SIAM J. SCI. COMPUT. © 2021 Society for Industrial and Applied Mathematics
Vol. 43, No. 5, pp. A3155–A3183

A DIFFUSION-DRIVEN CHARACTERISTIC MAPPING METHOD FOR PARTICLE MANAGEMENT[*]

XI-YUAN YIN[†], LINAN CHEN[†], AND JEAN-CHRISTOPHE NAVE[‡]

Abstract. We present a novel particle management method using the characteristic mapping framework. In the context of explicit evolution of parametrized curves and surfaces, the surface distribution of marker points created from sampling the parametric space is controlled by the area element of the parametrization function. As the surface evolves, the area element becomes uneven and the sampling suboptimal. In this method we maintain the quality of the sampling by precomposition of the parametrization with a deformation map of the parametric space. This deformation is generated by the velocity field associated to the diffusion process on the space of probability distributions and induces a uniform redistribution of the marker points. We also exploit the semigroup property of the heat equation to generate a submap decomposition of the deformation map which provides an efficient way of maintaining evenly distributed marker points on curves and surfaces undergoing extensive deformations.

Key words. particle management, equiareal parametrization, characteristic mapping method, heat equation

AMS subject classifications. 65D18, 65M25, 65M50, 65M75

DOI. 10.1137/20M1364357

1. Introduction.

The parametrization of a curve or surface has many applications in computer graphics, computational geometry, and geometric modeling. In scientific computing, where physics modeling and simulation require solving surface PDEs, a parametric representation is often useful for simplifying the surface equations or for generating computational meshes. This parametrization is often time-dependent, such as in the case of fluid interfaces in multiphase flows, where the extensive deformation of the surfaces can deteriorate their numerical accuracy. In this context, the numerical resolution of the surfaces are of special importance since the simulation of interfacial dynamics often relies on solving stiff, high order PDEs [16, 17]. Of particular interest to interface dynamics and phase separation is the Cahn–Hilliard equation [18, 19, 11], where proper spatial resolution is crucial for capturing topological transitions and for improving computational efficiency. In order to maintain a good representation of a parametric surface, current research largely focuses on two main desirable properties for the parametrization: angle-preservation and area-preservation.

An angle-preserving, or conformal, parametrization guarantees that the pulled back metric on the parametric space differs from the flat metric only by a scalar multiplicative factor. There has been extensive research in the field of conformal maps and application to surface parametrization. Conformal parametrization of genus zero surfaces and its application to patching more complicated surfaces has been

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section October 8, 2020; accepted for publication (in revised form) May 26, 2021; published electronically September 14, 2021.

https://doi.org/10.1137/20M1364357

Funding: The work of the first author was partially supported by FRQNT B2X (Fonds de recherche du Quebec) and by Hydro-Quebec. The work of the second and third authors was partially supported by the NSERC Discovery program (Natural Sciences and Engineering Research Council).

[†] Department of Mathematics and Statistics, McGill University, Montreal H3A 0B9, PQ, Canada (xi.yin@mail.mcgill.ca, lnchen@math.mcgill.ca).

[‡] Corresponding author. Department of Mathematics and Statistics, McGill University, Montreal H3A 0B9, PQ, Canada (jcnave@math.mcgill.ca).

studied in [21, 22]. For a given surface, conformal parametrizations are not unique and are generally not area-preserving. The area stretching will depend on the chosen conformal parametrization and on the intrinsic curvature of the surface; it may even grow exponentially in protruding regions of the surface. This motivated [26] to design a method which generates a global conformal parametrization minimizing some chosen energy functional of the area element.

For curved surfaces, it is generally not possible to find a parametrization which is both conformal and area-preserving. For certain applications such as surface sampling, an equiareal parametrization is preferred as the density of the sample points will be uniform over the surface. For many Lagrangian particle methods used in interface tracking, appropriate particle redistribution and reinitialization methods are necessary to maintain the accuracy of the surface representation and to prevent artificial topological changes [24, 9]. Methods for computing equiareal parametrizations include [4, 44], which propose a relaxation algorithm based on some stretch factor computed from mesh point distances. Winslow's rezoning algorithm [42] solves a system of variable diffusion equations on the coordinate functions in order to prescribe the Jacobian determinant. Many of these approaches can also be expressed in the framework of adaptive moving mesh methods where a moving mesh PDE is used to evolve a numerical mesh in order to better resolve the different scales that a solution to an evolution equation may exhibit [25]. For instance, in [8], a moving mesh PDE was employed as a redistribution map to reinitialize sample point positions in order to resolve the multiscale solutions generated by convection dominated flows.

We propose in this paper a novel framework for generating equiareal parametrizations of time-dependent curves or surfaces using the characteristic mapping (CM) method, a numerical framework for the advection equation based on the gradient-augmented level-set method [34] and the reference map technique [28]. The CM method was used in [32, 43] to solve the linear transport and the two-dimensional (2D) incompressible Euler equations; it consists in computing the backward-in-time deformation map generated by an incompressible velocity field. Quantities transported under this velocity can be directly obtained as the pullback of the initial condition by this map. In this paper, we use the CM method on a compressible velocity field in order to transport a density function evolving under its continuity equation. The evolving density distribution can be obtained as pullback by the map, i.e., as a volume form. For equiareal parametrization, we define a probability distribution on the parametric space corresponding to a constant scaling of the area element associated with the parametrization. This density is interpreted as the initial condition of a diffusion equation which we write in conservation form. The velocity field associated with this conservation law generates a characteristic map which transports the initial density toward a uniform density. This redistribution map is precomposed with the parametrization function to generate an equiareal parametrization. Note that since the image space of the parametrization is unchanged, this redistribution does not affect the accuracy of the surface location while improving its numerical representation.

The idea of diffusion flows has been used extensively outside the context of surface parametrization. For fluid simulations, the diffusion velocity method, as a generalization to particle methods, has seen many applications in numerical simulations of transport-dispersion and of viscous flows, for instance, in [29, 33]. Further extensive theoretical and numerical analysis for the blob-particle method for linear and non-linear diffusion can also be found in [13]. It is well-known that the diffusion equation can be viewed as the L^2-gradient descent of the Dirichlet energy functional; this allows for fast and robust methods that approximate an uniform redistribution of the density. In the context of optimal transport, the diffusion equation can also be seen as a gradient descent of the Gibbs–Boltzmann entropy under the Wasserstein-2 metric as analyzed in [27]. There has been extensive research in the field of optimal transport specifically concerning gradient descent and geodesic flows [7, 35, 12, 14, 7]. Notably, the fluid mechanics interpretation of the Monge–Kantorovich problem in [5, 6] is most closely related to the CM framework presented in this paper. An overview of the main concepts in optimal transport and gradient flows in the space of probability densities can be found in [37] and more complete surveys in [1, 36, 41]. In the context of generating equiareal parametrization, the optimal transport methods have also been investigated, for instance, in [45, 39, 40]. Recently, for the purpose of global weather forecasting and climate modeling, McRae, Cotter, and Budd presented a moving mesh method for generating r-adaptive triangular meshes on the sphere [31, 10]. This method uses the Monge–Ampère equation to equidistribute a monitor function; the generated meshes are adapted to represent the various scales of the problem on a spherical surface.

There are three main features of the method presented in this paper. First, we propose a novel framework for the application of density transport to the problem of maintaining an equiareal parametrization of a moving surface: the area element of the parametrization is interpreted as a probability density whose change in time is penalized by an L^2-gradient descent on its Dirichlet energy. Second, the reparametrization method proposed here acts purely on the parametric space, and the new parametrization function is obtained as a precomposition of a transport map with the original parametrization. Consequently, the reparametrization does not affect the precision of the surface and its movements but improves its numerical resolution. Third, the method provides arbitrary resolution of the parametrization. This is possible on one hand due to the functional definition of the transport maps in the parametric and the ambient spaces provided by the semi-Lagrangian approach of the CM method. Indeed, once the redistribution map is computed, resampling of the surface can be readily done by map evaluations; no additional computations are needed. On the other hand, the CM method uses the semigroup structures of the deformation maps to leverage the separation of scales: long-time deformations can be decomposed into short-time submaps which can be computed on coarser grids [43]. Since the parametrized surfaces can undergo arbitrary deformations of various scales, the method is effective in maintaining arbitrary resolution of the parametrization using coarse grid computations.

The paper is organized as follows. In section 2, we present the mathematical framework of the CM method for the transport of densities as well as the diffusion flow used for the area redistribution. Section 3 provides an overview of the numerical implementation of the method with some error estimates. We present in section 4 the application of the redistribution method to the maintenance of an equiareal parametrization for moving surfaces. This section also contains the CM framework for the evolution of surfaces in a 3D ambient flow, the explicit parametric counterpart of the level-set method used in [32]. Section 5 presents some numerical results for the application of this method to the evolution of parametric curves and surfaces. Finally, section 6 contains some concluding remarks and potential directions for future research.

2. Mathematical formulation.

2.1. Characteristic mapping method for density transport. The use of the CM method for linear advection and nonlinear advection in the case of the Euler equations has been studied in [32, 43]. In this section, we extend the CM framework to the transport of density distributions.

For a given velocity field $\boldsymbol{u}$ defined on some domain $U \subset \mathbb{R}^d$, characteristic curves of the velocity are given by the solution of the IVP

$$\frac{d}{dt}\boldsymbol{\gamma}(t) = \boldsymbol{u}(\boldsymbol{\gamma}(t), t), \qquad t \in \mathbb{R}_+, \tag{2.1a}$$

$$\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}_0, \qquad \boldsymbol{\gamma}_0 \in U. \tag{2.1b}$$

The backward characteristic map $\boldsymbol{X}$ is the backward solution operator of the characteristic ODE in the sense that

$$\boldsymbol{X}(\boldsymbol{\gamma}(t), t) = \boldsymbol{\gamma}_0 \tag{2.2}$$

for all initial conditions $\boldsymbol{\gamma}_0$ and all $t \in \mathbb{R}_+$.

We can check that $\boldsymbol{X}$ satisfies the PDE

$$\partial_t \boldsymbol{X} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{X} = 0 \qquad \forall (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{2.3a}$$

$$\boldsymbol{X}(\boldsymbol{x}, 0) = \boldsymbol{x}. \tag{2.3b}$$

The characteristic map possesses a semigroup structure which allows for the decomposition of a long-time map into the composition of several submaps. We denote by $\boldsymbol{X}_{[\tau_{i+1},\tau_i]}$ the backward characteristic map for the time interval $[\tau_i, \tau_{i+1}]$. This means that for any characteristic curve $\boldsymbol{\gamma}$ satisfying (2.1), we have that

$$\boldsymbol{X}_{[\tau_{i+1},\tau_i]}(\boldsymbol{\gamma}(\tau_{i+1})) = \boldsymbol{\gamma}(\tau_i). \tag{2.4}$$

Remark 1. Note that the notation for the characteristic maps used here is slightly different from the one used in [32]. The backward map in the interval $[\tau_i, \tau_{i+1}]$ is denoted $\boldsymbol{X}_{[\tau_{i+1},\tau_i]}$ with the endpoints of the interval inverted to highlight that it is the backward in time transformation.

The global-time characteristic map can then be split into submaps using the following decomposition:

$$\boldsymbol{X}(\cdot, t) = \boldsymbol{X}_{[t,0]} = \boldsymbol{X}_{[\tau_1,0]} \circ \boldsymbol{X}_{[\tau_2,\tau_1]} \circ \cdots \circ \boldsymbol{X}_{[\tau_{m-1},\tau_{m-2}]} \circ \boldsymbol{X}_{[t,\tau_{m-1}]}. \tag{2.5}$$

We will denote the global backward characteristic map $\boldsymbol{X}_{[t,0]}(\boldsymbol{x})$ by $\boldsymbol{X}_B(\boldsymbol{x}, t)$.

For smooth, divergence-free velocity fields, these characteristic maps are known to be diffeomorphisms for all times. Further details on this method and its application to the advection equation can be found in [43].

Without the divergence-free assumption on the velocity field, the diffeomorphism property is not guaranteed. The evolution of the Jacobian determinant of the characteristic maps can be computed along characteristic curves using Jacobi's formula:

$$\begin{aligned} \frac{d}{dt}\det\left(\nabla \boldsymbol{X}_B(\boldsymbol{\gamma}(t), t)\right) &= \operatorname{tr}\left(\operatorname{adj}(\nabla \boldsymbol{X}_B)\frac{d}{dt}\nabla \boldsymbol{X}_B(\boldsymbol{\gamma}(t), t)\right) \\ &= \operatorname{tr}\left(-\operatorname{adj}(\nabla \boldsymbol{X}_B) \cdot \nabla \boldsymbol{X}_B \cdot \nabla \boldsymbol{u}(\boldsymbol{\gamma}(t), t)\right) = -\det(\nabla \boldsymbol{X}_B)\nabla \cdot \boldsymbol{u}, \end{aligned} \tag{2.6}$$

where adj denotes the adjugate matrix. That is,

$$\frac{d}{dt}\log\det(\nabla \boldsymbol{X}_B) = (\partial_t + \boldsymbol{u} \cdot \nabla)\log\det(\nabla \boldsymbol{X}_B) = -\nabla \cdot \boldsymbol{u}, \tag{2.7}$$

and the absence of finite-time blow-up of the time-integral of the divergence on all characteristic curves is required for the characteristic maps to remain diffeomorphisms.

Under the assumption that the velocity $\boldsymbol{u}$ generates diffeomorphic characteristic maps, we can extend the CM method to the density transport problem. Consider the continuity equation

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \forall (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{2.8a}$$

$$\rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}). \tag{2.8b}$$

In this paper, for the purpose of particle management, we will restrict our attention to the case of positive densities ρ bounded away from 0. That is, we assume there exists some $a > 0$ such that $\rho(\boldsymbol{x}, t) \geq a$ for all $(\boldsymbol{x}, t) \in U \times \mathbb{R}_+$. We also assume that ρ_0 is a probability density, in particular, (2.8) implies that ρ will then remain a probability density for all t.

The solution $\rho(\boldsymbol{x}, t)$ can then be obtained from the characteristic map as follows:

$$\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{X}_B(\boldsymbol{x}, t)) \det \nabla \boldsymbol{X}_B. \tag{2.9}$$

The submap decomposition (2.5) can also be applied to this pullback.

