reference map

-
- Input:** A solid domain $D \subset \mathbb{R}^3$ of size $L \times M \times N$, a prescribed density ρ^0 , the error threshold ε , the maximum number of iterations allowed n_{\max} .
- Output:** A density-equalizing reference map ξ_{final} .
- 1 Set $\delta t = \frac{2h}{3 \max_{i,j,k} (|(v_x)_{i,j,k}^0| + |(v_y)_{i,j,k}^0| + |(v_z)_{i,j,k}^0|)}$;
 - 2 Compute $A = I - \delta t \Delta$;
 - 3 Set $n = 0$;
 - 4 **repeat**
 - 5 Update $n = n + 1$;
 - 6 Solve $\rho^n = A^{-1} \rho^{n-1}$ as discussed in Eq. (9) with an appropriate boundary condition;
 - 7 Compute the velocity field \mathbf{v}^n using Eq. (11);
 - 8 Update the reference map ξ^n using Eq. (13);
 - 9 **until** $\frac{\|\rho^n - \rho^{n-1}\|_2}{\text{mean}(\rho^{n-1})} \leq \varepsilon$ or $n \geq n_{\max}$;
 - 10 Obtain $\xi_{\text{final}} = \xi^n$;
-

where ε is the error threshold. Our algorithm is summarized in Algorithm 1.

The density-equalizing reference map provides us with the information of the diffusion-based deformation. In particular, if we denote $\xi_{\text{final}} = (\xi_1, \xi_2, \xi_3)$, one can obtain the forward mapping of the grid points of D by tracking the intersections of the contour planes $\xi_1 = ih$, $\xi_2 = jh$ and $\xi_3 = kh$. More specifically, after computing the density-equalizing reference map, we build an interpolant for each of the x -, y -, and z -coordinates of the undeformed grid using (ξ_1, ξ_2, ξ_3) , where the ξ values are the abscissa and the x, y, z values are the ordinate. In other words, we have three interpolant functions $I_1(x, y, z), I_2(x, y, z), I_3(x, y, z)$ where

$$\begin{cases} I_1(\xi_1(ih, jh, kh), \xi_2(ih, jh, kh), \xi_3(ih, jh, kh)) = ih, \\ I_2(\xi_1(ih, jh, kh), \xi_2(ih, jh, kh), \xi_3(ih, jh, kh)) = jh, \\ I_3(\xi_1(ih, jh, kh), \xi_2(ih, jh, kh), \xi_3(ih, jh, kh)) = kh, \end{cases} \quad (27)$$

for all i, j, k . The three interpolants allow us to compute a forward deformation of the given domain. In practice, we compute a Delaunay triangulation of the given points and then perform a linear interpolation in each tetrahedral element. It is noteworthy that throughout the density-equalizing iterations it suffices to work on a grid, and the interpolation is only needed once at the end. On the contrary, a direct extension of the original density-equalizing map [1] would involve interpolating the velocity field at every iteration. Therefore, the use of the reference map technique here is advantageous.

4.2 Boundary conditions

Note that boundary conditions are needed for solving the diffusion equation (8). Below, we discuss three possible boundary conditions.

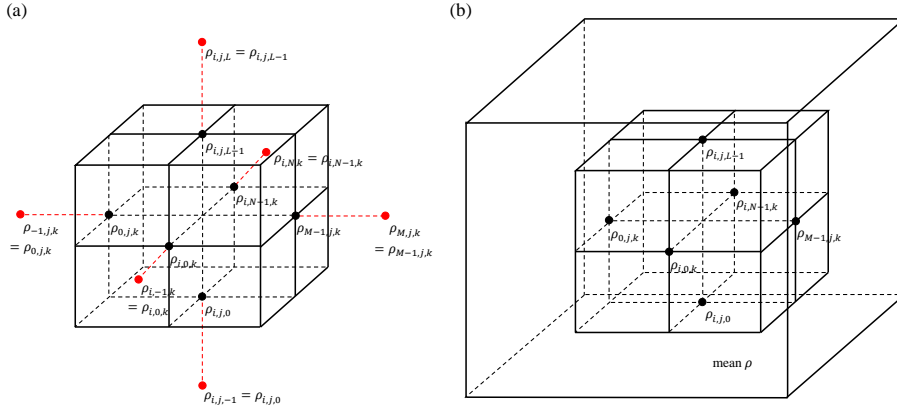


Fig. 4 Different choices of boundary conditions for a solid domain D . (a) The no-flux boundary condition can be imposed using the ghost node approach shown. A ghost node (red) is placed at the neighborhood of each boundary node on ∂D , with the density being the same as the density at the boundary node. This ensures that there is no density flux orthogonal to the boundary faces. (b) The free boundary condition can be imposed by constructing a larger solid domain \tilde{D} and setting the density at the “sea” (i.e. $\tilde{D} \setminus D$) to be the average value of the original density ρ . This ensures that the boundary ∂D of the original domain D can deform freely.

4.2.1 No-flux boundary condition

To maintain the rectangular boundary shape of the 3D domain, we can enforce the no-flux boundary condition

$$\mathbf{n} \cdot \nabla \rho = 0 \quad (28)$$

at the boundaries, where \mathbf{n} is the outward unit normal at the boundaries. More explicitly, this condition ensures that the density gradient at the boundaries is zero in the normal direction, and hence all the boundary planes remain planar throughout the density-equalization process.

In the discrete case, we incorporate the above no-flux boundary condition in the diffusion equation (8) using the following ghost node approach. At the six boundary faces $i = 0$, $i = L - 1$, $j = 0$, $j = M - 1$, $k = 0$, $k = N - 1$, we respectively replace the terms $\rho_{i-1,j,k}^n$, $\rho_{i+1,j,k}^n$, $\rho_{i,j-1,k}^n$, $\rho_{i,j+1,k}^n$, $\rho_{i,j,k-1}^n$, $\rho_{i,j,k+1}^n$ on the right hand side in the diffusion equation (8) by $\rho_{i,j,k}^n$. This ensures that there is no density flux orthogonal to the six boundary faces throughout the density diffusion process (Fig. 4(a)).

The effect is that for each boundary node, one of its coordinates will remain unchanged under the update by the advection equation (13), while the other coordinates can vary. Hence, the boundary nodes can freely slide along the six boundary faces to achieve density equalization while D will remain to be a rectangular box throughout the iterations.

