Characteristic Mapping Framework with Applications to Density Transport and Fluid Dynamics

Xi-Yuan Yin

Department of Mathematics and Statistics

McGill University
Montreal, Quebec

December 2021

A thesis submitted to McGill University in partial fulfilment of the requirements of the degree of Doctor of Philosophy in Science in Mathematics and Statistics

ACKNOWLEDGEMENTS

I would like to thank first and foremost my mentor and supervisor Professor Jean-Christophe Nave, whose guidance over all these years played the central role in the progress of this project. I am grateful for the training and opportunities he provided me in order to further my academic capabilities. I would also like to thank Professor Linan Chen and Professor Kai Schneider for their insightful inputs and discussions on several research topics. I also thank the Mathematics and Statistics department of McGill University, the Fonds de recherche du Québec, the Hydro-Québec fellowship, the Alexis D. and W. Charles Pelletier Fellowship and Professor Nave for supporting my studies with scholarships, teaching experiences and opportunities to participate in seminars and conferences.

The graduate students community in the department has also provided a great environment for my graduate studies and my research. I'd like to thank Olivier Mercier and Badal Yadav for their cooperation on the research featured in this thesis. I am also very appreciative of the Applied Mathematics group and the Applied Mathematics student seminars for fostering the camaraderie and mutual support between the students, among which I want to thank in particular: Geoff, Ivan, Damien, Seth, Matthieu, Manuel and Noah for their insightful discussions on various mathematical topics. The graduate students in the department have always been very welcoming and cheerful, they have brought a very enjoyable social aspect to my experience of graduate school.

Finally, I wish to thank my family and my friends who have always been there for me, and have supported me in various ways, whose company I always enjoy and who have made these years of graduate studies all the more memorable.

ABSTRACT

We study in this thesis the Characteristic Mapping (CM) framework and its application in various scientific computing problems involving transported quantities. The CM framework is a novel numerical approach for general transport equations; it consists in discretizing the deformation map generated by the transport flow instead of the transported quantities themselves. This computation yields the solution operator of the advection equation, called the characteristic map, which can then be used to transport any quantity evolving under the same transport flow. The CM method is distinguished by several unique numerical properties made possible by its geometric approach: any number of transported quantities can be simultaneously and coherently evolved using the same characteristic map, the evaluation of advected fields through map pullbacks guarantees relabelling symmetry and thus eliminates artificial viscous dissipation, the group structure of the characteristic maps can also be used to decompose long-time maps into a number of short-time submaps allowing for efficient coarse grid computations. The resulting numerical method is efficient, accurate and well-suited for multiscale problems such as the simulation of incompressible inviscid fluids. We will first motivate the CM approach from the linear advection equation. We will also study the some underlying theory and preliminaries on which the CM method is built, in particular, we will examine relevant differential calculus concepts which allow us to formulate the characteristic map as the solution operator to the Lie-advection equation. As a first application, we study the use of the CM method for the problem of equiareal surface parametrization in the context of passively advected surfaces. In this case, the CM method is used to produce a redistribution map on the parametric space which, when composed with the evolving parametrization, yields an equiareal parametrization of the surface. Next, we apply the CM method to the simulation of perfect fluids, first through the incompressible Euler equations in two-dimensional space, and then extending the method to the three-dimensional case. In two-dimensions, the characteristic map is used to evolve the advected scalar vorticity field; the velocity field used for the vorticity transport is obtained from the Biot-Savart law. This allows for a non-dissipative transport of the vorticity resulting in an arbitrary spatial resolution for the solution. In the three-dimensional case, the vorticity vector field is no longer a scalar advected quantity, instead using the Kelvin circulation theorem, we reformulate the equation as the Lie-advection of the vorticity field expressed as a differential 2-form. The time t vorticity field can then be constructed from the initial vorticity via pullback by the characteristic map. Error estimates are derived for these methods, in particular, a more geometric analysis in the three-dimensional case demonstrates the advective nature of the error as one can show that the numerical vorticity is related to the true vorticity through pullback by an error map. Numerical convergence tests are also performed and comparisons with results obtained using other current methods are made. Finally, we conclude by examining some promising directions for future investigation.

ABRÉGÉ

La matière à l'étude dans cette thèse est le cadre mathématique de la méthode de l'Application Caractéristique (AC) ainsi que son application aux divers problèmes de nature advective dans le domaine du calcul scientifique. La méthode de l'AC constitue une approche numérique nouvelle destinée aux équations de transport; elle consiste à discrétiser l'application de déformation générée par l'écoulement lié au transport au lieu de discrétiser les quantités transportées elles-mêmes. Le produit de cette discrétisation est l'opérateur de solution nommé l'application caractéristique, celle-ci peut être utilisée pour évoluer toute quantité transportée par le même écoulement. La méthode de l'AC se distingue par plusieurs propriétés numériques uniques grâce à son approche géométrique: toute quantité transportée peut être évoluée par la même application de façon simultanée et cohérente, l'évaluation des champs transportés comme étant l'image réciproque de la condition initiale par l'application caractéristique garantie la préservation de la symétrie de «réétiquetage» et prévient ainsi la dissipation visqueuse artificielle, la structure de groupe des applications caractéristiques permet aussi la décomposition des applications de long-terme en plusieurs sous-applications de court-terme, ces dernières pouvant être efficacement calculées sur des maillages de basse résolution. La méthode numérique qui en résulte est efficace, précise, et bien adaptée aux problèmes multiéchelles telle la simulation des fluides incompressibles et non visqueux. Nous motivons premièrement l'approche de l'AC par l'exemple de l'équation d'advection linéaire. Nous étudions aussi les théories et les notions préliminaires sur lesquelles cette méthode est bâtie, en particulier, les notions pertinentes du calcul différentiel nous permettant de formuler l'application caractéristique comme étant l'opérateur de solution pour une équation d'advection de Lie seront étudiées. Comme premier exemple, nous examinons l'utilisation de la méthode de l'AC au problème de l'uniformisation des surfaces paramétrées, évoluant passivement sous l'influence d'un écoulement. Dans ce contexte, la méthode de l'AC est utilisée pour générer sur le domaine paramétrique une application de redistribution qui, une fois composée avec la fonction de paramétrage, produit un nouveau paramétrage authalique de la surface. Ensuite, nous employons la méthode de l'AC à la simulation des fluides parfaits au moyen des équations d'Euler, d'abord dans le cas bidimensionnel, puis au cas tridimensionnel en apportant les extensions nécessaires à la méthode. En deux dimensions, l'application caractéristique est utilisée pour transporter le champ de vorticité scalaire; le champ de vitesse associé à ce transport provient de la loi de Biot-Savart. Cet approche permet un transport non dissipatif de la vorticité et une résolution spatiale arbitraire de la solution. Dans le cas tridimensionnel par contre, le champ de vorticité n'est plus une quantité scalaire advectée, plutôt, par le théorème de Kelvin sur la conservation de la circulation de la vitesse, nous reformulons l'équation de l'évolution de la vorticité en tant qu'advection de Lie du vecteur vorticité exprimé comme forme différentielle de degré 2. La vorticité à temps t peut alors être calculé via l'image réciproque de la vorticité initiale par l'application caractéristique. Nous présentons des estimations d'erreurs pour ces méthodes. Notamment, une analyse plus géométrique dans le cas tridimensionnel démontre la nature advective de l'erreur, puisque la vorticité obtenue numériquement est liée à la vorticité exacte à travers une application de transport. Nous effectuons des tests de convergence ainsi que des comparaisons avec les résultats obtenus utilisant d'autres méthodes actuelles. Finalement nous concluons avec une examination des directions potentielles pour les recherches futures.

STATEMENT OF CONTRIBUTION

The contents of this thesis are considered original scholarship and distinct contributions to knowledge. The particle management method for parametric surfaces discussed in Chapter 3 was published with Professor Linan Chen and Professor Jean-Christophe Nave in [121]. The work on the Characteristic Mapping method for the two-dimensional incompressible Euler equations in Chapter 4 is original and was inspired by preliminary investigations carried out by Olivier Mercier and Badal Yadav. The content of this chapter was published with Olivier Mercier, Badal Yadav, Professor Kai Schneider and Professor Jean-Christophe Nave in [122]. The work on the Characteristic Mapping method for the three-dimensional Euler equations is original and has been submitted for publication with Professor Kai Schneider and Professor Jean-Christophe Nave.

All chapters, figures, tables and numerical algorithms used to generate the results are written and implemented by Xi-Yuan Yin. The research in this thesis was conducted by Xi-Yuan Yin under the direction and guidance of Professor Jean-Christophe Nave.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

The advection equation plays an important role in many scientific and industrial applications; many physical phenomena are modelled using evolution equations which often involve an advected quantity. For instance, the advection-diffusion-reaction equation is used to model the behaviour of solute particles, densities or other physical properties evolving due to diffusion and reaction while being passively transported by an ambient velocity field. In the study of optimal transport, it is also possible to generate a transport map by advecting a probability density. In fluid dynamics, the Navier-Stokes and the Euler equations are used to model the behaviour of viscous and inviscid fluids; in these equations, the non-linear "self-advection" of the fluid velocity plays a central role.

The ubiquity of the advection equation and its wide range of applications motivated a great number of mathematical investigations in particular in the design and analysis of efficient numerical methods. Current methods can be roughly divided into two approaches, namely using the Lagrangian or the Eulerian descriptions of the flow. This duality is in fact a central theme in the framework presented in this thesis. These two approaches differ in the reference frame used to describe the fluid motion. Whereas Lagrangian methods aim at discretizing material parcels, for which the time 0 space can be used as reference frame to label each particle, Eulerian methods represent quantities on a fixed spatial domain, using the time t space as reference frame. Each approach leads to a set of unique numerical properties with their advantages and disadvantages. Lagrangian methods are generally less dissipative and computationally efficient while Eulerian methods have a more controlled spatial resolution and convenient numerical representation at the cost of higher artificial dissipation which must be reduced using larger computationally demanding discretizations. The Characteristic Mapping (CM) method studied in this thesis is a novel numerical framework which aims at exploiting the advantages of both the Lagrangian and Eulerian approaches; this is done by discretizing the backward flow map of the transport, called the characteristic map. This map is in fact the Eulerian to Lagrangian coordinate transformation and allows for a semi-Lagrangian discretization of the transport while keeping an Eulerian spatial representation of all transported quantities. The resulting scheme is non-dissipative, preserves all scales and provides arbitrary resolution of the solution at the cost of fast coarse grid computations.

We will study the application of the CM method to several problems in computational sciences involving a transport equation. The starting point of this investigation is the linear advection equation

$$\partial_t \phi + (\boldsymbol{u} \cdot \nabla)\phi = 0, \tag{1.1}$$

$$\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}), \tag{1.2}$$

where ϕ is a scalar advected quantity and $\boldsymbol{u}$ is a given advection velocity field. This equation models the passive transport of a quantity ϕ under the flow generated by $\boldsymbol{u}$. For instance, ϕ can represent a solute density or temperature field of a fluid, the equation then models the evolution of these fields when diffusion is negligible. The advection equation is hyperbolic in nature and has a characteristic structure which defines a domain of dependence for the solution at each specific point in time and space. It is therefore a stability requirement that the domain of dependence of the numerical scheme contains the true domain of dependence. Classical Eulerian schemes for the linear advection equation include the Courant-Isaacson-Rees upwind schemes [33], and the Lax-Wendroff and Lax-Friedrich methods. The size of the time steps in explicit Eulerian schemes are typically restricted by a Courant-Friedrichs-Lewy (CFL) condition, this condition guarantees that at each step, the exact characteristic curves of the solution at each grid point do not exit the domain of dependence of the numerical scheme. These schemes are based on low order spatial representations, indeed, as illustrated by Godunov's order barrier theorem, higher order linear schemes suffer from stability issues due to the formation of spurious oscillations near a discontinuity or high-gradient feature of the solution. The Essentially Non-Oscillatory [57] (ENO) schemes and its weighted variant [69,80] (WENO) as well as their corresponding formulations on unstructured meshes [44, 63] are a popular family of methods for cases where solutions may contain discontinuities; a thorough overview of these methods is given in chapter 5 of [9]. These methods achieve higher order convergence by using wide adaptive stencils which are spatially biased according to local smoothness of the solution in order to avoid discretizing a discontinuity. As a result, the ENO and WENO methods yield uniformly high order accurate solutions while resolving discontinuities and high gradients without generating spurious oscillations. These numerical properties make the methods suitable for solving hyperbolic equations where regions of discontinuities mix with smooth structures of the solution. For instance, the ENO and WENO schemes are used in the numerical solution of incompressible [117,125] and compressible [46,109,110,126] Euler and Navier-Stokes equations. The non-oscillatory schemes have also been applied to various advection dominated problems, and Hamilton-Jacobi equations [68, 93]. The ENO and WENO methods achieve high order convergence through the use of wide stencils which need to be adapted to avoid discontinuities, other approaches instead rely on compact stencils and achieve high accuracy using higher derivatives information. For instance, high degree piecewise Hermite polynomials are used in [2, 3, 56] to represent and evolve solutions of advection dominated and hyperbolic PDEs, in these cases, the high degree of the interpolating polynomials increase accuracy in smooth regions of the solution and the compact stencil and semi-norm reducing properties of the scheme provide stability even in the case of non-smooth solutions. The semi-Lagrangian gradient-augmented level-set (GALS) [89] and Jet schemes [108] methods also employ a compact discretization stencil and achieve high order accuracy by evolving a functional representation of the solution in higher regularity space. In particular, these methods use the characteristic structure of the advection equation to perform time evolution of the numerical solution and therefore have a natural upwinding property with an optimally local discretization and a superconsistent scheme for evolving the solution in a finite-dimensional function space. Additionally, the semi-Lagrangian structure imply that they are typically not restricted by the CFL constraint. This approach has since then been extended for adaptive multi-resolution grids [75] as well as for unstructured meshes [17]. The CM framework studied in this thesis is based on these methods; it is novel in that the discretized quantity is the backward-in-time flow map of the transport velocity which serves as the Eulerian to Lagrangian coordinate transformation. For the linear scalar advection equation, this has been studied in [74] and [84], where the time t solution of the advection is obtained from the function composition of the characteristic map with the initial condition. In particular, in [84], this approach was used to generalize the level-set formalism [92] in order to evolve implicitly defined surfaces and general spatial sets with arbitrary resolution using efficient coarse grid computations.

In this thesis, we study the application of the CM framework to more general transport equations, in particular to the Lie-advection of differential forms. This uses the fact that the characteristic map is the backward flow map of the transport velocity, therefore, all forms which are Lie-advected by the velocity can be written as the pullback of the initial condition by the map. The theoretical preliminaries for this formulation will be studied in chapter 2. The notion of Lie-advection for differential forms can be applied to various evolution equations arising from conservation laws. For instance, a positive density field evolving under its continuity equation can be seen as the Lie-transport of a volume-form. We will study this in chapter 3 in the context of equiareal parametrizations for moving surfaces. In contrast to [84], the surfaces in chapter 3 are defined explicitly through a parametrization function and are represented using marker points obtained by sampling the parametric space. The explicit representations can be useful in cases where the surface topology does not admit an implicit representation, for instance for non-orientable surfaces. However, due to the parametric nature of the approach, surface representation can deteriorate due to the ambient flow deformations and marker point density can become highly uneven. The main idea of chapter 3 is to penalize the resulting changes in the area-element of the parametrization by treating the surface area as a volume-form on the parametric space. An equiareal redistribution on the parametric space is then computed to uniformize the surface; in this case, a heat equation written as a continuity equation is used to equidistribute the area element, the characteristic map computed from the associated velocity field then acts as the redistribution map. This is related to the optimal transport problem concerned with finding an optimal mapping between two probability measures. In particular, the CM method uses an approach similar to the fluid mechanics intepretation of the Monge-Kantorovich problem studied in [12, 13] where a transport velocity is selected to generate the redistribution map. The diffusion velocity used in chapter 3 can be compared with the JKO scheme studied in [71], in this sense, the redistribution method studied here can be seen as a penalization on changes in area-element through a Wasserstein-2 gradient descent of the Gibbs-Boltzmann entropy.

An important application of the CM framework is the numerical study of fluid dynamics. Turbulence remains one of the oldest and most challenging research problems in both pure and applied science; the high Reynolds number limit of the Navier-Stokes equations is of particular interest and developments in scientific computing are useful in advancing the frontiers of our understanding of fluid dynamics in a highly nonlinear regime. However the multiscale nature of incompressible inviscid fluids requires the development of specialized algorithms. Among the Eulerian approaches using a fixed computational grid, Fourier pseudo-spectral methods are very popular due to their high accuracy and fast computational speeds [21,22,39,65,66,76,81,91]. However, purely Eulerian schemes are typically constrained by the CFL condition, this implies that complexity of simulations with N grid points per spatial dimension is proportional to N^3 in 2D and N^4 or even worse in 3D [106]. Furthermore, Eulerian methods are often prone to artificial dissipation or require some viscous or hyperviscous regularization for stability (see e.g. discussion in [40]). Special care needs to be taken in the spatial resolution of the solution and when using spectral dealiasing; spatial adaptivity and parallel implementation can play an important role in improving computational efficiency [55, 75, 85]. A comparative study of spatial discretization and spectral dealiasing methods for the Euler equations can be found in [51]. Generally, high-resolution computational grids are needed to reduce the effects of dissipation, currently pseudo-spectral simulations with up to 12288^3 grid points can be performed [67]. On the other hand, Lagrangian and semi-Lagrangian methods do not suffer from the time-step restrictions due to the CFL condition (see e.g. Staniford and Côté [112]) and can be less computationally demanding for advection dominated problems. Purely Lagrangian methods include the vortex blob [11, 27, 62, 90], vortex particle and vortex filament methods [30, 100, 101, 118]. These methods are characterized by a particle-based discretization of the vorticity field; the motion of the fluid is idealized as the transport of a collection of point vortices or compactly supported vortex blobs and the velocity of each particle can be computed from vortices using the Biot-Savart law. These methods are inherently spatially adaptive since the representation of the vorticity field is reduced to a collection of point vortices concentrated where the vorticity is important. Furthermore, they are more effective in avoiding artificial viscous dissipation compared to their Eulerian counterparts. For instance, Bowman, Yassaei and Basu [18] recently proposed a fully Lagrangian scheme based on the relabelling symmetry of the advection problem, this scheme is characterised by a non-dissipative transport of the solution and the conservation of all Casimir invariants. Some drawbacks of the Lagrangian approach include the difficulty in the representation and controlled resolution of Eulerian quantities. Methods for transferring Lagrangian quantities to fixed Eulerian grids include the vortex-in-cell methods [28, 31, 32, 104] and the Cauchy-Lagrangian frameworks [97]. The approach we study in this thesis uses the CM method as solution operator for the transport of the vorticity field in the Euler equations. Indeed, through the Kelvin circulation theorem, by writing the vorticity as a differential 2-form, the Euler equations can be interpreted as the Lie-advection of the vorticity under the fluid velocity. This allows for the evaluation of the time t vorticity field using pullback by the characteristic map. In the 2D case, for incompressible flows, the vorticity 2-form can be identified with an advected scalar function and the method consists of a coupling of the CM method for scalar advection and a Biot-Savart law yielding the velocity field from a given vorticity. In the 3D case, the Lie-advection of the vorticity 2-form depends on the local material deformation of the flow which involves the Jacobian matrix of the characteristic map. In both cases, a key feature of this approach is that numerical viscous dissipation is avoided since the vorticity is reconstructed by map pullback at each step; we will show that the error is instead dispersive and the modified equation is related to the Euler-α equations. As a consequence of the CM approach, the numerical solution preserves all fine scales and provides arbitrary spatial resolution of the solution.

The remainder of this thesis is organized as follows. In chapter 2, we go over some theoretical and numerical preliminaries needed for the formulation of the CM method as well as its application to the linear advection equation previously studied in [84]. In chapter 3 we examine the application of the CM method to a density transport problem in the context of generating equiareal parametrizations for a moving surface. In chapter 4, we extend the CM method for linear advection to the incompressible Euler equations in two-dimensional space. In its vorticity form, the 2D Euler equation consists in a scalar vorticity advection under the corresponding velocity field obtained from Biot-Savart law. In chapter 5, we study the extension of the CM method for 2D Euler equations to the 3D case where the vorticity is no longer an advected scalar quantity. Instead, as a differential 2-form, the vorticity is Lie-advected as consequence of the Kelvin circulation theorem. Lastly, chapter 6 contains some concluding remarks and proposes potential directions for future research.

CHAPTER 2
Preliminaries

2.1 Mathematical formulation of the Characteristic Mapping method

In this section, we summarize some of the preliminary notions regarding the Characteristic Mapping framework based on which the later chapters are built. The starting point of the Characteristic Mapping (CM) framework is the linear advection equation. We consider an advection equation on a domain $U \subset \mathbb{R}^d$ with a given time-dependent velocity field $\boldsymbol{u} : U \times \mathbb{R}_+ \to \mathbb{R}^d$ satisfying a no penetration boundary condition $\boldsymbol{u} \cdot \boldsymbol{n}_{\partial U} = 0$ on ∂U, where $\boldsymbol{n}_{\partial U}$ is the boundary normal. For most of this thesis, the domain is simply assumed to be a periodic d-torus and no boundary conditions apply. The linear advection equation under this velocity field is given by

$$(\partial_t + \boldsymbol{u} \cdot \nabla)\phi = 0, \tag{2.1a}$$

$$\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}), \tag{2.1b}$$

for some initial condition ϕ_0. In this case, ϕ is a transported quantity under the flow of $\boldsymbol{u}$.

Solutions to this linear advection equation can be expressed using the method of characteristics. We consider characteristic curves $\boldsymbol{\gamma} : \mathbb{R}_+ \to U$ satisfying the following ODE

$$\frac{d}{dt}\boldsymbol{\gamma} = \boldsymbol{u}(\boldsymbol{\gamma}, t), \tag{2.2a}$$

$$\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}_0, \tag{2.2b}$$

for some initial position $\boldsymbol{\gamma}_0 \in U$.

For a solution of equation (2.1), one can check that the advected quantity ϕ is constant along characteristic curves, that is

$$\frac{d}{dt}\phi(\boldsymbol{\gamma}(t),t)=0, \tag{2.3}$$

or equivalently

$$\phi(\boldsymbol{\gamma}(t),t)=\phi_0(\boldsymbol{\gamma}_0). \tag{2.4}$$

We define the characteristic map $\boldsymbol{X}_{[t_1,t_2]}$ associated with the velocity $\boldsymbol{u}$ to be the solution operator for the characteristic curves in the interval $[t_1,t_2]$, that is,

$$\boldsymbol{X}_{[t_1,t_2]}(\boldsymbol{\gamma}(t_1))=\boldsymbol{\gamma}(t_2) \tag{2.5}$$

for all times t_1,t_2 and for all characteristic curves $\boldsymbol{\gamma}$. One can check that, keeping t_0 fixed, the characteristic map satisfies the following equations

$$\partial_t\boldsymbol{X}_{[t_0,t]}=\boldsymbol{u}(\boldsymbol{X}_{[t_0,t]},t), \tag{2.6a}$$

$$(\partial_t+\boldsymbol{u}\cdot\nabla)\boldsymbol{X}_{[t,t_0]}=0. \tag{2.6b}$$

The map $\boldsymbol{X}_{[t_1,t_2]}$ can be thought of as a transformation of the space from time t_1 to t_2 following the flow. There is no requirement that $t_1<t_2$, if $t_1<t_2$, the map is forward in time and we will call it the forward map, if $t_1>t_2$, we call it the backward map. It is straightforward to check the following properties of the characteristic maps:

$$\boldsymbol{X}_{[t_1,t_2]}\circ\boldsymbol{X}_{[t_0,t_1]}=\boldsymbol{X}_{[t_0,t_2]}, \tag{2.7a}$$

$$\boldsymbol{X}^{-1}_{[t_0,t_1]}=\boldsymbol{X}_{[t_1,t_0]}, \tag{2.7b}$$

$$\boldsymbol{X}_{[t_0,t_0]}=\boldsymbol{id}, \tag{2.7c}$$

for arbitrary t_0,t_1,t_2. Indeed, for a given divergence-free velocity field $\boldsymbol{u}$, the characteristic maps $\boldsymbol{X}$ are elements of the Lie group of volume-preserving diffeomorphisms

SDiff(U), with the space of divergence-free vector fields as its Lie algebra. For simplicity of notation, we will denote the forward map $\boldsymbol{X}_{[0,t]}(\boldsymbol{x})$ by $\boldsymbol{X}_F(\boldsymbol{x},t)$ and the backward map $\boldsymbol{X}_{[t,0]}(\boldsymbol{x})$ by $\boldsymbol{X}_B(\boldsymbol{x},t)$. As a function of t, we can formally see $\boldsymbol{X}_F$ as the integral curve of the time-dependent velocity field $\boldsymbol{u}$ on *SDiff*(U) starting from the identity map; $\boldsymbol{X}_B$ is the corresponding inverse element of the group for each time t. In the context of the Euler equations, the treatment of incompressible inviscid flow as geodesic curves of the Lie group of volume preserving diffeomorphisms can be found in the work of Arnold [4].

Remark 2.1.1. We note that the notation for the backward map $\boldsymbol{X}_{[t,0]}$ or more generally, $\boldsymbol{X}_{[t_2,t_1]}$ for $t_2 > t_1$ refers to a map going backward in time. The corresponding interval in the subscript is therefore written in reverse order to remind the fact that the map is backwards in time. For most of the methods in this thesis we will only make use of the backward map in the numerical methods. The forward map would correspond instead to a Lagrangian approach in contrast to the Eulerian and semi-Lagrangian methods presented here.