2.2. Density redistribution. In this section we apply the deformation map $\boldsymbol{X}_B$ in the context of measure transport. Let m denote the uniform probability measure on U, that is, $m(U) = 1$ with constant density with respect to the Lebesgue measure. Let μ be the initial probability measure continuous with respect to uniform with density $\rho_0 \; dm$. We define μ_t as the pullback measure of μ by $\boldsymbol{X}_B(\cdot, t)$. That is, $\boldsymbol{X}_B$ is a deterministic coupling between the probability spaces $(U, \mathcal{B}, \mu)$ and $(U, \mathcal{B}, \mu_t)$. Throughout this paper, the subscript t following a variable will only denote the evaluation at time t. From (2.8), we have that μ_t has density $d\mu_t = \rho_t \; dm$.

The characteristic map is generated from a chosen velocity field $\boldsymbol{u}$. For instance, one can use the Moser flow [15] given by

$$\boldsymbol{u}(\boldsymbol{x}, t) = \frac{\nabla \Delta^{-1}(\rho_0 - \rho_1)}{(1-t)\rho_0 + t\rho_1} \tag{2.10}$$

to generate a coupling between two strictly positive probability densities ρ_0 and ρ_1 such that $\boldsymbol{X}_{[1,0]}$ moves the base density ρ_1 to a target ρ_0.

For the method presented in this paper, $\boldsymbol{X}_B$ will be a coupling between the target density ρ_0 and the uniform density. As a result, pullback by $\boldsymbol{X}_B$ will "uniformize" ρ_0. For parametrized curves and surfaces, ρ_0 will correspond to the arclength or area-element functions defined on U. Precomposition of the parametrization with $\boldsymbol{X}_B$ will then yield an equiareal parametrization. We will examine this in further details in section 4.

The transport map in this paper is generated from a heat equation for the densities, $\partial_t \rho = \nu \Delta \rho$ (with Neumann or periodic boundary condition, ν is a constant diffusion constant), which we write as a continuity equation:

$$\partial_t \rho + \nu \nabla \cdot (-\nabla \rho) = \partial_t \rho + \nabla \cdot (\rho \cdot -\nu \nabla \log \rho) = 0 \qquad \forall (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{2.11a}$$

$$\partial_{\boldsymbol{n}} \rho = 0 \qquad \forall (\boldsymbol{x}, t) \in \partial U \times \mathbb{R}_+, \tag{2.11b}$$

$$\rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}). \tag{2.11c}$$

The redistribution map is then obtained from

$$\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{X}_B(\boldsymbol{x}, t)) \det \nabla \boldsymbol{X}_B(\boldsymbol{x}, t), \tag{2.12a}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = -\nu \nabla \log \rho(\boldsymbol{x}, t), \tag{2.12b}$$

$$\partial_t \boldsymbol{X}_B + (\boldsymbol{u} \cdot \nabla) \boldsymbol{X}_B = 0. \tag{2.12c}$$

The above flow has the property that ρ follows a heat equation; therefore, the maximum principle guarantees that the density stays bounded away from zero at all times. Furthermore, in the limit as $t \to \infty$, we have that ρ tends to its average $\bar{\rho} = 1$, and hence formally, $\boldsymbol{X}_B$ maps the uniform density to the target ρ_0. Finally the diffusion flow consists of an L^2-gradient descent of the energy or a W^2-gradient descent of the entropy. In fact, an implicit Euler step is equivalent to a minimizing movement scheme in the Wasserstein metric [27]. In this sense, the time evolution of the map can be seen as an iterative process which contracts to the desired transport map.

2.3. Energy estimates. The density ρ follows an L^2-gradient descent of the Dirichlet energy with the usual energy estimate:

$$E(t) = \frac{1}{2}\int_U (\rho(\boldsymbol{x},t) - 1)^2 \, dx \quad \text{with} \quad \frac{d}{dt}E(t) = -\int_U \nu \, |\nabla\rho|^2 \, dx. \tag{2.13}$$

Since ρ is a probability density, $\rho - 1$ has zero average for all t, so we can apply the Poincaré–Wirtinger inequality $\|\rho - 1\|_{L^2} \le C\|\nabla\rho\|_{L^2}$ to get the exponential decay in the L^2 energy:

$$\frac{d}{dt}E(t) \le -\alpha\nu E(t) \tag{2.14}$$

for some constant α.

In practice, we will compute the backward map to a sufficiently large time t to obtain a transport map between ρ_0 and a "close to uniform" distribution ρ_t. In terms of random variables, we let Y and Y_t be random variables taking value in U with probability densities $\rho_0(\boldsymbol{x})$ and $\rho(\boldsymbol{x},t)$, respectively. We have by construction that $\boldsymbol{X}_{[t,0]}(Y_t)$ has density ρ_0. Therefore, trivially, the random variables $\boldsymbol{X}_{[t,0]}(Y_t)$ converge in distribution to Y as $t \to \infty$. However, we are interested in the above convergence when Y_t is replaced by a fixed random variable with uniform distribution, as this allows us to use $\boldsymbol{X}_B$ to redistribute uniform random variables according to density ρ_0.

THEOREM 2.1. *Let Z be a random variable with uniform distribution on U and also define $Z_t = \boldsymbol{X}_{[t,0]}(Z)$ for each t. We have that Z_t converges to Y in distribution as $t \to \infty$.*

Proof. We have that Y has law μ. We also denote by ν_t the law of Z_t; this is the pushforward measure of m by $\boldsymbol{X}_{[t,0]}$. From definition we have

$$\nu_t(A) = m(\boldsymbol{X}^{-1}_{[t,0]}(A)) \quad \text{and} \quad \mu(A) = \mu_t(\boldsymbol{X}^{-1}_{[t,0]}(A)), \tag{2.15}$$

where μ_t is the law of Y_t and is the pullback measure of μ by $\boldsymbol{X}_{[t,0]}$. Therefore, we have

$$|\nu_t(A) - \mu(A)| = \left| m\left(\boldsymbol{X}^{-1}_{[t,0]}(A)\right) - \mu_t\left(\boldsymbol{X}^{-1}_{[t,0]}(A)\right)\right|. \tag{2.16}$$

Given that $\boldsymbol{X}_{[t,0]}$ is a diffeomorphism on U, we have established that

$$Y_t \xrightarrow{d} Z \iff Z_t \xrightarrow{d} Y, \tag{2.17}$$

since the total variation norms $\|\nu_t - \mu\|_{TV}$ and $\|\mu_t - m\|_{TV}$ are equal for any given t.

More precise estimates can also be obtained by looking at the measure densities from taking a Radon–Nikodym derivative with respect to m. We have that the measure densities are given by $d\nu_t = \frac{d\nu_t}{dm}\, dm$ and $d\mu = \frac{d\mu}{dm}\, dm$ with

$$\frac{d\mu}{dm} = \rho_0 = \rho_t(\boldsymbol{X}_{[t,0]}^{-1}) \det \nabla \boldsymbol{X}_{[t,0]}^{-1} \quad \text{and} \quad \frac{d\nu_t}{dm} = \det \nabla \boldsymbol{X}_{[t,0]}^{-1}. \tag{2.18}$$

The L^2 distance between the densities is then given by

$$\begin{aligned}\left\| \frac{d\nu_t}{dm} - \frac{d\mu}{dm} \right\|_{L^2} &= \left(\int_U \left| \rho_0 - \det \nabla \boldsymbol{X}_{[t,0]}^{-1} \right|^2 \, dm \right)^{\frac{1}{2}} \\ &= \left(\int_U \left| \rho_t \circ \boldsymbol{X}_{[t,0]}^{-1} - 1 \right|^2 (\det \nabla \boldsymbol{X}_{[t,0]}^{-1})^2 \, dm \right)^{\frac{1}{2}} \\ &= \left(\int_{\boldsymbol{X}_{[t,0]}^{-1}(U)} |\rho_t - 1|^2 \det \nabla \boldsymbol{X}_{[t,0]}^{-1} \, dm \right)^{\frac{1}{2}} \\ &\leq \|\rho_t - 1\|_{L^2} \| \det \nabla \boldsymbol{X}_{[t,0]}^{-1} \|_{L^\infty}^{\frac{1}{2}}. \end{aligned} \tag{2.19}$$

We can further bound $\det \nabla \boldsymbol{X}_{[t,0]}^{-1}$ using the maximum principle on ρ_t,

$$\det \nabla \boldsymbol{X}_{[t,0]}^{-1} = \frac{\rho_0(\boldsymbol{x})}{\rho_t(\boldsymbol{X}_{[t,0]}^{-1}(\boldsymbol{x}))} \leq \frac{\sup \rho_0}{\inf \rho_t} \leq \frac{\sup \rho_0}{\inf \rho_0}. \tag{2.20}$$

Using the decay rate (2.14), this effectively gives us an a priori estimate on the L^2 density error of $\sqrt{2 \frac{\sup \rho_0}{\inf \rho_0} E(0)} \cdot e^{-\alpha \nu t/2}$ as well as an a posteriori estimate $\sqrt{\frac{\sup \rho_0}{\inf \rho_t}} \cdot \|\rho_t - 1\|_{L^2}$. ∎

3. Numerical implementation.

3.1. Characteristic mapping method. In this section we briefly describe the CM method for a given velocity field $\boldsymbol{u}$. We follow the same framework as in [34, 38, 43], where more details can be found.

The CM method is essentially composed of a spatial interpolation operator and a time-stepping operator. For the method presented in this paper, we will use a Hermite interpolation in space. Let $\boldsymbol{G}$ be a grid on the domain U with grid points $\boldsymbol{x}_{i,j}$ and cells $C_{i,j}$. We define $\mathcal{V}_{\boldsymbol{G},m} \subset C^m(U)$ to be the space of Hermite interpolants on $\boldsymbol{G}$ of order $2m+1$. The interpolation operator $\mathcal{H}_{\boldsymbol{G}} : C^m(U) \to \mathcal{V}_{\boldsymbol{G},m}$ is then the projection operator defined by choosing the unique piecewise Hermite polynomial which matches the derivatives $\partial^{\vec{\alpha}} f$ at grid points of $\boldsymbol{G}$ for $\vec{\alpha} \in \{0, 1, \ldots, m\}^2$. In this paper, we will work exclusively with Hermite linear or cubic interpolants, i.e., $m = 0$ or 1. For a function f that is at least $2m+2$ times continuously differentiable in space, we have that

$$\left\| \partial^{\vec{\alpha}} \left(f - \mathcal{H}_{\boldsymbol{G}}[f] \right) \right\|_\infty = \mathcal{O}\left({h_x}^{2m+2-|\vec{\alpha}|} \right), \tag{3.1}$$

where h_x is the cell width of the grid $\boldsymbol{G}$ and $|\vec{\alpha}|$ denotes $\max(\alpha_1, \alpha_2)$. Inside each cell of $\boldsymbol{G}$, $\mathcal{H}_{\boldsymbol{G}}[f]$ is obtained from a tensor product of degree $2m+1$ polynomials in each dimension. From Taylor expansion, we see that the leading order term of the difference between f and this polynomial is $(x - x_i)^{m+1}(x - x_{i+1})^{m+1}$. This means that interpolation or extrapolation of the polynomial at a point $\epsilon < h_x$ away from a grid point yields $\mathcal{O}(\epsilon^{m+1-|\alpha|} {h_x}^{m+1})$ error.

Finally, it is important to note that we gain an order of accuracy when interpolating the first derivative at cell centers. Indeed, the leading order term in the error in the cell $[x_i, x_{i+1}]$ can be rewritten as $\left((x - \frac{1}{2}(x_i + x_{i+1}))^2 - \frac{1}{4}{h_x}^2\right)^{m+1}$. At $x = \frac{1}{2}(x_i + x_{i+1})$ this function has vanishing first derivatives for all m. In particular, this means that evaluation of the gradient and first mixed derivatives of a Hermite cubic interpolant is order $\mathcal{O}(h_x^4)$ accurate and $\mathcal{O}(h_x^2)$ for linear interpolants.

We outline below the algorithm for the time evolution of the characteristic map. We will denote by $\boldsymbol{\mathcal{X}}_B \in \mathcal{V}_{\boldsymbol{G},m}$ the numerical discretization of the exact characteristic map $\boldsymbol{X}_B$.

Given discrete time steps t_n, the time-discretization consists in approximating the "one-step map" $\boldsymbol{X}_{[t_{n+1},t_n]}$ obtained from integrating the velocity backward in time. We denote by $\tilde{\boldsymbol{u}}$ the approximate numerical velocity. This velocity is defined to be a constant-in-time interpolation for each interval $[t_n, t_{n+1}]$ and is obtained from approximating the true velocity at time t_{n+1}. The one-step map is defined using an Euler step:

$$\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}) = \boldsymbol{x} - h_t\tilde{\boldsymbol{u}}(\boldsymbol{x}), \tag{3.2}$$

and the characteristic map is updated by taking the Hermite interpolant of the composition

$$\boldsymbol{\mathcal{X}}_B^{n+1} = \mathcal{H}_{\boldsymbol{G}}\left[\boldsymbol{\mathcal{X}}_B^n \circ \tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}\right]. \tag{3.3}$$

The definition of $\tilde{\boldsymbol{u}}$ is presented in the next section. We note here that once $\tilde{\boldsymbol{u}}$ is defined for the interval $[t_n, t_{n+1}]$, the computation of $\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}$ can be split up into time subintervals using the same $\tilde{\boldsymbol{u}}$. This allows us to pick a small enough time step for the map updates to satisfy the CFL condition while reducing frequency of the calculations for $\tilde{\boldsymbol{u}}$.

The error of the map evolution is globally first order in time. The contribution of spatial error to the time stepping comes from the projection of the map composition on the space of Hermite interpolants $\mathcal{V}_{\boldsymbol{G}}$. The following lemma provides an estimate on this error.