4.2.2 Free boundary condition

In case it is desirable to let the domain deform freely without any boundary constraints, the “sea” approach by Gastner and Newman [1] can be used. Consider putting the entire solid domain D with the prescribed density ρ^0 at the center of a larger rectangular solid domain \tilde{D} . Define the density $\tilde{\rho}^0$ on \tilde{D} by

$$\tilde{\rho}^0 = \begin{cases} \rho^0 & \text{on } D, \\ \text{mean}(\rho^0) & \text{on } \tilde{D} \setminus D. \end{cases} \quad (29)$$

Now, we impose the no-flux boundary condition on $\partial\tilde{D}$ and compute the density-equalizing map on \tilde{D} (Fig. 4(b)). As there is no boundary condition imposed on ∂D , D can deform freely under the deformation of \tilde{D} without any components of the reference map ξ on ∂D being pinned. Note that the purpose of setting $\tilde{\rho}^0 = \text{mean}(\rho^0)$ at the “sea” $\tilde{D} \setminus D$ is to prevent D from expanding infinitely.

4.2.3 Mixed boundary condition

It is also possible to combine different boundary conditions to achieve other deformations. For instance, in case it is desired to keep the top and the bottom boundary planes planar while allowing the other boundary planes to deform freely, one can place the $L \times M \times N$ domain in a larger $\tilde{L} \times \tilde{M} \times N$ domain, where $\tilde{L} > L$ and $\tilde{M} > M$. Then, using the above-mentioned “sea” approach, one can enforce the no-flux boundary condition on the z -boundary planes and the free boundary condition on the x - and y -boundary planes, thereby producing the desired effect.

4.3 Numerical results

The proposed iterative scheme is implemented in C++ with OpenMP parallelization (with grid size $h = 1$, maximum number of iterations $n_{\max} = 10000$, and error threshold $\varepsilon = 10^{-2}$). The sparse linear systems are solved using the conjugate gradient method (`ConjugateGradient`) in the C++ library Eigen, with the preconditioner option `DiagonalPreconditioner` used. All experiments are performed on a PC with an Intel i7-6700K quad-core processor and 16 GB RAM. The statistics and the visualization are done using MATLAB. In the following, we assess the performance of the proposed method with numerical experiments.

We first consider a cubic solid domain D of grid size $L \times M \times N = 32 \times 32 \times 32$ with a smooth input density

$$\rho^0(i, j, k) = 10 + 9.99 \sin\left(\frac{4\pi i}{L-1}\right) \cos\left(\frac{2\pi j}{M-1}\right) \cos\left(\frac{2\pi k}{N-1}\right), \quad (30)$$

with $i, j, k = 0, 1, \dots, 31$, as shown in Fig. 5(a). We compute the volumetric density-equalizing maps with the no-flux boundary condition (Fig. 5(b)), the free boundary condition (Fig. 5(c)), and the mixed-boundary condition (Fig. 5(d)). The one with the free boundary condition is done by placing the solid in a circumscribed $48 \times 48 \times 48$ cubic grid, and the one with the mixed boundary condition is achieved with the aid of a circumscribed $48 \times 48 \times 32$ rectangular grid. From Fig. 5(b), it can be observed that different regions are enlarged or shrunk according to the prescribed density. Also, the deformed domain remains a cube, while the boundary nodes are free to slide on the boundary planes to achieve density-equalization. For the map with the free boundary condition as shown in Fig. 5(c), it can be observed that the domain deforms freely while not maintaining a cubic shape. For the map with the mixed boundary condition as shown in Fig. 5(d), it can be observed that the top and the bottom of the solid domain remain to be planar, while the other sides of the domain are deformed freely.

To assess the accuracy of the mappings, we define the *volume-density mismatch error* to be the logged ratio of the volumetric scale factor to the prescribed density with a normalization:

$$e(\mathbf{x}) = \log \frac{\det(F(\mathbf{x})) / \iiint_D \det(F(\mathbf{x}))}{\rho^0(\xi_{\text{final}}(\mathbf{x})) / \iiint_D \rho^0}, \quad (31)$$

Number of vertices ($L \times L \times L$)	Time (s)	# iterations	mean($ e $)
$8 \times 8 \times 8 = 512$	0.02	105	0.8868
$16 \times 16 \times 16 = 4096$	0.20	323	0.4468
$24 \times 24 \times 24 = 13824$	1.14	504	0.2774
$32 \times 32 \times 32 = 32768$	4.43	637	0.1972
$40 \times 40 \times 40 = 64000$	12.13	727	0.1577
$48 \times 48 \times 48 = 110592$	26.65	784	0.1356
$56 \times 56 \times 56 = 175616$	59.59	816	0.1262
$64 \times 64 \times 64 = 262144$	111.53	856	0.1241

Table 1 The performance of the proposed method in terms of the time required for the iterative scheme, the number of iterations needed, and the mean of the absolute volume-density mismatch error mean($|e|$). In all experiments, four OpenMP threads are used.

where $F(\mathbf{x}) = \left(\frac{\partial \xi_{\text{final}}}{\partial \mathbf{x}}\right)^{-1}$ is the Jacobian, and $\iiint_D \det(F(\mathbf{x}))$ and $\iiint_D \rho^0$ are computed using the trapezoidal integration method. Note that $e = 0$ if and only if the final volume distribution matches the prescribed density distribution. As shown in Fig. 5, the histograms of e for all different boundary conditions highly concentrate at 0, implying that the computation is accurate regardless of the choice of the boundary conditions.

Our method can also handle prescribed densities with discontinuity. Fig. 6(a) shows another example of a 3D grid of size $L \times M \times N = 32 \times 32 \times 32$ with eight different density values defined on eight regions

$$\rho^0(i, j, k) = \begin{cases} 1 & \text{if } i < L/2, j < M/2, k < N/2, \\ 3 & \text{if } i \geq L/2, j < M/2, k < N/2, \\ 5 & \text{if } i < L/2, j \geq M/2, k < N/2, \\ 7 & \text{if } i \geq L/2, j \geq M/2, k < N/2, \\ 9 & \text{if } i < L/2, j < M/2, k \geq N/2, \\ 11 & \text{if } i \geq L/2, j < M/2, k \geq N/2, \\ 13 & \text{if } i < L/2, j \geq M/2, k \geq N/2, \\ 15 & \text{if } i \geq L/2, j \geq M/2, k \geq N/2. \end{cases} \quad (32)$$

Again, we consider computing the volumetric density-equalizing maps with the no-flux boundary condition (Fig. 6(b)), the free boundary condition (Fig. 6(c)) and the mixed boundary condition (Fig. 6(d)). The one with the free boundary condition is done by placing the solid in a larger $48 \times 48 \times 48$ grid, and the one with the mixed boundary condition is achieved with the aid of a $48 \times 48 \times 32$ grid. In all cases, it can be observed that the regions are appropriately magnified or shrunk according to the input density, either with the cubic boundary shape preserved (Fig. 6(b)), with the boundary shape freely deformed (Fig. 6(c)), or with the planarity of only the top and the bottom boundaries preserved (Fig. 6(d)). Also, while the input density is highly discontinuous, it can be observed from the histograms of the volume-density mismatch error e that the error is concentrated at 0. This suggests that our method is capable of handling discontinuous input densities.