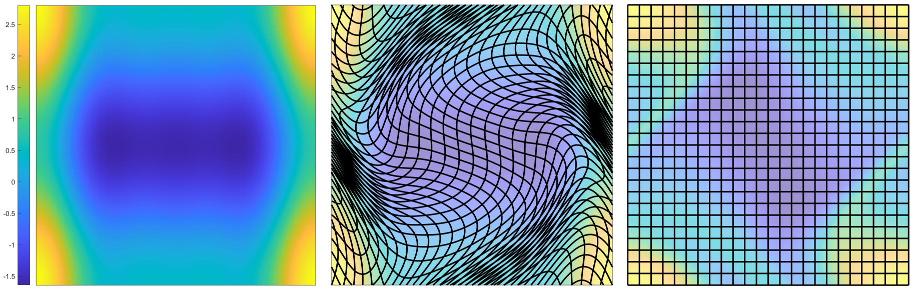
The characteristic maps act as solution operators to the transport equations. Consider the advection equation (2.1). From the method of characteristics, we know that $\frac{d}{dt}\phi(\boldsymbol{\gamma}(t),t) = 0$ for any characteristic $\boldsymbol{\gamma}$ given by (2.2). It follows that

$$\phi(\boldsymbol{x},t) = \phi_0(\boldsymbol{X}_{[t,0]}(\boldsymbol{x},t)). \tag{2.8}$$

The above approach for solving the linear advection equation was studied in [84] where it was applied to the linear advection of level-set surfaces as well as general implicitly defined arbitrary sets.

As illustrated in figure 2–1, the time t values of the advected quantity (shown in background colors) is obtained by picking the value of ϕ_0 at the time 0 location $\boldsymbol{X}_B(\boldsymbol{x},t)$. Indeed, $\boldsymbol{X}_B$ allows us to switch between Lagrangian and Eulerian frames. One can think of $\boldsymbol{X}_B$ as identifying the characteristic curve passing through $\boldsymbol{x}$ at time t and returning the location of the corresponding particle in U at time 0 which, by convention, we use as the Lagrangian reference space. In geometric terms, since

ϕ is Lie-advected by $\boldsymbol{u}$, i.e. $(\partial_t + \mathcal{L}_{\boldsymbol{u}})\phi = 0$, we have that ϕ is given by the pullback ${\boldsymbol{X}_B}^*\phi_0$. This is also called the relabelling symmetry or back-to-label map. Here, ϕ_0 is a 0-form, however this is true for higher order forms and will be our main tool for solving the linear and nonlinear transport equations in this thesis.



(a) Initial condition.

(b) Domain deformation from characteristic map.

(c) Pullback of the initial condition by the map.

Figure 2–1: Evolution of the characteristic map acting on the initial condition.

The main approach of the CM framework is to use the fact that the pullback action by the map $\boldsymbol{X}_B$ is the solution operation for the Lie-advection equation under the velocity field $\boldsymbol{u}$, in the sense that for any initial condition ϕ_0, the solution ϕ to $(\partial_t + \mathcal{L}_{\boldsymbol{u}})\phi = 0$ is given by the pullback ${\boldsymbol{X}_B}^*\phi_0$. The numerical method then consists in discretizing the evolution equation (2.6b) in order to approximate the map $\boldsymbol{X}_B$. The numerical scheme is based on the group property (2.7a) which is discretized according to the following general framework. Throughout this thesis, we will denote by the script letter $\boldsymbol{\mathcal{X}}$ the numerical approximation of the characteristic map in some finite-dimensional function space, the superscript n on a variable will denote the evaluation of said variable at time t_n and the tilde indicates an approximation or modified equation. The numerical framework comprises of the following parts:

1. A one-step map is approximated using a time-integration scheme: $\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}) \approx \boldsymbol{x} - \Delta t \tilde{\boldsymbol{u}}^n(\boldsymbol{x})$. In the case of linear advection $\tilde{\boldsymbol{u}}^n$ is provided, and in the case of the Euler equations or nonlinear advection, $\tilde{\boldsymbol{u}}^n$ will be computed and will depend on $\boldsymbol{\mathcal{X}}_B$.

2. A time update for the characteristic map based on the group property (2.7a). The map $\boldsymbol{\mathcal{X}}_{[t_{n+1},0]}$ at time t_{n+1} will be given by

$$\boldsymbol{\mathcal{X}}_{[t_{n+1},0]} = \mathcal{H}\left[\boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}\right], \tag{2.9}$$

using some spatial interpolation operator $\mathcal{H}$ which projects smooth function to a discretization function space $\mathcal{V}$. When $\mathcal{V}$ is the space of Hermite cubic splines, equation (2.9) is the Gradient-Augmented Level-Set [89] (GALS) method.

The numerical solution to the advection problem (2.1) is then given by

$$\tilde{\phi}^n(\boldsymbol{x}) = \phi_0(\boldsymbol{\mathcal{X}}_{[t_n,0]}(\boldsymbol{x})). \tag{2.10}$$

We can further use the group property (2.7a) to decompose the characteristic map. For some fixed remapping times $0 = T_0 < T_1 < \ldots < T_m = t$, we have

$$\boldsymbol{X}_{[t,0]} = \boldsymbol{X}_{[T_1,0]} \circ \boldsymbol{X}_{[T_2,T_1]} \circ \cdots \boldsymbol{X}_{[t,T_{m-1}]}, \tag{2.11}$$

which we can use to define the following numerical solution

$$\tilde{\phi}^n = \phi_0 \circ \boldsymbol{\mathcal{X}}_{[T_1,0]} \circ \boldsymbol{\mathcal{X}}_{[T_2,T_1]} \circ \cdots \boldsymbol{\mathcal{X}}_{[t,T_{m-1}]}. \tag{2.12}$$

We note the difference between this decomposition and the time-stepping scheme in (2.9): In (2.9) the group property is used to write the time stepping as the composition of a time t map with a one-step map over Δt, the result of this composition is immediately projected into a finite dimensional function space by interpolation. In (2.12), each submap is a function in $\mathcal{V}$, the composition is not projected, rather it is kept in this expanded form to increase the numerical resolution. We will study this in more details in chapter 4.

The numerical schemes for the methods presented in this thesis all follow the above framework, the specific schemes used for defining the velocity and the one-step map will be dependent on the needs of each problem. In all cases, the interpolation operator $\mathcal{H}$ plays a central role, we examine this in detail in the next section.

2.2 Hermite Interpolation

In this section, we summarize the Hermite cubic interpolation scheme in d-dimensional space. For simplicity, we assume that the domain is the d-dimensional torus, $U = \mathbb{T}^d$ which is discretized using a Cartesian meshgrid denoted $\boldsymbol{G}$. We label the grid points of $\boldsymbol{G}$ by $\boldsymbol{x_i}$, with each $\boldsymbol{x_i}$ located at the lower corner of the cell $C_{\boldsymbol{i}}$; $\boldsymbol{i}$ stands for the index vector $\boldsymbol{i} = (i_1, i_2, \ldots, i_d)$. Given a grid $\boldsymbol{G}$, we define the Hermite cubic interpolation operator using piecewise smooth basis functions which are d-cubic in each cell $C_{\boldsymbol{i}}$. These basis functions are constructed by tensor product of the 1D Hermite cubic basis functions Q_j, with Q_0 interpolating function value, and Q_1, the derivative:

$$q_0(s) = (1 + 2|s|)(1 - |s|)^2, \tag{2.13a}$$

$$q_1(s) = s(1 - |s|)^2, \tag{2.13b}$$

$$Q_j(s) = q_j(s)\mathbf{1}_{[-1,1]}(s), \tag{2.13c}$$

where $\mathbf{1}$ denotes the indicator function. The Q_0 and Q_1 functions, shown in figure 2–2, form the shape functions corresponding to the function value and derivative interpolation on a 1D grid. They have the property that $\partial^a Q_b(s) = \delta^a_b \delta^0_s$ for $a, b \in \{0, 1\}$ and $s \in \{-1, 0, 1\}$. The d-dimensional shape functions on the grid $\boldsymbol{G}$ are then defined by the tensor product

$$H^{\boldsymbol{i}}_{\boldsymbol{a}}(\boldsymbol{x}) = \prod_{m=1}^{d} Q_{a_m}\left(\frac{x^m - x^m_{\boldsymbol{i}}}{\Delta x^m}\right)(\Delta x^m)^{a_m} \tag{2.14}$$

where $\boldsymbol{x} = (x^1, x^2, \ldots, x^d)$, $\boldsymbol{i} = (i_1, i_2, \ldots, i_d)$ and $\boldsymbol{a} = (a_1, a_2, \ldots, a_d)$.

On grid points $\boldsymbol{x_r}$ of $\boldsymbol{G}$, these basis functions satisfy

$$\partial^{\boldsymbol{b}} H^{\boldsymbol{i}}_{\boldsymbol{a}}(\boldsymbol{x_r}) = \delta^{\boldsymbol{b}}_{\boldsymbol{a}} \delta^{\boldsymbol{i}}_{\boldsymbol{r}}. \tag{2.15}$$

The $H^{\boldsymbol{i}}_{\boldsymbol{a}}$ functions are supported on the 2^d cells surrounding the grid point $\boldsymbol{x_i}$, they are d-cubic in each cell and are globally $\mathcal{C}^1$ with continuous mixed derivatives of the form $\partial^{\boldsymbol{b}}$ with $b_m \leq 1$ for $m = 1, 2, \ldots, d$. We define the Hermite cubic

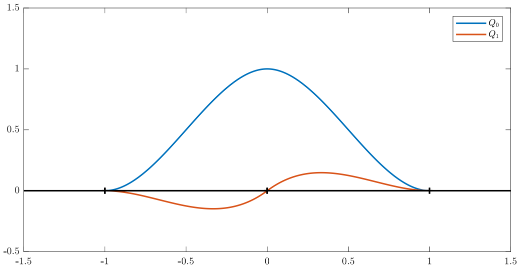


Figure 2–2: 1D Hermite cubic basis functions.

interpolation operator for a smooth function f to be

$$\mathcal{H}_{\boldsymbol{G}}[f](\boldsymbol{x}) = \sum_{\boldsymbol{a}\in\{0,1\}^d} \sum_{\boldsymbol{s}\in\boldsymbol{G}} \partial^{\boldsymbol{a}} f(\boldsymbol{x}_{\boldsymbol{i}}) H_{\boldsymbol{a}}^{\boldsymbol{i}}(\boldsymbol{x}). \tag{2.16}$$

The minimum regularity requirement for this to be well-defined is that $\partial^{\boldsymbol{b}} f$ is continuous for all mixed partials involving at most one derivative in each Cartesian coordinate. We denote $\mathcal{K}^1(U) = \left\{f \in \mathcal{C}^1(U) \mid \partial^{\boldsymbol{b}} f \in \mathcal{C}^0(U) \text{ for } \boldsymbol{b} \in \{0,1\}^d\right\}$, then $\mathcal{H}_{\boldsymbol{G}} : \mathcal{K}^1(U) \to \mathcal{K}^1(U)$ is a projection operator. Furthermore, for $f \in \mathcal{C}^4(U)$, it is known that

$$\|\partial^{\boldsymbol{a}} f - \partial^{\boldsymbol{a}} \mathcal{H}_{\boldsymbol{G}}[f]\|_\infty \lesssim \Delta x^{4-|\boldsymbol{a}|} \|f\|_{\mathcal{C}^4}, \tag{2.17}$$

where $|\boldsymbol{a}| = a_1 + a_2 + \cdots + a_d$.

Remark 2.2.1. The Hermite cubic interpolation is a special case of the degree $2m+1$ piecewise Hermite interpolants. Higher order interpolants have a similar construction to the Hermite cubic case and the interpolation error is given by

$$\|\partial^{\boldsymbol{a}} f - \partial^{\boldsymbol{a}} \mathcal{H}_{\boldsymbol{G}}[f]\|_\infty \lesssim \Delta x^{2m+2-|\boldsymbol{a}|} \|f\|_{\mathcal{C}^{2m+2}}, \tag{2.18}$$

For the characteristic maps, $\boldsymbol{\mathcal{X}}$ is given by d coordinate functions. We define the discretization subspace $\mathcal{V} = \left(\mathcal{K}^1(U)\right)^d$ and use coordinate-wise Hermite cubic interpolation as interpolation operator for the evolution of the characteristic maps. Here we note that $\mathcal{V}$ is not necessarily contained in *SDiff*(U) or even *Diff*(U) as there is no *a priori* constraints on the determinant or invertibility of the interpolated map. However, when interpolating a volume-preserving diffeomorphism, the error on the Jacobian determinant is $\mathcal{O}(\Delta x^3)$ and therefore, for sufficiently well-behaved maps, the interpolant will be a diffeomorphism. For longer time simulations, the characteristic map will develop strong small scale features which cannot be resolved using a fixed grid, in those cases, a remapping method will be employed to decompose the transformation; we will examine this in later chapters.

The time-stepping scheme for the characteristic map given in equation (2.9) relies on the interpolation of a composition of one-step map $\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}$ with the $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ map which is defined as a Hermite interpolant. The following lemma from [121] for the general order $2m+1$ Hermite interpolation allows us to estimate the local truncation error of this scheme.

Lemma 2.2.2. *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,*

$$\left\|\partial^{\vec{\alpha}}\left(\mathcal{H}_{\boldsymbol{G}}[f\circ\boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f]\circ\boldsymbol{T}\right)\right\|_\infty = \mathcal{O}(\epsilon\Delta x^{2m+1-|\vec{\alpha}|}) \tag{2.19}$$

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 $\boldsymbol{x}$:

$$f(\boldsymbol{x}+\epsilon\boldsymbol{v}(\boldsymbol{x})) = f(\boldsymbol{x}) + \sum_{i\in\{1,2\}}\epsilon v^i(\boldsymbol{x})\partial_i f(\boldsymbol{x}) + \sum_{i,j\in\{1,2\}} v^i(\boldsymbol{x})v^j(\boldsymbol{x})\frac{\epsilon^2}{2}\partial_{i,j}f(\boldsymbol{x}) + H.O.T.. \tag{2.20}$$

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

$$\mathcal{H}_{\boldsymbol{G}}[f\circ\boldsymbol{T}] = \mathcal{H}_{\boldsymbol{G}}[f] + \sum_{i\in\{1,2\}}\epsilon\mathcal{H}_{\boldsymbol{G}}[v^i\partial_i f] + \frac{\epsilon^2}{2}\mathcal{H}_{\boldsymbol{G}}[v^iv^j\partial_{i,j}f] + H.O.T., \tag{2.21a}$$

$$\mathcal{H}_{\boldsymbol{G}}[f]\circ\boldsymbol{T} = \mathcal{H}_{\boldsymbol{G}}[f] + \sum_{i\in\{1,2\}}\epsilon v^i\partial_i\mathcal{H}_{\boldsymbol{G}}[f] + \frac{\epsilon^2}{2}v^iv^j\partial_{i,j}\mathcal{H}_{\boldsymbol{G}}[f] + H.O.T.. \tag{2.21b}$$

We omit the interpolation on (2.21a) for all order ϵ and higher terms as replacing the interpolant with the interpolated function contributes a $\mathcal{O}(\epsilon^{|\vec{\alpha}|}\Delta x^{2m+2})$, $|\vec{\alpha}| \geq 1$ term which we absorb in the higher order terms. Similarly, we can use a Taylor expansion of $\mathcal{H}_G[f]$ in (2.21b) since the error incurred from extending an order $|\vec{\alpha}|$ derivative of the Hermite interpolant outside a cell is of order $\Delta x^{2m+2-|\vec{\alpha}|}$ and hence can also be absorbed in the higher order terms. Taking the difference of the two equations in (2.21), we get

$$\mathcal{H}_{\boldsymbol{G}}[f \circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f] \circ \boldsymbol{T} = \epsilon \partial_i (f - \mathcal{H}_{\boldsymbol{G}}[f]) v^i + \frac{\epsilon^2}{2} \partial_{i,j} (f - \mathcal{H}_{\boldsymbol{G}}[f]) v^i v^j + H.O.T.. \tag{2.22}$$

In light of (2.18), we have that

$$\mathcal{H}_{\boldsymbol{G}}[f \circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f] \circ \boldsymbol{T} = \mathcal{O}(\epsilon \Delta x^{2m+1}). \tag{2.23}$$

and therefore

$$\left\| \partial^{\vec{\alpha}} \left(\mathcal{H}_{\boldsymbol{G}}[f \circ \boldsymbol{T}] - \mathcal{H}_{\boldsymbol{G}}[f] \circ \boldsymbol{T} \right) \right\|_{\infty} = \mathcal{O}(\epsilon \Delta x^{2m+1-|\vec{\alpha}|}) \tag{2.24}$$

□

2.3 Tools from Differential Calculus

The CM method takes a geometric approach to the advection problem, the central heuristic of the approach is to express conservation laws in evolution equations as a Lie-advection under some velocity field. For this purpose, the language of differential forms provides a powerful tool to simplify the formulation as well as to make the intuition more apparent. We present in this section a short summary and formal exposition of the differential calculus tools used in this thesis. Detailed study of these notions can be found in standard textbooks (e.g. Lang [78]).

Differentiable Manifold: A differentiable manifold is a topological manifold M endowed with an atlas of charts (U_i, ϕ_i) where $i \in I$, $U_i \subset M$ open, form an open cover of M. The charts $\phi_i : U_i \to \mathbb{R}^d$ are homeomorphisms mapping local patches of M to Euclidean space. A manifold is smooth if the transition functions $\phi_i \circ \phi_j^{-1} : \phi_j(U_i \cap U_j) \subset \mathbb{R}^d \to \phi_i(U_i \cap U_j) \subset \mathbb{R}^d$ are smooth diffeomorphisms of the corresponding subsets in $\mathbb{R}^d$. In this thesis, the manifolds studied are either the flat d-torus or subsets of Euclidean space, we will use an atlas comprising of a single global chart (U, ϕ) with ϕ being the identity map (or inclusion map).

Tangent and Cotangent Spaces: For a point $\boldsymbol{p} \in M$, the tangent space $T_{\boldsymbol{p}}M$ at $\boldsymbol{p}$ is a vector space of tangent vectors. A tangent vectors at $\boldsymbol{p}$ is defined as an equivalence class of differentiable curves passing through $\boldsymbol{p}$; two curves are equivalent if for all smooth functions $f : M \to \mathbb{R}$, the 1^{st} derivative of f along the curve are the same at $\boldsymbol{p}$. In this thesis, the tangent spaces are identified with $\mathbb{R}^d$. The dual space of $T_{\boldsymbol{p}}M$, the space of linear functionals on the tangent space is the cotangent space $T^*_{\boldsymbol{p}}M$. A vector field assigns a vector to each point on the manifold, $\boldsymbol{u} : M \to TM$, $\boldsymbol{u} : \boldsymbol{p} \mapsto \boldsymbol{u}(\boldsymbol{p}) \in T_{\boldsymbol{p}}M$ is a section of the tangent bundle $TM = \cup_{\boldsymbol{p} \in M}\{\boldsymbol{p}\} \times T_{\boldsymbol{p}}M$. Similarly, a 1-form assigns a covector to each point, $\eta : M \to T^*M$ is a section of the cotangent bundle. For an oriented curve Γ with parametrization $\boldsymbol{\gamma} : (0,1) \to \Gamma \subset M$, a 1-form η can be integrated against Γ:

$$\int_\Gamma \eta = \int_0^1 \eta(\dot{\boldsymbol{\gamma}}) dt, \tag{2.25}$$

which heuristically yields a "measurement" of the 1-dimensional oriented submanifold Γ by η as this definition is parametrization-independent. The pairing $\eta(\dot{\boldsymbol{\gamma}})$ is also denoted $\iota_{\dot{\boldsymbol{\gamma}}}\eta$ where ι is the interior product.

Pushforward and Pullback: Given two manifolds M and N and a smooth map $F : M \to N$, the pushforward F_* at $\boldsymbol{p} \in M$ is a linear map $F_* : T_{\boldsymbol{p}}M \to T_{F(\boldsymbol{p})}N$. In coordinates this corresponds to left-multiplication by the Jacobian matrix of F. The pullback $F^* : T^*_{F(\boldsymbol{p})}N \to T^*_{\boldsymbol{p}}M$ is the dual map of F_* given by

$$(F^*\eta)(\boldsymbol{u}) = \eta(F_*\boldsymbol{u}). \tag{2.26}$$

It follows that for a curve Γ,

$$\int_{\Gamma} F^* \eta = \int_{F(\Gamma)} \eta. \tag{2.27}$$

***k*-forms:** Point-wise, a 1-form is a linear functional on 1-vectors, similarly, a k-form is a linear functional on k-vectors. One can see k-forms as the alternating parts of general k-tensors in the sense that for a k-vector $(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k)$, the k-form ω satisfies

$$\omega(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k) = \frac{1}{k!} \sum_{\sigma \in S_k} \operatorname{sgn}(\sigma) \omega(\boldsymbol{u}_{\sigma(1)}, \boldsymbol{u}_{\sigma(2)}, \ldots, \boldsymbol{u}_{\sigma(k)}). \tag{2.28}$$

Let K be the oriented k-dimensional subspace of $\mathbb{R}^d$ spanned by $\{\boldsymbol{u}_i\}_{i=1}^k$ and K_σ the oriented subspace spanned by $\{\boldsymbol{u}_{\sigma(i)}\}_{i=1}^k$, we note that $\operatorname{sgn}(\sigma)$ is the determinant of the permutation matrix mapping K to K_σ, i.e. that the above alternating sum removes all dependence on the ordering of the k vectors except the orientation of the spanned subspace. This allows us to view k-forms as linear functionals on oriented k-dimensional parallelograms $\mathbb{R}^d$. Higher degree forms can be obtained from combinations of lower degree forms, the operation which preserves the orientation property above is the wedge product. The wedge product is bilinear and for given η a k-form, and ω an l-form, the product $\eta \wedge \omega$ is a $(k+l)$-form and satisfies $\eta \wedge \omega = (-1)^{kl} \omega \wedge \eta$. Generally, for arbitrary k and l-forms η and ω, we have

$$\begin{gathered}(\eta \wedge \omega)(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{k+l}) \\ = \sum_{\sigma \in S_{k,k+l}} \operatorname{sgn}(\sigma) \eta(\boldsymbol{u}_{\sigma(1)}, \boldsymbol{u}_{\sigma(2)}, \ldots, \boldsymbol{u}_{\sigma(k)}) \omega(\boldsymbol{u}_{\sigma(k+1)}, \boldsymbol{u}_{\sigma(k+2)}, \ldots, \boldsymbol{u}_{\sigma(k+l)}).\end{gathered} \tag{2.29}$$

The pullback integration also generalizes to k-forms, given a k-dimensional oriented submanifold S, we have

$$\int_{S} F^* \eta = \int_{F(S)} \eta. \tag{2.30}$$

One can similarly deduce that pullback distributes through the wedge product: $F^*(\eta \wedge \omega) = F^*\eta \wedge F^*\omega$.

Exterior derivative: The exterior derivative d maps a k-form to a $(k+1)$-form. It satisfies the following axioms: For f a 0-form (scalar function), $\boldsymbol{u}$ a 1-vector, $df(\boldsymbol{u}) = \boldsymbol{u}(f)$. For arbitrary forms η and ω, $d(d\eta) = 0$ and $d(\eta \wedge \omega) = d\eta \wedge \omega + (-1)^k \eta \wedge d\omega$ where k is the degree of η. A form η is called *exact* if there exists a ω such that $\eta = d\omega$; η is called *closed* is $d\eta = 0$. The key result we will use on the exterior derivatives is the generalized Stokes' theorem:

$$\int_S d\omega = \int_{\partial S} \omega. \tag{2.31}$$

This also gives a geometric interpretation for the commutativity of pullbacks and d: $F^* d\eta = dF^*\eta$ since $F(\partial S) = \partial F(S)$.

Lie derivative of k-forms: The Lie derivative allows us to compare quantities at different locations on the manifold without a metric. Given a vector field $\boldsymbol{u}$ on M, for a short time t, $\boldsymbol{u}$ generates a flow map Φ_t on M, $\Phi_t : M \to M$ such that $\frac{d}{dt}\Phi_t(\boldsymbol{p}) = \boldsymbol{u}(\boldsymbol{p})$. The Lie derivative of a k-form ω under $\boldsymbol{u}$ (or the flow map Φ_t) is given by

$$\mathcal{L}_{\boldsymbol{u}}\omega = \frac{d}{dt}\Phi_t^*\omega. \tag{2.32}$$

The Lie derivative is linear in $\boldsymbol{u}$ and ω with $-\mathcal{L}_{\boldsymbol{u}}\omega = \mathcal{L}_{-\boldsymbol{u}}\omega = \frac{d}{dt}\Phi_{-t}^*\omega$. From the definition by pullback, we also have that Lie derivatives commute with exterior derivatives. If we consider a time-dependent k-form given by $\omega_t = \Phi_{-t}^*\omega_0$ where Φ_{-t} is the $-t$ flow of $\boldsymbol{u}$, then we have $\partial_t\omega + \mathcal{L}_{\boldsymbol{u}}\omega = 0$; this gives us that the pullback by the backward-in-time flow map is the solution operator for Lie advection. This is a main idea of the characteristic mapping method, with the exception that $\boldsymbol{u}$ will also be time-dependent.

Cartan's Homotopy Formula: The Cartan formula allows for the computation of a Lie derivative in terms of exterior derivatives and interior products:

$$\mathcal{L}_{\boldsymbol{u}}\omega = \iota_{\boldsymbol{u}} d\omega + d\iota_{\boldsymbol{u}}\omega, \tag{2.33}$$

where the interior product $\iota_{\boldsymbol{u}}$ maps k forms to $(k-1)$ forms by inserting $\boldsymbol{u}$ as first vector of the k-vector the k-form will act on.