Lemma 3.1. *For $f : U \to \mathbb{R}$ smooth and $\boldsymbol{T} : U \to U$ a diffeomorphism with $\boldsymbol{T} - \boldsymbol{I} = \mathcal{O}(\epsilon)$, that is, $\boldsymbol{T}$ is an order ϵ perturbation of the identity map. We have that, in the limit of small ϵ,*

$$\left\|\partial^{\vec{\alpha}}\left(\mathcal{H}_{\boldsymbol{G}}[f \circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f] \circ \boldsymbol{T}\right)\right\|_\infty = \mathcal{O}(\epsilon {h_x}^{2m+1-|\vec{\alpha}|}). \tag{3.4}$$

Proof. We write $\boldsymbol{T}$ as $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{x} + \epsilon\boldsymbol{v}(\boldsymbol{x})$. We consider first the Taylor expansion of f at $\boldsymbol{x}$:

$$f(\boldsymbol{x} + \epsilon\boldsymbol{v}(\boldsymbol{x})) = f(\boldsymbol{x}) + \sum_{i\in\{1,2\}} \epsilon v^i(\boldsymbol{x})\partial_i f(\boldsymbol{x}) + \sum_{i,j\in\{1,2\}} v^i(\boldsymbol{x})v^j(\boldsymbol{x})\frac{\epsilon^2}{2}\partial_{i,j}f(\boldsymbol{x}) + H.O.T. \tag{3.5}$$

Applying $\mathcal{H}_{\boldsymbol{G}}$ to $f \circ \boldsymbol{T}$ and to f, we get

$$\mathcal{H}_{\boldsymbol{G}}[f \circ \boldsymbol{T}] = \mathcal{H}_{\boldsymbol{G}}[f] + \sum_{i\in\{1,2\}} \epsilon\mathcal{H}_{\boldsymbol{G}}[v^i\partial_i f] + \frac{\epsilon^2}{2}\mathcal{H}_{\boldsymbol{G}}[v^iv^j\partial_{i,j}f] + H.O.T., \tag{3.6a}$$

$$\mathcal{H}_{\boldsymbol{G}}[f] \circ \boldsymbol{T} = \mathcal{H}_{\boldsymbol{G}}[f] + \sum_{i\in\{1,2\}} \epsilon v^i\partial_i\mathcal{H}_{\boldsymbol{G}}[f] + \frac{\epsilon^2}{2}v^iv^j\partial_{i,j}\mathcal{H}_{\boldsymbol{G}}[f] + H.O.T. \tag{3.6b}$$

We omit the interpolation on (3.6a) for all order ϵ and higher terms as replacing the interpolant with the interpolated function contributes an $\mathcal{O}(\epsilon^{|\vec{\alpha}|}{h_x}^{2m+2})$, $|\vec{\alpha}| \geq 1$, term which we absorb in the higher order terms. Similarly, we can use a Taylor expansion of $\mathcal{H}_G[f]$ in (3.6b) since the error incurred from extending an order $|\vec{\alpha}|$ derivative of the Hermite interpolant outside a cell is of order ${h_x}^{2m+2-|\vec{\alpha}|}$ and hence can also be absorbed in the higher order terms. Taking the difference of the two equations in (3.6), we get

$$\mathcal{H}_{\boldsymbol{G}}[f\circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f]\circ \boldsymbol{T} = \epsilon\partial_i(f-\mathcal{H}_{\boldsymbol{G}}[f])v^i + \frac{\epsilon^2}{2}\partial_{i,j}(f-\mathcal{H}_{\boldsymbol{G}}[f])v^iv^j + H.O.T. \tag{3.7}$$

In light of (3.1), we have that

$$\mathcal{H}_{\boldsymbol{G}}[f\circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f]\circ \boldsymbol{T} = \mathcal{O}(\epsilon {h_x}^{2m+1}) \tag{3.8}$$

and therefore

$$\left\|\partial^{\vec{\alpha}}\left(\mathcal{H}_{\boldsymbol{G}}[f\circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f]\circ \boldsymbol{T}\right)\right\|_\infty = \mathcal{O}(\epsilon {h_x}^{2m+1-|\vec{\alpha}|}). \tag{3.9}$$ □

3.2. Diffusion flow velocity.

The velocity $\boldsymbol{u} = -\nu\nabla\log\rho$ is used to evolve the characteristic map $\boldsymbol{X}_B$, whereas the density ρ is a volume form obtained from pullback by $\boldsymbol{X}_B$. For the numerical method, we will use a natural staggered grids approach for the discretization of these two quantities in order to obtain a spatially compact scheme. Similar primal-dual grid approaches for solving hyperbolic and parabolic equations using Hermite interpolation have been explored in [2, 3, 23], where the H^{m+1}-seminorm decreasing property of Hermite interpolation was used to design stable methods with high order accuracy.

Here, we use the grid $\boldsymbol{G}$ for the definition of the characteristic map $\boldsymbol{X}_B$ and the velocity field $\boldsymbol{u}$. We define $\boldsymbol{D}$ to be the staggered grid of $\boldsymbol{G}$ with grid points placed at the cell centers of $\boldsymbol{G}$, where the density at time t_n is sampled. We define the grid function $\rho^n_{\boldsymbol{D}}$ to be the evaluation of the following function on the grid $\boldsymbol{D}$:

$$\rho^n(\boldsymbol{x}) = \rho_0(\boldsymbol{\mathcal{X}}^n_B(\boldsymbol{x}))\det\nabla\boldsymbol{\mathcal{X}}^n_B(\boldsymbol{x}). \tag{3.10}$$

Since the evaluation of $\det\nabla\boldsymbol{\mathcal{X}}^n_B(\boldsymbol{x})$ occurs at the cell centers of $\boldsymbol{G}$, we gain an order of accuracy on the gradient. Therefore, $\rho^n_{\boldsymbol{D}}$ is accurate to order $\mathcal{O}({h_x}^{2m+2})$ when exact map values are provided. This staggered grid approach is similar in spirit to the primal-dual grid method developed by Appelo, Hagstrom, and Vargas in [3] to solve the scalar wave equation. In that method, a full time step update of the displacement function goes through two half-step integrations of the velocity function, where in each half-step, the velocity function is computed on a grid dual to the one where the previous displacement function was defined. In that case, the smoothing property of the Hermite interpolation from primal to dual grid yielded stable schemes using very high order interpolation.

In the method proposed here, the velocity field is defined from $\rho^n_{\boldsymbol{D}}$ by taking the log-gradient. In order to avoid the time step constraint of an explicit heat step, we compute the velocity corresponding to an implicit Euler step of the heat equation,

$$\tilde{\rho}_{\boldsymbol{D}} = (I - h_t\nu\Delta)^{-1}\rho^n_{\boldsymbol{D}}, \tag{3.11a}$$

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) = -\nu\nabla\mathcal{H}_{\boldsymbol{D}}\left[\log\tilde{\rho}_{\boldsymbol{D}}\right]. \tag{3.11b}$$

Indeed, the velocity field $-\nu\nabla\log\rho$ extracted from the heat equation is stiff in time and therefore the characteristic ODEs (2.1) are stiff. A fully implicit time-stepping method would be complicated and costly due to the coupling of ρ and $\boldsymbol{X}_B$; instead, we replace the true velocity $\boldsymbol{u}$ with the above constant-in-time implicit Euler approximation. The approach can therefore be thought of as a time-regularization of the characteristic ODEs using a relaxation which is consistent with the underlying heat equation. As reference and clarification, we summarize the redistribution algorithm in the pseudocode of Algorithm 3.1.

Algorithm 3.1 CM method for the heat equation on the space of probability densities.

```
Input: Initial density $\rho_0$, staggered grids $\boldsymbol{G}$ and $\boldsymbol{D}$, time step $h_t$, final time $T$
function HeatFlowMap($\rho_0$, $\nu$, $[0,T]$)
    Initialize $t \leftarrow 0$, $\boldsymbol{\mathcal{X}}_B \leftarrow \boldsymbol{id}$        ▷ $\boldsymbol{id}$ is the identity map
    while $t < T$ do
        $\rho^n_{\boldsymbol{D}} \leftarrow [\rho_0(\boldsymbol{\mathcal{X}}_B) \det \nabla \boldsymbol{\mathcal{X}}_B]_{\boldsymbol{D}}$        ▷ evaluation of (3.10) on $\boldsymbol{D}$
        $\tilde{\rho}_{\boldsymbol{D}} \leftarrow (I - h_t \nu \Delta)^{-1} \rho^n_{\boldsymbol{D}}$
        Define $\boldsymbol{\mathcal{X}}_{[t+h_t,t]}(\boldsymbol{x}) := \boldsymbol{x} + h_t \nu \nabla \mathcal{H}_{\boldsymbol{D}} [\log \tilde{\rho}_{\boldsymbol{D}}] (\boldsymbol{x})$        ▷ combining (3.11b) and (3.2)
        $\boldsymbol{\mathcal{X}}_B \leftarrow \mathcal{H}_{\boldsymbol{G}} [\boldsymbol{\mathcal{X}}_B \circ \boldsymbol{\mathcal{X}}_{[t+h_t,t]}]$
        $t \leftarrow t + h_t$
    end while
    return $\boldsymbol{\mathcal{X}}_B$
end function
```

Using the velocity defined in (3.11), we have that the one-step map in (3.2) has $\mathcal{O}({h_t}^2 + h_t {h_x}^3)$ local truncation error for Hermite cubic interpolants and $\mathcal{O}({h_t}^2 + h_t {h_x}^2)$ for linear. Indeed, the Hermite grid data for the linear interpolation have $\mathcal{O}({h_x}^2)$ error due to being evaluated at cell centers. For cubic interpolation, the derivative data of the velocity field are only available to $\mathcal{O}({h_x}^2)$, hence the $\mathcal{O}({h_x}^3)$ spatial error for the one-step map. Given the accuracy of the above one-step maps, we have from Lemma 3.1 that the global truncation errors for $\rho_{\boldsymbol{D}}$ are $\mathcal{O}(h_t + h_x)$ for linear interpolants and $\mathcal{O}(h_t + {h_x}^2)$ for cubics.

We apply the redistribution algorithm on a toy problem to illustrate the L^2 energy decay. For this test, we take the domain U to be the flat torus $[0,1] \times [0,1]$. The target density ρ_0 is concentrated in a band of width $w = 0.15$ around a circle of radius $r = 0.25$ centered at $\boldsymbol{x}_c = (0.5, 0.5)$. The minimum density is set at 0.75.

$$\rho_0(\boldsymbol{x}) = 1 + 0.25\eta(\boldsymbol{x}), \tag{3.12a}$$

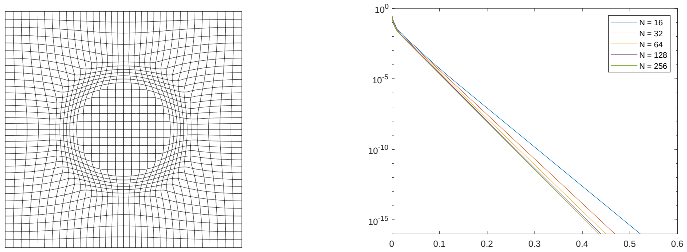
$$\eta(\boldsymbol{x}) = c(\eta_0(|\boldsymbol{x} - \boldsymbol{x}_c| - r) - \bar{\eta}_0), \tag{3.12b}$$

$$\eta_0(s) = \exp(-(1 - (2s/w)^2)^{-1}). \tag{3.12c}$$

Here, the constant $\bar{\eta}_0$ is the average of $\eta_0(|\boldsymbol{x} - \boldsymbol{x}_c| - r)$ and c is chosen so that $\min_{\boldsymbol{x}} \eta(\boldsymbol{x}) = -1$.

We test the CM method for density redistribution by running the algorithm described in section 3 to time 1 with $\nu = 1$, using Hermite cubic interpolation on various grids of size N and tracking the decay of $\mathcal{E}$, h_t is chosen to be $0.1/N$. The results are shown in Figure 3.1; the backward map deforms the domain to concentrate in the selected annulus, and the residual energy also exhibits the exponential decay from the diffusion equation.

3.2.1. Moving initial condition. The initial condition ρ_0 is the density to be redistributed toward uniform. In the context of surface reparametrization, ρ_0 is the area element of the surface which we try to uniformize. If the surface is moving, then ρ_0 will be different at different times. We consider the problem of redistribution of a



(a) Domain deformation from $\boldsymbol{\mathcal{X}}_{[1,0]}$. (b) Decay of $\mathcal{E}$ for various grids.

Fig. 3.1. *Diffusion flow density redistribution using CM method.*

"moving initial condition," that is, we let $\rho_0(\boldsymbol{x}, t)$ be a function which also depends on t. We can apply the same method as previously described, that is, defining

$$\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{X}_B(\boldsymbol{x}, t), t) \det \nabla \boldsymbol{X}_B(\boldsymbol{x}, t), \tag{3.13a}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = -\nu \nabla \log \rho(\boldsymbol{x}, t). \tag{3.13b}$$

At the implementation level, we simply replace line 4 of Algorithm 3.1 by

$$\rho^n_{\boldsymbol{D}} \leftarrow [\rho_0(\boldsymbol{X}_B, t) \det \nabla \boldsymbol{X}_B]_{\boldsymbol{D}}.$$

Taking a time derivative of $\rho(\boldsymbol{x}, t)$, we see that the density evolving simultaneously from the moving initial condition and the diffusion by the redistribution map satisfies the following inhomogeneous heat equation:

$$\partial_t \rho = \nu \Delta \rho + \partial_t \log \rho_0|_{\boldsymbol{X}_B} \rho. \tag{3.14}$$

From this, we can view the diffusion flow as a relaxation to a source term introduced by a moving initial condition. The additional diffusivity coefficient ν plays the role of balancing the strength of the source term and that of the diffusion. It can also be seen as a ratio of the two time scales involved, that is, if the source term is strong, then the diffusion must act on a faster time scale to control the growth in ρ. The form of the source term will be subject to further constraints in the case of a moving surface, a diffusion coefficient ν can then be picked accordingly to prevent growth in the density. We will examine this in further details in the following section.

4. Application to surface advection.

In this section, we apply the density transport method described above to the advection of parametric curves and surfaces in 3D ambient space. The density transport map will be used to maintain the proper sampling of the moving surface as it stretches and shrinks during its evolution. The general idea is to define a reparametrization of the surface as the function composition of the original parametrization with an area-redistributing map. The redistribution map will be computed from the heat flow and the area element of the redistributed parametrization will follow the inhomogeneous heat description of section 3.2.1.

4.1. Evolution of parametric surfaces using the CM method.

We briefly describe here the algorithm used to evolve parametric surfaces under a given velocity field in three dimensions.