We further examine the performance of the proposed method at different resolutions $L \times L \times L$ with the density given by Eq. (30). For a fair assessment, we introduce a diffusion coefficient $\kappa = L/64$ on the right-hand side of Eq. (7) and Eq. (3) to ensure that the rate of diffusion is proportionally rescaled. From Table 1, it can be observed that the volume-density mismatch error mean($|e|$) is inversely proportional to L . Therefore, the accuracy of

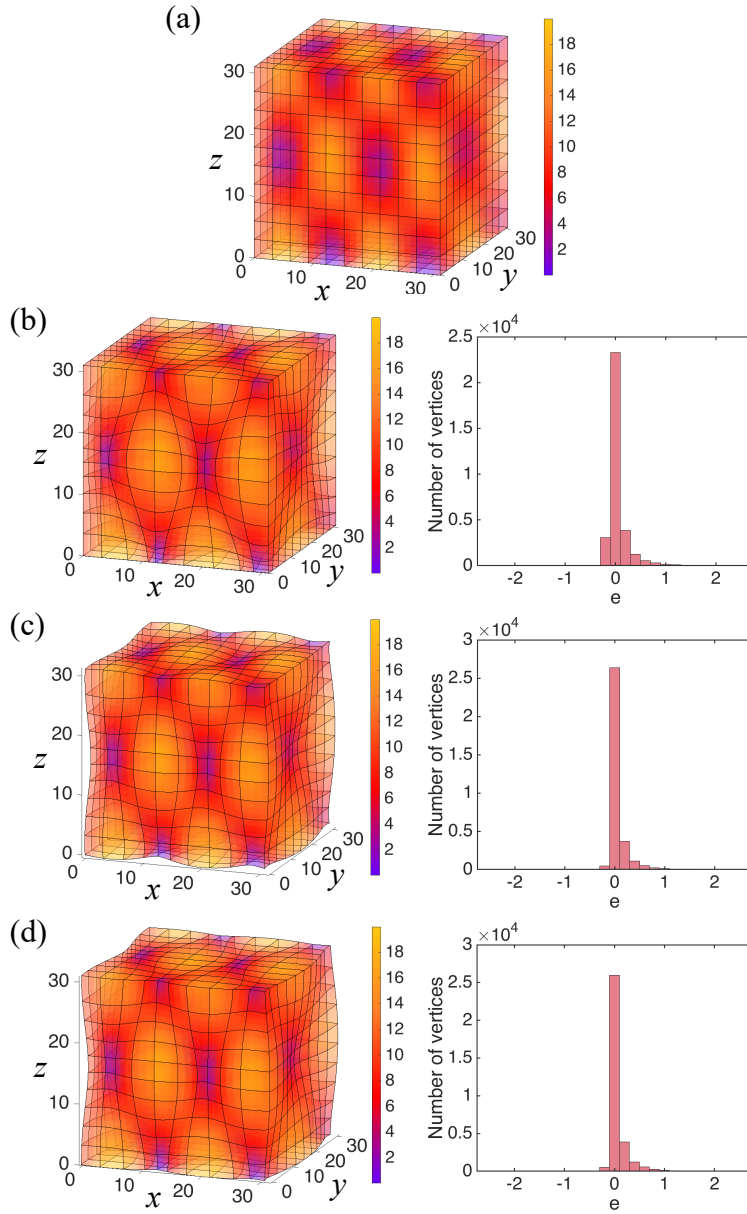


Fig. 5 An example of volumetric density-equalizing maps with the prescribed density being a periodic function with multiple peaks on a $32 \times 32 \times 32$ grid. The deformed contour planes at the x -, y -, and z -directions are plotted for visualizing the mapping results. (a) The initial state colored with the prescribed density. (b) The forward mapping result with the no-flux boundary condition and the histogram of the volume-density mismatch error e . (c) The forward mapping result with the free boundary condition and the histogram of e . (d) The forward mapping result with the mixed boundary condition where only the top and bottom boundary planes are required to remain planar, and the histogram of e . All the results are colored with the prescribed density. For better visualization, only part of the grid lines are drawn.

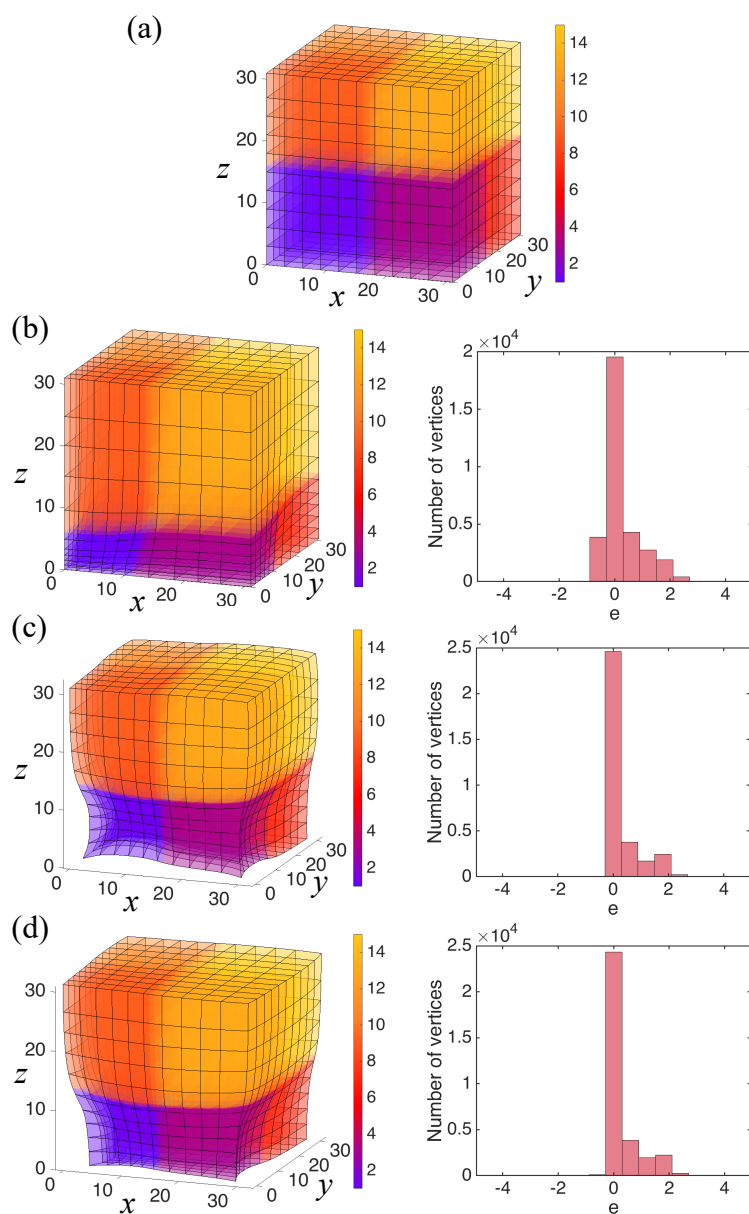


Fig. 6 An example of volumetric density-equalizing maps with 8 different density values defined on 8 regions on a $32 \times 32 \times 32$ grid. (a) The initial state colored with the prescribed density. (b) The forward mapping result with the no-flux boundary condition and the histogram of the volume-density mismatch error e . (c) The forward mapping result with the free boundary condition and the histogram of e . (d) The forward mapping result with the mixed boundary condition where only the top and bottom boundary planes are required to remain planar, and the histogram of e . All the results are colored with the prescribed density. For better visualization, only part of the grid lines are drawn.