We sketch the idea of this formula from a geometric point of view (see Arnold [5], section 36) as it shows some of the heuristics of the CM approach. Let ω be a k-form, let S be a k-dimensional oriented submanifold in M. We consider the product manifold $N = M \times \mathbb{R}$ and a $(k+1)$-dimensional submanifold $S_{[0,\epsilon]}$ obtained from the homotopy generated by Φ_t, the forward flow of $\boldsymbol{u}$, that is, $S_{[0,\epsilon]} = \cup_{t\in[0,\epsilon]}\Phi_t(S) \times \{t\}$. We can decomposed the boundary of $S_{[0,\epsilon]}$ into the following parts:

$$\partial S_{[0,\epsilon]} = (\Phi_\epsilon(S) \times \{\epsilon\}) \cup (S \times \{0\}) \cup D,$$

where $D = \cup_{t\in[0,\epsilon]}\Phi_t(\partial S) \times \{t\}$, that is the top and bottom slices plus the "cylindrical" boundary D generated by the homotopy of ∂S under the flow Φ_t.

Let $\pi : N \to M$ be the projection map $\pi : (\boldsymbol{p}, t) \mapsto \boldsymbol{p}$. We have that $\pi^*\omega$ is a k-form on N obtained by extending ω in the t dimension (constant in t). We integrate the $(k+1)$-form $d\pi^*\omega$ on $S_{[0,t]}$ and use Stokes' theorem to get

$$\int_{S_{[0,\epsilon]}} d\pi^*\omega = \int_{\partial S_{[0,\epsilon]}} \pi^*\omega = \int_{\pi(\Phi_\epsilon(S)\times\{\epsilon\})} \omega - \int_{\pi(S\times\{0\})} \omega + \int_D \pi^*\omega. \tag{2.34}$$

The t-dimension is the last dimension and is therefore the outer integral, we change order of integration to write the above as an operator on S, changing the order of integration corresponds to a permutation of the dimensions so the sign of the permutation is multiplied.

$$\int_{S_{[0,\epsilon]}} d\pi^*\omega = (-1)^k \int_S \int_0^\epsilon \iota_{\boldsymbol{u}} d\omega = (-1)^k \int_S (\Phi_\epsilon^*\omega - \omega) + (-1)^{k-1} \int_{\partial S} \int_0^\epsilon \iota_{\boldsymbol{u}}\omega$$

$$\Rightarrow \int_S (\iota_{\boldsymbol{u}} d\omega + d\iota_{\boldsymbol{u}}\omega) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_S (\Phi_\epsilon^*\omega - \omega) = \int_S \mathcal{L}_{\boldsymbol{u}}\omega, \tag{2.35}$$

where the inner integral is parametrized using $t \mapsto (t, \Phi_t(\boldsymbol{p}))$, and we used that the tangent vector of this parametrization satisfies $\pi_*(\frac{\partial}{\partial t} + \partial_t \Phi_t) = \boldsymbol{u}$ to express the integral using the interior product $\iota_{\boldsymbol{u}}$.

In regard to the CM method, one key fact we notice is that the term $d\iota_{\boldsymbol{u}}\omega$ comes from the integral over $\pi(D)$ which is the region "swept" by the boundary ∂S under the flow of $\boldsymbol{u}$. The integral over this region corresponds to the "net flux" of ω though the moving boundary (the boundary term in the Reynolds transport theorem). This suggest that if ω is closed, then it is Lie advected by $\boldsymbol{u}$ if $\boldsymbol{u}$ is a "flux-cancelling" velocity field in the sense that $d\iota_{\boldsymbol{u}}\omega = 0$ (or $d\iota_{\boldsymbol{u}}\omega = -\partial_t\omega$ in time-dependent equations). This is in fact the initial motivation for the method presented in chapter 3 where the volume form ρ evolving under the heat equation is Lie-advected by the "flux-cancelling" or "flux-matching" velocity $\boldsymbol{u} = -\nabla \log(\rho)$.

CHAPTER 3
Diffusion-driven Particle Management

The CM method for linear advection can be used for the evolution of implicitly defined level-set surfaces through the advection of the level-set function, this was studied in [84]. In this chapter, we will instead consider an explicit parametrized representation of evolving surfaces.

The parametrization of a curve or surface has many applications in computer graphics, computational geometry and geometric modelling. In scientific computing, where physics modelling 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 such as the Cahn-Hilliard equations [47, 48], 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 focus 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 studied in [53, 54]. 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 [70] 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 [16,58]. Methods for computing equiareal parametrizations include [7, 123], which propose a relaxation algorithm based on some stretch factor computed from mesh point distances. Winslow's rezoning algorithm [119] 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 evolved a numerical mesh in order to better resolve the different scales that a solution to an evolution equation may exhibit [64]. For instance, in [15], a moving-mesh PDE was employed as a redistribution map to reinitialize sample point positions in order to resolve the multiscale solutions generated from convection dominated flows.

In this chapter, we study the application of the CM framework to evolving an advected surface while preserving the equiareal property of the parametrization. The approach comprises mainly of two parts: First, we will compute the forward characteristic map of the velocity field in the ambient space. This gives a Lagrangian description of the moving interface as points on the initial surface can be mapped to their time t location using the map. From a parametric point of view, the parametrization function of the time t surface is simply given by the composition of the initial parametrization with the forward map. Secondly, we will compute a deformation map of the reference space of the parametrization in order to redistribute the sample points. In order to achieve this, we define a probability density function on the parametric space, this density is a scalar multiple of the area element associated to the surface parametrization and will be interpreted as the initial condition of a diffusion equation whose solution will tend to the uniform distribution. Our approach is then to compute the backward characteristic map associated to this diffusion equation, written in conservation form; the resulting redistribution map is then a transport map moving the initial density, i.e. the surface area-element, to a uniform density. This redistribution map is pre-composed 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 but improves 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 [77, 88]. Further extensive theoretical and numerical analysis for the blob-particle method for linear and non-linear diffusion can also be found in [24]. 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 analysed in [71]. There has been extensive research in the field of optimal transport specifically concerning gradient descent and geodesic flows [14, 14, 23, 25, 96]. Notably, the fluid mechanics interpretation of the Monge-Kantorovich problem in [12,13] 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 [103] and more complete surveys in [1, 102, 116]. In the context of generating equiareal parametrization, the optimal transport methods have also been investigated, for instance in [113,114,127].

The rest of this chapter is organized as follows. Section 3.1 presents the formulation of the CM method for the transport of densities. Section 3.2 contains an overview of the numerical implementation of the method along with some error estimates. We present in section 3.3 the application of this method for parametric surface advection, and numerical results are presented in section 3.4. Finally, section 3.5 contains some concluding remarks.

3.1 Mathematical Formulation

3.1.1 Characteristic Mapping Method for Density Transport

We examine in this section the application of the CM method to the transport of densities, that is, using the characteristic map as the solution operator for the Lie transport of a volume-form. In this case, the assumption that the transport velocity $\boldsymbol{u}$ is divergence-free is dropped since in the divergence-free case, the transport of volume-forms will coincide with scalar advection.

For smooth, divergence-free velocity fields, the backward characteristic maps given by (2.6b) are known to be diffeomorphisms for all times. 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:

$$\begin{aligned}\frac{d}{dt}\det\left(\nabla \boldsymbol{X}_B(\boldsymbol{\gamma}(t),t)\right) &= \operatorname{tr}\left(\operatorname{adj}(\nabla \boldsymbol{X}_B)\frac{d}{dt}\nabla \boldsymbol{X}_B(\boldsymbol{\gamma}(t),t)\right) \\ &= \operatorname{tr}\left(-\operatorname{adj}(\nabla \boldsymbol{X}_B)\cdot\nabla \boldsymbol{X}_B\cdot\nabla \boldsymbol{u}(\boldsymbol{\gamma}(t),t)\right) = -\det(\nabla \boldsymbol{X}_B)\nabla\cdot\boldsymbol{u},\end{aligned} \tag{3.1}$$

where adj denotes the adjugate matrix. That is

$$\frac{d}{dt}\log\det(\nabla \boldsymbol{X}_B) = (\partial_t + \boldsymbol{u}\cdot\nabla)\log\det(\nabla \boldsymbol{X}_B) = -\nabla\cdot\boldsymbol{u}, \tag{3.2}$$

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

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \forall (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{3.3a}$$

$$\rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}). \tag{3.3b}$$

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 $\rho(\boldsymbol{x}, t) \geq a \ \forall (\boldsymbol{x}, t) \in U \times \mathbb{R}_+$. We also assume that ρ_0 is a probability density, in particular, (3.3) 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:

$$\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{X}_B(\boldsymbol{x}, t)) \det \nabla \boldsymbol{X}_B. \tag{3.4}$$

The submap decomposition (2.11) can also be applied to this pullback.

3.1.2 Density Redistribution

In this section we consider 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 $\rho_0 \mathrm{d}m$. 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 chapter, the subscript t following a variable will only denote the evaluation at time t. From (3.3), we have that μ_t has density $\mathrm{d}\mu_t = \rho_t \mathrm{d}m$.

The characteristic map is generated from a chosen velocity field $\boldsymbol{u}$. For instance, one can use the Moser flow [34] given by

$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{\nabla\Delta^{-1}(\rho_0 - \rho_1)}{(1-t)\rho_0 + t\rho_1}, \tag{3.5}$$

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 here, $\boldsymbol{X}_B$ will be a coupling between the target density ρ_0 and the uniform density. As a result, pullback by $\boldsymbol{X}_B$ will "equidistribute" ρ_0. For parametrized curves and surfaces, ρ_0 will correspond to the arclength or area-element functions defined on U. Pre-composition of the parametrization with $\boldsymbol{X}_B$ will then yield an equiareal parametrization. We will examine this in further details in section 3.3.

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

$$\partial_t\rho + \nu\nabla\cdot(-\nabla\rho) = \partial_t\rho + \nabla\cdot(\rho\cdot -\nu\nabla\log\rho) = 0 \qquad \forall(\boldsymbol{x},t)\in U\times\mathbb{R}_+, \tag{3.6a}$$

$$\partial_{\boldsymbol{n}}\rho = 0 \qquad \forall(\boldsymbol{x},t)\in\partial U\times\mathbb{R}_+, \tag{3.6b}$$

$$\rho(\boldsymbol{x},0) = \rho_0(\boldsymbol{x}). \tag{3.6c}$$

The redistribution map is then obtained from:

$$\rho(\boldsymbol{x},t) = \rho_0(\boldsymbol{X}_B(\boldsymbol{x},t))\det\nabla\boldsymbol{X}_B(\boldsymbol{x},t), \tag{3.7a}$$

$$\boldsymbol{u}(\boldsymbol{x},t) = -\nu\nabla\log\rho(\boldsymbol{x},t), \tag{3.7b}$$

$$\partial_t\boldsymbol{X}_B + (\boldsymbol{u}\cdot\nabla)\boldsymbol{X}_B = 0. \tag{3.7c}$$

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. Lastly 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 [71]. In this sense, the time-evolution of the map can be seen as an iterative process which contracts to the desired transport map.

3.1.3 Energy Estimates

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

$$E(t) = \frac{1}{2}\int_U (\rho(\boldsymbol{x},t)-1)^2 \mathrm{d}x \quad \text{with} \quad \frac{d}{dt}E(t) = -\int_U \nu \left|\nabla\rho\right|^2 \mathrm{d}x. \tag{3.8}$$

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

$$\frac{d}{dt}E(t) \leq -\alpha\nu E(t), \tag{3.9}$$

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 3.1.1. *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

$$\nu_t(A) = m(\boldsymbol{X}_{[t,0]}^{-1}(A)) \quad \text{and} \quad \mu(A) = \mu_t(\boldsymbol{X}_{[t,0]}^{-1}(A)), \tag{3.10}$$

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

$$|\nu_t(A) - \mu(A)| = \left| m\left(\boldsymbol{X}_{[t,0]}^{-1}(A)\right) - \mu_t\left(\boldsymbol{X}_{[t,0]}^{-1}(A)\right)\right|. \tag{3.11}$$

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

$$Y_t \xrightarrow{d} Z \iff Z_t \xrightarrow{d} Y, \tag{3.12}$$

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 $\mathrm{d}\nu_t = \frac{\mathrm{d}\nu_t}{\mathrm{d}m}\mathrm{d}m$ and $\mathrm{d}\mu = \frac{\mathrm{d}\mu}{\mathrm{d}m}\mathrm{d}m$, with

$$\frac{\mathrm{d}\mu}{\mathrm{d}m} = \rho_0 = \rho_t(\boldsymbol{X}_{[t,0]}^{-1}) \det \nabla \boldsymbol{X}_{[t,0]}^{-1} \quad \text{and} \quad \frac{\mathrm{d}\nu_t}{\mathrm{d}m} = \det \nabla \boldsymbol{X}_{[t,0]}^{-1}. \tag{3.13}$$

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

$$\begin{aligned} \left\| \frac{\mathrm{d}\nu_t}{\mathrm{d}m} - \frac{\mathrm{d}\mu}{\mathrm{d}m} \right\|_{L^2} &= \left(\int_U \left| \rho_0 - \det \nabla \boldsymbol{X}_{[t,0]}^{-1} \right|^2 \mathrm{d}m \right)^{\frac{1}{2}} \\ = \left(\int_U \left| \rho_t \circ \boldsymbol{X}_{[t,0]}^{-1} - 1 \right|^2 (\det \nabla \boldsymbol{X}_{[t,0]}^{-1})^2 \mathrm{d}m \right)^{\frac{1}{2}} &= \left(\int_{\boldsymbol{X}_{[t,0]}^{-1}(U)} |\rho_t - 1|^2 (\det \nabla \boldsymbol{X}_{[t,0]})^{-1} \mathrm{d}m \right)^{\frac{1}{2}} \\ &\leq \|\rho_t - 1\|_{L^2} \|(\det \nabla \boldsymbol{X}_{[t,0]})^{-1}\|_{L^\infty}^{\frac{1}{2}}. \end{aligned} \tag{3.14}$$

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

$$(\det \nabla \boldsymbol{X}_{[t,0]})^{-1} = \frac{\rho_0(\boldsymbol{X}_{[t,0]})}{\rho_t(\boldsymbol{x})} \leq \frac{\sup_U \rho_0}{\inf_U \rho_t} \leq \frac{\sup_U \rho_0}{\inf_U \rho_0}. \tag{3.15}$$

Using the decay rate (3.9), 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 $\sqrt{\frac{\sup \rho_0}{\inf \rho_t}} \cdot \|\rho_t - 1\|_{L^2}$. □

3.2 Numerical Implementation

The numerical scheme used for the diffusion-driven characteristic mapping method largely follows the framework outlined in section 2.1. We present in this section the general approach using Hermite interpolation of order $2m+1$, the schemes in this chapter will use $m = 0$ or 1, i.e. piecewise linear or cubic interpolation. We will also make use of a staggered grid discretization, in this case, it is important to note that the Hermite interpolation gains an order of accuracy on the 1^{st} derivative when evaluated at cell centers. Indeed, the leading order term in the error in the cell $[x_i, x_{i+1}]$ can be rewritten as $\left((x - \frac{1}{2}(x_i + x_{i+1}))^2 - \frac{1}{4}\Delta x^2\right)^{m+1}$. At $x = \frac{1}{2}(x_i + x_{i+1})$ this function has vanishing 1^{st} derivatives for all m. In particular, this means that evaluation of the gradient and first mixed derivatives of a Hermite cubic interpolant is order $\mathcal{O}(\Delta x^4)$ accurate and $\mathcal{O}(\Delta x^2)$ for linear interpolants.

3.2.1 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 staggered grids approach for the discretization of these two quantities in order to obtain a spatially compact scheme. Similar primal-dual grids approaches for solving hyperbolic and parabolic equations using Hermite interpolation have been explored in [2,3,56] 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 $\boldsymbol{D}$:

$$\rho^n(\boldsymbol{x}) = \rho_0(\boldsymbol{\mathcal{X}}^n_B(\boldsymbol{x})) \det \nabla \boldsymbol{\mathcal{X}}^n_B(\boldsymbol{x}). \tag{3.16}$$

Since the evaluation of $\det \nabla \boldsymbol{\mathcal{X}}_B^n(\boldsymbol{x})$ occurs at the cell centers of $\boldsymbol{G}$, we gain an order of accuracy on the gradient. Therefore, $\rho_{\boldsymbol{D}}^n$ is accurate to order $\mathcal{O}(\Delta x^{2m+2})$ when exact map values are provided. This staggered grid approach is similar in spirit to the primal-dual grid method developed by Appelo et al. 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 $\rho_{\boldsymbol{D}}^n$ 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.

$$\tilde{\rho}_{\boldsymbol{D}} = (I - \Delta t \nu \Delta)^{-1} \rho_{\boldsymbol{D}}^n, \tag{3.17a}$$

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) = -\nu \nabla \mathcal{H}_{\boldsymbol{D}} \left[\log \tilde{\rho}_{\boldsymbol{D}} \right]. \tag{3.17b}$$

Indeed, the velocity field $-\nu \nabla \log \rho$ extracted from the heat equation is stiff in time and therefore the characteristic ODEs (2.2) 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 redistribution algorithm in the pseudocode Algorithm 1.

Using the velocity defined in (3.17), we have that the one-step map in has $\mathcal{O}(\Delta t^2 + \Delta t \Delta x^3)$ local truncation error for Hermite cubic interpolants and $\mathcal{O}(\Delta t^2 + \Delta t \Delta x^2)$ for linear. Indeed, the Hermite grid data for the linear interpolation have $\mathcal{O}(\Delta x^2)$ error due to being evaluated at cell centres. For cubic interpolation, the derivative data of the velocity field are only available to $\mathcal{O}(\Delta x^2)$, hence the $\mathcal{O}(\Delta x^3)$

Algorithm 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 $\Delta t$, 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 (3.16) on $\boldsymbol{D}$
        $\tilde{\rho}_{\boldsymbol{D}} \leftarrow (I - \Delta t \nu \Delta)^{-1} \rho^n_{\boldsymbol{D}}$
        Define $\boldsymbol{\mathcal{X}}_{[t+\Delta t, t]}(\boldsymbol{x}) := \boldsymbol{x} + \Delta t \nu \nabla \mathcal{H}_{\boldsymbol{D}} [\log \tilde{\rho}_{\boldsymbol{D}}] (\boldsymbol{x})$      ▷ using (3.17)
        $\boldsymbol{\mathcal{X}}_B \leftarrow \mathcal{H}_{\boldsymbol{G}} \left[\boldsymbol{\mathcal{X}}_B \circ \boldsymbol{\mathcal{X}}_{[t+\Delta t, t]}\right]$
        $t \leftarrow t + \Delta t$
    end while
    return $\boldsymbol{\mathcal{X}}_B$
end function
```

spatial error for the one-step map. Given the accuracy of the above one-step maps, we have from lemma 2.2.2 that the global truncation errors for $\rho_{\boldsymbol{D}}$ are $\mathcal{O}(\Delta t + \Delta x)$ for linear interpolants, and $\mathcal{O}(\Delta t + \Delta x^2)$ 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 $r = 0.25$ centred at $\boldsymbol{x}_c = (0.5, 0.5)$. The minimum density is set at 0.75.

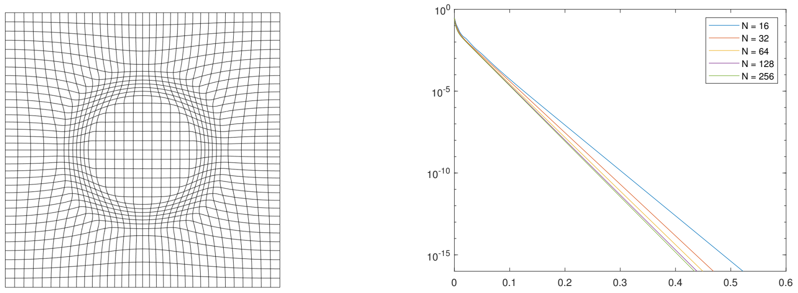
$$\rho_0(\boldsymbol{x}) = 1 + 0.25\eta(\boldsymbol{x}) \tag{3.18a}$$

$$\eta(\boldsymbol{x}) = c(\eta_0(|\boldsymbol{x} - \boldsymbol{x}_c| - r) - \bar{\eta}_0) \tag{3.18b}$$

$$\eta_0(s) = \exp(-(1 - (2s/w)^2)^{-1}) \tag{3.18c}$$

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 3.2 to time 1 with $\nu = 1$, using Hermite cubic interpolation on various grids of size N and tracking the decay of $\mathcal{E}$, Δt 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, the residual energy also exhibits the exponential decay from the diffusion equation.



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

Figure 3–1: Diffusion flow density redistribution using CM method.

3.2.2 Moving Initial Condition

The initial condition ρ_0 is the density to be redistributed towards 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 "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

$$\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{X}_B(\boldsymbol{x}, t), t) \det \nabla \boldsymbol{X}_B(\boldsymbol{x}, t), \tag{3.19a}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = -\nu \nabla \log \rho(\boldsymbol{x}, t). \tag{3.19b}$$

In the implementation, we simply replace line 4 of 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:

$$\partial_t \rho = \nu \Delta \rho + \partial_t \log \rho_0|_{\boldsymbol{X}_B} \rho. \tag{3.20}$$

From this, we can view the diffusion flow as a relaxation to a source term introduced by a moving initial condition. The 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.

3.3 Application to Surface Advection

In this section, we apply the density transport method described above to the advection of parametric curves and surfaces in 3-dimensional 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 3.2.2.

3.3.1 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 3D.

Let Ω be a 3-dimensional 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. We denote by $\boldsymbol{\Phi}_F$ and $\boldsymbol{\Phi}_B$ the forward and backward characteristic maps, they satisfy the following equations:

$$\partial_t \boldsymbol{\Phi}_F(\boldsymbol{p}, t) = \boldsymbol{v}(\boldsymbol{\Phi}_F(\boldsymbol{p}, t), t) \qquad \forall (\boldsymbol{p}, t) \in \Omega \times \mathbb{R}_+, \tag{3.21a}$$

$$(\partial_t + \boldsymbol{v} \cdot \nabla) \boldsymbol{\Phi}_B(\boldsymbol{p}, t) = 0 \qquad \forall (\boldsymbol{p}, t) \in \Omega \times \mathbb{R}_+, \tag{3.21b}$$

$$\boldsymbol{\Phi}_F(\boldsymbol{p}, 0) = \boldsymbol{\Phi}_B(\boldsymbol{p}, 0) = \boldsymbol{p} \qquad \forall \boldsymbol{p} \in \Omega. \tag{3.21c}$$

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

$$\boldsymbol{\Phi}_F(\gamma(0),t)=\gamma(t), \quad \boldsymbol{\Phi}_B(\gamma(t),t)=\gamma(0) \qquad \forall t\in\mathbb{R}_+, \tag{3.22}$$

for any characteristic curve γ satisfying

$$\partial_t\gamma(t)=\boldsymbol{v}(\gamma(t),t). \tag{3.23}$$

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

$$\boldsymbol{\Phi}_F(\boldsymbol{\Phi}_B(\boldsymbol{p},t),t)=\boldsymbol{\Phi}_B(\boldsymbol{\Phi}_F(\boldsymbol{p},t),t)=\boldsymbol{p} \qquad \forall t\geq 0. \tag{3.24}$$

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 $\boldsymbol{\varphi}$ is used for the numerical discretization of the $\boldsymbol{\Phi}$ maps. The time-stepping can be summarized as

$$\boldsymbol{\varphi}_{[0,t+\Delta t]}=\mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\varphi}_{[t,t+\Delta t]}\circ\boldsymbol{\varphi}_{[0,t]}\right], \tag{3.25a}$$

$$\boldsymbol{\varphi}_{[t+\Delta t,0]}=\mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\varphi}_{[t,0]}\circ\boldsymbol{\varphi}_{[t+\Delta t,t]}\right]. \tag{3.25b}$$

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

$$\boldsymbol{\Phi}_F\circ\boldsymbol{\Phi}_B=\boldsymbol{\Phi}_B\circ\boldsymbol{\Phi}_F=\boldsymbol{id}. \tag{3.26}$$

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 group 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 given in (2.11) to represent the global-time maps:

$$\boldsymbol{\varphi}_F = \boldsymbol{\varphi}_{[0,t]} = \boldsymbol{\varphi}_{[T_{n-1},t]} \circ \boldsymbol{\varphi}_{[T_{n-2},T_{n-1}]} \circ \cdots \circ \boldsymbol{\varphi}_{[0,T_1]}, \tag{3.27a}$$

$$\boldsymbol{\varphi}_B = \boldsymbol{\varphi}_{[t,0]} = \boldsymbol{\varphi}_{[T_1,0]} \circ \cdots \circ \boldsymbol{\varphi}_{[T_{n-1},T_{n-2}]} \circ \boldsymbol{\varphi}_{[t,T_{n-1}]}. \tag{3.27b}$$

The error on the inverse property (3.24) stems from the time-stepping and interpolation (3.25). This gives us an alternative *a posteriori* error estimate for choosing the remapping times T_i. Here we use the following composition error as remapping criterion.

$$\epsilon = \max\left(\|\boldsymbol{\varphi}_{[T_{i-1},t]} \circ \boldsymbol{\varphi}_{[t,T_{i-1}]} - \boldsymbol{id}\|_\infty, \|\boldsymbol{\varphi}_{[t,T_{i-1}]} \circ \boldsymbol{\varphi}_{[T_{i-1},t]} - \boldsymbol{id}\|_\infty\right). \tag{3.28}$$

We then define T_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:

$$S_t = \{\boldsymbol{p} \in \Omega \mid \boldsymbol{\Phi}_B(\boldsymbol{p}, t) \in S_0\}, \tag{3.29a}$$

$$\text{or} \quad S_t = \{\boldsymbol{\Phi}_F(\boldsymbol{p}, t) \mid \boldsymbol{p} \in S_0\}. \tag{3.29b}$$

Equation (3.29a) 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 [84], a review of recent advances in level-set methods can be found in [49].

Equation (3.29b) 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 non-orientable 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

$$\boldsymbol{P}_t = \boldsymbol{\Phi}_{[0,t]} \circ \boldsymbol{P}_0. \tag{3.30}$$

Numerically speaking this method evolves the parametrization function in time using the solution operator $\boldsymbol{\varphi}_F$. 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 [84], 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 $\boldsymbol{\varphi}_F$. In this case, the numerical accuracy of the diffeomorphism $\boldsymbol{\varphi}_F$ 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.

3.3.2 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 $I_t = \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t$.

$$A_t = \sqrt{\det I_t} = \sqrt{\det \nabla \boldsymbol{P}_0^T \nabla \boldsymbol{\Phi}_F^T \nabla \boldsymbol{\Phi}_F \nabla \boldsymbol{P}_0}. \tag{3.31}$$

We can compute the time-evolution of A_t. Using

$$\partial_t I_t = \nabla \boldsymbol{P}_t^T (\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v}) \nabla \boldsymbol{P}_t, \tag{3.32}$$

and applying Jacobi's rule for derivatives of matrix determinants. We have

$$\partial_t A_t = \frac{1}{2} A_t \mathrm{tr}\left(I^{-1}\partial_t I\right) = \frac{1}{2} A_t \mathrm{tr}\left((\nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t)^{-1} \nabla \boldsymbol{P}_t^T (\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v}) \nabla \boldsymbol{P}_t\right). \tag{3.33}$$

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

$$\partial_t A_t = A_t \mathrm{tr}\left(\nabla \boldsymbol{P}_t (\nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t)^{-1} \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{v}\right) = A_t \mathrm{tr}\left(\Pi_{S_t} \nabla \boldsymbol{v}\right) = A_t \, \mathrm{div}|_{S_t} \boldsymbol{v}, \tag{3.34}$$

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 3.1.2 to obtain a mapping $\boldsymbol{X}_B : U \to U$ which pushes the uniform density to the density A_t. The modified parametrization

$$\boldsymbol{Q}_t = \boldsymbol{P}_t \circ \boldsymbol{X}_{[t,0]} : U \to S_t \subset \Omega \tag{3.35}$$

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 3.2.2 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:

$$\rho_0(\boldsymbol{x},t) = \frac{1}{\rho_A}\sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t}, \tag{3.36a}$$

$$\rho(\boldsymbol{x},t) = \frac{1}{\rho_A}\sqrt{\det \nabla \boldsymbol{Q}_t^T \nabla \boldsymbol{Q}_t}, \tag{3.36b}$$

$$\text{with} \quad \rho_A(t) = \int_U \sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t} \mathrm{d}\boldsymbol{x} = |S_t|. \tag{3.36c}$$

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

$$\rho_A(t) = \int_{\boldsymbol{X}_B^{-1}(U)} \sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t}|_{\boldsymbol{X}_B} \det \nabla \boldsymbol{X}_B \mathrm{d}\boldsymbol{x} = \int_U \sqrt{\det \nabla \boldsymbol{Q}_t^T \nabla \boldsymbol{Q}_t} \mathrm{d}\boldsymbol{x}. \tag{3.37}$$

We have the following equations for their time evolution:

$$\partial_t \rho_A(t) = \int_U \operatorname{div}|_{S_t} \boldsymbol{v} \sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t} \mathrm{d}\boldsymbol{x} = \rho_A \int_U \rho_0 \operatorname{div}|_{S_t} \boldsymbol{v} \mathrm{d}\boldsymbol{x}, \tag{3.38a}$$

$$\partial_t \rho_0(\boldsymbol{x},t) = \frac{1}{\rho_A} \operatorname{div}|_{S_t} \boldsymbol{v} \sqrt{\det \nabla \boldsymbol{P}_t^T \nabla \boldsymbol{P}_t} - \frac{\rho_0}{\rho_A} \partial_t \rho_A = \left(\operatorname{div}|_{S_t} \boldsymbol{v} - \partial_t \log \rho_A \right) \rho_0(\boldsymbol{x},t), \tag{3.38b}$$

$$\rho(\boldsymbol{x},t) = \rho_0\left(\boldsymbol{X}_B(\boldsymbol{x},t),t\right) \det \nabla \boldsymbol{X}_B. \tag{3.38c}$$

Using the results in section 3.1.2, 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

$$\partial_t \rho(\boldsymbol{x},t) = \nu \Delta \rho(\boldsymbol{x},t) + \partial_t \rho_0|_{\boldsymbol{X}_B} \det \nabla \boldsymbol{X}_B = \nu \Delta \rho(\boldsymbol{x},t) + \lambda_{\boldsymbol{v}}(\boldsymbol{x},t)\rho(\boldsymbol{x},t), \tag{3.39}$$

where $\lambda_{\boldsymbol{v}}(\boldsymbol{x},t) = \partial_t \log \rho_0|_{\boldsymbol{X}_B} = \left(\operatorname{div}|_{S_t} \boldsymbol{v}\right)_{(\boldsymbol{X}_B)} - \int_U \rho_0 \operatorname{div}|_{S_t} \boldsymbol{v} \mathrm{d}\boldsymbol{x}$. 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 2. The notation $\boldsymbol{P}_t$ implicitly assumes the definition and computation of the non-redistributed time-dependent parametrization function (3.30) using the numerical characteristic map $\boldsymbol{\varphi}_F$ of the ambient advection. The computation of $\boldsymbol{\varphi}_F$ is given in(3.25a) and (3.27). In comparison with the algorithm described in section 3.2.1, 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 2 CM method for equiareal time-dependent surface parametrization