Let Ω be a 3D domain. For simplicity we assume $\Omega = \mathbb{T}^3$ the flat 3-torus. Let $\boldsymbol{v} : \Omega \times \mathbb{R}_+ \to \mathbb{R}^3$ be a given velocity field. Here we assume that $\boldsymbol{v}$ is smooth and divergence-free so that the domain transformation it generates is a smooth diffeomorphism of Ω for all times. We have the same characteristic structure as described in section 2.1 and in [32, 43]. We denote by $\boldsymbol{\Phi}_F$ and $\boldsymbol{\Phi}_B$ the forward and backward characteristic maps, which satisfy the following equations:

$$\partial_t \boldsymbol{\Phi}_F(\boldsymbol{p},t) = \boldsymbol{v}(\boldsymbol{\Phi}_F(\boldsymbol{p},t),t) \qquad \forall(\boldsymbol{p},t) \in \Omega \times \mathbb{R}_+, \tag{4.1a}$$
$$(\partial_t + \boldsymbol{v}\cdot\nabla)\boldsymbol{\Phi}_B(\boldsymbol{p},t) = 0 \qquad \forall(\boldsymbol{p},t) \in \Omega \times \mathbb{R}_+, \tag{4.1b}$$
$$\boldsymbol{\Phi}_F(\boldsymbol{p},0) = \boldsymbol{\Phi}_B(\boldsymbol{p},0) = \boldsymbol{p} \qquad \forall \boldsymbol{p} \in \Omega. \tag{4.1c}$$

The diffeomorphisms are the forward and backward flow maps of the velocity field $\boldsymbol{v}$ and have the characteristic structure

$$\boldsymbol{\Phi}_F(\gamma(0),t) = \gamma(t), \quad \boldsymbol{\Phi}_B(\gamma(t),t) = \gamma(0) \qquad \forall t \in \mathbb{R}_+ \tag{4.2}$$

for any characteristic curve γ satisfying

$$\partial_t \gamma(t) = \boldsymbol{v}(\gamma(t),t). \tag{4.3}$$

The forward and backward maps are inverse transformations for all times, i.e.,

$$\boldsymbol{\Phi}_F(\boldsymbol{\Phi}_B(\boldsymbol{p},t),t) = \boldsymbol{\Phi}_B(\boldsymbol{\Phi}_F(\boldsymbol{p},t),t) = \boldsymbol{p} \qquad \forall t \geq 0. \tag{4.4}$$

We denote by $\boldsymbol{\Phi}_{[t_1,t_2]}$ the forward map in the time interval $[t_1,t_2]$ with $\boldsymbol{\Phi}_F := \boldsymbol{\Phi}_{[0,t]}$, and we use $\boldsymbol{\Phi}_{[t_2,t_1]}$ to denote the backward map in the same time interval, with $\boldsymbol{\Phi}_B := \boldsymbol{\Phi}_{[t,0]}$. The notation $\mathit{\Phi}$ is used for the numerical discretization of the $\boldsymbol{\Phi}$ maps. The time-stepping can be summarized as

$$\mathit{\Phi}_{[0,t+h_t]} = \mathcal{H}_{\boldsymbol{M}}\left[\mathit{\Phi}_{[t,t+h_t]} \circ \mathit{\Phi}_{[0,t]}\right], \tag{4.5a}$$
$$\mathit{\Phi}_{[t+h_t,0]} = \mathcal{H}_{\boldsymbol{M}}\left[\mathit{\Phi}_{[t,0]} \circ \mathit{\Phi}_{[t+h_t,t]}\right] \tag{4.5b}$$

for some grid $\boldsymbol{M}$ on Ω. The identity $\mathit{\Phi}_{[t+h_t,t]} = \mathit{\Phi}^{-1}_{[t,t+h_t]}$ can be guaranteed to high precision by employing higher order ODE integration schemes, and hence the error on the property

$$\boldsymbol{\Phi}_F \circ \boldsymbol{\Phi}_B = \boldsymbol{\Phi}_B \circ \boldsymbol{\Phi}_F = \boldsymbol{I}, \tag{4.6}$$

stems mainly from the representation quality of the interpolation operator $\mathcal{H}_{\boldsymbol{M}}$, which we will control using a dynamic remapping technique described below.

It is well-known that the flow map possesses a semigroup property which allows for the decomposition of a long-time map into several submaps; this can be used to achieve high resolution representation of the deformation using coarse grid computations. We use the same time-decomposition strategy as in [43] to represent the global-time maps:

$$\mathit{\Phi}_F = \mathit{\Phi}_{[0,t]} = \mathit{\Phi}_{[\tau_{n-1},t]} \circ \mathit{\Phi}_{[\tau_{n-2},\tau_{n-1}]} \circ \cdots \circ \mathit{\Phi}_{[0,\tau_1]}, \tag{4.7a}$$
$$\mathit{\Phi}_B = \mathit{\Phi}_{[t,0]} = \mathit{\Phi}_{[\tau_1,0]} \circ \cdots \circ \mathit{\Phi}_{[\tau_{n-1},\tau_{n-2}]} \circ \mathit{\Phi}_{[t,\tau_{n-1}]}. \tag{4.7b}$$

The error on the inverse property (4.4) stems from the time-stepping and interpolation (4.5). This gives us an alternative a posteriori error estimate for choosing the remapping times τ_i. In [43], in the context of the incompressible Euler equations, the remapping time is chosen to be the first time the volume preservation error of the map exceeds some threshold. For surface advection, the velocity is not strictly constrained to divergence free, and volume preservation is not a main concern. Therefore we use the following composition error as the remapping criterion:

$$\epsilon = \max\left(\|\mathit{\Phi}_{[\tau_{i-1},t]} \circ \mathit{\Phi}_{[t,\tau_{i-1}]} - \boldsymbol{I}\|_\infty, \|\mathit{\Phi}_{[t,\tau_{i-1}]} \circ \mathit{\Phi}_{[\tau_{i-1},t]} - \boldsymbol{I}\|_\infty\right). \tag{4.8}$$

We then define τ_i to be the first time t where the above error exceeds some chosen threshold.

The forward and backward characteristic maps give us solution operators to the advection problem. Let $S_0 \subset \Omega$ be an initial curve or surface moving following the velocity field $\boldsymbol{v}$. Let S_t be the surface at time t. Using the characteristic maps, we have two equivalent definitions for S_t, one implicit and the other explicit:

$$S_t = \{\boldsymbol{p} \in \Omega \mid \boldsymbol{\Phi}_B(\boldsymbol{p}, t) \in S_0\} \tag{4.9a}$$

$$\text{or} \quad S_t = \{\boldsymbol{\Phi}_F(\boldsymbol{p}, t) \mid \boldsymbol{p} \in S_0\}. \tag{4.9b}$$

Equation (4.9a) uses the same Eulerian definition as in level-set methods. When S_0 is expressed as the zero-level-set of a function ψ, we have that S_t is given by the zero-level-set of the advected function $\psi \circ \boldsymbol{\Phi}_B$. This approach is studied in [32]; a review of recent advances in level-set methods can be found in [20].

Equation (4.9b) is a Lagrangian definition, where surfaces are defined explicitly through a parametrization function. This is suitable for curves and surfaces which do not admit a level-set representation, for instance, for open curves or nonorientable surfaces. Let U be the parameter space and $\boldsymbol{P}_0 : U \to S_0 \subset \Omega$ a regular parametrization of the surface at time 0, that is, we assume the mapping between U and S_0 to be a diffeomorphism. The parametrization of S_t is then given by

$$\boldsymbol{P}_t = \boldsymbol{\Phi}_{[0,t]} \circ \boldsymbol{P}_0. \tag{4.10}$$

Numerically speaking this method evolves the parametrization function in time using the solution operator $\mathit{\Phi}_F$. Compared to traditional methods where the surface is sampled and sample points evolved individually, the CM method provides a functional definition of the parametrization defined everywhere in U. Similar to the backward map method in [32], this approach also provides arbitrary resolution of the parametrization function $\boldsymbol{P}_t$, and hence of S_t. The surface S_t can be arbitrarily sampled at any time by evaluating the pushforward operator $\mathit{\Phi}_F$. In this case, the numerical accuracy of the diffeomorphism $\mathit{\Phi}_F$ is the determining factor in the accuracy of the surface markers. The accuracy of each marker point location is therefore independent of the overall sampling; however, there is no guarantee on the overall resolution of the surface since the ambient flow provides no control over the density of the sample points along the surface throughout its evolution.

4.2. Equiareal redistribution.

A curve or surface S_t can be sampled by choosing sample points $\boldsymbol{y}_i \in U$ and evaluating $\boldsymbol{p}_i(t) = \boldsymbol{P}_t(\boldsymbol{y}_i)$ to represent S_t discretely. However, a uniform distribution of $\boldsymbol{y}_i$ in U does not necessarily lead to well-distributed marker points $\boldsymbol{p}_i$ on S_t. Indeed, as the surface stretches and deforms under the flow, some regions may expand and become sparsely sampled. We can quantify this by evaluating the area element A_t from the first fundamental form $I_t = \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t$,

$$A_t = \sqrt{\det I_t} = \sqrt{\det \nabla \boldsymbol{P}_0^T \nabla \boldsymbol{\Phi}_F^T \nabla \boldsymbol{\Phi}_F \nabla \boldsymbol{P}_0}. \tag{4.11}$$

We can compute the time evolution of A_t. Using

$$\partial_t I_t = \nabla \boldsymbol{P}_t^T (\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v}) \nabla \boldsymbol{P}_t, \tag{4.12}$$

and applying Jacobi's rule for derivatives of matrix determinants, we have

$$\partial_t A_t = \frac{1}{2}A_t \mathrm{tr}\left(I^{-1}\partial_t I\right) = \frac{1}{2}A_t \mathrm{tr}\left((\nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t)^{-1}\nabla \boldsymbol{P}_t^T(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})\nabla \boldsymbol{P}_t\right). \tag{4.13}$$

Using the cyclic property of the trace operator, we can rewrite this as

$$\partial_t A_t = A_t \mathrm{tr}\left(\nabla \boldsymbol{P}_t(\nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t)^{-1}\nabla \boldsymbol{P}_t^T \nabla \boldsymbol{v}\right) = A_t \mathrm{tr}\left(\Pi_{S_t}\nabla \boldsymbol{v}\right) = A_t\, \mathrm{div}|_{S_t}\boldsymbol{v}, \tag{4.14}$$

where $\Pi_{S_t} = \nabla \boldsymbol{P}_t(\nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t)^{-1}\nabla \boldsymbol{P}_t^T$ is the orthogonal projection operator onto the tangent space of S_t. Therefore, the area element grows exponentially at rate $\mathrm{div}|_{S_t}\boldsymbol{v}$ corresponding to the divergence of the flow along the tangent space of S_t.

The approach we take to control the growth of the area element is to modify the parametrization function $\boldsymbol{P}_t$ by applying a transformation on the parametric space U. We take A_t as a probability density on U and apply the method described in section 2.2 to obtain a mapping $\boldsymbol{X}_B : U \to U$ which pushes the uniform density to the density A_t. The modified parametrization

$$\boldsymbol{Q}_t = \boldsymbol{P}_t \circ \boldsymbol{X}_{[t,0]} : U \to S_t \subset \Omega \tag{4.15}$$

should then have area element $\sqrt{\det I}$ equal to a constant. That is to say, if the points $\boldsymbol{y}_i$ are uniformly distributed in U, then the sample points $\boldsymbol{p}_i$ are uniform on S_t. In practice, it'll be wasteful to recompute a redistribution map at each time; t instead, the advection and redistribution are evolved alongside each other. We will apply the method of section 3.2.1 as a relaxation to any area deformation that occurs in one time step of advection. The evolution global-time redistribution $\boldsymbol{X}_{[t,0]}$ will be obtained by composing the one-step relaxation maps. This can be formalized as follows.

We define probability densities on U:

$$\rho_0(\boldsymbol{x}, t) = \frac{1}{\rho_A}\sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t}, \tag{4.16a}$$

$$\rho(\boldsymbol{x}, t) = \frac{1}{\rho_A}\sqrt{\det \nabla \boldsymbol{Q}_t^T \nabla \boldsymbol{Q}_t}, \tag{4.16b}$$

$$\text{with} \quad \rho_A(t) = \int_U \sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t}\, d\boldsymbol{x} = |S_t|. \tag{4.16c}$$

We note that both ρ_0 and ρ are the area elements corresponding to the parametrizations $\boldsymbol{P}$ and $\boldsymbol{Q}$ normalized to a probability density since

$$\rho_A(t) = \int_{\boldsymbol{X}_B^{-1}(U)} \sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t}|_{\boldsymbol{X}_B} \det \nabla \boldsymbol{X}_B\, d\boldsymbol{x} = \int_U \sqrt{\det \nabla \boldsymbol{Q}_t^T \nabla \boldsymbol{Q}_t}\, d\boldsymbol{x}. \tag{4.17}$$

We have the following equations for their time evolution:

$$\partial_t \rho_A(t) = \int_U \mathrm{div}|_{S_t}\boldsymbol{v}\sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t}\, d\boldsymbol{x} = \rho_A \int_U \rho_0\, \mathrm{div}|_{S_t}\boldsymbol{v}\, d\boldsymbol{x}, \tag{4.18a}$$

$$\begin{aligned}\partial_t \rho_0(\boldsymbol{x}, t) &= \frac{1}{\rho_A}\, \mathrm{div}|_{S_t}\boldsymbol{v}\sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t} - \frac{\rho_0}{\rho_A}\partial_t \rho_A \\ &= \left(\mathrm{div}|_{S_t}\boldsymbol{v} - \partial_t \log \rho_A\right)\rho_0(\boldsymbol{x}, t),\end{aligned} \tag{4.18b}$$

$$\rho(\boldsymbol{x}, t) = \rho_0\left(\boldsymbol{X}_B(\boldsymbol{x}, t), t\right)\det \nabla \boldsymbol{X}_B. \tag{4.18c}$$

Using the results in section 2.2, we have that the pullback of ρ_0 by $\boldsymbol{X}_B$ generates a diffusion process in ρ. Therefore, the governing equation for the area element of the redistributed parametrization $\boldsymbol{Q}_t$ is

$$\partial_t \rho(\boldsymbol{x},t) = \nu\Delta\rho(\boldsymbol{x},t) + \partial_t \rho_0|_{\boldsymbol{X}_B} \det \nabla \boldsymbol{X}_B = \nu\Delta\rho(\boldsymbol{x},t) + \lambda_{\boldsymbol{v}}(\boldsymbol{x},t)\rho(\boldsymbol{x},t), \tag{4.19}$$

where $\lambda_{\boldsymbol{v}}(\boldsymbol{x},t) = \partial_t \log \rho_0|_{\boldsymbol{X}_B} = (\,\mathrm{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)} - \fint_U \rho_0 \,\mathrm{div}|_{S_t}\boldsymbol{v}\ d\boldsymbol{x}$. The diffusion coefficient ν in the redistribution step can be used to increase control over the growth of area density from surface deformation. It effectively corresponds to accelerating the time scale of the diffusion-flow redistribution, thereby introducing the area diffusion as a fast time-scale relaxation.