Preconditioner	# iterations per CG solve	Total time for VDERM (s)
DiagonalPreconditioner	17	3.09
LeastSquareDiagonalPreconditioner	19	3.37
IncompleteLUT	4	10.38
IncompleteCholesky	9	6.47

Table 2 The performance of the proposed VDERM method for the example in Fig. 5, with different preconditioners in the Eigen library used for solving Eq. (9).

the density-equalizing reference map can be improved by increasing the resolution of the numerical domain.

4.4 Alternative numerical schemes

In our implementation of the proposed VDERM method, we solve the diffusion equation (9) using the Conjugate Gradient (CG) solver in the C++ Eigen library with the diagonal preconditioner, which neglects all off-diagonal entries of the input matrix A and approximates A^{-1} as the inverse of the matrix diagonal. One may wonder whether using some other preconditioners will lead to a better performance. Here we consider other built-in preconditioners in Eigen including `LeastSquareDiagonalPreconditioner` (a preconditioner which approximately solves for $A^T Ax = A^T b$ assuming $A^T A$ is a diagonal matrix), `IncompleteLUT` (the incomplete LU factorization with dual-threshold strategy [29]), and `IncompleteCholesky` (the modified Incomplete Cholesky method with dual threshold [30]). Table 2 shows the performance of the VDERM method with different preconditioners. While some of these alternative options take fewer iterations to converge for each CG solve, we find `DiagonalPreconditioner` costing the lowest total computation time among all of them.

This is because our choice of δt by Eq. (25) is comparatively small (in all our experiments we have $\delta t \sim 10^{-1}$), and so the operator $A = I - \delta t \Delta$ has a fundamental solution with a significant exponential decay. In the CG method with the diagonal preconditioner, information can only propagate by one grid point per iteration. However, since our fundamental solution is relatively compact, we achieve convergence in a small number of iterations per CG solve. While some other preconditioners allow information to propagate more than one grid point per iteration and hence require fewer iterations to converge per CG solve, their increased computational complexity outweighs the reduced iteration count. We remark that in case the problem involves much larger grids and/or some input densities with an extremely small density gradient such that the resulting δt by Eq. (25) is large enough, the alternative preconditioners may offer computational advantages. However, for the typical grid sizes and input densities considered in this work, the CG method with the diagonal preconditioner gives the best performance.

Besides the CG method, one may also consider other numerical methods such as the Fast Fourier Transform method (FFT) for solving the diffusion equation (9). Here we use the FFTW fast Fourier transform library [31,32] to compute the forward and inverse Fourier transforms for solving Eq. (9). For the example in Fig. 5, the FFT method results in a total computation time of 3.35s, which is slightly slower than the CG method with the diagonal preconditioner as shown in Table 2. The result can again be explained by the comparatively small δt set by Eq. (25) as discussed above. While the FFT would be effective for solving ∇^2

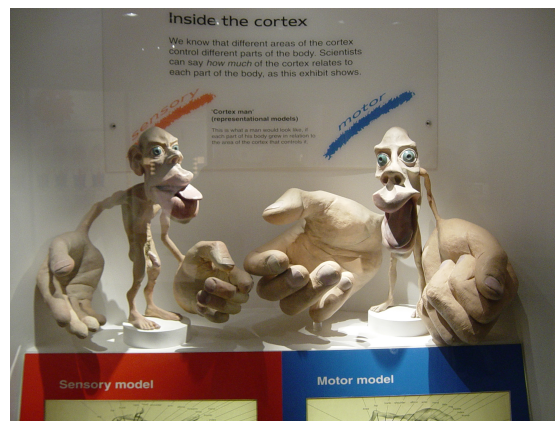


Fig. 7 Sculptures of sensory and motor homunculi at the Museum of Natural History, London, UK. The image is adopted online [21] under the CC BY-SA 3.0 license.

on a large grid, our case has a compact fundamental solution that is solved most efficiently with CG.

5 Applications

5.1 Volumetric data visualization

Recall that the original density-equalizing map for 2D domains was developed primarily for sociological and biological data visualization on the world map. Instead of simply representing certain information using colors on a normal world map, one can convey the information geometrically by distorting the map via density equalization. Analogously, with the aid of our proposed VDERM method, we can provide an alternative, geometrical visualization for volumetric data. We demonstrate this idea using two medical and sociological examples.

5.1.1 Medical data visualization

In neurology, a *cortical homunculus* (also known as a *cortex man*) is a 3D human model with distortions at different parts of the body representing the proportion of the human brain that is used for processing sensory or motor functions for those parts [19]. Fig. 7 shows sculptures of sensory and motor homunculi at the Museum of Natural History, London, UK. Such distorted representations enable us to easily understand the ratios between the levels of nerve control for different parts of the human body.

Our proposed VDERM method is well-suited for this medical data visualization task. To illustrate this idea, we adopt medical data of the spatial acuity for pain in human body from Ref. [20]. The spatial acuity was assessed by measuring the 2-point discrimination (2PD) thresholds (in cm) at different parts of the body, where a smaller threshold implies a higher spatial acuity. To represent the data geometrically, we deform a 3D human body model using our proposed volumetric density-equalizing reference map method, with the density being

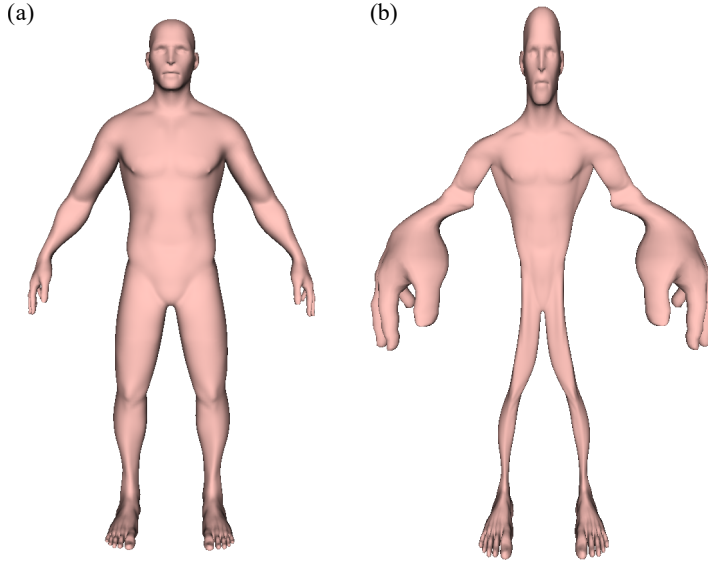


Fig. 8 Producing a cortical homunculus using our proposed VDERM method. (a) A normal 3D human body model. (b) A deformed human model produced by our proposed method representing the spatial acuity of pain [20].