```
Input: Parametrization $\boldsymbol{P}_t$, staggered grids $\boldsymbol{G}$ and $\boldsymbol{D}$, diffusivity $\nu$, time step $\Delta t$,
    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 (3.36)
 6:         $\boldsymbol{\mathcal{X}}_{[t+\Delta t,t]} \leftarrow$ HeatFlowMap($\rho^n$, $\nu$, $[0, \Delta t]$)        ▷ subroutine 1 on local-time problem
 7:         $\boldsymbol{\mathcal{X}}_{[t+\Delta t,0]} \leftarrow \mathcal{H}_{\boldsymbol{G}}\left[\boldsymbol{\mathcal{X}}_{[t,0]} \circ \boldsymbol{\mathcal{X}}_{[t+\Delta t,t]}\right]$
 8:         $t \leftarrow t + \Delta t$
 9:     end while
10:     return $\boldsymbol{Q}_T = \boldsymbol{P}_T \circ \boldsymbol{\mathcal{X}}_{[T,0]}$
11: end function
```

Remark 3.3.1. Line 6 of algorithm 2 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 $\boldsymbol{\mathcal{X}}_{[t+\Delta t,t]}$ would then be computed over several redistribution substeps.

Theorem 3.3.1. *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 (3.39). We have that the energy estimate in this case is

$$\begin{aligned}\frac{d}{dt}E(t) = \frac{d}{dt}\|\rho - 1\|_{L^2}^2 &= -\nu\|\nabla(\rho-1)\|_{L^2}^2 + \int_U (\rho-1)^2\lambda_{\boldsymbol{v}}\mathrm{d}\boldsymbol{x} + \int_U (\rho-1)\lambda_{\boldsymbol{v}}\mathrm{d}\boldsymbol{x} \\ &\leq -\nu\|\nabla(\rho-1)\|_{L^2}^2 + \|\lambda_{\boldsymbol{v}}\|_\infty\|\rho-1\|_{L^2}^2 - \int_U \lambda_{\boldsymbol{v}}\mathrm{d}\boldsymbol{x}, \end{aligned} \tag{3.40}$$

where $\int_U \rho\lambda_{\boldsymbol{v}}\mathrm{d}\boldsymbol{x}$ vanishes since $\int_U \partial_t\rho\mathrm{d}\boldsymbol{x} = 0$ by construction and $\int_U \Delta\rho\mathrm{d}\boldsymbol{x} = 0$ 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 $\int_U \lambda_{\boldsymbol{v}}\mathrm{d}\boldsymbol{x}$ on the current density:

$$\begin{aligned}-\int_U \lambda_{\boldsymbol{v}}\mathrm{d}\boldsymbol{x} &= -\int_U (\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\mathrm{d}\boldsymbol{x} + \int_{\boldsymbol{X}_B^{-1}(U)} \rho\,(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\mathrm{d}\boldsymbol{x} \\ &= \int_U (\rho-1)\,(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\mathrm{d}\boldsymbol{x} \leq \left\|(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\right\|_{L^2}\|\rho-1\|_{L^2}. \end{aligned} \tag{3.41}$$

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

$$\frac{d}{dt}\|\rho-1\|_{L^2} \leq 2(-\alpha\nu + \|\lambda_{\boldsymbol{v}}\|_\infty)\|\rho-1\|_{L^2} + 2\left\|(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\right\|_{L^2}, \tag{3.42}$$

meaning that the L^2 norm can be controlled by

$$\|\rho-1\|_{L^2} \leq \frac{\beta}{\alpha\nu - \|\lambda_{\boldsymbol{v}}\|_\infty} + c\exp\left(-2t\left(\alpha\nu - \|\lambda_{\boldsymbol{v}}\|_\infty\right)\right), \tag{3.43}$$

where $\beta = \max_t \left\|(\operatorname{div}|_{S_t}\boldsymbol{v})_{(\boldsymbol{X}_B)}\right\|_{L^2}$ 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 (3.39), 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 $\boldsymbol{\varphi}_{[T_{i-1},T_i]}$ 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 (3.27) and obtain the full reparametrization as the composition of all redistribution maps. Combining (3.27), (3.30) and (4.5), we have

$$\boldsymbol{Q}_t = \boldsymbol{\varphi}_{[T_{n-1},t]} \circ \boldsymbol{\varphi}_{[T_{n-2},T_{n-1}]} \circ \cdots \circ \boldsymbol{\varphi}_{[0,T_1]} \circ \boldsymbol{P}_0 \circ \boldsymbol{\mathcal{X}}_{[T_1,0]} \circ \cdot \circ \boldsymbol{\mathcal{X}}_{[T_{n-1},T_{n-2}]} \circ \boldsymbol{\mathcal{X}}_{[t,T_{n-1}]}. \tag{3.44}$$

Numerically, each submap is computed independently, sequentially using algorithm 2. To be consistent with the reinitialized problem, the input parametrization for the $i+1^{st}$ map is defined to be $\boldsymbol{\varphi}_{[T_i,t]} \circ \boldsymbol{Q}_{T_i}$.

Remark 3.3.2. For the submap computations, the parametrization of the surface S_t is given by $\boldsymbol{\varphi}_{[T_i,t]} \circ \boldsymbol{Q}_{T_i}$ where $\boldsymbol{Q}_{T_i}$ is given by (3.44). However, to save computational time, for the purpose of computing the density ρ, it is sufficient to replace $\boldsymbol{Q}_{T_i}$ by an interpolant on a fine enough grid similar to the approach in [84]. This is because the method maintains the sampling density of $\boldsymbol{Q}$ near uniform for all times and therefore $\boldsymbol{Q}_{T_i}$ can be accurately represented by interpolation.

Remark 3.3.3. 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, this method is unique in that its computational efficiency does not depend on the number of sample points. The algorithm computes a functionally defined parametrization, 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.

3.4 Numerical Results

3.4.1 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 3.2.2.

We define the following density ρ_0:

$$\rho_0(x, y, t) = 1 + 0.25 \sin^2(1.5\pi t)\, \eta\left(x + 0.25 \sin(0.5\pi t) \sin(2\pi y), y\right), \tag{3.45}$$

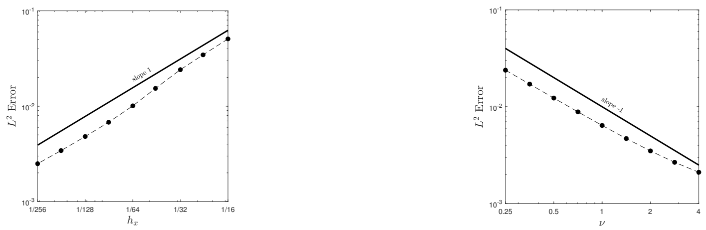
where η is defined in (3.18b).

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 3–2. As expected, the maximum L^2 norm of $\rho - 1$ is linear with respect to Δx, Δt and ν^{-1}.



(a) $\nu = 10$, $\Delta t = \Delta x/4$.

(b) $\Delta x = 1/768$, $\Delta t = 1/2048$.

Figure 3–2: L^2 error with respect to Δx, Δt and ν.

3.4.2 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 [79]:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \begin{pmatrix} 2\cos\left(\frac{\pi t}{P}\right)\left(\sin(\pi x)\right)^2 \sin(2\pi y)\sin(2\pi z) \\ -\cos\left(\frac{\pi t}{P}\right)\sin(2\pi x)\left(\sin(\pi y)\right)^2 \sin(2\pi z) \\ -\cos\left(\frac{\pi t}{P}\right)\sin(2\pi x)\sin(2\pi y)\left(\sin(\pi z)\right)^2 \end{pmatrix} \tag{3.46}$$

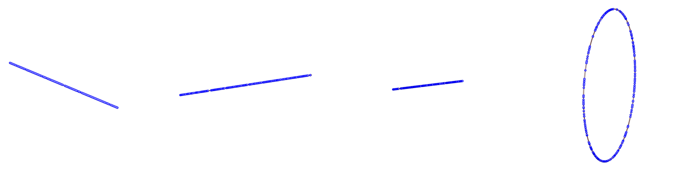
defined on a periodic cube $[0, 1]^3$, with $\boldsymbol{x} = (x, y, z)^T$. P is the period of the velocity field, the deformation it generates reaches its maximum extent at $t = P/2$ then returns to identity at $t = P$. For the following tests, we will choose $P = 3$.

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 sides. 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 $\Delta t = 1/96$ for the computation of the forward characteristic map in the 3-dimensional ambient space, 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 are implemented in Matlab with C-Mex subroutines for the interpolation operations. As reference, the 3D ambient characteristic maps was decomposed into 6 submaps in the interval $[0, 1.5]$, each map being stored on hard drive. The total computational time for calculating the ambient characteristic map was 511 seconds.

3.4.3 Evolution of Curves

We apply the redistribution method in 1-dimensional parametric space to maintain an arclength parametrization of curves evolving under the flow given in (3.46). We use 4 curves in this test, shown in figure 3–3.



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

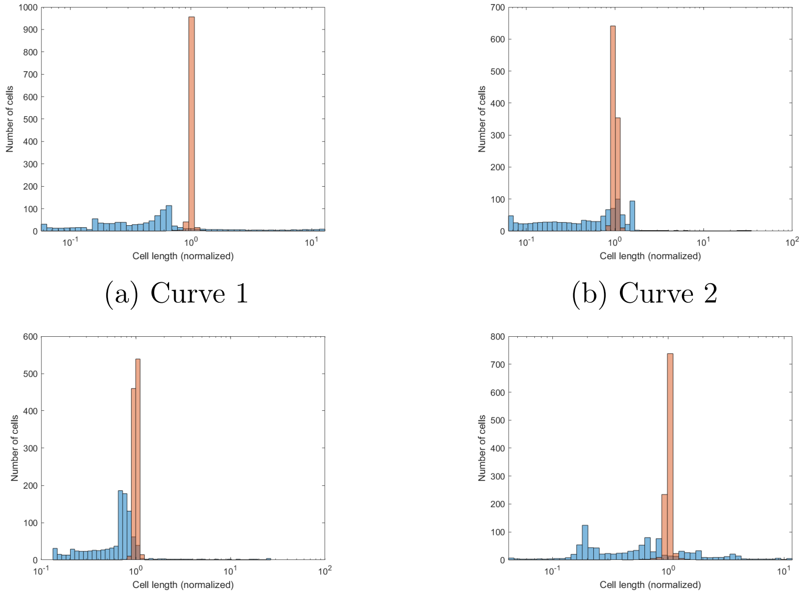
Figure 3–3: Initial curves with random sampling.

To illustrate the effect of the redistribution, we show in figure 3–5 the final states of the curves at various resolutions (the initial curves and their time evolution are shown in Appendix A). In figure 3–5, 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 3–1 shows that interpolating the $\boldsymbol{P}$ parametrization yields and 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 between marker points that were equidistant at $t = 0$. This effect can be clearly seen in figure 3–4 where the histogram of the cell lengths for the curves are 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.

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

Table 3–1: Error of linear interpolation for the parametrization functions at different grid resolutions, using the $N = 1024$ tests as reference.

We can also measure the uniformity of the marker points distribution quantitatively: table 3–2 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 indicates that the



(a) Curve 1

(b) Curve 2

(c) Curve 3

(d) Curve 4

Figure 3–4: Arclength distribution of the parametrizations $\boldsymbol{P}$ (blue) and $\boldsymbol{Q}$ (red).

arclength distance between two sample points are more uniform thereby avoiding the undersampling of the more deformed parts of the curves or redundancy of markers in compressed parts.

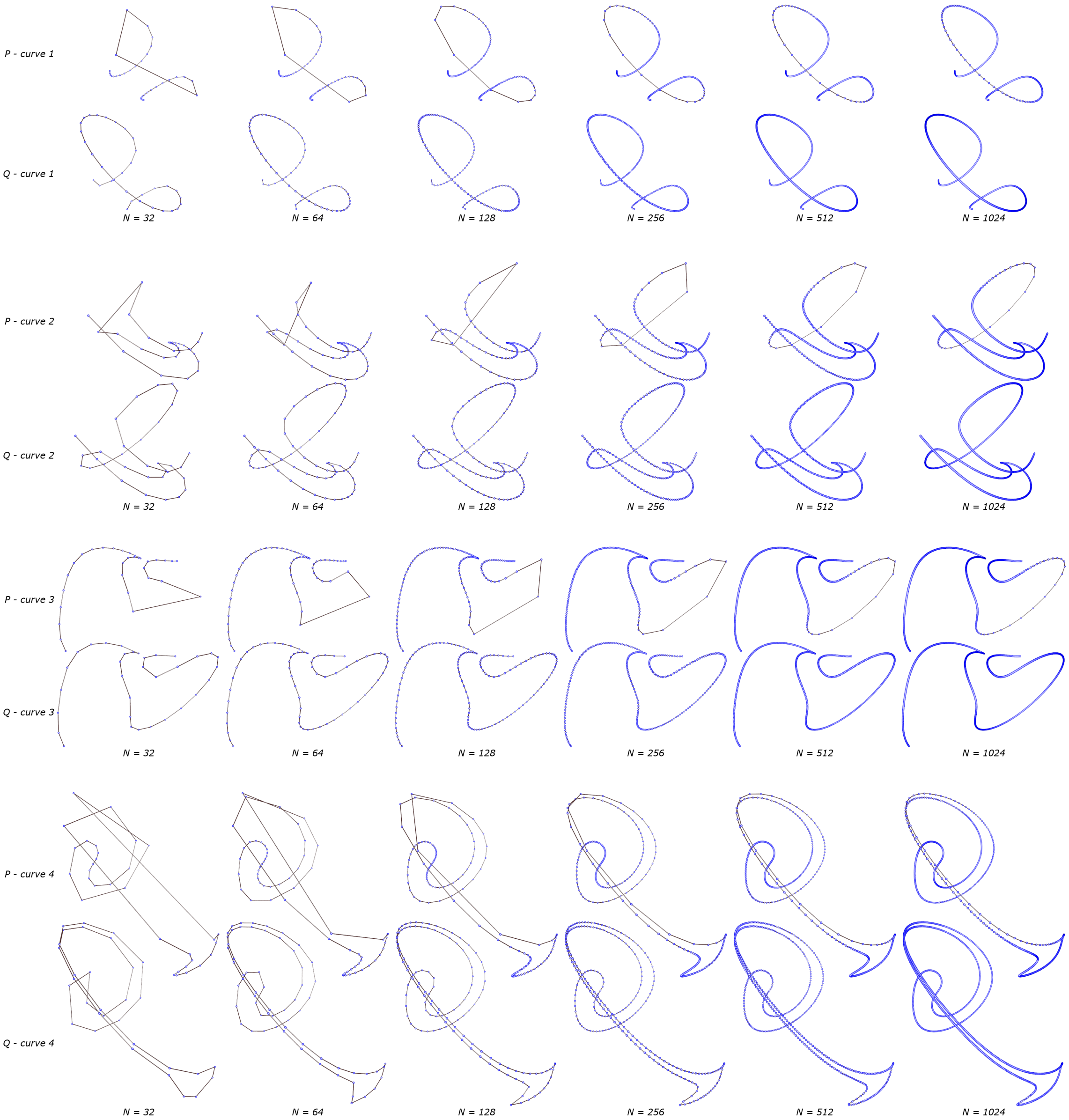


Figure 3–5: $\boldsymbol{P}$ and $\boldsymbol{Q}$ parametrizations of curves 1 to 4 using gradually finer grids.

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.

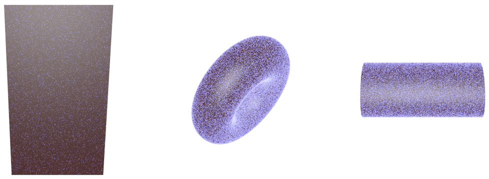
Curves	1	2	3	4
Evaluating $\boldsymbol{Q}$	9.96 s	9.32 s	9.61 s	9.62 s
Defining $\rho^n(\boldsymbol{x})$	0.04 s	0.04 s	0.05 s	0.04 s
Updating $\boldsymbol{\mathcal{X}}_B$	0.74 s	0.70 s	0.67 s	0.59 s
Number of remappings	2	3	3	3

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

Table 3–2: Parametrizations $\boldsymbol{P}$ and $\boldsymbol{Q}$ at $t = 1.5$.

3.4.4 Evolution of Surfaces

We test the redistribution method on three different topologies for 2-dimensional surfaces: rectangle, torus and cylinder. The parametric spaces are taken to be $U = [0,1]^2$ with Neumann, periodic and mixed Neumann-periodic boundary conditions respectively. These surfaces will move under the flow (3.46) 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 3–7 to 3–9, 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 3.1.2. The initial surfaces with marker points are shown in 3–6, the time evolution of each surface is shown in figures 3–7 to 3–9. We also show the standard deviation and median of the normalized area densities and the computational times for the redistribution maps in figure 3–10 and table 3–3.



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

Figure 3–6: Initial surfaces with uniformly distributed random sample points.

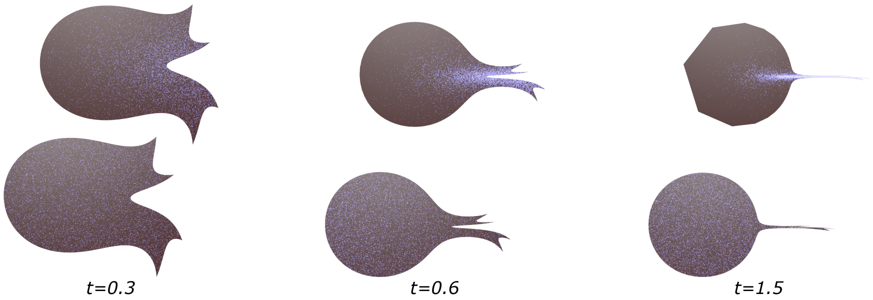


Figure 3–7: Comparison of the surface sampling of the evolved rectangle using the original parametrization $\boldsymbol{P}$ (top images) and the redistributed parametrization $\boldsymbol{Q}$ (bottom images).

Figure 3–7 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

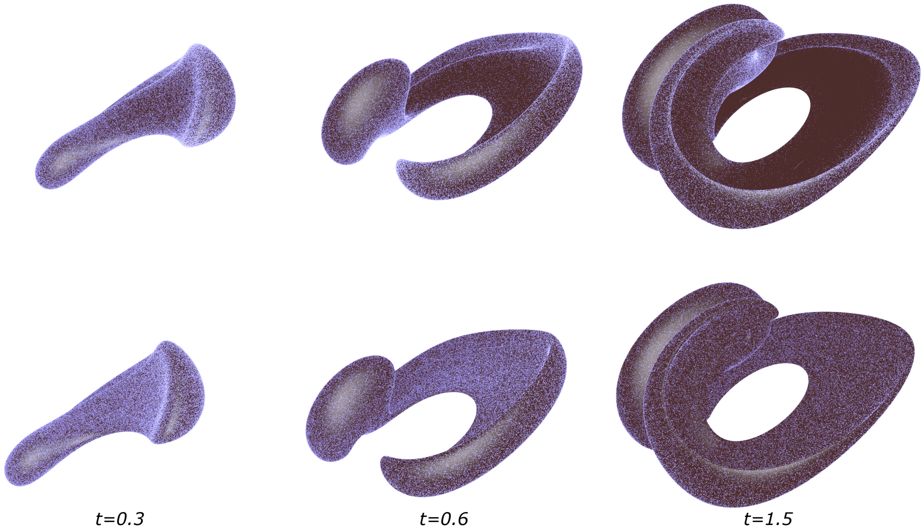


Figure 3–8: Comparison of the surface sampling of the evolved torus using the original parametrization $\boldsymbol{P}$ (top images) and the redistributed parametrization $\boldsymbol{Q}$ (bottom images).

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 an uniform sampling and a smooth, well-resolved surface interpolation. Similar observations can be made in figure 3–8 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 3–9, 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

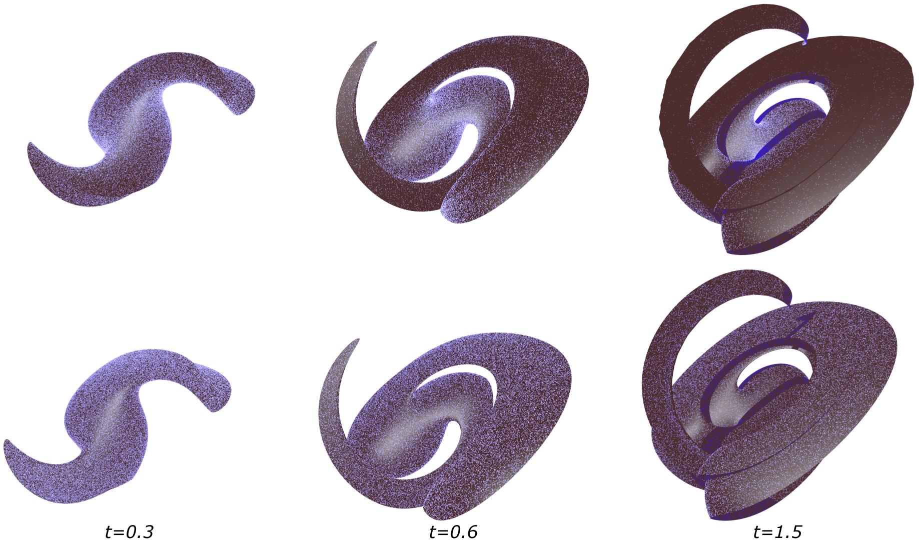
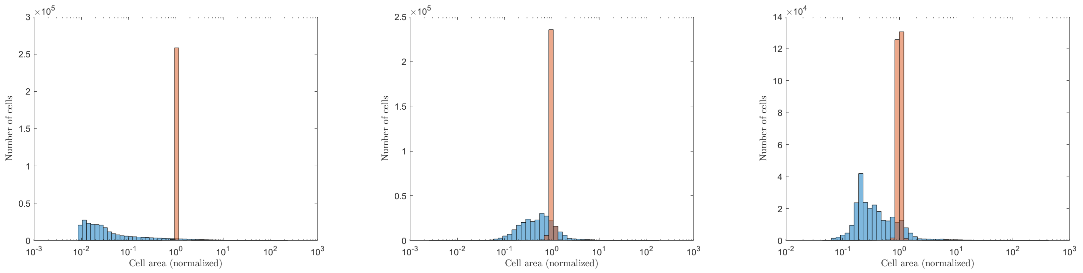


Figure 3–9: Comparison of the surface sampling of the evolved cylinder using the original parametrization $\boldsymbol{P}$ (top images) and the redistributed parametrization $\boldsymbol{Q}$ (bottom images).

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 exhibit poor resolution of the boundary. The piecewise linear interpolation of $\boldsymbol{P}$ is jagged at the 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, smooth.

We quantify the effect of the redistribution by plotting the histogram of the cell area population in figure 3–10. 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 non-uniform, suboptimal distribution of marker points. In particular, for the rectangle, we see in figure 3–10a, 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 3–7. The reparametrization $\boldsymbol{Q}$ is more uniform, with almost all cells having the average area. These properties are also shown quantitatively in table 3–3 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 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 3.4.2); this reparametrization time includes the intermittent evaluation of the full $\boldsymbol{Q}$ parametrization (as defined in (3.44)) 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.



(a) Rectangle

(b) Torus

(c) Cylinder

Figure 3–10: Area distribution of the parametrizations $\boldsymbol{P}$ (blue) and $\boldsymbol{Q}$ (red)

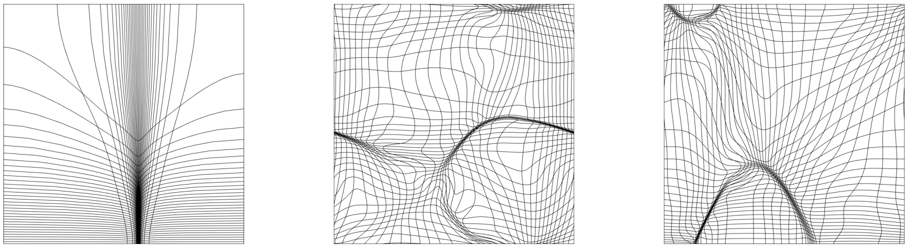
Surfaces	Rect.	Torus	Cyl.
$\sigma_{\boldsymbol{P}}$	1.6749	0.8828	1.098
$\sigma_{\boldsymbol{Q}}$	0.0094	0.0485	0.0321
$M_{\boldsymbol{P}} - 1$	-0.9722	-0.4754	-0.6718
$M_{\boldsymbol{Q}} - 1$	-0.0007	-0.0001	-0.0019

(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 $\boldsymbol{Q}$.

Table 3–3: Parametrizations $\boldsymbol{P}$ and $\boldsymbol{Q}$ at $t = 1.5$.



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

Figure 3–11: The redistribution map on the parametric space for each surface at $t = 1.5$.

Indeed, figure 3–11 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 group structure of the 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 (3.44). The advantage of using the decomposition method for the redistribution map is two-fold: first, if we assume that at time T_i, the redistributed parametrization $\boldsymbol{Q}_{T_i}$ is close to equiareal, then the variations in $\rho(\boldsymbol{x}, T_i)$ are small and therefore can be resolved on a coarse grid for the computation of $\boldsymbol{\mathcal{X}}_{[T_{i+1},T_i]}$. 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 push-forward 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 an uniform sampling of the surface and area features above a certain spatial scale as indicated in theorem 3.3.1 are guaranteed to be resolved.

Remark 3.4.1. 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:

$$\partial_t \rho = \Delta(\rho - \rho_\kappa) \quad \text{with} \quad \boldsymbol{u} = -\frac{\nabla(\rho - \rho_\kappa)}{\rho}, \tag{3.47}$$

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.

3.5 Concluding Remarks for Chapter 3

In this chapter, we have presented a 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 initial condition and extract a velocity field from the continuity equation, flowing the initial density towards 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 3.1.2 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 3.3. 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 push-forward 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 3.4. This method is unique in that the changes in the parametrization function are made by pre-composition 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 characteristic mapping method allows us to exploit the group structures of the diffeomorphisms associated to 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 equidistribute the resulting large variations in area density on the parametric space.

CHAPTER 4
Two-dimensional incompressible Euler Equations

The modelling of the movement of an ideal fluid by the Euler equations is one of the most famous and important applications of the advection equation. In the two-dimensional case, written in its vorticity form, the incompressible Euler equation essentially states that the scalar vorticity is advected by the fluid velocity. In this chapter, we study the application of the CM framework to the 2D incompressible Euler equations. The content of this chapter contains joint work with Olivier Mercier and Badal Yadav.

The study of the incompressible Euler equations in two-dimensional (2D) space poses many challenges. It is well known that solutions of the 2D Euler equations, although smooth, have fast growing gradients. The growth of the sup-norm of the vorticity gradient can be bounded by a double exponential in time [124], this bound was proven to be sharp in the case of smooth initial data on a disk [73]. Furthermore, on a flat periodic 2-torus, it has been shown that for an arbitrary time interval, there exists a vorticity field whose gradient exhibits exponential growth within the chosen interval [37]. For a summary on the advances in mathematics for the Euler equations, we refer to a 2013 review by Bardos and Titi [8]. These theoretical results suggest that the spatial discretization for the numerical solution of the Euler equations can be challenging. As the gradient of the solution grows, the numerical resolution required to correctly capture the solution also increases. The resources necessary to avoid excessive spatial truncation errors can thus become prohibitive for long time simulations. On the other hand, allowing for spatial truncation errors by undersampling the solution can generate numerical dissipation akin to a viscosity term which qualitatively affects the simulation. In some cases, truncation errors resonate with the solution, creating spurious oscillations and numerical instability.

This phenomenon is analysed in detail by Ray et al. [99] in the case of a conservative Fourier-Galerkin scheme, and a wavelet regularization technique has been proposed by Pereira et al. in [95].

Many numerical solvers use a mix of Lagrangian and Eulerian descriptions to exploit the advantages of each approach. For instance, Arbitrary Lagrangian-Eulerian [38, 59] methods use reference coordinates that are neither fully Lagrangian nor fully Eulerian to describe the fluid configuration. Fully Lagrangian methods are typically formulated to discretize the solution as a collection of Lagrangian particles, these methods are characterized by a non-dissipative evolution of the solution. For instance, the fully Lagrangian scheme proposed by Bowman, Yassaei and Basu [18] achieves a non-dissipative transport of the solution and thus preserving all Casimir invariants. Another large family of numerical solvers are the semi-Lagrangian methods. In these methods, relevant quantities are represented on an Eulerian frame, however, the evolution equations are discretized from a Lagrangian description. Methods of this type include the Cauchy-Lagrangian method [97] where the vorticity field is evolved by transport along a moving mesh following the flow, the result is then periodically projected back on an Eulerian grid. Level-set methods are another popular semi-Lagrangian framework used for fluid simulation and implicit interface tracking, e.g. [92]. In these methods, characteristics are traced backwards in time and Eulerian interpolation schemes are used to update the solution. Semi-Lagrangian schemes have also been used to generate fast simulations intended for computer graphics [43, 111]. One main property of the semi-Lagrangian approach is that it tries to capture the characteristic structure of the equations. In particular, in 2D inviscid flow, the vorticity field is transported along characteristic curves, hence accuracy and stability can be improved by taking the geometric approach of following these characteristic curves in order to propagate the solution in time.