The method is summarized in pseudocode in Algorithm 4.1. The notation $\boldsymbol{P}_t$ implicitly assumes the definition and computation of the nonredistributed time-dependent parametrization function (4.10) using the numerical characteristic map $\mathit{\Phi}_F$ of the ambient advection. The computation of $\mathit{\Phi}_F$ is given in (4.5a) and (4.7). In comparison with the algorithm described in section 3.2, the redefinition of ρ^n at each t_n at line 5 corresponds to a source term $\lambda_{\boldsymbol{v}}\rho$ in the density evolution arising from the deformation of the surface. A typical energy argument for the reaction-diffusion equation provides some estimates on the evolution of the density. Here we use the special structure of the $\lambda_{\boldsymbol{v}}$ term to write a more specific estimate.

Algorithm 4.1 CM method for equiareal time-dependent surface parametrization.

```
Input: Parametrization $\boldsymbol{P}_t$, staggered grids $\boldsymbol{G}$ and $\boldsymbol{D}$, diffusivity $\nu$, time step $h_t$,
    final time $T$
 1: function REPARAMETRIZATION($\boldsymbol{P}_t$, $\nu$, $T$)
 2:     Initialize $t \leftarrow 0$ ,$\boldsymbol{\mathcal{X}}_{[t,0]} \leftarrow \boldsymbol{id}$                ▷ $\boldsymbol{id}$ is the identity map
 3:     while $t < T$ do
 4:         Define $\boldsymbol{Q}_t := \boldsymbol{P}_t \circ \boldsymbol{\mathcal{X}}_{[t,0]}$
 5:         Define $\rho^n(\boldsymbol{x}) := \frac{1}{\rho_A(t)}\sqrt{\det \nabla \boldsymbol{Q}_t^T \nabla \boldsymbol{Q}_t}$                ▷ from (4.16)
 6:         $\boldsymbol{\mathcal{X}}_{[t+h_t,t]} \leftarrow$ HeatFlowMap($\rho^n, \nu, [0, h_t]$)       ▷ subroutine Algorithm 3.1 on
    local time problem
 7:         $\boldsymbol{\mathcal{X}}_{[t+h_t,0]} \leftarrow \mathcal{H}_{\boldsymbol{G}}\left[\boldsymbol{\mathcal{X}}_{[t,0]} \circ \boldsymbol{\mathcal{X}}_{[t+h_t,t]}\right]$
 8:         $t \leftarrow t + h_t$
 9:     end while
10:     return $\boldsymbol{Q}_T = \boldsymbol{P}_T \circ \boldsymbol{\mathcal{X}}_{[T,0]}$
11: end function
```

Remark 2. Line 6 of Algorithm 4.1 allows for the HeatFlowMap one-step redistribution and the ambient advection to be implemented using two different time step sizes. This is not necessary; however, it is useful if ν is large: the surface area diffusion and the ambient advection would then essentially evolve on two different time scales. In that case, shrinking the diffusion time steps is necessary to reduce numerical error; the one-step redistribution map $\boldsymbol{\mathcal{X}}_{[t+h_t,t]}$ would then be computed over several redistribution substeps.

THEOREM 4.1. *Choosing ν large enough, the L^2 distance between the area element of the $\boldsymbol{Q}$ parametrization and the uniform distribution can be controlled to order $\mathcal{O}(\nu^{-1})$.*

Proof. The algorithm is consistent with the reaction-diffusion equation (4.19). We have that the energy estimate in this case is

$$\begin{aligned}\frac{d}{dt}E(t) = \frac{d}{dt}\|\rho - 1\|_{L^2}^2 &= -\nu\|\nabla(\rho-1)\|_{L^2}^2 + \int_U (\rho-1)^2\lambda_{\boldsymbol{v}}\ d\boldsymbol{x} + \int_U (\rho-1)\lambda_{\boldsymbol{v}}\ d\boldsymbol{x} \\ &\le -\nu\|\nabla(\rho-1)\|_{L^2}^2 + \|\lambda_{\boldsymbol{v}}\|_\infty\|\rho-1\|_{L^2}^2 - \int_U \lambda_{\boldsymbol{v}}\ d\boldsymbol{x}, \end{aligned} \tag{4.20}$$

where $\int_U \rho\lambda_{\boldsymbol{v}}\ d\boldsymbol{x}$ vanishes since $\int_U \partial_t\rho\ d\boldsymbol{x} = 0$ by construction and $\int_U \Delta\rho\ d\boldsymbol{x} = 0$ with periodic or Neumann boundary conditions.

The $\lambda_{\boldsymbol{v}}$ term is the growth rate of the area density normalized so that the density stays in the space of probability distributions. Implicit in this normalization is a dependence of the integral $\int_U \lambda_{\boldsymbol{v}}\ d\boldsymbol{x}$ on the current density:

$$\begin{aligned} -\int_U \lambda_{\boldsymbol{v}}\ d\boldsymbol{x} &= -\int_U (\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\ d\boldsymbol{x} + \int_{\boldsymbol{X}_B^{-1}(U)} \rho\,(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\ d\boldsymbol{x} \\ &= \int_U (\rho-1)\,(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\ d\boldsymbol{x} \le \left\|(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\right\|_{L^2} \|\rho-1\|_{L^2}. \end{aligned} \tag{4.21}$$

We obtain the following bound on the growth of the L^2 norm:

$$\frac{d}{dt}\|\rho-1\|_{L^2} \le 2(-\alpha\nu + \|\lambda_{\boldsymbol{v}}\|_\infty)\|\rho-1\|_{L^2} + 2\left\|(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\right\|_{L^2}; \tag{4.22}$$

This means that the L^2 norm can be controlled by

$$\|\rho-1\|_{L^2} \le \frac{\beta}{\alpha\nu - \|\lambda_{\boldsymbol{v}}\|_\infty} + c\exp\left(-2t\left(\alpha\nu - \|\lambda_{\boldsymbol{v}}\|_\infty\right)\right), \tag{4.23}$$

where $\beta = \max_t \|(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\|_{L^2}$ and α the constant from the Poincaré inequality.

Therefore, by choosing $\nu > \alpha^{-1}\|\lambda_{\boldsymbol{v}}\|_\infty$ sufficiently large, we can guarantee that the L^2 norm of the deviation from uniform of the $\boldsymbol{Q}$ area element stays of order $\mathcal{O}(\nu^{-1})$ for all times. □

According to the governing equation (4.19), for larger enough ν, the diffusion should limit the fine scale spatial features present in ρ and hence in the velocity field $-\nu\nabla\log\rho$. High frequency modes are generated by $\lambda_{\boldsymbol{v}}$ and hence it is sufficient to compute the local-time deformation map on a grid fine enough to resolve the source term. Similar to the advection problem where the local maps $\mathit{\Phi}_{[\tau_{i-1},\tau_i]}$ are computed on coarse grids which resolve well enough the local-time velocity field, the submaps in the density transport map can also benefit from the computational savings of coarser grids. All characteristic maps involved share the same semigroup structure and can be decomposed into submaps in order to achieve higher spatial resolution at low computational cost. We can therefore apply the above reparametrization algorithm to each subinterval in the submap decomposition (4.7) and obtain the full reparametrization as the composition of all redistribution maps. Combining (4.7), (4.10), and (2.5), we have

$$\boldsymbol{Q}_t = \mathit{\Phi}_{[\tau_{n-1},t]} \circ \mathit{\Phi}_{[\tau_{n-2},\tau_{n-1}]} \circ \cdots \circ \mathit{\Phi}_{[0,\tau_1]} \circ \boldsymbol{P}_0 \circ \boldsymbol{\mathcal{X}}_{[\tau_1,0]} \circ \cdot \circ \boldsymbol{\mathcal{X}}_{[\tau_{n-1},\tau_{n-2}]} \circ \boldsymbol{\mathcal{X}}_{[t,\tau_{n-1}]}. \tag{4.24}$$

Numerically, each submap is computed independently, sequentially using Algorithm 4.1. To be consistent with the reinitialized problem, the input parametrization for the $i+1$st map is defined to be $\mathit{\Phi}_{[\tau_i,t]} \circ \boldsymbol{Q}_{\tau_i}$.

Remark 3. For the submap computations, the parametrization of the surface S_t is given by $\mathit{\Phi}_{[\tau_i,t]} \circ \boldsymbol{Q}_{\tau_i}$ where $\boldsymbol{Q}_{\tau_i}$ is given by (4.24). However, to save computational time, for the purpose of computing the density ρ, it is sufficient to replace $\boldsymbol{Q}_{\tau_i}$ by an interpolant on a fine enough grid similar to the approach in [32]. This is because the method maintains the sampling density of $\boldsymbol{Q}$ near uniform for all times and therefore $\boldsymbol{Q}_{\tau_i}$ can be accurately represented by interpolation.

Remark 4. The primary purpose of this method is to maintain an almost uniform parametrization of the surface and to prevent growth in the density distribution of the area element. In this sense, the computational efficiency of this method is unique in that it does not depend on the number of sample points. The algorithm computes a functionally defined parametrization, and sampling is done by interpolation and has linear complexity. The exact computational cost of the reparametrization will be application dependent. For instance, if we require that the area distribution be constant up to an L^2 tolerance of $\mathcal{E}$, then we would typically observe polynomial complexity with respect to $\mathcal{E}^{-1}$. Indeed, according to the results in this section, the L^2 error is controlled by ν^{-1}. Hence, roughly $\mathcal{E}^{-1}$ time steps of redistriubtion will be required. Additional computational cost will be incurred for reducing the numerical error of the algorithm below the desired threshold; this will be dependent on the order of accuracy of the interpolation and time-stepping schemes. We give further details on the computational time in Appendix A.

5. Numerical results.

5.1. Density redistribution on flat domains. Here we first show a simple numerical test of the density redistribution algorithm for a moving initial condition on a flat 2D periodic domain, as described in section 3.2.1.

We define the following density ρ_0:

$$\rho_0(x,y,t) = 1 + 0.25\sin^2(1.5\pi t)\,\eta\left(x + 0.25\sin(0.5\pi t)\sin(2\pi y), y\right), \tag{5.1}$$

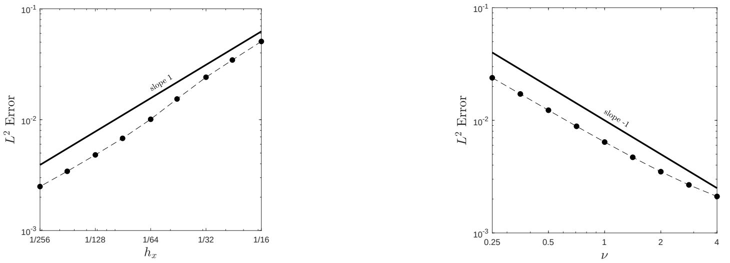
where η is defined in (3.12b).

This density is initially constant; it then concentrates around a circle of radius 0.25. The overall density is also advected by the volume preserving transformation

$$(x,y) \mapsto (x + 0.25\sin(0.5\pi t)\sin(2\pi y), y).$$

The amplitude of the density is scaled by $0.25\sin^2(1.5\pi t)$.

We test the redistribution of this moving density in the time interval $t \in [0,3]$ with various diffusion coefficients ν and with various grid sizes for the redistribution map which we represent using piecewise linear interpolation. We ran convergence tests of the maximum L^2 error for $t \in [0,3]$ with respect to the grid size and ν. The results are shown in Figure 5.1. As expected, the maximum L^2 norm of $\rho - 1$ is linear with respect to h_x, h_t, and ν^{-1}.



(a) $\nu = 10$, $h_t = h_x/4$. (b) $h_x = 1/768$, $h_t = 1/2048$.

FIG. 5.1. *L^2 error with respect to h and ν.*

5.2. Equidistributing parametrizations of curves and surfaces. In this section, we apply the redistribution algorithm to the evolution of several curves and surfaces in a 3D ambient flow. Starting from a given initial parametrization, we use the redistribution characteristic map to maintain an equiareal parametrization of the evolving curves and surfaces at all times during the simulations. For the tests in this section, we use the following 3D ambient velocity taken from [30]:

$$\boldsymbol{v}(\boldsymbol{x},t) = \begin{pmatrix} 2\cos\left(\frac{\pi t}{P}\right)(\sin(\pi x))^2\sin(2\pi y)\sin(2\pi z) \\ -\cos\left(\frac{\pi t}{P}\right)\sin(2\pi x)(\sin(\pi y))^2\sin(2\pi z) \\ -\cos\left(\frac{\pi t}{P}\right)\sin(2\pi x)\sin(2\pi y)(\sin(\pi z))^2 \end{pmatrix} \tag{5.2}$$

defined on a periodic cube $[0,1]^3$, with $\boldsymbol{x} = (x,y,z)^T$. P is the period of the velocity field; the deformation it generates reaches its maximum extent at $t = P/2$, then returns to identity at $t = P$. For the following tests, we will choose $P = 3$.

The velocity field also has reflection symmetries across the planes $z = y$ and $z = 1 - y$; the flow is planar along these two planes. We know therefore that the flow will not cross these planes and will have mirror symmetric motion on either side. It follows that any initial curve or surfaces crossing these planes will undergo extensive deformation. Without any maintenance on the parametrization function, one can expect the resulting arclength and area elements to grow exponentially, resulting in poor representation of the curves and surfaces.