the reciprocal of the 2PD threshold of each part of the body divided by the volume of it:

$$\rho^0 = \begin{cases} (1/0.6)/\text{Volume of the fingers} & \text{for the fingers,} \\ (1/1.1)/\text{Volume of the hand palms} & \text{for the palms,} \\ (1/2.7)/\text{Volume of the dorsum of the hands} & \text{for the dorsum of the hands,} \\ (1/2.1)/\text{Volume of the arms} & \text{for the arms,} \\ (1/1.2)/\text{Volume of the head} & \text{for the head,} \\ (1/1.4)/\text{Volume of the shoulders} & \text{for the shoulders,} \\ (1/2.5)/\text{Volume of the lower back} & \text{for the lower back,} \\ (1/2.9)/\text{Volume of the thighs} & \text{for the thighs,} \\ (1/3.3)/\text{Volume of the calves} & \text{for the calves,} \\ (1/3.5)/\text{Volume of the dorsum of the foot} & \text{for the dorsum of the foot,} \\ (1/1.3)/\text{Volume of the foot soles} & \text{for the foot soles.} \end{cases} \quad (33)$$

The density at the other parts of the body is defined using the closest parts with data available. The density at the remaining regions of the volumetric domain is set to be the average density of the human body, making those regions a “sea” to allow for a free-boundary deformation of the human body model.

Fig. 8 shows the deformed human body model obtained by our proposed VDERM method. It can be observed that the hands are expanded significantly, as they are with the highest spatial acuity. The deformed model and the sculptures in Fig. 7 are highly similar, except for the mouth/lips at which the acuity data is not provided in Ref. [20]. This example shows that our proposed method is useful for medical data visualization.

5.1.2 Sociological data visualization

Our proposed VDERM method can also be used for visualizing sociological data. For instance, airlines have multiple flight classes, where passengers of different flight classes enjoy service and accommodation at different levels. It is well known that the ticket price of different travel classes can be significantly different. Here, we apply our proposed method for visualizing the ticket price of different travel classes on an aircraft. We collect the ticket price data of a round-trip direct flight between New York (JFK) and Hong Kong (HKG) (departure date: March 1, 2020; returning date: March 8, 2020; data retrieved on December 3, 2019 from the American Airlines website [22]). Both the departure flight and the returning flight are operated on a Boeing 777-300ER aircraft, in which there are in total 6 first class seats, 53 business class seats, 34 premium economy class seats and 182 economy class seats [23]. The ticket prices for a first, business, premium economy, and economy class seat are \$16039, \$6922, \$2542, and \$941 respectively.

To give a geometrical interpretation of the ticket price difference, we deform a 3D model of the Boeing 777-300ER aircraft using our proposed volumetric density-equalizing reference map method, with the density being the ticket price divided by the volume of the cabin:

$$\rho^0 = \begin{cases} 16039/\text{Volume of the first class cabin} & \text{for the first class cabin,} \\ 6922/\text{Volume of the business class cabin} & \text{for the business class cabin,} \\ 2542/\text{Volume of the premium economy class cabin} & \text{for the premium economy class cabin,} \\ 941/\text{Volume of the economy class cabin} & \text{for the economy class cabin.} \end{cases} \quad (34)$$

Again, the density at the remaining regions of the volumetric domain is set to be the average density of the above to allow for a free-boundary deformation. Note that the division by the volume of the cabins is to ensure that the ticket price ratio is equal to the ratio of the volume of the entire cabins. Fig. 9(a),(b) show the original Boeing 777-300ER aircraft and the deformed aircraft. It can be observed that the first class cabin expands in all directions, while the economy class cabin shrinks significantly. This reflects the large ticket price difference between the first class seats and the economy class seats.

We can also estimate the total revenue of different ticket classes by multiplying the ticket price by the total number of seats in each class. This gives a total revenue of \$96234, \$366866, \$86428, and \$171262 for the four classes respectively. Similarly, we obtain a volumetric density-equalizing map using the following density:

$$\rho^0 = \begin{cases} 96234/\text{Volume of the first class cabin} & \text{for the first class cabin,} \\ 366866/\text{Volume of the business class cabin} & \text{for the business class cabin,} \\ 86428/\text{Volume of the premium economy class cabin} & \text{for the premium economy class cabin,} \\ 171262/\text{Volume of the economy class cabin} & \text{for the economy class cabin.} \end{cases} \quad (35)$$

Fig. 9(c) shows the deformed aircraft. Unlike the previous result, this deformed aircraft is not significantly different from the original model. We observe that the economy class cabin is slightly shrunk, while the business class cabin is slightly expanded. This suggests that the total revenue of each ticket class, with the amount of cabin space used taken into account, is in fact similar for the four classes.

The above examples show that our proposed VDERM method is capable of producing deformations to convey sociological information in volumetric domains.