The CM approach for solving the Euler equations in two-dimensional space consists in splitting the evolution equations into the advection of the vorticity using the CM method and the computation of the velocity through the Biot-Savart law. These two parts are connected in that the advection provides the vorticity field for the Biot-Savart kernel, whose resulting velocity field is then used to advect the characteristic map. In doing so we achieve a separation of scales: under the assumption that the flow is governed by large scale features of the velocity field, the characteristic map can be accurately evolved on a coarse grid. Furthermore, since the vorticity solution is constructed through the pullback of the initial condition by the characteristic map, the functional definition of the solution provides arbitrarily fine spatial resolution.

One main issue in inviscid fluid simulation is the artificial viscosity incurred from the spatial truncations during the evolution of the solution. Some methods such as [97] approach this problem by designing high order methods in order to take extremely large time steps, hence minimizing the accumulation of the diffusive error. Others employ an adaptive multi-resolution mesh refinement to efficiently resolve fine scale features [35, 55, 75, 115]. One unique property of the CM method for 2D Euler is that it eliminates artificial viscosity by never directly evolving a discretized vorticity field: the vorticity changes as a consequence of the evolving characteristic map used to compute the pullback. In particular, a straightforward consequence is that the extrema of the vorticity field are conserved for all times. Additionally, in order to correctly represent the arbitrarily fine scales generated by the flow, we use a time decomposition of the characteristic map based on its group structure. This allows the CM method to represent exponentially growing vorticity gradients while only computing on a fixed coarse grid. The resulting scheme achieves arbitrary subgrid resolution, high long term enstrophy conservation and is non-dissipative.

The rest of this chapter is structured as follows. In section 4.1 we lay out the mathematical framework for the CM method and carry out some heuristic analysis of its properties. In section 4.2, we present in detail the numerical implementation of the method and provide some error bounds. Section 4.3 contains several numerical tests and discussions on the accuracy and qualitative properties of the solutions. Finally, we make some concluding remarks and propose future directions of work in section 4.4.

4.1 Mathematical Framework

The equations governing the motion of a perfect fluid in two-dimensional space are:

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p \qquad (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{4.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{4.1b}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), \tag{4.1c}$$

where U is some fixed spatial domain, $\boldsymbol{u}$ is a vector field describing the instantaneous velocity of a fluid element and p is the pressure. For this chapter, we assume for simplicity that U is the flat 2-torus and hence there are no boundary conditions. In general, for a boundary ∂U, the no penetration boundary condition for inviscid flow is $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on ∂U where $\boldsymbol{n}$ is the normal vector to the boundary. This will require further extensions to the framework and is not covered in this work.

Define $\omega = \nabla \times \boldsymbol{u}$ the scalar vorticity of the fluid, and taking the 2D curl of (4.1), we obtain the vorticity equations:

$$\partial_t \omega + (\boldsymbol{u} \cdot \nabla)\omega = 0 \qquad (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{4.2a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{4.2b}$$

$$\omega(\boldsymbol{x}, 0) = \omega_0(\boldsymbol{x}). \tag{4.2c}$$

Using the incompressibility assumption, $\boldsymbol{u}$ can be obtained from ω by solving a Helmholtz-Hodge problem, $\boldsymbol{u}$ is then given by the Biot-Savart law

$$\boldsymbol{u} = -\Delta^{-1} \nabla \times \omega. \tag{4.3}$$

From equation (4.2), we see that the scalar vorticity ω satisfies an advection equation under the velocity field $\boldsymbol{u}$. If we assume for now that the velocity field $\boldsymbol{u}$ is given, then the ω can be computed using the CM method described in section 2.1, that is

$$\omega(\boldsymbol{x},t) = \omega_0(\boldsymbol{X}_{[t,0]}(\boldsymbol{x})), \tag{4.4}$$

where $\boldsymbol{X}_{[t,0]}$ is the time t backward characteristic map generated by $\boldsymbol{u}$. The solution of the Euler equations can be obtained by coupling the above vorticity pullback definition with a Biot-Savart computation of the velocity field used to evolve the map. This treatment of the Euler equations essentially reformulates the equations as a geodesic flow on the Lie group of volume preserving diffeomorphisms $\mathit{SDiff}(U)$ on the domain U.

4.1.1 Group structure

The treatment of incompressible inviscid flow from the point of view of differential geometry and geodesic flow on the group of volume preserving diffeomorphisms was examined by Arnold in 1966 [4]. The backward characteristic maps are the inverse maps of the elements of the one-parameter semigroup of these volume preserving forward flow maps parametrized by t. This allows for the time decomposition of the characteristic map according to equation (2.7a).

Numerically, this group property allows us to adaptively adjust the numerical resolution of the map. In particular, a backward characteristic map $\boldsymbol{X}_B(\cdot,t)$ for the time interval $[0,t]$ can be split into arbitrarily many submaps in the following way:

We subdivide the interval $[0,t]$ into m subintervals $[T_{i-1},T_i]$ with $0 = T_0 < T_1 < \cdots < T_m = t$. One can then check that the following decomposition holds:

$$\boldsymbol{X}_B(\cdot,t) = \boldsymbol{X}_{[t,0]} = \boldsymbol{X}_{[T_1,0]} \circ \boldsymbol{X}_{[T_2,T_1]} \circ \cdots \circ \boldsymbol{X}_{[T_{m-1},T_{m-2}]} \circ \boldsymbol{X}_{[t,T_{m-1}]} \tag{4.5}$$

Each of the submaps $\boldsymbol{X}_{[T_i,T_{i-1}]}$ can be computed individually and stored. The global time map is then defined as the composition of all the stored submaps. This decomposition will provide several important numerical advantages which we will discuss in section 4.2. In particular, each submap has the identity map as initial condition and the subdivision allows for dynamic and adaptive spatial resolution without changing the computational grid.

In 2D incompressible Euler, the vorticity gradient can grow exponentially in time. Using the characteristic map, we can write

$$\omega(\boldsymbol{x},t) = \omega_0(\boldsymbol{X}_B(\boldsymbol{x},t)). \tag{4.6}$$

We observe from this that the advection operator is responsible for the formation of high vorticity gradient since

$$\nabla\omega(\boldsymbol{x},t) = \nabla\omega_0 \cdot \nabla\boldsymbol{X}_B(\boldsymbol{x},t). \tag{4.7}$$

In cases where the vorticity gradient grows exponentially, we infer that $\nabla\boldsymbol{X}$ must also grow exponentially. The semigroup decomposition is analogous to the exponential function in one variable, where the natural instantaneous evolution is multiplicative instead of additive in the sense that $\exp(c(t+\Delta t)) \approx (1+c\Delta t)\exp(ct)$ is more natural than $\exp(c(t+\Delta t)) \approx \exp(ct) + c\Delta t\exp(ct)$ (the latter requiring tracking an integrand which grows exponentially). Similarly, taking the gradient of equation (4.5), we have

$$\nabla\boldsymbol{X}_{[t,0]} = \nabla\boldsymbol{X}_{[T_1,0]}\nabla\boldsymbol{X}_{[T_2,T_1]}\cdots\nabla\boldsymbol{X}_{[T_{m-1},T_{m-2}]}\nabla\boldsymbol{X}_{[t,T_{m-1}]}. \tag{4.8}$$

This means that an exponential growth in gradient can be achieved by the composition of submaps each having bounded gradient.

4.1.2 Advection-Vorticity Coupling

We use the backward characteristic map to rewrite the vorticity equation (4.2a) as the following coupling of $\boldsymbol{u}, \omega$ and $\boldsymbol{X}_B$:

$$\omega(\boldsymbol{x},t) = \omega_0(\boldsymbol{X}_B(\boldsymbol{x},t)), \tag{4.9a}$$

$$\boldsymbol{u} = -\Delta^{-1}\nabla\times\omega, \tag{4.9b}$$

$$(\partial_t + \boldsymbol{u}\cdot\nabla)\boldsymbol{X}_B = 0. \tag{4.9c}$$

Equation (4.9b) is known as the Biot-Savart law and can be obtained from $\nabla \cdot \boldsymbol{u} = 0$ and the definition of the vector Laplacian:

$$\Delta \boldsymbol{F} = \nabla (\nabla \cdot \boldsymbol{F}) - \nabla \times (\nabla \times \boldsymbol{F}) \tag{4.10}$$

for $\boldsymbol{F}$ a $\mathbb{R}^2 \to \mathbb{R}^2$ vector field. We let $\boldsymbol{u} = \boldsymbol{F}$ and commute Δ^{-1} and $\nabla\times$ in (4.9b), the velocity is then obtained from the stream function ψ:

$$\psi = -\Delta^{-1}\omega, \qquad \boldsymbol{u} = \nabla \times \psi. \tag{4.11}$$

The CM method for Euler equations then consists of numerically evolving $\boldsymbol{X}$ in time using equation (4.9c); the velocity and vorticity are defined using (4.9a), and (4.11). Additionally, if we employ the submap decomposition from the group property, we would solve equation (4.9c) for the submaps $\boldsymbol{X}_{[t,T_i]}$ for $t \in [T_i, T_{i+1}]$ and initial condition $\boldsymbol{X}_{[T_i,T_i]}(\boldsymbol{x}) = \boldsymbol{x}$; the pullback (4.9a) would be computed from the composition of all submaps.

4.1.3 Modified Equations

In order to qualify the properties of the numerical solution obtained from this method, we look at the modified equations.

Let $\boldsymbol{\mathcal{X}}_B^n$ be the numerical characteristic map at time step t_n. Let $\omega^n = \omega_0(\boldsymbol{\mathcal{X}}^n)$ and let $\boldsymbol{u}^n$ be the velocity field associated to ω^n. This velocity $\boldsymbol{u}^n$ is used to advance the map to time t_{n+1} from which we obtain the velocity $\boldsymbol{u}^{n+1}$ from pullback and Biot-Savart; the process continues so on. Here, we look at the modified equation which arises from replacing the true velocity $\boldsymbol{u}$ by some modified $\tilde{\boldsymbol{u}}$ which is better approximated by the discrete evolution of the characteristic map.

We look at a modified velocity $\tilde{\boldsymbol{u}}$ defined at all times which approximates the velocities $\boldsymbol{u}^n$ at discrete time steps t_n. The corresponding modified equation for the characteristic map is then

$$\partial_t \tilde{\boldsymbol{X}}_B + (\tilde{\boldsymbol{u}} \cdot \nabla)\tilde{\boldsymbol{X}}_B = 0. \tag{4.12}$$

From here on, we will use the tilde notation for variables associated to the modified equation. Discretized variables will be denoted by script letters with a superscript n referring to the corresponding time step t_n.

We will estimate the difference between the true solution $\boldsymbol{X}_B$ and the solution to the modified equation $\tilde{\boldsymbol{X}}_B$. This is a useful strategy as it will allow us to estimate the error of the numerical solution $\boldsymbol{\mathcal{X}}_B$ using $|\boldsymbol{X}_B - \boldsymbol{\mathcal{X}}_B| \leq |\boldsymbol{X}_B - \tilde{\boldsymbol{X}}_B| + |\tilde{\boldsymbol{X}}_B - \boldsymbol{\mathcal{X}}_B|$.

Consider the evolution of $\boldsymbol{X}_B$ and $\tilde{\boldsymbol{X}}_B$ in some time interval $[t_0, t]$ (for $t_0 < t$). We can write $\boldsymbol{X}_B$ and $\tilde{\boldsymbol{X}}_B$ in integral form:

$$\boldsymbol{X}_{[t,t_0]}(\boldsymbol{x}) = \boldsymbol{x} + \int_t^{t_0} \boldsymbol{u}\big(\boldsymbol{X}_{[t,r]}(\boldsymbol{x}), r\big) dr, \tag{4.13a}$$

$$\tilde{\boldsymbol{X}}_{[t,t_0]}(\boldsymbol{x}) = \boldsymbol{x} + \int_t^{t_0} \tilde{\boldsymbol{u}}\big(\tilde{\boldsymbol{X}}_{[t,r]}(\boldsymbol{x}), r\big) dr. \tag{4.13b}$$

By Grönwall's lemma, we have that

$$\left|\boldsymbol{X}_{[t,t_0]}(\boldsymbol{x}) - \tilde{\boldsymbol{X}}_{[t,t_0]}(\boldsymbol{x})\right| \lesssim C(t-t_0)e^{C(t-t_0)}\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_\infty, \tag{4.14}$$

for some $C \approx \|\nabla \boldsymbol{u}\|_\infty$.

Heuristically, we note from this that if a numerical solution $\boldsymbol{\mathcal{X}}_B$ approximates well the solution $\tilde{\boldsymbol{X}}_B$ of the modified equation generated by the discretized velocity field $\tilde{\boldsymbol{u}}$ (or some approximation of it), that is, if the scheme is "self-consistent", then it is sufficient to control the difference between $\tilde{\boldsymbol{u}}$ and the true velocity $\boldsymbol{u}$. This tool makes the error analysis more straightforward and allows us to obtain better bounds on the error in the conservation of various advected quantities. In particular, we can write the modified equation approximated by the numerical vorticity solution:

$$\partial_t(\omega_0 \circ \tilde{\boldsymbol{X}}_B) = \nabla\omega_0 \cdot \partial_t \tilde{\boldsymbol{X}}_B = -\nabla\omega_0 \cdot (\tilde{\boldsymbol{u}} \cdot \nabla)\tilde{\boldsymbol{X}}_B = -(\tilde{\boldsymbol{u}} \cdot \nabla)(\omega_0 \circ \tilde{\boldsymbol{X}}_B). \tag{4.15}$$

That is, the numerical vorticity approximates an advection equation under the modified flow generated by $\tilde{\boldsymbol{u}}$, making the error "advective" rather than diffusive. The exact nature of the modified equation is unclear and depends on the leading order term of $\boldsymbol{u} - \tilde{\boldsymbol{u}}$. In cases where spatial truncation errors dominate, we shall see that this is closely related to the Lagrangian-Averaged Euler equations.

4.1.4 Multiscale Evolution

Due to the presence of different scales in the solution and to limited computational resources, we need to make choices on the degree of spatial truncation appropriate to each evolved quantity. In particular, the Biot-Savart law implies that $\boldsymbol{u}$ has a faster decay in its Fourier coefficients compared to ω and $\boldsymbol{X}_B$. Therefore, it can be represented on a coarser grid without incurring excessive L^∞ error. In this method, we make a similar assumption as in Lagrangian-Averaged Euler (LAE-α), that is, that the low frequency features of the velocity dictate the global evolution of the flow and high frequency small features do not need to be solved for exactly: it is sufficient to resolve the coarse scales of the velocity field as long as the fine scale features of the transported vorticity are not lost. With this assumption in mind, we design a method where the representation of the instantaneous velocity as well as the short time deformation map (by "short time", we refer to the submaps in the decomposition (4.5)) are done on a grid much coarser than the fine scales present in the vorticity solution. However, since the vorticity is defined as the pullback/rearrangement by $\boldsymbol{X}_B$, all scales in the vorticity are preserved and coherently transported under the smoothed velocity field. Indeed, in the CM method, ω is defined as a function $\omega_0 \circ \boldsymbol{X}_B$ over the entire domain and evolves in time as a consequence of the changes in the deformation map $\boldsymbol{X}_B$. The absence of a grid-based discretization of ω means that we do not incur a dissipative error due to spatial truncation and preserve all small scale features. As a result, the CM method can achieve arbitrary resolution on the vorticity field and allows us to separate the scales involved in the problem: the large scales are computed and accurately represented whereas small scales are preserved and passively transported.

Here we make a parallel between the CM method and the LAE-α equations. The LAE-α equations aim at modelling the flow of an incompressible inviscid fluid on a spatial scale larger than α by taking a Lagrangian average of the velocity field. For more details on the LAE-α and LANS-α formulation, readers can refer to [82,83,87]. In its vorticity form, the LAE-α can be written as

$$\partial_t \omega + (\boldsymbol{u} \cdot \nabla)\omega = 0, \tag{4.16a}$$

$$\boldsymbol{u} = -\nabla \times \Delta^{-1}(1 - \alpha\Delta)^{-1}\omega. \tag{4.16b}$$

The above equation also models the flow of a second-grade non-Newtonian fluid; an analysis of the relation between second-grade non-Newtonian fluids, the vortex blob method [27] and the LAE-α equations can be found in [90].

We see from (4.16) that the vorticity is transported by $\boldsymbol{u}$ and that $\boldsymbol{u}$ is obtained from the Biot-Savart law on a smoothed vorticity field $(1-\alpha\Delta)^{-1}\omega$. This smoothing effect can also be achieved by a spatial filter during the same Biot-Savart computation in CM, resulting in a smoothed modified velocity $\tilde{\boldsymbol{u}}$ (see section 4.1.3). The difference is that the quantity of interest in LAE-α is the averaged velocity whereas in CM, we are interested in the vorticity field which contains arbitrarily fine scales. It remains that in cases where the sampling of the velocity in CM corresponds to the $(1-\alpha\Delta)^{-1}$ smoothing in LAE-α, the flow maps from both formulations are the same and hence generate the same dynamics.

4.2 Numerical Implementation

The CM method for the 2D Euler equations uses the scheme presented in section 2.1 for the advection component. The definition of a velocity field in the interval $[t_n, t_{n+1}]$ as well as the numerical time-integration schemes are required to compute the one-step map $\boldsymbol{X}_{[t_{n+1},t_n]}$ in (5.22). We will detail these in this section.

4.2.1 Spatial representation of the velocity field

In 2D Euler, the vorticity field solves a transport equation under the velocity $\boldsymbol{u}$. Assuming we have the global time characteristic map, $\boldsymbol{X}_B(\cdot, t)$, and that the initial condition for the vorticity ω_0 is given analytically, we have that the vorticity at time t is given by

$$\omega(\boldsymbol{x}, t) = \omega_0(\boldsymbol{X}_B(\boldsymbol{x}, t)). \tag{4.17}$$