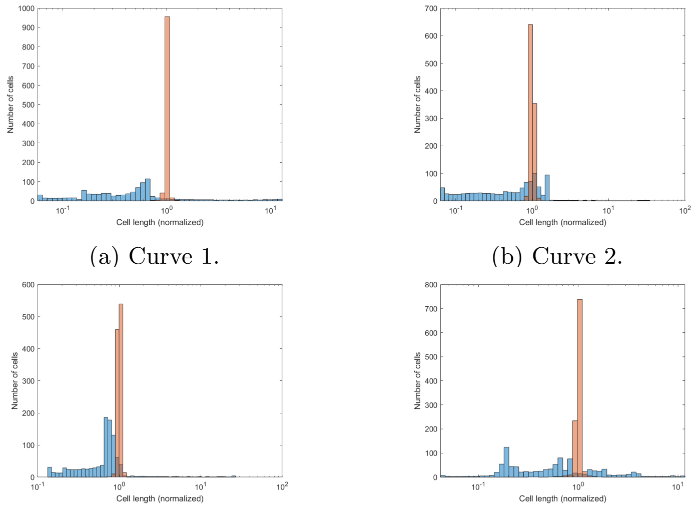
For all results in this section, we used a 64^3 grid with $h_t = 1/96$ for the computation of the forward characteristic map in the 3D ambient space, and we used a 128^2 grid for the computation of the 1D and 2D redistribution maps on the parametric space. The ambient map uses Hermite cubic interpolation and redistribution maps use linear interpolation. The diffusion coefficient for all redistribution maps were fixed at $\nu = 2$. All characteristic maps use the submap decomposition method and all curves and surfaces are pushed forward using the same forward ambient space map. The computations were carried out on a laptop with an Intel i5-3210 duo-core 2.50 GHz CPU with 8 GB of RAM. The routines were implemented in MATLAB with C-Mex subroutines for the interpolation operations. As reference, the 3D ambient characteristic maps were decomposed into six submaps in the interval $[0, 1.5]$, each map being stored on the hard drive. The total computational time for calculating the ambient characteristic map was 511 seconds.

5.2.1. Evolution of curves. We apply the redistribution method in 1D parametric space to maintain an arclength parametrization of curves evolving under the flow given in (5.2). We use four curves in this test; the first three are line segments and the last one is a circle.

To illustrate the effect of the redistribution, we show in Figure 5.3 the final states of the curves at various resolutions (the initial curves and their time evolution are shown in Appendix B). In Figure 5.3, the parametrization $\boldsymbol{P}$ as well as its redistributed version $\boldsymbol{Q}$ are drawn using a piecewise linear interpolation on a gradually refined grid from 32 equidistant grid points to 1024 grid points. As we can see, in all cases, to capture the features of a given curve, the $\boldsymbol{P}$ parametrization requires a roughly 10× finer grid to obtain the same quality as the $\boldsymbol{Q}$ parametrization. Moreover, Table 5.1 shows that interpolating the $\boldsymbol{P}$ parametrization yields an L^∞ error roughly an order of magnitude larger than that of $\boldsymbol{Q}$. This is due to the high variations in the speed of the $\boldsymbol{P}$ parametrization. Indeed, the large distortions created by the ambient flow cause some regions of the curves to undergo large stretching, whereas others are compressed. At $t = 1.5$, this results in high amplitude variations in the distances

TABLE 5.1
Error of linear interpolation for the parametrization functions at different grid resolutions, using the $N = 1024$ tests as reference.

N	$\boldsymbol{P}_1$	$\boldsymbol{Q}_1$	$\boldsymbol{P}_2$	$\boldsymbol{Q}_2$	$\boldsymbol{P}_3$	$\boldsymbol{Q}_3$	$\boldsymbol{P}_4$	$\boldsymbol{Q}_4$
32	1.32e-1	1.83e-2	4.38e-1	3.77e-2	2.58e-1	1.32e-2	3.87e-1	7.72e-2
64	5.64e-2	1.08e-2	3.43e-1	1.77e-2	1.81e-1	6.91e-3	1.74e-1	3.39e-2
128	1.60e-2	2.92e-3	2.23e-1	5.87e-3	1.03e-1	3.25e-3	6.96e-2	1.51e-2
256	4.14e-3	1.00e-3	8.52e-2	1.69e-3	4.19e-2	8.52e-4	1.83e-2	3.63e-3
512	1.05e-3	2.96e-4	2.39e-2	6.59e-4	1.18e-2	2.76e-4	4.91e-3	1.30e-3



(a) Curve 1.

(b) Curve 2.

(c) Curve 3.

(d) Curve 4.

FIG. 5.2. *Arclength distribution of the parametrizations $\boldsymbol{P}$ (blue) and $\boldsymbol{Q}$ (red).*

between marker points that were equidistant at $t = 0$. This effect can be clearly seen in Figure 5.2, where the histogram of the cell lengths for the curves is shown. As we can see, the cell lengths of the original parametrization $\boldsymbol{P}$ (shown in blue) are rather spread out, with a majority of very short cells covering one part of the curves and few very large cells covering the rest. In terms of sampling, this is suboptimal since the marker points in the oversampled regions are redundant. In contrast, the redistributed parametrizations $\boldsymbol{Q}$ (shown in red/orange) have a much more uniform distribution of cell lengths: almost all cell lengths are concentrated around the average, meaning that marker points are uniformly distributed along the curve. We can also measure the uniformity of the marker points distribution quantitatively: Table 5.2 shows the standard deviation, i.e., $\|\rho^n - 1\|_{L^2}$, and the median of the normalized area densities for each parametrization as well as the computational time required for generating the redistribution map. Ideally, for a perfectly uniform distribution, the median cell length should be 1 and the standard deviation 0. A median closer to 1 and a smaller standard deviation in the $\boldsymbol{Q}$ case indicate that the arclength distance between two sample points is more uniform, thereby avoiding the undersampling of the more deformed parts of the curves or redundancy of markers in compressed parts.

5.2.2. Evolution of surfaces. We test the redistribution method on three different topologies for 2D surfaces: rectangle, torus, and cylinder. The parametric spaces are taken to be $U = [0, 1]^2$ with Neumann, periodic, and mixed Neumann-

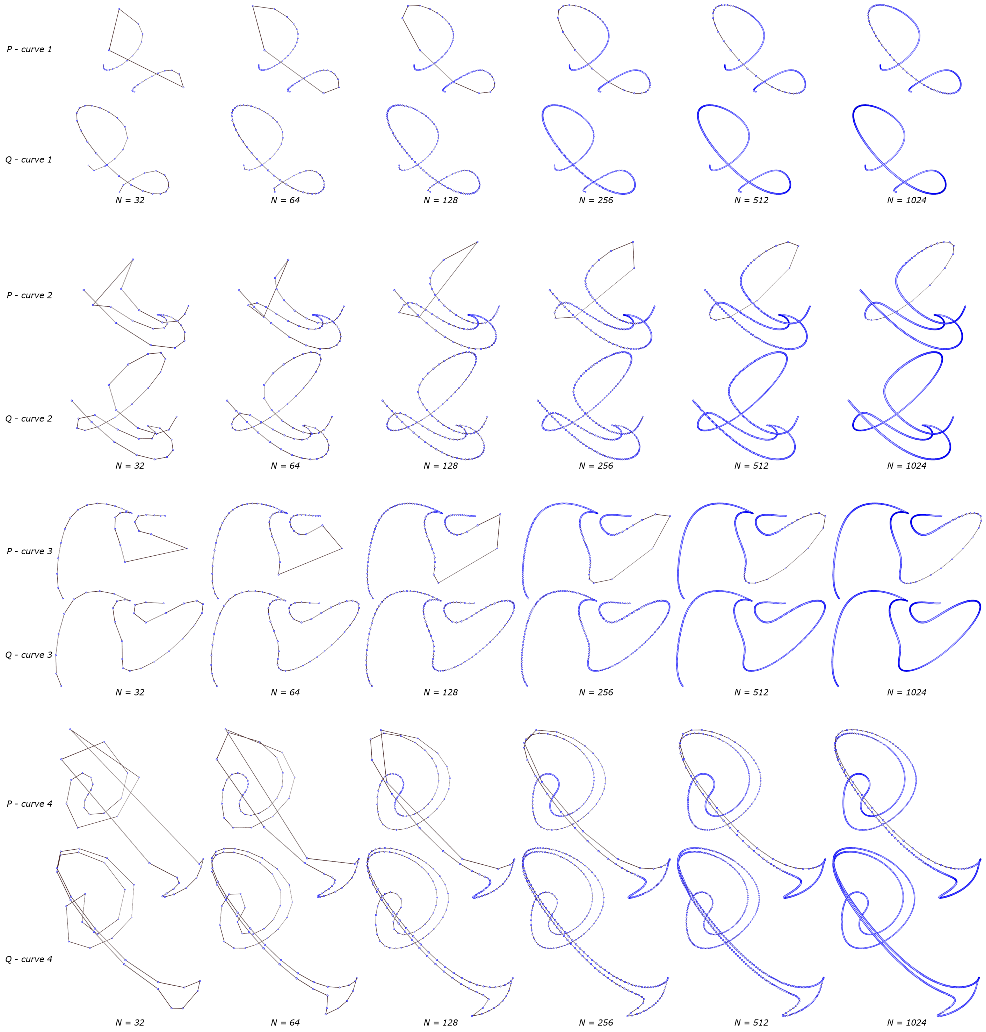


Fig. 5.3. *$\boldsymbol{P}$ and $\boldsymbol{Q}$ parametrizations of curves* 1 *to* 4 *using gradually finer grids.*

periodic boundary conditions, respectively. These surfaces will move under the flow (5.2) and we will compute the two parametrizations $\boldsymbol{P}$ and $\boldsymbol{Q}$ as in the 1D case. For each given time t shown in Figures 5.5 to 5.7, the parametrizations are represented by a linear interpolation on a uniform mesh grid of 512^2 points. For $\boldsymbol{P}$ the grid data is obtained by solving the ODEs forward in time for each grid point, $\boldsymbol{Q}$ is obtained by evaluating the redistribution map $\boldsymbol{P}_0$ and the forward ambient characteristic maps at grid points on U. In order to illustrate the effect of the redistribution, we sample each parametrization with 200, 000 randomly generated marker points. The distributions of these random points over the surfaces are expected to follow the random variables description in section 2.2. The initial surfaces with marker points are shown in Figure 5.4, and the time evolution of each surface is shown in Figures 5.5 to 5.7. We also show the standard deviation and median of the normalized area densities and the computational times for the redistribution maps in Figure 5.8 and Table 5.3.

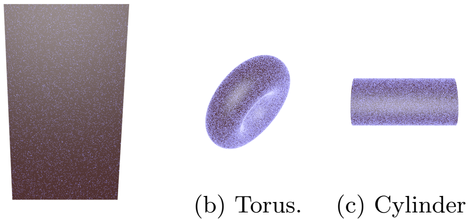
TABLE 5.2
Parametrizations $\boldsymbol{P}$ and $\boldsymbol{Q}$ at $t = 1.5$.

Curves	1	2	3	4
$\sigma_{\boldsymbol{P}}$	1.0479	0.8214	0.7088	0.8858
$\sigma_{\boldsymbol{Q}}$	0.0195	0.0318	0.0248	0.0353
$M_{\boldsymbol{P}} - 1$	-0.5358	-0.3742	-0.2927	-0.4435
$M_{\boldsymbol{Q}} - 1$	0.0013	0.0012	-0.0002	0.0005

(a) Standard deviation (σ) and median (M) errors of the length density.

Curves	1	2	3	4
Evaluating $\boldsymbol{Q}$	9.96 s	9.32 s	9.61 s	9.62 s
Defining $\rho^n(\boldsymbol{x})$	0.04 s	0.04 s	0.05 s	0.04 s
Updating $\boldsymbol{\mathcal{X}}_B$	0.74 s	0.70 s	0.67 s	0.59 s
Number of remappings	2	3	3	3

(b) Total computation times for the evolution of the parametrization $\boldsymbol{Q}$.



(a) Rectangle. (b) Torus. (c) Cylinder.

FIG. 5.4. *Initial surfaces with uniformly distributed random sample points.*

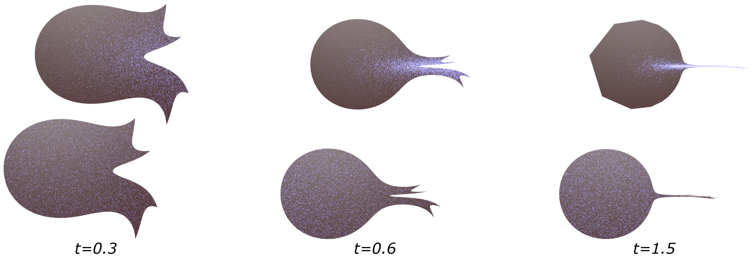


FIG. 5.5. *Comparison of the surface sampling of the evolved rectangle using the original parametrization $\boldsymbol{P}$ (top images) and the redistributed parametrization $\boldsymbol{Q}$ (bottom images).*

Figure 5.5 shows a clear example of the benefits of the reparametrization method. In this case, the initial rectangle is placed on one of the symmetry planes of the flow, hence the deformation is applied fully on the tangential directions of the surface. At $t = 1.5$, we see from the $\boldsymbol{P}$ parametrization that almost the entirety of the surface is compressed in a thin sliver. The sample points distribution is unnecessarily dense on the thin protrusion and very sparse on the rest of the surface. In fact, using a 512^2 grid for the interpolation of $\boldsymbol{P}$, we still see that the boundary of the surface is jagged and visibly piecewise-linear as opposed to the smooth circular shape shown by the same resolution interpolation of $\boldsymbol{Q}$. Indeed, each line segment on the boundary corresponds to a cell edge of length $\mathcal{O}(1/512)$ at time 0. This is an indication that the large scale deformation the surface has undergone between $t = 0$ and 1.5 makes the $\boldsymbol{P}$ parametrization inefficient; the numerical difficulties from this deformation are mitigated by the redistribution method resulting in a reparametrization $\boldsymbol{Q}$ which provides a uniform sampling and a smooth, well-resolved surface interpolation. Similar