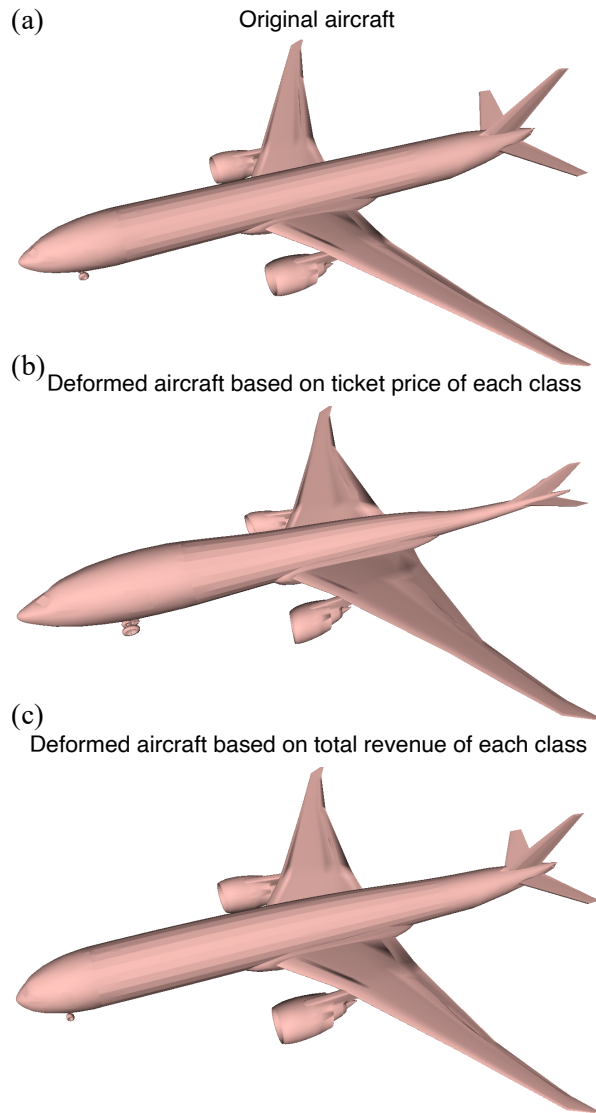


Fig. 9 Visualization of the economics of airline class using our proposed VDERM method. (a) A 3D model of the Boeing 777-300ER aircraft. (b) The deformed aircraft produced by our proposed method based on the ticket price for each ticket class. The magnification at the front and the shrinkage at the end (in x -, y -, and z -directions) reflect the large ticket price difference between the first class seats and the economy class seats. (c) The deformed aircraft produced by our proposed method based on the total revenue for each ticket class. The relatively small deformation suggests that the total revenue of each ticket class, with the amount of cabin space taken into account, is similar for the four classes.

5.2 Deformation-based shape modeling

Besides volumetric data visualization, our proposed VDERM method can also be used for deformation-based shape modeling. Suppose we are given a mesh in \mathbb{R}^3 . By defining different

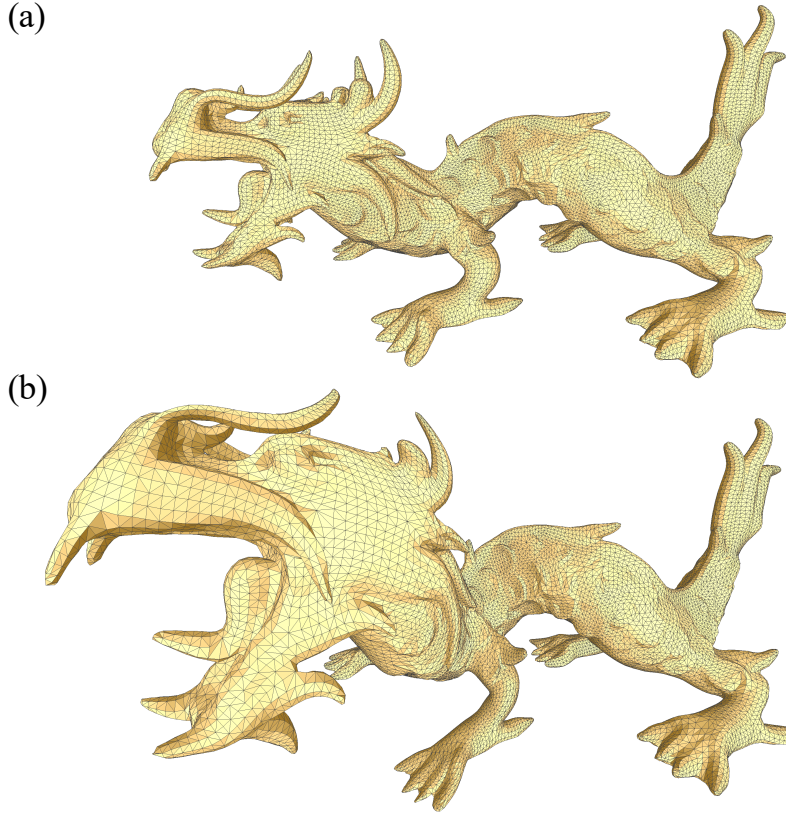


Fig. 10 Deformation of a dragon model using our volumetric density-equalizing reference map. (a) The original dragon. (b) The deformed dragon with the head enlarged.

density values at different parts of an underlying 3D grid and computing the volumetric density-equalizing map, we obtain a deformation of the grid, which induces a deformation of the object. Using this idea, we can easily achieve different shape modeling effects.

Fig. 10 shows an experiment on a dragon model adapted from The Stanford 3D Scanning Repository [18]. In this experiment, we aim to magnify the head of the dragon while keeping the body shape unchanged. On an underlying 3D grid of size $32 \times 32 \times 32$, we set the density to be

$$\rho^0 = \begin{cases} 10 & \text{around the head of the dragon,} \\ 1 & \text{elsewhere.} \end{cases} \quad (36)$$

We then compute the volumetric density-equalizing map on the underlying grid and obtain the induced deformation of the dragon. It can be observed that the head of dragon is effectively enlarged, while the main body of the dragon is almost unchanged. This demonstrates the advantage of our approach in accurately producing a deformed model that satisfies the desired effect. Also, while the input density ρ is with a sharp discontinuity, the density diffusion process produces a smooth transition from the high-density region to the low-density region. As shown in Fig. 10, the neck of the dragon is also slightly enlarged under the deformation because of the density diffusion process. This demonstrates another advantage of our approach in producing a naturally and smoothly deformed model.

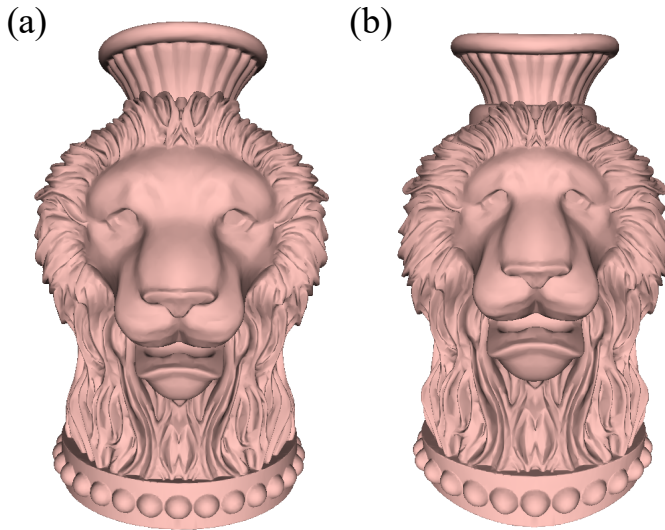


Fig. 11 Deformation of a lion vase model using our proposed VDERM method. (a) The original lion vase. (b) The deformed lion vase with the facial expression changed.

Fig. 11 shows another experiment of shape modeling using our approach. This time, we consider a lion vase model adapted from the 3D Segmentation Benchmark [24]. To change the facial expression of the lion, we define a smaller density value around the forehead of the lion on the underlying grid and a larger density value around the chin of it. Then we compute the volumetric density-equalizing map and obtain the induced deformation of the lion vase. It can be observed that the facial expression of the lion is changed naturally under our deformation-based approach.