This evaluation is necessary at every time step for the Biot-Savart law. Numerically, the spatial resolution is limited by the grid's Nyquist frequency resulting in a spatial filter when defining the velocity $\boldsymbol{u}$. This grid needs to be fine enough to avoid Fourier aliasing from lack of resolution. Here we propose a sampling scheme based on spatial filtering by defining a mollified velocity

$$\boldsymbol{u}_\epsilon = \mu_\epsilon * \boldsymbol{u}, \tag{4.18}$$

where μ_ϵ is a mollifier supported in a neighbourhood of size ϵ. Commuting derivatives and mollification, it is sufficient to compute the mollified vorticity. We pick ϵ to be smaller than the cellwidth of $\boldsymbol{V}$, the evaluation of the mollified vorticity at grid points $\boldsymbol{x}_{i,j}$ can then be expressed as the sum of the convolution integrals in all 4 cells adjacent to $\boldsymbol{x}_{i,j}$:

$$(\mu_\epsilon * \omega^n)(\boldsymbol{x}_{i,j}) = \sum_{\boldsymbol{r} \in \{-1,0\}^2} \int_{C_{i+r_1,j+r_2}} \mu_\epsilon(\boldsymbol{x}_{i,j} - \boldsymbol{y}) \omega^n(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{4.19}$$

The integral in each cell is computed using numerical quadrature of sufficiently high accuracy. The order of the quadrature could be in principle adapted in each cell; in this chapter, quadrature order is fixed for simplicity. The resulting algorithm effectively computes a mollified vorticity field where subgrid oscillations are filtered out, by choosing an appropriate mollification scale, the $\boldsymbol{V}$ grid can resolve the mollified vorticity without aliasing errors and the pointwise evaluation of $\mu_\epsilon * \omega^n$ on $\boldsymbol{V}$ is accurate as long as the sampling in each cell is sufficiently dense.

Remark 4.2.1. We only use this mollified vorticity to generate a mollified velocity, the true numerical solution is still defined as $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}_B^n$.

The velocity field can be obtained from ω_ϵ^n through the stream function as in equation (4.11). Numerically we will solve this using a spectral method with Fast Fourier transforms. Let $\mathcal{F}$ denote the Discrete Fourier transform operator, we have:

$$\psi_\epsilon^n := -\mathcal{F}^{-1}\left[\Delta^{-1}\mathcal{F}\left[\omega_\epsilon^n\right]\right], \tag{4.20}$$

where the above Δ^{-1} solves the Poisson equation in Fourier space and is a diagonal operator.

The divergence-free property of the velocity should be preserved in order for the characteristic map to be volume preserving. Therefore $\boldsymbol{u}^n$ should be defined as the curl of some scalar function. To achieve this, we use a grid $\boldsymbol{V}$ for the representation of the velocity field and define

$$\boldsymbol{u}_\epsilon^n := \nabla \times \mathcal{H}_{\boldsymbol{V}}[\psi_\epsilon^n]. \tag{4.21}$$

This definition guarantees that $\boldsymbol{u}_\epsilon^n \in \text{curl}(\mathcal{V}_{\boldsymbol{V}}) \subset \{f \in \mathcal{C}^0(U) | \nabla \cdot f \equiv 0\}$. Indeed, $\boldsymbol{u}^n$ is C^∞ in each cell and continuous across cell boundaries, its divergence is however continuous everywhere and identically 0 due to the continuity of the mixed partials $\partial^{(1,1)}\mathcal{H}_{\boldsymbol{V}}[\psi_\epsilon^n]$. As a result, the numerical flow is also divergence-free which allows us to control the error on the volume-preserving property of the characteristic map.

Remark 4.2.2. The grid $\boldsymbol{V}$ and the parameter ϵ used to represent the velocity field are independent of the grid used for the evolution of $\boldsymbol{\mathcal{X}}_B$, the interpolant is also not restricted to Hermite cubics. Consequently, $\boldsymbol{V}$ can be made arbitrarily fine and ϵ arbitrarily small such that $\boldsymbol{u}_\epsilon^n$ approach the exact Biot-Savart velocity associated to the numerical vorticity $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}_B$. Furthermore, it is important to have a fine enough $\boldsymbol{V}$ so that the interpolation of ψ_ϵ^n does not create new extrema from undersampling, as this can cause spurious oscillations and pollute the solution. In fact, choosing a fine grid does not require large computational resources as it only involves an inverse FFT of a zero-padded $\hat{\psi}_\epsilon^n$.

4.2.2 Time Discretization

Section 4.2.1 gives us a discretization of the velocity field at time t_n given the characteristic map. In this section, we provide the time discretization for the evolution of the characteristic map based on this $\boldsymbol{u}_\epsilon^n$.

The characteristic map can be evolved using the group property (2.7a). Let t_n be the discrete time steps, with $\Delta t = t_{n+1} - t_n$, we have

$$\boldsymbol{X}_{[t_{n+1},0]} = \boldsymbol{X}_{[t_n,0]} \circ \boldsymbol{X}_{[t_{n+1},t_n]}. \tag{4.22}$$

In order to approximate the one-step map $\boldsymbol{X}_{[t_{n+1},t_n]}$, we extend the velocity $\boldsymbol{u}_\epsilon^n$ to the time interval $[t_n, t_{n+1}]$ using an order p Lagrange polynomial in time. This is similar to a multistep method. Let

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}, t) := \sum_{i=0}^{p-1} l_i(t) \boldsymbol{u}_\epsilon^{n-i}(\boldsymbol{x}), \tag{4.23}$$

where l_i are the Lagrange basis polynomials for time steps t_{n-i}. We note that for each fixed time t, $\tilde{\boldsymbol{u}}$ is a linear combination of $\boldsymbol{u}_\epsilon^{n-i}$. In particular, this implies that if $\nabla \cdot \boldsymbol{u}_\epsilon^{n-i} = 0$ then $\nabla \cdot \tilde{\boldsymbol{u}} = 0$.

The one-step map $\boldsymbol{X}_{[t_{n+1},t_n]}$ is then approximated from (4.13b) by a k-stage Runge-Kutta integration of the velocity $\tilde{\boldsymbol{u}}$ along characteristic curves, for Δt backward in time:

$$\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}) = \boldsymbol{x} - \Delta t \sum_{j=1}^{s} b_j k_j, \tag{4.24}$$

with

$$k_j = \tilde{\boldsymbol{u}} \left(\boldsymbol{x} - \Delta t \sum_{m=1}^{j-1} a_{jm} k_m, t_{n+1} - c_j \Delta t \right), \tag{4.25}$$

and $k_0 := 0$. The coefficients a, b, c are those of the Butcher tableau corresponding to the explicit Runge-Kutta scheme.

Remark 4.2.3. The smoothness of $\tilde{\boldsymbol{u}}$ is required for the convergence of RK schemes. However, the Hermite cubic definition of $\boldsymbol{u}_\epsilon^n$ in section 4.2.1 is $\mathcal{C}^0$. Nonetheless, this does not pose a problem since $\mathcal{H}_{\boldsymbol{V}}[\psi_\epsilon^n]$ is a piecewise polynomial approximation of ψ_ϵ^n, which is smooth: the moduli of smoothness for $\boldsymbol{u}_\epsilon^n$ scale with the cell width of

$\boldsymbol{V}$ (in the limit of infinitely fine grid $\boldsymbol{V}$, $\boldsymbol{u}_\epsilon^n$ is smooth). Therefore, by appropriately scaling $\boldsymbol{V}$ with Δt the lack of smoothness does not affect the convergence.

4.2.3 Characteristic Mapping Method for 2D Incompressible Euler

We combine the time and space discretizations in the previous sections to generate the Characteristic Mapping method for 2D Euler. The evolution of $\boldsymbol{X}_{[t,0]}$ at discrete time steps t_n is given by (4.22). We construct the numerical approximation of $\boldsymbol{X}_B$ by evolving a map in the space of Hermite cubic C^1 diffeomorphisms in the sense that each coordinate function is a piecewise Hermite cubic polynomial defined on some grid $\boldsymbol{M}$.

$$\boldsymbol{\mathcal{X}}_B^{n+1} = \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_B^n \circ \boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}\right], \tag{4.26a}$$

$$\boldsymbol{\mathcal{X}}_B^0(\boldsymbol{x}) = \boldsymbol{x}. \tag{4.26b}$$

where the one step map $\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}$ is given in (4.23) and (4.24) and equation (4.26a) is the Gradient-Augmented Level-Set [89] (GALS) time stepping scheme. The velocities $\boldsymbol{u}_\epsilon^n$ required to define the one step map are given by (4.19), (4.20) and (4.21). The velocities $\boldsymbol{u}_\epsilon^{n-i}$ at previous time steps used in (4.23) are stored until no longer needed (for p steps where p is the order of the Lagrange polynomial chosen to represent $\tilde{\boldsymbol{u}}$).

4.2.4 Error Estimates

We give some estimates on the characteristic map error:

$$\mathcal{E}^n := \|\boldsymbol{X}(\boldsymbol{x}, t_n) - \boldsymbol{\mathcal{X}}^n(\boldsymbol{x})\|_\infty. \tag{4.27}$$

First, we consider some given numerical solutions $\boldsymbol{\mathcal{X}}^n$ for $n = 0, 1, \dots N$ for some unspecified N. The characteristic map at each time step t_n generates a velocity field $\boldsymbol{u}_\epsilon^n$. We take the velocity field $\tilde{\boldsymbol{u}}(\boldsymbol{x}, t)$ given in (4.23) to be the velocity field of the modified equation (4.12). We notice that the numerical solution $\boldsymbol{\mathcal{X}}$ is exactly the CM discretization of the advection operator $\tilde{\boldsymbol{X}}$ for the velocity $\tilde{\boldsymbol{u}}$ (taking $\tilde{\boldsymbol{u}}$ as given). That is $\boldsymbol{\mathcal{X}}$ is a CM method approximation of the advection operator generated by the modified velocity it engenders.

Theorem 4.2.1. *Using an s-stage explicit RK integrator with Hermite cubic spatial interpolation, the numerical characteristic map $\boldsymbol{\mathcal{X}}^n(\boldsymbol{x})$ is consistent with $\tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n)$ to order*

$$\|D^\alpha(\tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n) - \boldsymbol{\mathcal{X}}^n(\boldsymbol{x}))\|_\infty = \mathcal{O}(t_n(\Delta x^{2-|\alpha|} \min(\Delta t, \Delta x^2 \Delta t^{-1}) + \Delta t^s)), \qquad (4.28)$$

for $\alpha \in \{0, 1\}^2$.

Proof. Taking $\tilde{\boldsymbol{u}}$ as a fixed velocity, $\boldsymbol{\mathcal{X}}$ is simply CM method applied to $\tilde{\boldsymbol{u}}$. The error estimates are given in [84]. It is a property of jet-schemes with Hermite cubic interpolants that we lose one order of convergence for the first mixed derivative only in the spatial error term. This is because time integration in jet-schemes computes the function values and mixed derivatives of degree 1 in each dimension and all functions evaluated in the method are at least everywhere C^1.

We note that the velocity field $\tilde{\boldsymbol{u}}$ is smooth in space (see remark 4.2.3), however it may be discontinuous in time at t_n. This does not cause an issue as the smoothness of the velocity is only required in the time step intervals $[t_n, t_{n+1}]$. The local truncation error estimates still hold for the one-step maps $\boldsymbol{\mathcal{X}}_{[t_{n+1}, t_n]}$ and $\tilde{\boldsymbol{X}}_{[t_{n+1}, t_n]}$; the global truncation error can be obtained from the composition and Hermite interpolation of the one-step maps. □

Remark 4.2.4. The above error bounds are valid within the stability requirements that the modified velocity $\tilde{\boldsymbol{u}}$ remains regular and bounded independently of Δx and Δt. In the CM method for Euler equations, the spatial and time grids used to defined $\tilde{\boldsymbol{u}}$ are in fact independent of those for the map $\boldsymbol{\mathcal{X}}$. It is possible stabilize the flow to various degrees by choosing an appropriate spatial resolution and smoothing for $\boldsymbol{u}_\epsilon^n$ to control the numerical behaviour of $\tilde{\boldsymbol{u}}$. Again, this is akin to the α-regularization of the velocity in the LAE-α equations.

Corollary 1. *The CM method for 2D incompressible Euler conserves enstrophy to order $\mathcal{O}(t_n(\Delta x^2 \min(\Delta t, \Delta x^2 \Delta t^{-1}) + \Delta t^s))$.*

Proof. Since $\tilde{\boldsymbol{u}}$ is by definition divergence-free, we have that $\tilde{\boldsymbol{X}}$ is a volume preserving map, i.e. $\det(\nabla\tilde{\boldsymbol{X}}) = 1$. We get by a change of variable that, with $U = \tilde{\boldsymbol{X}}(U)$,

$$\int_U f(\omega_0(\boldsymbol{x}))dx = \int_{\tilde{\boldsymbol{X}}(U)} f(\omega_0(\tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n)))\det(\nabla\tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n))dx = \int_U f(\omega_0(\tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n)))dx, \tag{4.29}$$

for any measurable f. Therefore, we have

$$\int_U f(\omega_0(\boldsymbol{\mathcal{X}}^n(\boldsymbol{x})))dx - \int_U f(\omega_0(\boldsymbol{x}))dx \approx \int_U \nabla(f \circ \omega_0)(\boldsymbol{\mathcal{X}}^n(\boldsymbol{x}) - \tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n))dx \tag{4.30}$$

$$\leq \|\nabla(f \circ \omega_0)\|_{L^2}\|\boldsymbol{\mathcal{X}}^n - \tilde{\boldsymbol{X}}(\cdot, t_n)\|_{L^2} = \mathcal{O}(t_n(\Delta x^2 \min(\Delta t, \Delta x^2\Delta t^{-1}) + \Delta t^s))$$

In particular, taking $f(\omega) = \omega^2$ gives us conservation of enstrophy, and for higher order monomials, this implies that the moments of the vorticity are conserved, as they are in the continuous setting. □

To obtain a full error bound, it is sufficient to bound the difference between the true characteristic map and the map from the modified equation. Let

$$\tilde{\mathcal{E}}^n := \|\boldsymbol{X}(\boldsymbol{x}, t_n) - \tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n)\|_\infty. \tag{4.31}$$

From theorem 4.2.1, we then have that

$$\mathcal{E}^n \leq \tilde{\mathcal{E}}^n + \mathcal{O}(t_n(\Delta x^2 \min(\Delta t, \Delta x^2\Delta t^{-1}) + \Delta t^s)). \tag{4.32}$$

Theorem 4.2.2. *From the above error decomposition we can deduce that the global truncation error for the characteristic map is*

$$\mathcal{E}^n = \mathcal{O}\left(\Delta t^s + \Delta x^2 \min(\Delta t, \Delta x^2\Delta t^{-1}) + \Delta t^p\right). \tag{4.33}$$

Proof. It is sufficient to control the evolution of $\tilde{\mathcal{E}}^n$. We note that $\boldsymbol{X}_{[t_n,0]} = \boldsymbol{X}_{[t_{n-1},0]} \circ \boldsymbol{X}_{[t_n,t_{n-1}]}$ and $\tilde{\boldsymbol{X}}_{[t_n,0]} = \tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}$, hence

$$\boldsymbol{X}_{[t_n,0]} - \tilde{\boldsymbol{X}}_{[t_n,0]} = (\boldsymbol{X}_{[t_{n-1},0]} - \tilde{\boldsymbol{X}}_{[t_{n-1},0]}) \circ \boldsymbol{X}_{[t_n,t_{n-1}]} + \mathcal{O}(\boldsymbol{X}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}). \tag{4.34}$$

From our estimates in (4.14), we have that $\boldsymbol{X}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]} = \mathcal{O}(\Delta t \|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_\infty)$. Given that $\omega(\boldsymbol{x}, t_{n-1}) - \omega^{n-1}(\boldsymbol{x}) = \mathcal{O}(\mathcal{E}^{n-1})$, we have

$$\boldsymbol{X}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]} = \mathcal{O}(\Delta t \mathcal{E}^{n-1} + \Delta t^{p+1}), \tag{4.35}$$

where we incurred an extra order $p+1$ error from the Lagrange interpolation.

Therefore,

$$\tilde{\mathcal{E}}^n \leq \tilde{\mathcal{E}}^{n-1} + \mathcal{O}(\Delta t \mathcal{E}^{n-1} + \Delta t^{p+1}) \tag{4.36}$$

$$= \tilde{\mathcal{E}}^{n-1} + \mathcal{O}\left(\Delta t \tilde{\mathcal{E}}^{n-1} + \Delta t (t_n (\Delta x^2 \min(\Delta t, \Delta x^2 \Delta t^{-1}) + \Delta t^s)) + \Delta t^{p+1}\right), \tag{4.37}$$

which implies that

$$\tilde{\mathcal{E}}^n = \mathcal{O}\left(\Delta x^2 \min(\Delta t, \Delta x^2 \Delta t^{-1}) + \Delta t^s + \Delta t^p\right). \tag{4.38}$$

Together with (4.32), we obtain the desired error estimate. □

Remark 4.2.5. Equation (4.35) in fact omits a sampling error incurred when defining $\boldsymbol{u}^n_\epsilon$, due to sampling the vorticity at discrete points. In this method, we define ω^n_ϵ by convolution with a pullback mollifier instead of directly evaluating $\omega_0 \circ \boldsymbol{\mathcal{X}}^n$ at sample points. This allows us to justify a Fourier truncation at low number of modes, however, we incur an error term which is second order in the width of the mollifier. This is omitted from the analysis since the sampling grid for ω^n_ϵ is independent of the computational grid for the map, hence the sampling error can be controlled separately.

4.2.5 Convergence Tests

We provide here some numerical evidence for the error estimates derived above. We will more extensively test the full method in section 4.3.

We use a standard four-modes initial condition (4.48) to test the convergence. For the spatial error, we fix Δt at 1/512 using third order Lagrange interpolation in time and third order explicit Runge-Kutta for time-integration. We then vary the spatial grid size between 32 to 512. To test the error in the time variable, we fix the spatial grid at 1024 and vary Δt between 1/8 and 1/128. This time, we test both a second and third order Lagrange interpolant while keeping the same Runge-Kutta scheme as before. This aims to show the independence of the conservation error from the solution error. For all tests, we sample the vorticity on a 1024 grid and represent the stream function as a piecewise Hermite cubic interpolant on a 2048 grid. We run the simulation to time $t = 1$ and calculate the errors in the following quantities:

$$\text{Map error } := \|\boldsymbol{\mathcal{X}}^n - \boldsymbol{X}(\cdot, t_n)\|_\infty, \tag{4.39a}$$

$$\text{Vorticity error } := \|\omega^n - \omega(\cdot, t_n)\|_\infty, \tag{4.39b}$$

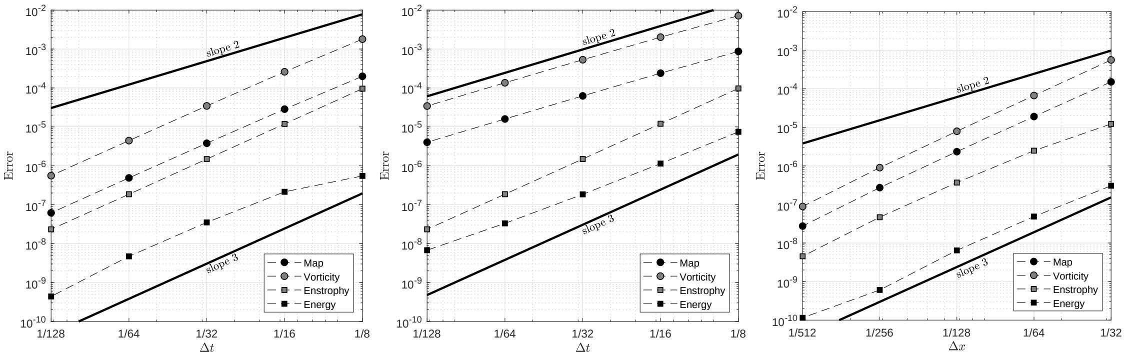
$$\text{Enstrophy conservation error } := \|\omega^n\|_{L^2}^2 - \|\omega_0\|_{L^2}^2, \tag{4.39c}$$

$$\text{Energy conservation error } := \|\boldsymbol{u}^n\|_{L^2}^2 - \|\boldsymbol{u}_0\|_{L^2}^2, \tag{4.39d}$$

where for the map error, the maximum error of the two coordinate functions is recorded.

Conservation errors are calculated directly, the sup-norm map and vorticity errors are estimated by comparing each result to the $\Delta x = 1/1024$, $\Delta t = 1/512$ test. The functions are evaluated on a 2048 grid, both the L^∞ and L^2 norms are approximated by their discrete variants on this grid. The results are shown in figure 4–1.

Figures 4–1a and 4–1b are both time convergence plots, the difference is that one uses a third order Lagrange interpolant for the definition of $\tilde{\boldsymbol{u}}$ and the other, second order. We see that as expected, the enstrophy conservation error is independent of the choice of Lagrange interpolation and is third order in both cases due to the use



(a) Time convergence test, 3^{rd} order Lagrange. $\Delta x = 1/1024$.

(b) Time convergence test, 2^{nd} order Lagrange. $\Delta x = 1/1024$.