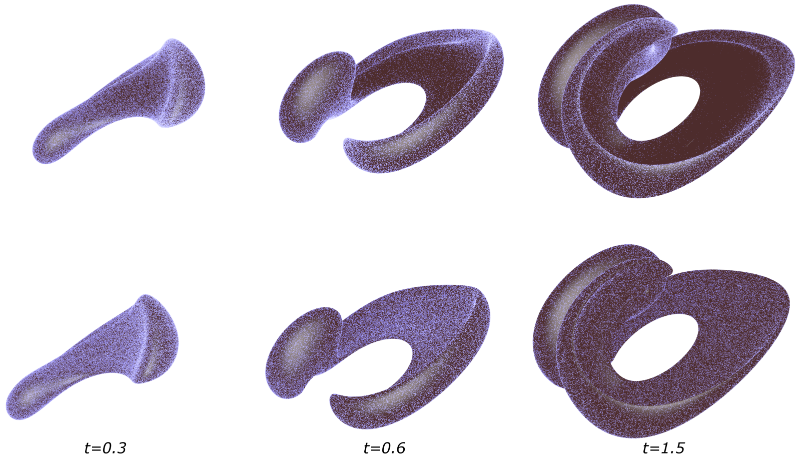


Fig. 5.6. *Comparison of the surface sampling of the evolved torus using the original parametrization* $\boldsymbol{P}$ *(top images) and the redistributed parametrization* $\boldsymbol{Q}$ *(bottom images).*

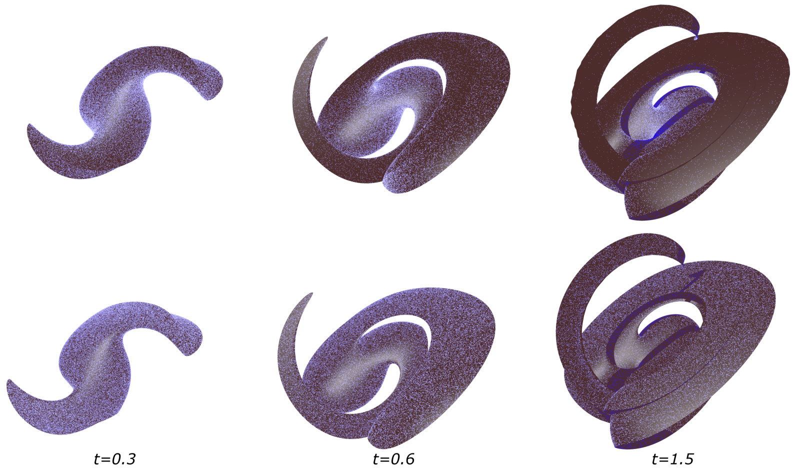
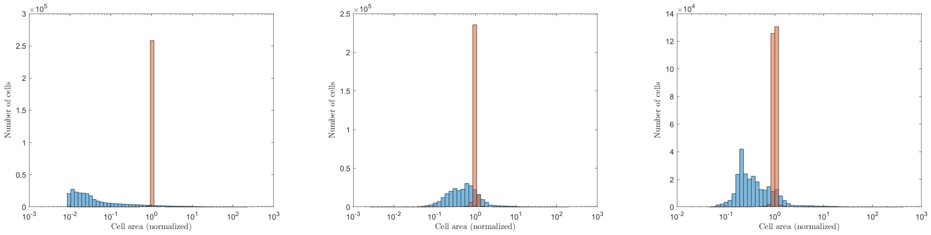


Fig. 5.7. *Comparison of the surface sampling of the evolved cylinder using the original parametrization* $\boldsymbol{P}$ *(top images) and the redistributed parametrization* $\boldsymbol{Q}$ *(bottom images).*

observations can be made in Figure 5.6, where, at $t = 1.5$, a region of the torus is essentially not sampled under the $\boldsymbol{P}$ parametrization. The clear demarcation line between the sampled and empty regions is in fact produced by the perspective of the view angle on the hole of the torus after the flow deformation. Hence, the $\boldsymbol{P}$ sampling indicates that the marker points failed to represent a handle of the genus-1 surface; if a "pure" particles method were used, this can cause errors in the topology of the evolved shapes. For the evolution of the cylinder in Figure 5.7, we also see the above issues in the $\boldsymbol{P}$ parametrization. At $t = 1.5$, the "top face" of the surface, which consists of two diametrically opposite portions of the cylinder that were brought close together by the flow, is poorly sampled by $\boldsymbol{P}$. Without some underlying interpolation of the parametrization, such undersampling could fail to indicate the presence of two distinct pieces of surface. Furthermore, the protruding arc-like part also exhibits poor resolution of the boundary. The piecewise linear interpolation of $\boldsymbol{P}$ is jagged at the



(a) Rectangle. (b) Torus. (c) Cylinder.

FIG. 5.8. *Area distribution of the parametrizations $\boldsymbol{P}$ (blue) and $\boldsymbol{Q}$ (red).*

TABLE 5.3
Parametrizations $\boldsymbol{P}$ and $\boldsymbol{Q}$ at $t = 1.5$.

Surfaces	Rect.	Torus	Cyl.
$\sigma_{\boldsymbol{P}}$	1.6749	0.8828	1.098
$\sigma_{\boldsymbol{Q}}$	0.0094	0.0485	0.0321
$M_{\boldsymbol{P}} - 1$	-0.9722	-0.4754	-0.6718
$M_{\boldsymbol{Q}} - 1$	-0.0007	-0.0001	-0.0019

(a) Standard deviation (σ) and median (M) errors of the length density.

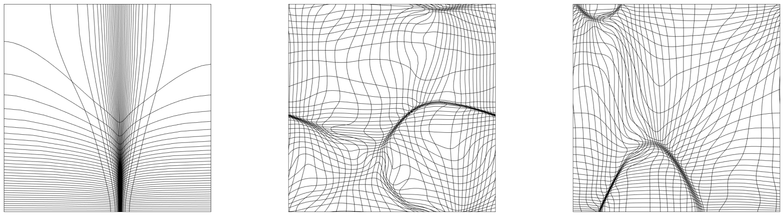
Surfaces	Rect.	Torus	Cyl.
Evaluating $\boldsymbol{Q}$	47.94 s	64.68 s	54.57 s
Defining $\rho^n(\boldsymbol{x})$	9.15 s	9.36 s	9.25 s
Updating $\boldsymbol{\mathcal{X}}_B$	68.86 s	44.29 s	49.06 s
Number of remappings	5	11	5

(b) Total computation times for the evolution of the parametrization $\boldsymbol{Q}$.

boundary, which indicates that $\boldsymbol{P}$ is not smooth enough (in the sense of the growth in magnitude of the higher derivatives) to be accurately represented on a 512^2 grid. With the redistributed parametrization, the marker points generated from $\boldsymbol{Q}$ are uniform and the interpolated surface is smooth.

We quantify the effect of the redistribution by plotting the histogram of the cell area population in Figure 5.8. In all cases, we see that the $\boldsymbol{Q}$ parametrization (in red/orange) generates cells that have almost all the same area, concentrated at the normalized average 1. The $\boldsymbol{P}$ parametrization (in blue), on the other hand, generates large disparity between cell areas, evidential of a nonuniform, suboptimal distribution of marker points. In particular, for the rectangle, we see in Figure 5.8(a) that the area distribution exhibits a Pareto distribution, where the large amount of the surface area is represented by a minority of the cells; this observation clearly reflects the illustrations in Figure 5.5. The reparametrization $\boldsymbol{Q}$ is more uniform, with almost all cells having the average area. These properties are also shown quantitatively in Table 5.3, where we've computed the sample standard deviation and median of the cell areas. In all cases, the standard deviation from the $\boldsymbol{Q}$ parametrization is 1 to 2 orders of magnitude smaller than the one from $\boldsymbol{P}$, and the median error is about 3 to 4 orders of magnitude smaller. In all cases, these improvements came at a cost of a roughly 20% increase over the computational time of the advection (given in section 5.2); this reparametrization time includes the intermittent evaluation of the full $\boldsymbol{Q}$ parametrization (as defined in (4.24)) which is also used to render the surface. The computation of the redistribution map itself accounts for about 10% of the total computation time. The resulting $\boldsymbol{Q}$, however, is a functionally defined parametrization which offers arbitrary resolution at uniform area density.

Indeed, Figure 5.9 shows the redistribution map on the parametric space for each surface at full deformation. As we can see, since the surfaces undergo violent deformations, the transport maps needed to redistribute evenly the area element will also contain large deformations and small scale features: typically this would require a fine grid to compute and resolve. Instead, we use the semigroup structure of the



(a) Rectangle (b) Torus (c) Cylinder

FIG. 5.9. *The redistribution map on the parametric space for each surface at* $t = 1.5$.

characteristic maps to carry out short-time computations on coarse grids; the full time redistribution map is then obtained from the composition of submaps given in (4.24). The advantage of using the decomposition method for the redistribution map is twofold: first, if we assume that at time τ_i, the redistributed parametrization $\boldsymbol{Q}_{\tau_i}$ is close to equiareal, then the variations in $\rho(\boldsymbol{x}, \tau_i)$ are small and therefore can be resolved on a coarse grid for the computation of $\boldsymbol{\mathcal{X}}_{[\tau_{i+1},\tau_i]}$. The resolution of this grid essentially acts as a frequency cut-off selecting the highest frequency in the area element visible to the algorithm. Second, since the evolution of the area density is unknown a priori and can grow in an arbitrary fashion, the semigroup property of the map allows us to decompose the transformation into manageable short-time maps and achieve arbitrary resolution in the global-time map obtained from the composition. This permits the redistribution map to represent large deformations and resolve small scale features in order to compensate for the arbitrary changes in area element.

The numerical experiments in this section demonstrate several practical advantages of representing a moving surface using an equiareal parametrization. Compared to a particle-based method, the parametrization of a moving surface defined by the pushforward of the initial parametrization by the ambient flow map provides a functional definition of the surface at all times. Therefore, resampling can be done by simply evaluating the parametrization function at new sample points. In particle methods, new sample points need to be generated by interpolation which can affect the position of the surface whose accuracy will depend on existing sample points and the curvature of the surface. With the CM method, the accuracy of the parametrization function in respect to the surface shape and location depends only on the forward characteristic maps which we control separately. The precision of the marker locations is therefore independent of existing sample points and of the shape of the surface. Furthermore, coupled with the redistribution method, the equiareal property of the parametrization function is maintained. This means that the sampling density of the parameter space is directly mapped to that of the surface without needing extra computation. Adaptive sampling methods based on the area element of the surface may become inaccurate or inefficient when the variations in area become large. In comparison, the method proposed in this paper maintains an equiareal parametrization at all times, therefore there are no extra computations necessary when computing a uniform sampling of the surface, and area features above a certain spatial scale as indicated in Theorem 4.1 are guaranteed to be resolved.

Remark 5. The redistribution generates an equiareal parametrization; from a sampling point of view, the number of sample points per unit area on the surface should be asymptotically constant. This does not directly translate to a property on the distance between sample points. Indeed, the ratio between the geodesic radius of a disk and its area on the surface is given by the scalar curvature. As a consequence, an equiareal sampling of a surface will have sample points that are more distant from each other where the surface has positive scalar curvature. One way to remedy this would be to require that the equilibrium density of the diffusion be given by the curvature of the surface:

$$\partial_t \rho = \Delta(\rho - \rho_\kappa) \quad \text{with} \quad \boldsymbol{u} = -\frac{\nabla(\rho - \rho_\kappa)}{\rho}, \tag{5.3}$$

where ρ_κ is the target density given by the curvature. The redistribution will then generate an L^2-gradient descent on the difference between ρ and ρ_κ. However, since we no longer have the maximum principle due to the source term, the solution of the heat equation is not guaranteed to stay in the space of probability distributions for all times as densities can temporarily become negative, potentially making the redistribution map singular. So special consideration needs to be taken when the initial and equilibrium densities are far. However, this should generally not be an issue if the curvature changes gradually as the source term will be too small to generate a singularity.

6. Conclusion.

In this work, we have presented a novel method for computing an equiareal parametrization of a curve or surface. We have applied this method to the problem of surface advection and presented an algorithm for evolving the parametrization of a time-dependent moving surface while maintaining an uniform area distribution. The area element of a surface is first pulled back to the parametric space in order to define a density distribution. We define a diffusion process using this density as an initial condition and extract a velocity field from the continuity equation, flowing the initial density toward the uniform density. The backward characteristic map generated from this velocity is used to equidistribute the area element of the surface in the parametric space. In the context of optimal transport, this can be seen as a W^2-gradient descent of the entropy landscape. We studied the mathematical construction of the map in section 2.2 and demonstrated the convergence in distribution of the random variables generated from it. The evolution of the surface under a velocity field is obtained by computing the forward characteristic map in the ambient space as studied in section 4. Combined with the redistribution method, we have that the composition of the backward redistribution map on the parametric space with the initial parametrization followed by the pushforward by the ambient space forward map generates an equiareal parametrization for all times. We then tested this method and provided numerical examples of evolving surfaces in 3D ambient space in section 5. This method is novel and unique in that the changes in the parametrization function are made by precomposition with a deformation of the parametric domain. As a consequence, the image space of the parametrization function is unaffected and we preserve the correct position and shape of the surface. Furthermore, the CM method allows us to exploit the semigroup structures of both the surface advection in the ambient space and the density transport on the parametric space. This allows for the decomposition of both maps into coarse grid submaps of smaller deformations, while maintaining high resolution for the parametrization obtained from the composition. The resulting method is able to track large deformations of the curves and surfaces in the ambient space and redistribute the resulting large variations in area density on the parametric space.

The use of the CM method for evolving equiareal parametrizations of surfaces opens a novel framework with many possibilities for future research. Currently, the method does not provide any mechanism to control the conformity of any meshes generated by the parametrization, and the numerical experiments do not provide any evidence of significant effect on the conformity. Although it is generally impossible to generate a parametrization which is both equiareal and conformal, there is possibility to maintain a parametrization with a trade-off between equidistribution and conformity or mesh-alignment using the CM framework by devising an appropriate redistribution velocity. There may also be interesting methods that couple the equiareal parametrization evolution with a meshing algorithm to generate high quality triangulation meshes on moving surfaces; application of such methods to interface problems such as the Cahn–Hilliard equations [18] could also be of interest. Additionally, since the redistribution framework generalizes directly to any number of dimensions, it may be interesting to investigate its application to volume redistributing flows in 3D space. Last, from a geometric point of view, it would be interesting to study the relation between the diffusion-driven redistribution flow and the geometric flows in the theory of manifold uniformization. We think that these directions of research could be interesting as we believe that the CM method provides a novel and unique approach for solving the problem of surface parametrization and sampling.