5.3 Shape morphing

It is noteworthy that our proposed approach deforms the underlying grid continuously with an iterative scheme, and hence the intermediate states of the deformation of the grid can be used for producing a continuous change of the 3D model. Fig. 12 shows multiple snapshots of the facial expression change produced by computing the induced deformation of the lion vase with respect to the intermediate states of the deformed 3D grid at different time points. It can be observed that the facial expression changes continuously. This experiment suggests that our deformation-based method is also advantageous for applications which focus not only on the final shape but also the intermediate states, such as object morphing.

5.4 Adaptive remeshing

Our proposed VDERM method can also be applied for remeshing, which aims at improving the discretization of meshes [25, 26]. More specifically, suppose our goal is to construct a tetrahedral mesh for a genus-0 closed surface. Furthermore, the tetrahedral mesh density of some prescribed regions in the volumetric domain D is desired to be higher. To achieve this,

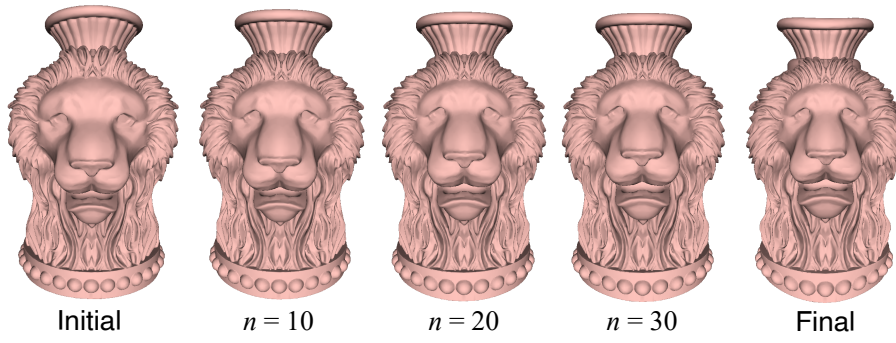


Fig. 12 An animation of the facial expression change of the lion produced using the intermediate states of the volumetric density-equalizing reference maps.

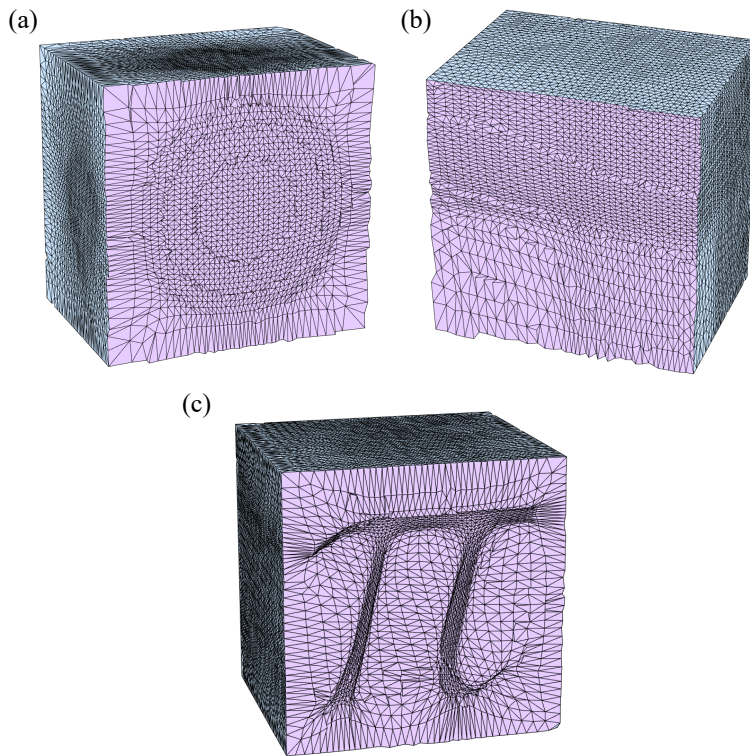


Fig. 13 Examples of adaptive remeshing achieved by our proposed VDERM method. (a) A cube with the central part refined. (b) A cube with eight different tetrahedral mesh densities. (c) A cube with a π shape having a higher mesh density. For better visualization, the surface is colored in blue and the interior tetrahedral elements are colored in purple.

we run our proposed algorithm with a larger ρ at those regions and obtain the volumetric density-equalizing reference map ξ_{final} . Then, we construct a uniform tetrahedral mesh in the deformed domain using DistMesh [27], and map it back to the original domain D using

ξ_{final} . This puts more tetrahedral elements in the regions with a larger prescribed ρ and less elements in the regions with a smaller ρ , thereby achieving a tetrahedral mesh with adaptive mesh density.

Three examples are given in Fig. 13. For Fig. 13(a), the central part of the domain is desired to be with a higher tetrahedral mesh density. To achieve this, we set

$$\rho^0 = \begin{cases} 10 & \text{around the center of the domain,} \\ 1 & \text{elsewhere.} \end{cases} \quad (37)$$

For Fig. 13(b), we set ρ^0 by Eq. (32) in order to achieve eight different tetrahedral mesh densities at the eight corners. For Fig. 13(c), we set ρ^0 to be 10 inside the π shape and 1 otherwise, in order to achieve a higher mesh density at the π shape. In all examples, the resulting volumetric density-equalizing reference map ξ_{final} is capable of mapping a uniform tetrahedral mesh to the original domain and producing the desired tetrahedral meshes with adaptive mesh density.

DistMesh [27] also allows for adaptive remeshing, by specifying an edge length function expression that yields the target adaptive resolution. This differs from our approach, where we can exactly specify the volumes of mesh elements throughout the domain. Our approach allows us to deform a mesh to precisely control element volume while maintaining mesh topology, although it can result in elongated triangles such as those shown in Fig. 13(c). A hybrid approach combining both DistMesh and VDERM provides a flexible platform for generating meshes for numerical computations (e.g. solving PDEs) with added precision in certain regions.

6 Discussion

In this work, we have proposed a method for computing volumetric density-equalizing reference maps with a prescribed density defined in a solid domain in \mathbb{R}^3 . This is the first work on the formulation of density-equalizing map in 3D for diffusion-based deformation of volumetric domains. We have demonstrated the effectiveness of the proposed VDERM method via numerical experiments and discussed potential applications of the proposed method to volumetric data visualization, remeshing and deformation-based shape modeling. We have further introduced a novel use of the entire density-diffusion process for time-dependent applications such as object morphing.