(c) Spatial convergence test, 3^{nd} order Lagrange. $\Delta t = 1/512$.

Figure 4–1: Map and vorticity error and conservation errors of enstrophy and energy for the Characteristic Mapping method without remapping.

of RK3 integration for $\tilde{\boldsymbol{u}}$. The error on the map values and vorticity however, do depend on the accuracy of $\tilde{\boldsymbol{u}}$ and have third and second order convergence for the respective tests.

Figure 4–1c shows the convergence with respect to Δx. Our error estimates suggest a convergence between $\mathcal{O}(\Delta t \Delta x^2)$ and $\mathcal{O}(\Delta t^{-1} \Delta x^4)$. For a fixed Δt, this is between second and fourth order. This ambiguity comes from the GALS time stepping scheme. The grid data at time t_{n+1} are obtained by evaluating the time t_n Hermite interpolants at $\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}_g)$ for $\boldsymbol{x}_g$ a grid point; $\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}_g)$ is commonly called the “foot-point”. The interpolation error depends on the location of the foot-point relative to grids points. In each dimension, Hermite cubic interpolation errors scales quadratically with both closest grid points. This implies that the interpolation error is $\mathcal{O}(\Delta x^2 \Delta t^2)$ if $\Delta t \ll \Delta x$ and $\mathcal{O}(\Delta x^4)$ otherwise. This is consistent with the third order convergence we see in figure 4–1c.

The experiments in this section suggest that the CM method with Hermite cubic spatial interpolation, third order Lagrange time interpolation and RK3 time integration yields a globally third order method. This is to provide some support for the error estimates in section 4.2.4. In practice, since a precise representation of fine scale features in the velocity field does not contribute very much to the global dynamics and deformation of the domain, we use a coarse grid to represent short time characteristic maps in order to improve efficiency. In this regard, the remapping step presented in the next section will play an important role in maintaining an accurate resolution of the fine scale features in the deformation map generated by long term advection. We will then provide more numerical results and benchmark tests in section 4.3.

4.2.6 Adaptive Remapping and Arbitrary Resolution

In the absence of a viscosity term, solutions of Euler equations tend to develop arbitrarily small scale spatial features. As a result, a fixed grid for representing the characteristic map is only valid for a limited amount of time before spatial resolution needs to be increased. Changing the computational grid during simulations can be cumbersome and adversely affect the speed of all computations thereafter. We use instead a decomposition method based on the group property of the characteristic map mentioned in section 4.1.1. Numerically, the time t characteristic map can be constructed as the composition of several submaps of time subintervals:

$$\boldsymbol{\mathcal{X}}_B(\cdot, t) := \boldsymbol{\mathcal{X}}_{[T_1,0]} \circ \boldsymbol{\mathcal{X}}_{[T_2,T_1]} \circ \cdots \circ \boldsymbol{\mathcal{X}}_{[T_{m-1},T_{m-2}]} \circ \boldsymbol{\mathcal{X}}_{[t,Y_{m-1}]}, \tag{4.40}$$

for some subdivision $0 < T_1 < T_2 < \cdots < T_{m-1} < t$. Each of these submaps are computed using the CM method described in previous sections. We initialize $\boldsymbol{\mathcal{X}}_{[t,T_i]}$ with the identity map at $t = T_i$ and evolved until a remapping time $t = T_{i+1}$ which can be determined dynamically. Once the remapping time is reached, we store $\boldsymbol{\mathcal{X}}_{[T_{i+1},T_i]}$ in memory and start computing the map for the next subinterval.

Heuristically speaking, each of the subintervals $[T_i, T_{i+1}]$ should be short enough such that the grid used to discretize $\boldsymbol{\mathcal{X}}_{[T_{i+1},T_i]}$ can correctly represent the deformation generated by the velocity $\tilde{\boldsymbol{u}}$ in this interval.

In this implementation, we use the error in the Jacobian determinant

$$e^n_{\det} := \| \det \nabla \boldsymbol{\mathcal{X}}_{[t_n,T_i]} - 1 \|_\infty \tag{4.41}$$

as a measurement of the map quality, based on which we choose the remapping times. We pick an error threshold δ_{det} for the submaps. The i^{th} submap is initialized with the identity map. After each time step, we compute the Jacobian determinant of $\boldsymbol{\mathcal{X}}_{[T_i+n\Delta t,T_i]}$ at off-grid sample points. If for some n, the Jacobian error exceeds δ_{det} for the first time, we define $T_{i+1} := T_i + n\Delta t$ and store $\boldsymbol{\mathcal{X}}_{[T_{i+1},T_i]}$. The same process is repeated for the $i+1^{\text{st}}$ submap using T_{i+1} as initial time.

Although $\|\det \nabla\boldsymbol{\mathcal{X}}_{[t_n,T_i]} - 1\|_\infty < \delta_{\text{det}}$ does not constitute a proper bound on the error in $\boldsymbol{\mathcal{X}}_B$, we can use this as an approximate *a posteriori* error estimate for the gradient of the map compared to that of the map from the modified equation. Indeed, since $\|\boldsymbol{\mathcal{X}}_B - \tilde{\boldsymbol{X}}_B\|_\infty$ is the error from using the CM method on the velocity $\tilde{\boldsymbol{u}}$, it is globally third order when $\Delta x \sim \Delta t$. Assuming that $\nabla\boldsymbol{\mathcal{X}}_B - \nabla\tilde{\boldsymbol{X}}_B$ is small, we can justify the following first order expansion

$$\begin{aligned} e_{\text{det}} = \|\det \nabla\boldsymbol{\mathcal{X}}_B - \det \nabla\tilde{\boldsymbol{X}}_B\|_\infty &\approx \left\|\text{tr}\left(\nabla\tilde{\boldsymbol{X}}_B^{-1}(\nabla\boldsymbol{\mathcal{X}}_B - \nabla\tilde{\boldsymbol{X}}_B)\right)\right\|_\infty \\ &= \mathcal{O}(\|\nabla\boldsymbol{\mathcal{X}}_B - \nabla\tilde{\boldsymbol{X}}_B\|_\infty) \end{aligned} \tag{4.42}$$

Choosing the remapping times such that for each subinterval, we have $e_{\text{det}} < \delta_{\text{det}}$ implies that each submap has $\mathcal{O}(\Delta x \delta_{\text{det}})$ error with respect to $\tilde{\boldsymbol{X}}_B$ and hence volume preservation and enstrophy conservation error of the same order. In turn, since the error is 0 at the initial time and we remap at the first time step where this threshold is exceeded, we know that we can assume the error to be small enough to justify the above first order expansion (at least for all previous time steps).

This remapping technique is key in the accurate, dissipation-free resolution of the vorticity field. Qualitatively speaking, there are two types of errors in this method, one is dissipative in nature, and the other, "advective". Dissipative error refers to artificial diffusion (or diffusion-like) terms that we incur from spatial truncation of the solution. When we represent an evolving quantity on a fixed spatial grid, the high frequency features of the solution, namely those above the grid's Nyquist frequency, are lost. When these spatial truncations are directly applied to the Euler equations, we get artificial dissipation of ω or $\boldsymbol{u}$ resulting in loss of enstrophy or energy. In the case of Fourier-Galerkin truncation, the dissipative errors can resonate with the solution resulting in numerical artefacts and spurious oscillations [99].

Due to the discrete nature of numerical computations, truncation errors are inevitable. In the present method, the evolution of $\boldsymbol{\mathcal{X}}_B^n$ (and only $\boldsymbol{\mathcal{X}}_B^n$) contains a diffusive type error since at each GALS update step, the Hermite cubic interpolation consists of a 4th order averaging of grid values. The leading order error is a 4th order spatial derivative acting like a squared Laplacian. Over time, this accumulated averaging error artificially smooths out the map and resists fine scale deformations which might be present in the true solution. However, since $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}_B^n$, the error in the vorticity is not dissipative in nature. The vorticity is not directly obtained from the previous step ω^{n-1} and there is no averaging involved. Instead, the error occurs only at the evaluation of ω^n and is produced by evaluating ω_0 at a wrong position. In fact, since $\boldsymbol{X}_B$ and $\boldsymbol{\mathcal{X}}_B^n$ are both diffeomorphisms of U, there exists a diffeomorphism $\boldsymbol{\Psi}^n = (\boldsymbol{\mathcal{X}}_B^n)^{-1} \circ \boldsymbol{X}_{[t_n,0]}$ such that

$$\boldsymbol{X}_{[t_n,0]} = \boldsymbol{\mathcal{X}}_B^n \circ \boldsymbol{\Psi}^n, \tag{4.43}$$

The error for ω^n can then be seen as an advective error in the sense that

$$\omega(\boldsymbol{x}, t_n) = \omega^n(\boldsymbol{\Psi}^n(\boldsymbol{x})), \tag{4.44}$$

where ω on the left-hand side refers to the true solution.

This means that qualitatively speaking, the global dynamics of the solution are not obtained from a viscous approximation: the numerical fluid is still inviscid. We make an error on the position of the vortices, controlled by the error of the characteristic map. In particular, it is a straightforward consequence that the numerical solution has the correct L^∞-norm. Moreover, all L^p-norms for $1 \leq p < \infty$ are controlled by $\|\boldsymbol{\mathcal{X}}_B - \tilde{\boldsymbol{X}}_B\|_\infty$. Essentially, the CM method places the inevitable diffusive truncation error on the deformation map so that by composition with ω_0, the dissipative error in $\boldsymbol{\mathcal{X}}_B$ manifests itself in ω as an advective error, hence preserving the inviscid quality of the numerical solution.

Going back to the remapping routine, we apply the same principle. In limiting the length of the submap intervals $[T_i, T_{i+1}]$ by choosing a small δ_{det}, we limit the amount of artificial diffusion a single submap can accumulate. This prevents the dissipative error from smoothing out the map and smearing out the fine scale deformations generated by the advection: the global time map is constructed by composition of the short time submaps, also resulting in an advection type error for the map.

In practice, we can interpret the choice of δ_{det} in several ways. On one hand, since the use of a smoother $\boldsymbol{u}_\epsilon^n$ velocity can be seen as coarse scale discretization of the fluid velocity in the sense of the LAE-α and non-Newtonian fluid equations, we can view the choices of the remapping threshold δ_{det} as control on the artificial elasticity of the numerical flow due to spatial truncations. In this sense, the CM method does not make an error on the viscosity, rather it allows for some small controlled elasticity in the fluid. On the other hand, δ_{det} controls the error on the volume preserving property of the map. The numerical deformation map is not exactly volume preserving, hence the characteristic paths approximate those of a fluid that is slightly compressible. Therefore, the CM method avoids numerical dissipation by allowing for a small compressibility in particle paths. It is important to note however that the vorticity field is still advected and is not stretched by the compressibility relaxation of the characteristic map.

Lastly, one can also look at the spatial resolution of the remapping routine from the point of view of the gradient of the represented quantities. Heuristically speaking, the maximum gradient that can be accurately represented on a grid of cell width Δx is $\mathcal{O}(\Delta x^{-1})$, that is, the gradient scales roughly with N, the number of grid points per dimension. It follows that for an exponentially growing vorticity gradient, the required grid size to avoid excessive truncation errors grows exponentially also. For methods where the evolution of ω is carried out additively, i.e. methods of the type

$$\omega^{n+1} = \omega^n + \Delta t \partial_t \omega^n, \tag{4.45}$$

it implies that computations for $\partial_t \omega^n$ must be carried out on an exponentially growing grid.

On the other hand, ω is not evolved additively in the CM method, the gradient is instead generated by the characteristic map:

$$\nabla\omega(\cdot, t_n) = \nabla\omega_0 \nabla\boldsymbol{\mathcal{X}}_{[t_n,0]}. \tag{4.46}$$

Here, the exponential quantity is $\nabla\boldsymbol{\mathcal{X}}_B^n$. This growth is however a natural property of the characteristic map (the map being itself the exponential flow map of the backward time velocity). In fact, the semigroup decomposition (4.8) is the intrinsic generating process of the gradient just as multiplication is the generating process of the exponential function. We have

$$\nabla\boldsymbol{\mathcal{X}}_{[t,0]} = \prod_{j=1}^{m} \nabla\boldsymbol{\mathcal{X}}_{[\tau_j,\tau_{j-1}]}, \tag{4.47}$$

where the ℓ^2 operator norm $\|\nabla\boldsymbol{\mathcal{X}}_{[\tau_j,\tau_{j-1}]}\|_2$ of each submap gradient is expected to scale exponentially with $\Delta\tau_j = \tau_j - \tau_{j-1}$. Therefore, by appropriately choosing the remapping criterion, one can make $\Delta\tau$ small enough that the gradient of each submap is bounded of order $\mathcal{O}(1+\Delta\tau)$, hence representable on a coarse grid. This means that through the semigroup property, we can generate exponential growth in the vorticity gradient without having to do computations on an exponentially growing grid. As we shall see in the next section, this yields a computationally efficient method which captures arbitrarily fine scales and arbitrarily large gradients in the solution.

4.3 Numerical Tests

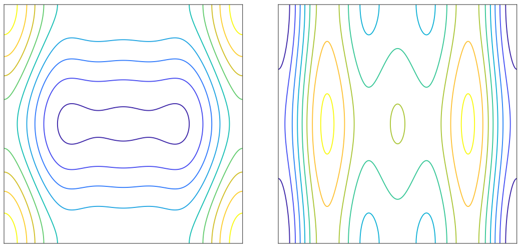
The simulations in this section were carried out on Matlab on a computer with an Intel Core i5-2320 3.00GHz 4 cores processor and 8GB of RAM. For the current simulations, we have not used any parallelization routines. However, almost all the computational time is spent on Hermite interpolations. Parallel and GPU implementation of these operations are standard and could drastically improve the speed of the simulations. Parallelization and application of domain decomposition techniques may be of interest for future work. The larger computations in section 4.3.4 are performed using a cluster computer.

4.3.1 “4-modes” test

We test the CM method for 2D incompressible Euler using the “4-modes” initial condition. We use the tests performed in [97] as reference. The initial vorticity is given by:

$$\omega_0(x, y) = \cos(x) + \cos(y) + 0.6\cos(2x) + 0.2\cos(3x), \tag{4.48}$$

This flow can roughly be characterised as two vortices of opposite signs partitioning a flat torus. The contour plot of ω_0 and $\Delta\omega_0$ are shown in figure 4–2, all contour plots in this paper show 9 isolines at values uniformly distributed in the range of the plotted data. For this flow, we used a 128^2 grid for the evolution of the maps with a 512^2 grid to represent $\tilde{\boldsymbol{u}}$. The time step Δt was set at $1/32$ and the remapping determinant error threshold to 10^{-4}. The simulation was run to times 3.5 and 4 for the enstrophy spectrum plot in figure 4–14, and for several times until $t = 8$ for the long time simulation in figures 4–3 an 4–4.



(a) ω_0 (b) $\Delta\omega_0$

Figure 4–2: Contour plot of the 4-modes initial vorticity and its Laplacian. Isolines are shown for 9 values uniformly distributed in the range of each functions.

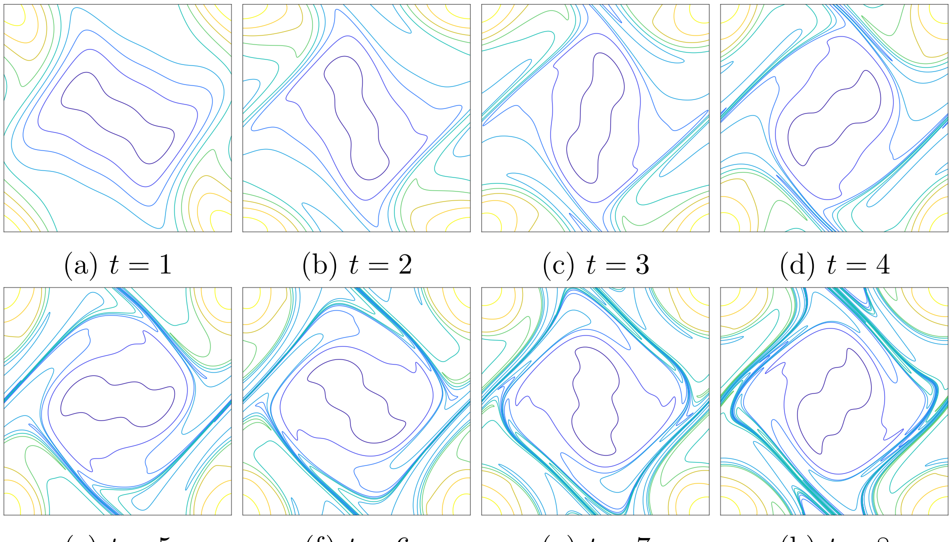
The same tests were performed in [97] up to time $t = 5$ using the Cauchy-Lagrange method of various truncation order in time on spatial grids up to 8192^2. This was necessary due to the presence of large high frequency components in the

t	1	2	3	4	5	6	7	8
Number of remaps	1	4	14	30	47	65	88	105
Total CPU time	20 s	41 s	66 s	100 s	145 s	202 s	274 s	359 s

Table 4–1: Number of remaps and CPU times for the 4-modes test.

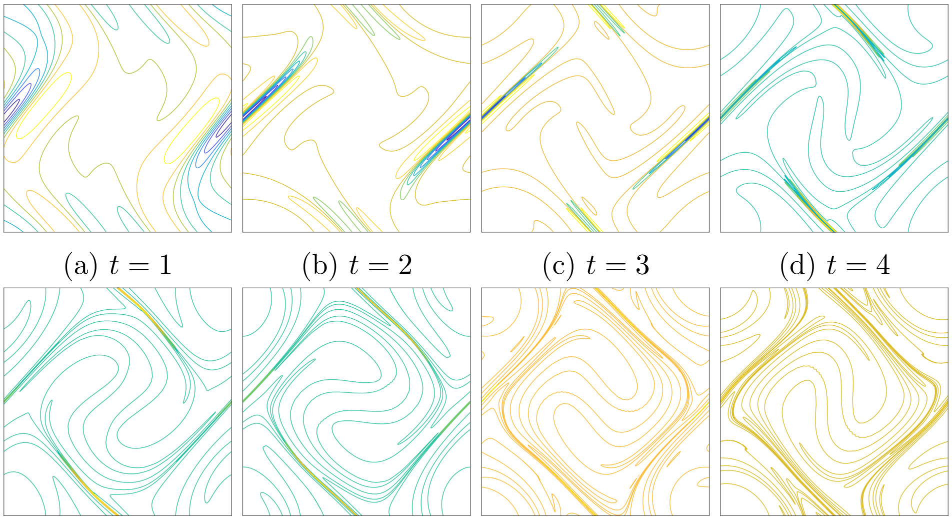
vorticity at large times and to the necessary anti-aliasing routines in Fourier pseudo-spectral methods. On the other hand, one can justify using a coarser 128^2 grid for the submap evolution in the CM method since the submaps are remapped and reset to identity before large high frequency features can form. Furthermore, these maps are evolved using the velocity field $\tilde{\boldsymbol{u}}$ which, by the Biot-Savart law, has a faster decay in its Fourier coefficients than the vorticity field.

Figures 4–3 and 4–4 show the contour plots of the vorticity and its Laplacian at length 1 time intervals between 0 and 8. The characteristic maps are computed on a coarse gird, we only use a fine grid sampling of the vorticity to generate the figures. Table 4–1 shows the number of remaps and total computational times required to reach the various plotting times.


(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

(e) $t = 5$ (f) $t = 6$ (g) $t = 7$ (h) $t = 8$

Figure 4–3: Contour plot of the vorticity using 128^2 grid for $\boldsymbol{\mathcal{X}}^n$, 512^2 grid for representing ψ^n, $\Delta t = 1/32$ and $\delta_{\text{det}} = 10^{-4}$.



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

(e) $t = 5$ (f) $t = 6$ (g) $t = 7$ (h) $t = 8$

Figure 4–4: Contour plot of the Laplacian of the vorticity using 128^2 grid for $\boldsymbol{\mathcal{X}}^n$, 512^2 grid for representing ψ^n, $\Delta t = 1/32$ and $\delta_{\text{det}} = 10^{-4}$.

From figures 4–3 and 4–4, we see that for larger times, the flow forms very thin vortex sheets where the two vortices meet. These regions have high vortex gradient and present increasingly fine scale features. For methods employing a fixed grid to represent the solution ω^n, these fine features will eventually become smaller than the grid resolution after which they are lost. This can be interpreted as numerical diffusion or hyper-diffusion associated to the grid size and eventually destroys sharp features of the solution. In the case of conservative high resolution methods such as [97], the truncation of high frequency modes can provoke resonance in the solution leading to a type of spurious oscillations called "Tygers": essentially, for a given spatial resolution the numerical solution will reach a time after which numerical artefacts become visible and the solution becomes unstable.

The CM method circumvents this issue by obtaining the solution as a "rearrangement" of the initial condition using the backward map. The lack of spatial resolution due to the discrete representation of the map does not result in a dissipative error in the vorticity. This can be observed in figures 4–3 and 4–4 where we