Appendix A. Computational cost.

We give a rough estimate of the computational cost for maintaining the area density of a surface constant within an L^2 tolerance of $\mathcal{E}$. We assume that the time step size h_t for the ambient advection is fixed and we do not account for the advection cost below; these can be found in [32]. For simplicity, we do not use any submap decomposition and only estimate the total cost of the redistribution map obtained from the diffusion transport after each ambient advection step as described in Algorithm 4.1. According to the arguments in section 4.2, to achieve an L^2 tolerance of $\mathcal{E}$, the diffusion coefficient ν needs to be on the order of $\mathcal{E}^{-1}$. Equivalently, for each ambient advection step of length h_t, the redistribution map calculations must be carried out over a time interval of length $h_t\mathcal{E}^{-1}$. The size of the time steps for the redistribution map is virtual, however; for an order p time integrator, the time steps need to be of order $\mathcal{E}^{1/p}$ for the numerical error to be smaller than the tolerance. Therefore, $\mathcal{O}(h_t\mathcal{E}^{-\frac{p+1}{p}})$ redistribution steps are needed to achieve the error tolerance. Furthermore, the numerical map $\boldsymbol{\mathcal{X}}_B$ equidistributes a discretized density $\rho_{\boldsymbol{D}}$, which carries a spatial numerical error depending on the size of the computational grid. According to the derivations in section 3, the discretized density $\rho_{\boldsymbol{D}}$ has global spatial error $\mathcal{O}(N^{-\max(2m,1)})$ for a degree $2m+1$ Hermite interpolant on a 2D computational grid of size N^2. Additionally, due to the implicit heat step used to calculate the redistribution velocity, spatial complexity is $\mathcal{O}(N^2\log(N))$ if using FFT spectral methods. Therefore, overall, to achieve $\mathcal{O}(\mathcal{E})$ L^2 redistribution, computations need to be carried out over $\mathcal{O}(\mathcal{E}^{-\frac{1}{q}})$, $q=\max(m,\frac{1}{2})$ spatial grid points for $\mathcal{O}(h_t\mathcal{E}^{-\frac{p+1}{p}})$ time steps per ambient advection step, giving us a global cost of

$$\text{Global computational cost} = \mathcal{O}\left(\mathcal{E}^{-1-\frac{p+q}{pq}}\log(\mathcal{E})\right)$$

for maintaining an uniform parametrization throughout the entire simulation. This error is assuming that ν, cell width, and time steps are scaled at $\mathcal{O}(\mathcal{E}^{-1})$, $\mathcal{O}(\mathcal{E}^{\frac{1}{\max(2m,1)}})$ and $\mathcal{O}(\mathcal{E}^{\frac{1}{p}})$, respectively. However, in practice, we find that the global truncation error is not a very accurate estimate of the uniformity of the density. Indeed, the evolution of $\rho^n_{\boldsymbol{D}}$ approximates a heat equation with the Laplacian operator approximated to the order of the spatial local truncation error. Typically, any error term above that scale can be interpreted as additional source terms which will be decayed by the diffusion process. This means that in practice, it could be more efficient to pick h_x to order

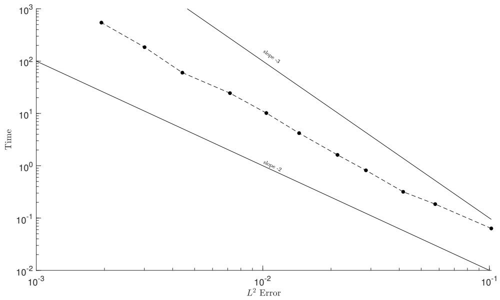
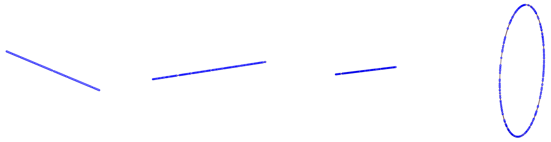


FIG. A.1. *Error versus computational time. The ith data point is given by* $\nu = \frac{1}{2} \times 2^{i/2}, h_x = \frac{1}{16} \times 2^{-i/4}, h_t = \frac{1}{64} \times 2^{-i/2}$.

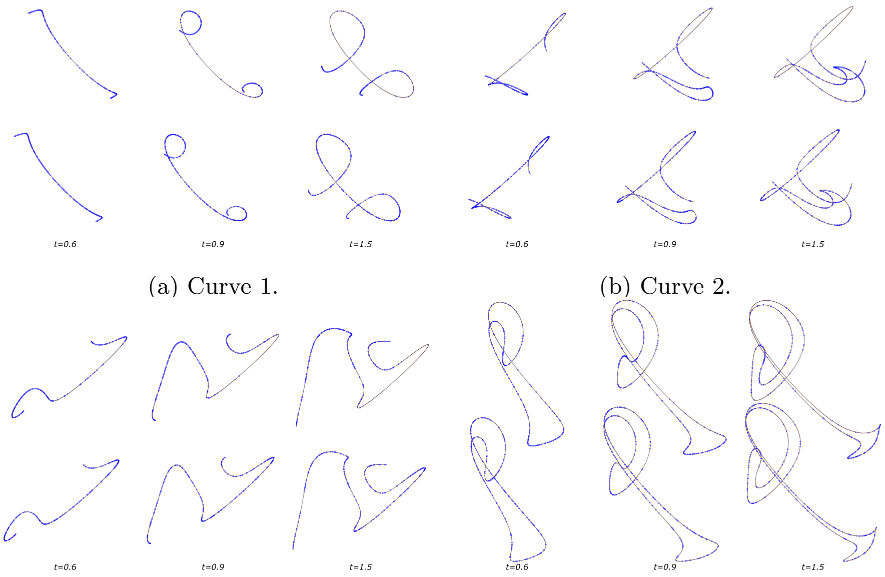
$\mathcal{O}(\mathcal{E}^{\frac{1}{\max(2m+1,2)}})$ instead. Figure A.1 shows an efficiency plot using the above scaling on the tests in section 5.1 for $t \in [0, 1]$.

Appendix B. Time evolution of 1D curves. The curves used in section 5.2.1 are shown here. The initial curves are shown in Figure B.1; the curves at times $t = 0.6$, 0.9, and 1.5 are shown in Figure B.2. We sample each parametrization



(a) Curve 1. (b) Curve 2. (c) Curve 3. (d) Curve 4.

FIG. B.1. *Initial curves with random sampling.*



(a) Curve 1. (b) Curve 2.

(c) Curve 3. (d) Curve 4.

FIG. B.2. *Comparison of the curves sampling using the original parametrization* $\boldsymbol{P}$ *(top images) and the redistributed parametrization* $\boldsymbol{Q}$ *(bottom images).*

function with 256 random marker points sampled from uniform distribution on U which we draw as blue dots over the underlying exact curve in black. The sampling using the $\boldsymbol{P}$ parametrization is shown in the top images and the redistributed $\boldsymbol{Q}$ parametrization in the bottom images.

REFERENCES

[1] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*, Springer, New York, 2008.

[2] D. Appelö and T. Hagstrom, *On advection by Hermite methods*, Pac. J. Appl. Math., 4 (2012), p. 125.

[3] D. Appelo, T. Hagstrom, and A. Vargas, *Hermite methods for the scalar wave equation*, SIAM J. Sci. Comput., 40 (2018), pp. A3902–A3927.

[4] L. Balmelli, G. Taubin, and F. Bernardini, *Space-optimized texture maps*, in Computer Graphics Forum, Vol. 21, Eurographics Association, 2002, pp. 411–420.

[5] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84 (2000), pp. 375–393.

[6] J.-D. Benamou, Y. Brenier, and K. Guittet, *The Monge-Kantorovitch mass transfer and its computational fluid mechanics formulation*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 21–30.

[7] J.-D. Benamou, G. Carlier, and M. Laborde, *An augmented Lagrangian approach to Wasserstein gradient flows and applications*, ESAIM Proc. Surveys, 54 (2016), pp. 1–17.

[8] M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos, *Multilevel adaptive particle methods for convection-diffusion equations*, Multiscale Model. Simul., 4 (2005), pp. 328–357.

[9] M. Bergdorf and P. Koumoutsakos, *A Lagrangian particle-wavelet method*, Multiscale Model. Simul., 5 (2006), pp. 980–995.

[10] C. J. Budd, A. T. McRae, and C. J. Cotter, *The scaling and skewness of optimally transported meshes on the sphere*, J. Comput. Phys., 375 (2018), pp. 540–564.

[11] D. Caetano and C. M. Elliott, *Cahn-Hilliard Equations on an Evolving Surface*, preprint, arXiv:2101.08203, 2021.

[12] G. Carlier, V. Duval, G. Peyré, and B. Schmitzer, *Convergence of entropic schemes for optimal transport and gradient flows*, SIAM J. Math. Anal., 49 (2017), pp. 1385–1418.

[13] J. A. Carrillo, K. Craig, and F. S. Patacchini, *A blob method for diffusion*, Calc. Var. Partial Differential Equations, 58 (2019), pp. 1–53.

[14] J. A. Carrillo, K. Craig, L. Wang, and C. Wei, *Primal Dual Methods for Wasserstein Gradient Flows*, preprint, arXiv:1901.08081, 2019.

[15] B. Dacorogna and J. Moser, *On a Partial Differential Equation Involving the Jacobian Determinant*, in Ann. Inst. H. Poincaŕe Anal. Non Lineaire 7 (1990), pp. 1–26.

[16] K. Deckelnick, G. Dziuk, and C. M. Elliott, *Computation of geometric partial differential equations and mean curvature flow*, Acta Numer., 14 (2005), p. 139.

[17] G. Dziuk and C. M. Elliott, *Finite element methods for surface PDEs*, Acta Numer., 22 (2013), pp. 289–396.

[18] P. Gera and D. Salac, *Cahn-Hilliard on surfaces: A numerical study*, Appl. Math. Lett., 73 (2017), pp. 56–61.

[19] P. Gera and D. Salac, *Modeling of multicomponent three-dimensional vesicles*, Comput. & Fluids, 172 (2018), pp. 362–383.

[20] F. Gibou, R. Fedkiw, and S. Osher, *A review of level-set methods and some recent applications*, J. Comput. Phys., 353 (2018), pp. 82–109.

[21] 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. Medical Imaging, 23 (2004), pp. 949–958.

[22] X. Gu and S.-T. Yau, *Global conformal surface parameterization*, in Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Eurographics Association, 2003, pp. 127–137.

[23] T. Hagstrom and D. Appelö, *Solving PDEs with Hermite interpolation*, in Spectral and High Order Methods for Partial Differential Equations, ICOSAHOM 2014, Springer, New York, 2015, pp. 31–49.

[24] S. E. Hieber and P. Koumoutsakos, *A Lagrangian particle level set method*, J. Comput. Phys., 210 (2005), pp. 342–367.

[25] W. Huang and R. D. Russell, *Adaptive Moving Mesh Methods*, Appl. Math. Sci. 174, Springer, New York, 2010.

[26] M. Jin, Y. Wang, S.-T. Yau, and X. Gu, *Optimal global conformal surface parameterization*, in Proceedings of IEEE Visualization 2004, IEEE, 2004, pp. 267–274.

[27] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the Fokker-Planck equation*, SIAM J. Math. Anal., 29 (1998), pp. 1–17.

[28] 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 (2012), pp. 1952–1969.

[29] G. Lacombe and S. Mas-Gallic, *Presentation and analysis of a diffusion-velocity method*, in ESAIM Proc., 7, (1999), pp. 225–233.

[30] R. J. LeVeque, *High-resolution conservative algorithms for advection in incompressible flow*, SIAM J. Numer. Anal., 33 (1996), pp. 627–665.

[31] A. T. McRae, C. J. Cotter, and C. J. Budd, *Optimal-transport–based mesh adaptivity on the plane and sphere using finite elements*, SIAM J. Sci. Comput., 40 (2018), pp. A1121–A1148.

[32] O. Mercier, X.-Y. Yin, and J.-C. Nave, *The characteristic mapping method for the linear advection of arbitrary sets*, SIAM J. Sci. Comput., 42 (2020), pp. A1663–A1685.

[33] P. Mycek, G. Pinon, G. Germain, and E. Rivoalen, *Formulation and analysis of a diffusion-velocity particle model for transport-dispersion equations*, Comput. Appl. Math., 35 (2016), pp. 447–473.

[34] J.-C. Nave, R. R. Rosales, and B. Seibold, *A gradient-augmented level set method with an optimally local, coherent advection scheme*, J. Comput. Phys., 229 (2010), pp. 3802–3827.

[35] G. Peyré, *Entropic approximation of Wasserstein gradient flows*, SIAM J. Imaging Sciences, 8 (2015), pp. 2323–2351.

[36] F. Santambrogio, *Optimal Transport for Applied Mathematicians*, Springer, New York, 2015.

[37] F. Santambrogio, {*Euclidean, metric, and Wasserstein*} *gradient flows: An overview*, Bull. Math. Sci., 7 (2017), pp. 87–154.

[38] B. Seibold, J.-C. Nave, and R. R. Rosales, *Jet schemes for advection problems*, Discrete Contin. Dyn. Sys. Ser. B, 17 (2012), pp. 1229–1259.

[39] 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.

[40] Z. Su, J. Sun, X. Gu, F. Luo, and S.-T. Yau, *Optimal mass transport for geometric modeling based on variational principles in convex geometry*, Engineering with Computers, 30 (2014), pp. 475–486.

[41] C. Villani, *Optimal Transport: Old and New*, Grundhehren Math. Wiss. 338, Springer, 2008.

[42] A. M. Winslow, *Adaptive-mesh Zoning by the Equipotential Method*, Tech. report, Lawrence Livermore National Laboratory, Livermore, CA, 1981.

[43] X.-Y. Yin, O. Mercier, B. Yadav, K. Schneider, and J.-C. Nave, *A characteristic mapping method for the two-dimensional incompressible Euler equations*, J. Comput. Phys., 424 (2021), 109781.

[44] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, *A fast and simple stretch-minimizing mesh parameterization*, in Proceedings of Shape Modeling Applications, IEEE, 2004, pp. 200–208.

[45] 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. Visualization and Computer Graphics, 19 (2013), pp. 2838–2847.