One possible future work is to extend our VDERM method for achieving more sophisticated shape modeling effects. Note that our approach currently focuses on deformations related to relative volume change of voxels. We may consider adding some extra steps with other deformation energies [28] throughout the density-equalization iterations to produce other type of deformations. For instance, we may enable rotations of the velocity vectors at certain regions throughout the iterations in order to achieve a rotational effect on the deformed object. It will also be interesting to explore if our method can be combined with landmark-matching mapping methods [33, 34] to achieve a wider range of shape deformations that satisfy prescribed landmark correspondences in 3D.

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability

The data are available from the corresponding author on request.

Acknowledgments

This work was supported in part by the Croucher Foundation (to G. P. T. Choi), the Harvard Quantitative Biology Initiative and the NSF-Simons Center for Mathematical and Statistical Analysis of Biology at Harvard, award number #1764269 (to G. P. T. Choi), and the Applied Mathematics Program of the U.S. Department of Energy (DOE) Office of Science Advanced Scientific Computing Research under contract DE-AC02-05CH11231 (to C. H. Rycroft). We thank Dr. Anselm Hui (Prince of Wales Hospital) for useful discussion on medical data visualization.

References

1. M. T. Gastner and M. E. J. Newman, *Diffusion-based method for producing density-equalizing maps*. Proc. Natl. Acad. Sci., 101(20), 7499–7504, 2004.
2. D. B. Wake and V. T. Vredenburg, *Are we in the midst of the sixth mass extinction? A view from the world of amphibians*. Proc. Natl. Acad. Sci., 105(1), 11466–11473, 2008.
3. D. Dorling, M. Newman, and A. Barford, *The atlas of the real world: mapping the way we live*. Thames & Hudson, 2010.
4. R. K. Pan, K. Kaski, and S. Fortunato, *World citation and collaboration networks: uncovering the role of geography in science*. Sci. Rep., 2, 902, 2012.
5. H. D. Matthews, T. L. Graham, S. Keeverian, C. Lamontagne, D. Seto, and T. J. Smith, *National contributions to observed global warming*. Environ. Res. Lett., 9(1), 014010, 2014.
6. P. J. Dodd, C. Sismanidis, and J. A. Seddon, *Global burden of drug-resistant tuberculosis in children: a mathematical modelling study*. Lancet Infect. Dis., 16(10), 1193–1201, 2016.
7. D. Ballas, D. Dorling, and B. Hennig, *Analysing the regional geography of poverty, austerity and inequality in Europe: a human cartographic perspective*. Reg. Stud., 51(1), 174–185, 2017.
8. G. P. T. Choi and C. H. Rycroft, *Density-equalizing maps for simply connected open surfaces*. SIAM J. Imaging Sci., 11(2), 1134–1178, 2018.
9. M. T. Gastner, V. Seguy, and P. More, *Fast flow-based algorithm for creating density-equalizing map projections*. Proc. Natl. Acad. Sci., 115(10), E2156–E2164, 2018.
10. G. P. T. Choi, B. Chiu, and C. H. Rycroft, *Area-preserving mapping of 3D carotid ultrasound images using density-equalizing reference map*. IEEE Trans. Biomed. Eng., 1–11, 2020.
11. K. Kamrin, C. H. Rycroft, and J.-C. Nave, *Reference map technique for finite-strain elasticity and fluid–solid interaction*. J. Mech. Phys. Solids, 60(11), 1952–1969, 2012.
12. B. Valkov, C. H. Rycroft, and K. Kamrin, *Eulerian method for multiphase interactions of soft solid bodies in fluids*. J. Appl. Mech., 82(4), 041011, 2015.
13. C. H. Rycroft, C.-H. Wu, Y. Yu, and K. Kamrin, *Reference map technique for incompressible fluid–structure interaction*. Preprint, arXiv:1810.03015.
14. M. E. Gurtin, E. Fried, and L. Anand, *The mechanics and thermodynamics of continua*. Cambridge University Press, 2010.
15. S. Govindjee and P. A. Mihalic, *Computational methods for inverse finite elastostatics*. Comput. Methods Appl. Mech. Eng., 136(1–2), 47–57, 1996.
16. V. D. Fachinotti, A. Cardona, and P. Jetteur, *Finite element modelling of inverse design problems in large deformations anisotropic hyperelasticity*. Int. J. Numer. Methods Eng., 74(6), 894–910, 2008.
17. R. Courant, K. Friedrichs, H. Lewy, *Über die partiellen differenzengleichungen der mathematischen physik*. Math. Ann. (in German), 100(1), 32–74, 1928.
18. The Stanford 3D Scanning Repository. <http://graphics.stanford.edu/data/3Dscanrep/>
19. E. Marieb and K. Hoehn, *Human anatomy and physiology*. Pearson Benjamin Cummings: San Francisco, 2007.
20. F. Mancini, A. Bauleo, J. Cole, F. Lui, C. A. Porro, P. Haggard, and G. D. Iannetti, *Whole-body mapping of spatial acuity for pain and touch*. Ann. Neurol., 75(6), 917–924, 2014.

21. *Cortical Homunculus*. https://en.wikipedia.org/wiki/Cortical_homunculus
22. *American Airlines*. <https://www.aa.com>
23. *SeatGuru*. <https://www.seatguru.com>
24. *3D Segmentation Benchmark*. <http://193.48.251.101/3dsegbenchmark/bust.html>
25. G. P.-T. Choi, K. T. Ho, and L. M. Lui, *Spherical conformal parameterization of genus-0 point clouds for meshing*. *SIAM J. Imaging Sci.*, 9(4), 1582–1618 (2016).
26. G. P.-T. Choi, M. H.-Y. Man, and L. M. Lui, *Fast spherical quasiconformal parameterization of genus-0 closed surfaces with application to adaptive remeshing*. *Geom. Imaging Comput.*, 3(1–2), 1–29, 2016.
27. P.-O. Persson and G. Strang, *A simple mesh generator in MATLAB*. *SIAM Rev.*, 46(2), 329–345, 2004.
28. K. Hildebrandt, C. Schulz, C. V. Tycowicz, and K. Polthier, *Interactive surface modeling using modal analysis*. *ACM Trans. Graph.*, 30(5), 119, 2011.
29. Y. Saad, *ILUT: A dual threshold incomplete LU factorization*. *Numer. Linear Algebra Appl.*, 1(4), 387–402, 1994.
30. C.-J. Lin and J. J. Moré, *Incomplete Cholesky factorizations with limited memory*. *SIAM J. Sci. Comput.*, 21(1), 24–45, 1999.
31. M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*. *Proc. IEEE*, 93(2), 216–231, 2005.
32. M. Frigo and S. G. Johnson, *FFTW*. <http://www.fftw.org>
33. 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(1), 67–94, 2015.
34. Y. T. Lee, K. C. Lam, and L. M. Lui, *Landmark-matching transformation with large deformation via n -dimensional quasi-conformal maps*. *J. Sci. Comput.*, 67(3), 926–954, 2016.