t	1	2	3	4
Enstrophy	$1.35 \cdot 10^{-6}$	$2.75 \cdot 10^{-6}$	$4.39 \cdot 10^{-6}$	$6.13 \cdot 10^{-6}$
Energy	$-3.21 \cdot 10^{-8}$	$-4.14 \cdot 10^{-8}$	$-5.17 \cdot 10^{-8}$	$4.66 \cdot 10^{-8}$
t	5	6	7	8
Enstrophy	$7.90 \cdot 10^{-6}$	$9.76 \cdot 10^{-6}$	$1.17 \cdot 10^{-5}$	$1.37 \cdot 10^{-5}$
Energy	$-3.51 \cdot 10^{-7}$	$-1.64 \cdot 10^{-6}$	$-3.98 \cdot 10^{-6}$	$-7.51 \cdot 10^{-6}$

Table 4–2: Conservation errors for the 4-modes test using the CM method.

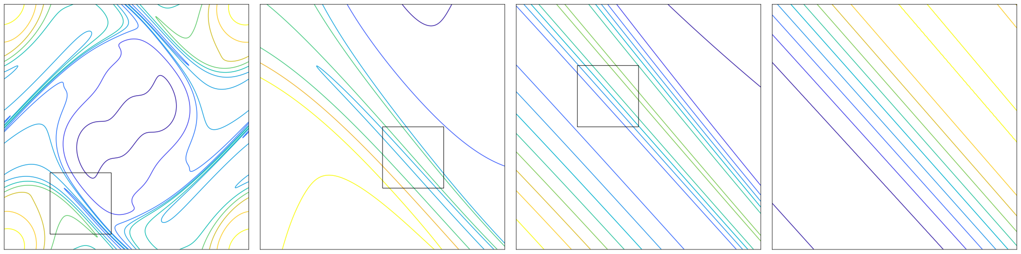
can see that for large times, the solution still contains fine scale features and there are no spurious oscillations. There is however some numerical dissipation in $\boldsymbol{\mathcal{X}}$ akin to an artificial elasticity term. From the point of view of vorticity, the effect of this dissipation in the map is that the vorticity will be transported along a less violent flow. This error is controlled by the remapping routine. Normally, if we evolve a single characteristic map on a 128^2 grid, the accumulated diffusive error will prevent sharp deformations to form. Using the remapping method with $\delta_{\text{det}} = 10^{-4}$ we limit the amount of diffusive error in each submap. The global map is constructed from the composition of the submaps and hence is able to represent large shears and the formation of thin vortex sheets. Indeed, we can see from the results that the vorticity develops scales much finer than the 128^2 grid used for the submap evolution. These scales were absent in the initial condition and are generated from the domain deformation represented by the composition of several submaps.

Another advantage of the remapping routine is that it offers some control over the growth of the enstrophy conservation error. Indeed, since each additional submap transports the vorticity at the previous remapping time, we incur the conservation error in corollary 1 with respect to the enstrophy at the previous remapping. This means that the enstrophy error accumulates additively when remapping. The error from each submap is controlled through the choice of the remapping tolerance δ_{det}, thereby providing better long term conservation. The enstrophy and energy conservation errors are shown in table 4–2.

Compared to a direct grid based representation of the vorticity, this growth is much slower. Indeed, in [97], using the 8^{th} order Cauchy-Lagragian method on a

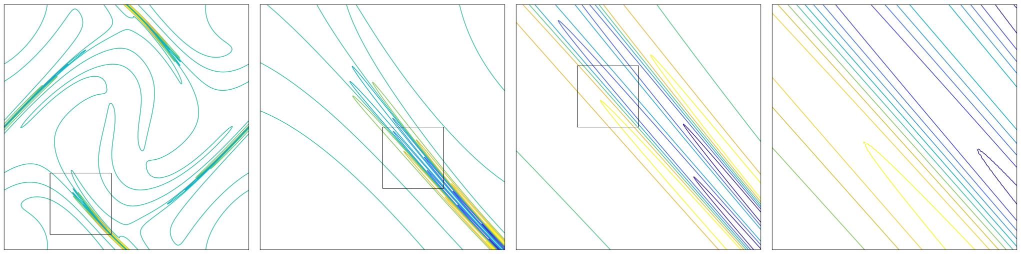
1024^2 grid, the enstrophy error increases from 10^{-14} to 10^{-12} to 10^{-6} for times 1, 3 and 5, whereas for the CM method, the enstrophy error seems to grow linearly with time.

The remapping method combined with the functional representation of the characteristic map offers the possibility of arbitrary spatial resolution of the solution. Indeed, since the interpolation structure of the submaps implies that the global map $\boldsymbol{\mathcal{X}}_{[t,0]}$ can be readily evaluated anywhere in the domain, it follows that the vorticity of any quantity transported by the flow can also be evaluated anywhere. With the accuracy control provided by the remapping method, this means that solutions can be faithfully represented at an arbitrary resolution. We illustrate this property by gradually zooming into the solution at times 4 and 8.



(a) $[0,1]\times[0,1]$ (b) $[\frac{3}{16},\frac{7}{16}]\times[\frac{1}{16},\frac{5}{16}]$ (c) $[\frac{5}{16},\frac{6}{16}]\times[\frac{2}{16},\frac{3}{16}]$ (d) $[\frac{21}{64},\frac{22}{64}]\times[\frac{10}{64},\frac{11}{64}]$

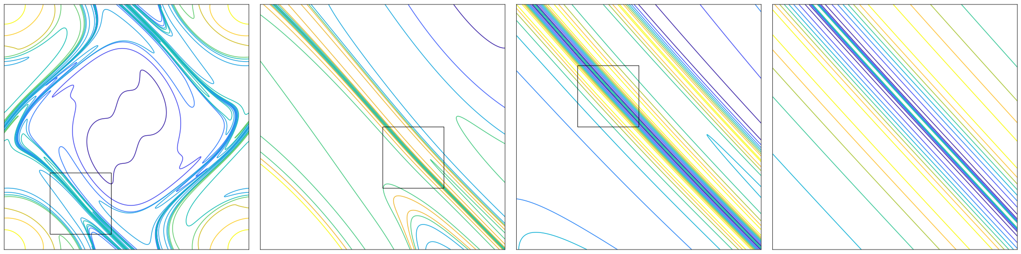
Figure 4–5: Gradual 64× zoom on the vorticity at $t = 4$.



(a) $[0,1]\times[0,1]$ (b) $[\frac{3}{16},\frac{7}{16}]\times[\frac{1}{16},\frac{5}{16}]$ (c) $[\frac{5}{16},\frac{6}{16}]\times[\frac{2}{16},\frac{3}{16}]$ (d) $[\frac{21}{64},\frac{22}{64}]\times[\frac{10}{64},\frac{11}{64}]$

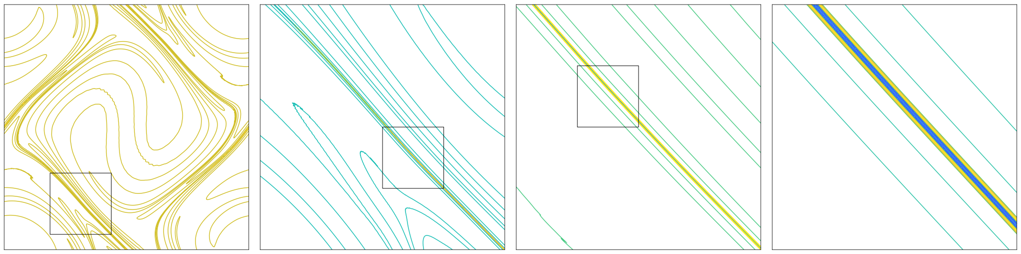
Figure 4–6: Gradual 64× zoom on the Laplacian of the vorticity at $t = 4$.

Figures 4–5, 4–6, 4–7 and 4–8 demonstrate the arbitrary spatial resolution provided by the characteristic map. Each zoomed plot is sampled with the same number



(a) $[0,1]\times[0,1]$ (b) $[\frac{3}{16},\frac{7}{16}]\times[\frac{1}{16},\frac{5}{16}]$ (c) $[\frac{5}{16},\frac{6}{16}]\times[\frac{2}{16},\frac{3}{16}]$ (d) $[\frac{21}{64},\frac{22}{64}]\times[\frac{10}{64},\frac{11}{64}]$

Figure 4–7: Gradual 64× zoom on the vorticity at $t = 8$.



(a) $[0,1]\times[0,1]$ (b) $[\frac{3}{16},\frac{7}{16}]\times[\frac{1}{16},\frac{5}{16}]$ (c) $[\frac{5}{16},\frac{6}{16}]\times[\frac{2}{16},\frac{3}{16}]$ (d) $[\frac{21}{64},\frac{22}{64}]\times[\frac{10}{64},\frac{11}{64}]$

Figure 4–8: Gradual 64× zoom on the Laplacian of the vorticity at $t = 8$.

of sample points, providing the same image resolution on gradually smaller subdomains. We observe that when we zoom in, we recover additional small scale features not visible on the original plot. The undersampling from the $[0,1]\times[0,1]$ plot fails to represent the complexity of the contour lines inside the thin vortex sheet (see figures 4–7a v.s. 4–7d). These features are recovered when using a finer sampling. This in fact shows that the solution provided by the CM method is not bound to a fixed set of sample points. Whereas in most methods, once a grid is chosen, any detail finer than this grid is lost, the CM method does not compute the solution on a fixed grid, rather it provides an algorithm to sample the vorticity field $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}^n$ defined as a function over the whole domain. This means that the solution can be evaluated anywhere, providing arbitrary spatial resolution. In practice, in case ω_0 is given numerically, this implies that we maintain the same resolution as that of ω_0 throughout the entire simulation: there is no loss of spatial features.

4.3.2 Random initial conditions

In this section, we perform the same tests as in section 4.3.1 on a randomly generated initial condition. The procedure to generate the random initial condition is given in [99]. In short, the vorticity is defined in Fourier space, which is divided into lattice shells, each containing all modes $\boldsymbol{k} = (k_1, k_2)$ such that $\sqrt{k_1^2 + k_2^2} \in [K, K+1)$. The K^{th} shell contains $N(K)$ modes, and for each of these modes, we assign a vorticity Fourier coefficient $\hat{\omega}_{\boldsymbol{k}}$ of fixed modulus $2K^{7/2}\exp(-K^2/4)/N(K)$ and a phase picked randomly from $[0, 2\pi)$ with uniform distribution. This guarantees that the total vorticity in the K^{th} shell decays like $2K^{7/2}\exp(-K^2/4)$. Further, to ensure that the vorticity is real, opposite wave vectors (i.e. $\hat{\omega}_{\boldsymbol{k}}$ and $\hat{\omega}_{-\boldsymbol{k}}$) are given opposite phases so that the resulting Fourier expansion is Hermitian. The same test was performed in [97] until $t = 1$ using the Cauchy-Lagrange methods with 2048^2 spatial Fourier modes.

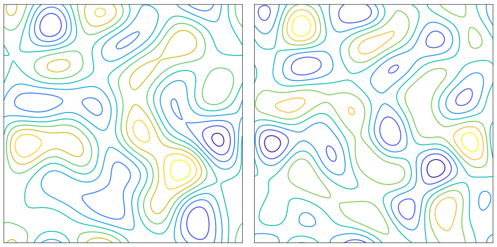

(a) ω_0 (b) $\Delta\omega_0$

Figure 4–9: Contour plot of the random initial vorticity and its Laplacian.

Since the CM method does not work directly with the Fourier transform of the initial vorticity, we defined our initial ω_0 as follows: we first generate $\hat{\omega}_{\boldsymbol{k}}$ as described above, we then sample the Fourier series on a 512^2 grid to obtain a Hermite cubic interpolant which we use as ω_0. We used total of 32 lattice shells, i.e. $K = 0, 1, \ldots 32$; in fact, due to the prescribed decay rate of the coefficients, $|\hat{\omega}_{\boldsymbol{k}}|$ is already well below machine precision for $|\boldsymbol{k}| = 32$ and is below machine underflow for 64. Using more shells would have no consequence on our ω_0. The initial vorticity and its Laplacian are shown in figure 4–9. We ran the CM method on this initial condition using a 256^2

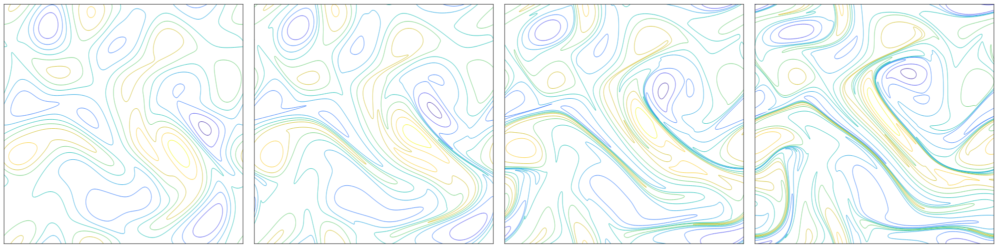
t	0.5	1	1.5	2
Number of remaps	3	18	38	69
Total CPU time	233 s	499 s	822 s	1226 s

Table 4–3: Number of remaps and CPU times for the random initial condition test using the CM method.

t	0.5	1	1.5	2
Enstrophy error	$-2.40 \cdot 10^{-6}$	$1.36 \cdot 10^{-6}$	$5.95 \cdot 10^{-6}$	$9.36 \cdot 10^{-6}$
Energy error	$-7.74 \cdot 10^{-7}$	$-4.71 \cdot 10^{-6}$	$-3.79 \cdot 10^{-5}$	$-9.36 \cdot 10^{-5}$

Table 4–4: Conservation errors for random initial condition test using the CM method.

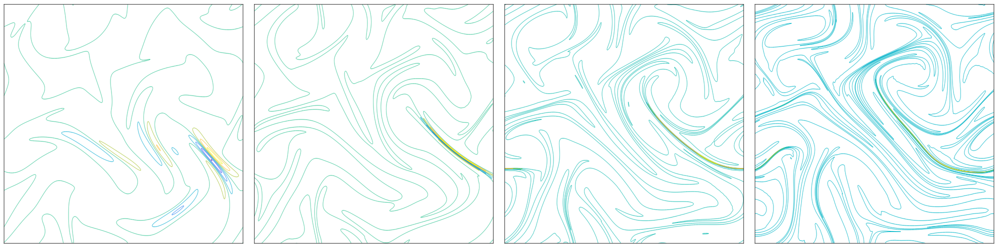
grid for the map, 1024^2 grid to represent ψ^n, $\Delta t = 1/64$ and $\delta_{\text{det}} = 10^{-4}$. Contour plots of the vorticity field and of the its Laplacian are shown at 0.5 time intervals in figures 4–10 and 4–11. The number of remaps and CPU times required are shown in table 4–3.



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$ (d) $t = 2$

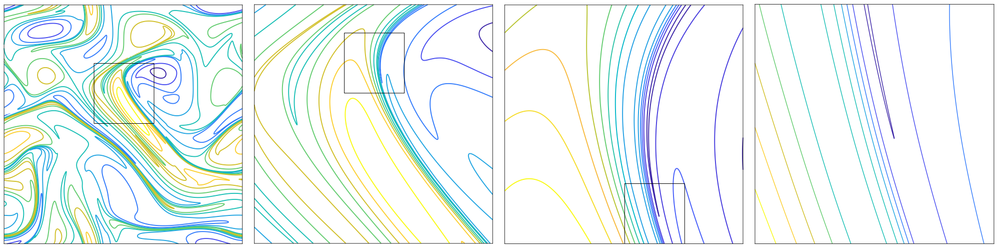
Figure 4–10: Contour plot of the vorticity for the random initial condition test using 256^2 grid for $\boldsymbol{\mathcal{X}}^n$, 1024^2 grid for representing ψ^n, $\Delta t = 1/128$ and $\delta_{\text{det}} = 10^{-4}$.

As in the 4-modes test, we also observe slow growth in conservation errors for the random initial data in table 4–4. In comparison, the enstrophy error for the 8th order, 2048^2 harmonics Cauchy-Lagrangian method grows from 10^{-14} to 10^{-13} to 10^{-8} for times 0.2, 0.6 and 1.



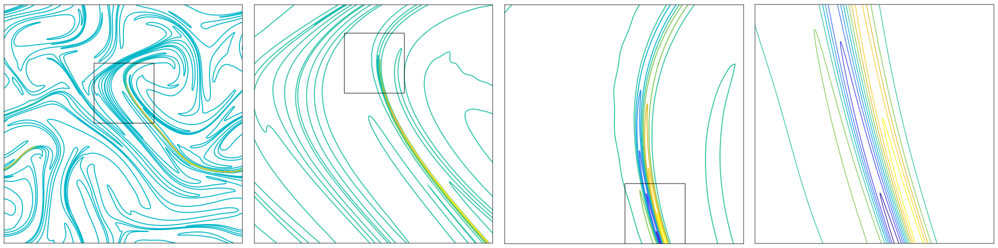
(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$ (d) $t = 2$

Figure 4–11: Contour plot of the Laplacian of the vorticity for the random initial condition test using 256^2 grid for $\boldsymbol{\mathcal{X}}^n$, 1024^2 grid for representing ψ^n, $\Delta t = 1/128$ and $\delta_{\text{det}} = 10^{-4}$.



(a) $[0, 1] \times [0, 1]$ (b) $[\frac{3}{8}, \frac{5}{8}] \times [\frac{1}{2}, \frac{3}{4}]$ (c) $[\frac{15}{32}, \frac{17}{32}] \times [\frac{21}{32}, \frac{23}{32}]$ (d) $[\frac{1}{2}, \frac{33}{64}] \times [\frac{21}{32}, \frac{43}{64}]$

Figure 4–12: Gradual 64× zoom on the vorticity at $t = 2$.



(a) $[0, 1] \times [0, 1]$ (b) $[\frac{3}{8}, \frac{5}{8}] \times [\frac{1}{2}, \frac{3}{4}]$ (c) $[\frac{15}{32}, \frac{17}{32}] \times [\frac{21}{32}, \frac{23}{32}]$ (d) $[\frac{1}{2}, \frac{33}{64}] \times [\frac{21}{32}, \frac{43}{64}]$

Figure 4–13: Gradual 64× zoom on the Laplacian of the vorticity at $t = 2$.

4.3.3 Spatial resolution

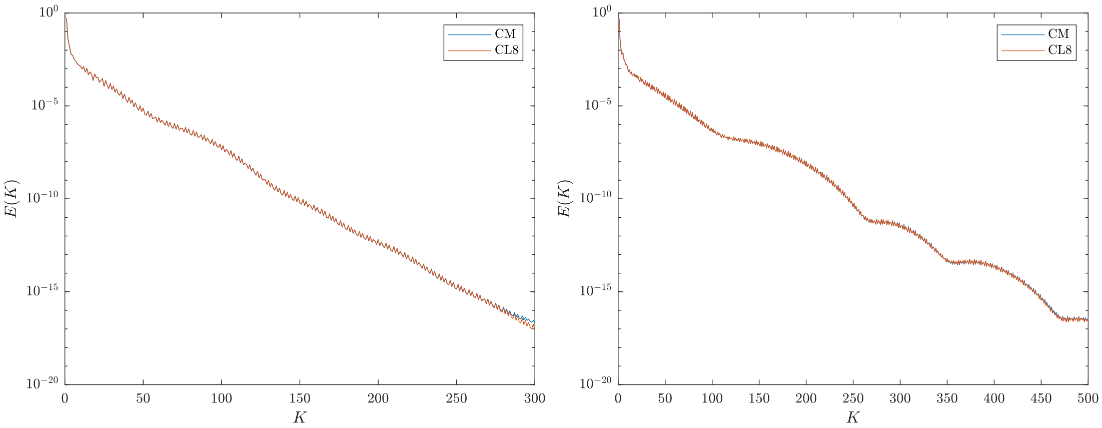
The vorticity solutions in both tests in this section are observed to have increasingly finer spatial features as time progresses. One way to quantify the evolution of the spatial scales is through the Fourier expansion of the solution, in particular, we look at the decay of the magnitudes of the high frequency coefficients. This can be seen from the enstrophy spectrum obtained by integrating the square of vorticity over circular shells in Fourier space. That is, let $\hat{\omega}_{\boldsymbol{k}}$ be the $\boldsymbol{k} = (k_1, k_2)$ coefficient of the Fourier transform of ω and denote $|\boldsymbol{k}| = \sqrt{k_1^2 + k_2^2}$, we have

$$E_\omega(K) := \frac{1}{2} \sum_{K \leq |\boldsymbol{k}| < K+1} |\hat{\omega}_{\boldsymbol{k}}|^2. \tag{4.49}$$

We compare the enstrophy spectrum from the CM method to that of Cauchy-Lagrangian (CL8) method presented in [97]. Figure 4–14 shows the overlay of the vorticity spectra obtained from both methods. The vorticity fields are sampled at times 3.45 and 3.95 (due to the nature of the time step in the CL8 method, the times presented in [97] did not land exactly on $t = 3.5$ and 4).

We see that in figure 4–14, the enstrophy spectra obtained from the CM method matches almost exactly, up to double precision, the high-fidelity results obtained from the 8096^2 harmonics Cauchy-Lagrangian simulation (The curves are on top of each other. The plots are provided in vector-graphics format, zooming in the tail of the spectrum shows some discrepancies between the two curves. This is in large part due to the two simulations not having exactly the same final times). This suggests that small time deformations can be accurately represented on a coarse 128^2 grid and the fine scale global time deformations can be reconstructed by the composition of the submaps without loss of resolution.

One measurement of the asymptotic decay of the Fourier coefficients is the radius of analyticity. The decay rate of the enstrophy spectrum at high frequency modes indicates the spatial scales present in the solution. Asymptotically, the decay of the



(a) $t \approx 3.5$ (b) $t \approx 4$

Figure 4–14: Decay in the enstrophy spectrum at times 3.5 and 4[1] .

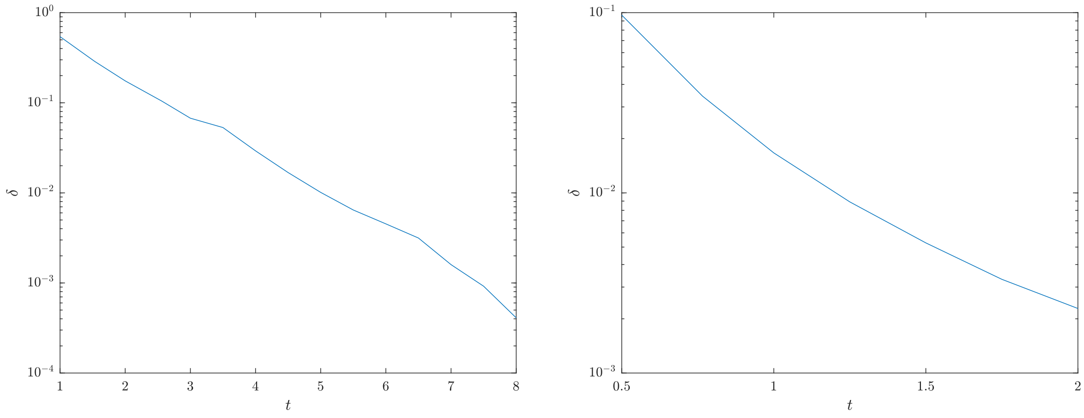
enstrophy spectrum is typically

$$E_\omega(K) \sim K^\alpha e^{-2\delta K}. \tag{4.50}$$

The rate of the exponential, δ, is the radius of analyticity and governs the spatial truncation error. For a grid which resolves a maximum frequency of $k_{\max}$, the spatial truncation error scales like $e^{-\delta k_{\max}}$.

Figure 4–15 shows the evolution of the radius of analyticity in time for both numerical tests. The radius is estimated at various times by taking a least-squared fit of the logarithm of the tail of the enstrophy spectrum $\log(E_\omega(K))$ with respect to the quantities $log(K)$, K and 1; we extract δ from the fitted coefficient for K. We see that the reduction in the radius of analyticity is exponential. This implies that in order to maintain a certain level of spatial truncation, the maximum resolved frequency $k_{\max}$ must grow exponentially. In particular, for the 4-modes test, at time 8, a

[1] The Cauchy-Lagrangian method employs variable length time steps. The enstrophy spectrum presented in [97] are computed at the last time step before reaching times 3.5 and 4. The final times turned out to be approximately 3.45 and 3.95. In this figure, the final times for the CM method are taken to be exactly 3.45 and 3.95.



(a) 4-modes

(b) Random initial condition

Figure 4–15: Radius of analyticity $\delta(t)$ vs time.

grid size of order 10^4 would be needed to properly resolve the solution. Carrying out computations with traditional methods on such grids would be difficult on a personal-use computer. We see that CM method allows us to evolve the solution for long times without having to use such large grids: through the submap decomposition, only local time coarse grids computations are required, the fine scale details can be recovered by the composition of the submaps. The CM method in fact dynamically adapts to the spatial resolution necessary to the problem. Through the remapping process, the available numerical resolutions autonomously grows as the spatial features in the solution increase.

4.3.4 Illustration of the Arbitrary Subgrid Resolution

Finally, in this section we provide an illustration of the power of the semigroup decomposition approach in achieving high subgrid resolution of the solution. For this, we simulate a 2 vortex merger problem. We use two identical Gaussian blobs of variance 0.07 placed 0.3 apart in a periodic domain of width 1 (see figure 4–16). The two vortices both have clockwise spins and are expected to start spinning around each other and almost merge into a single vortex blob. Due to the lack of viscosity, the vortices do not become a single vortex and will generate instabilities as time goes on.

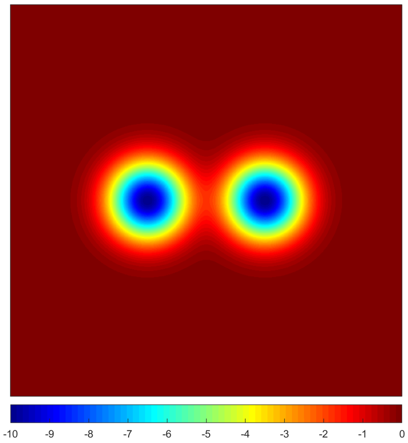


Figure 4–16: Initial vorticity for the vortex merger simulation.

The fine scale features produced by this flow requires a high amount of spatial resolution to evolve and represent. We performed the simulation using the CM method on a 512 grid for the map with time step 1/128 until $t = 20$. The results are shown in figures 4–17 and 4–18.

The final frame (figure 4–17) at time 20 is obtained using a composition of 605 submaps. This allows for the representation of a tremendous amount of fine scale subgrid structures. To illustrate this, we take a gradual zoom towards the position $(x, y) = (13/32, 13/32)$ in the last frame. Figure 4–19 shows the zoomed view on the $t = 20$ vorticity field. Each subfigure is obtained by evaluating the submap compositions on the subdomain corresponding to the zoomed view. Since the characteristic map has a functional definition, we can use the same number of sample points to generate each picture, therefore obtaining high resolution images of arbitrarily small regions in the domain. For instance, figure 4–19l shows the vorticity field in a region of size $1/8192 \times 1/8192$. The image is generated using 768^2 sample points, providing an accurate depiction of the details seen at the fine scale level.

The numerical experiments in section 4.3 showcase several advantageous properties of the CM method for the 2D incompressible Euler equations. Firstly, the

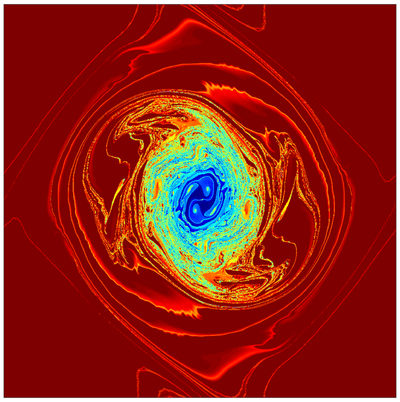
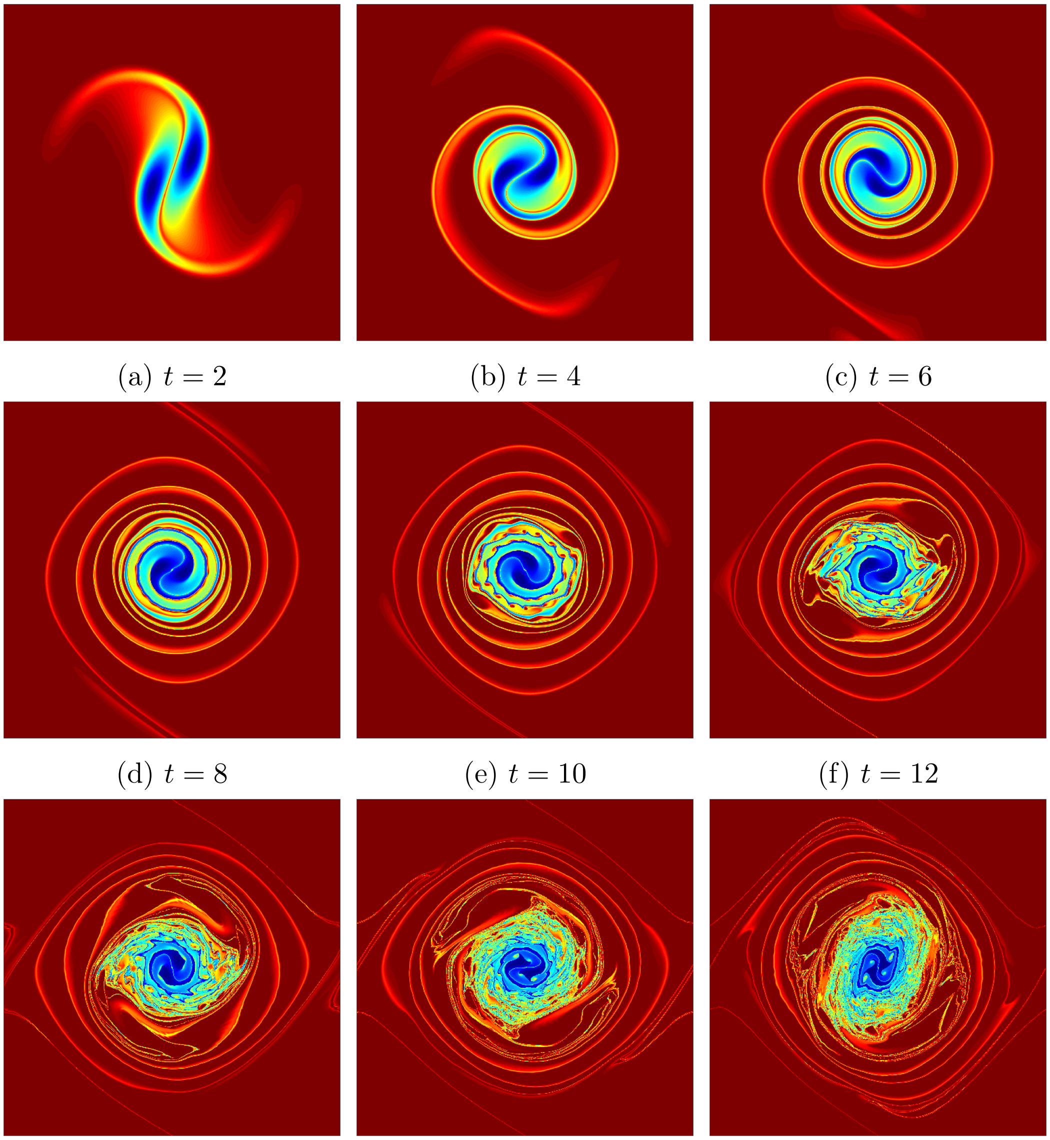


Figure 4–17: Vorticity field at time $t = 20$

submap decomposition using the group property of the characteristic map allows for quick and accurate computations on a coarse grid, circumventing the usual requirement of increasing spatial resolution due to exponential vorticity gradient growth. As evidence, solutions from a 128^2 grid CM solver achieves the same enstrophy spectrum as an 8192^2 grid direct vorticity solver (see figure 4–14). Furthermore, due to the volume preserving property of the characteristic map, the CM method achieves high accuracy enstrophy conservation for all times. Whereas other methods experience a spike in enstrophy error when the vorticity fields becomes complicated, the enstrophy conservation in CM is independent of the current time vorticity and is a direct result of volume preservation. This allows the enstrophy error to grow only linearly in time, regardless of the complexity of the vorticity field. Lastly, the functional definition of the numerical vorticity through composition with the backward map allows for an arbitrary spatial resolution of the solution. Furthermore, the submap decomposition generates the correct scales of gradients, ensuring that the increasing fine scale features in the vorticity solutions are properly represented as the resolution increases. This property is evidenced in figures 4–5, 4–6, 4–7, 4–8 and more thoroughly in 4–19 in section 4.3.4, where we zoom in on the solution to show the arbitrary spatial resolution achieved by the CM method.

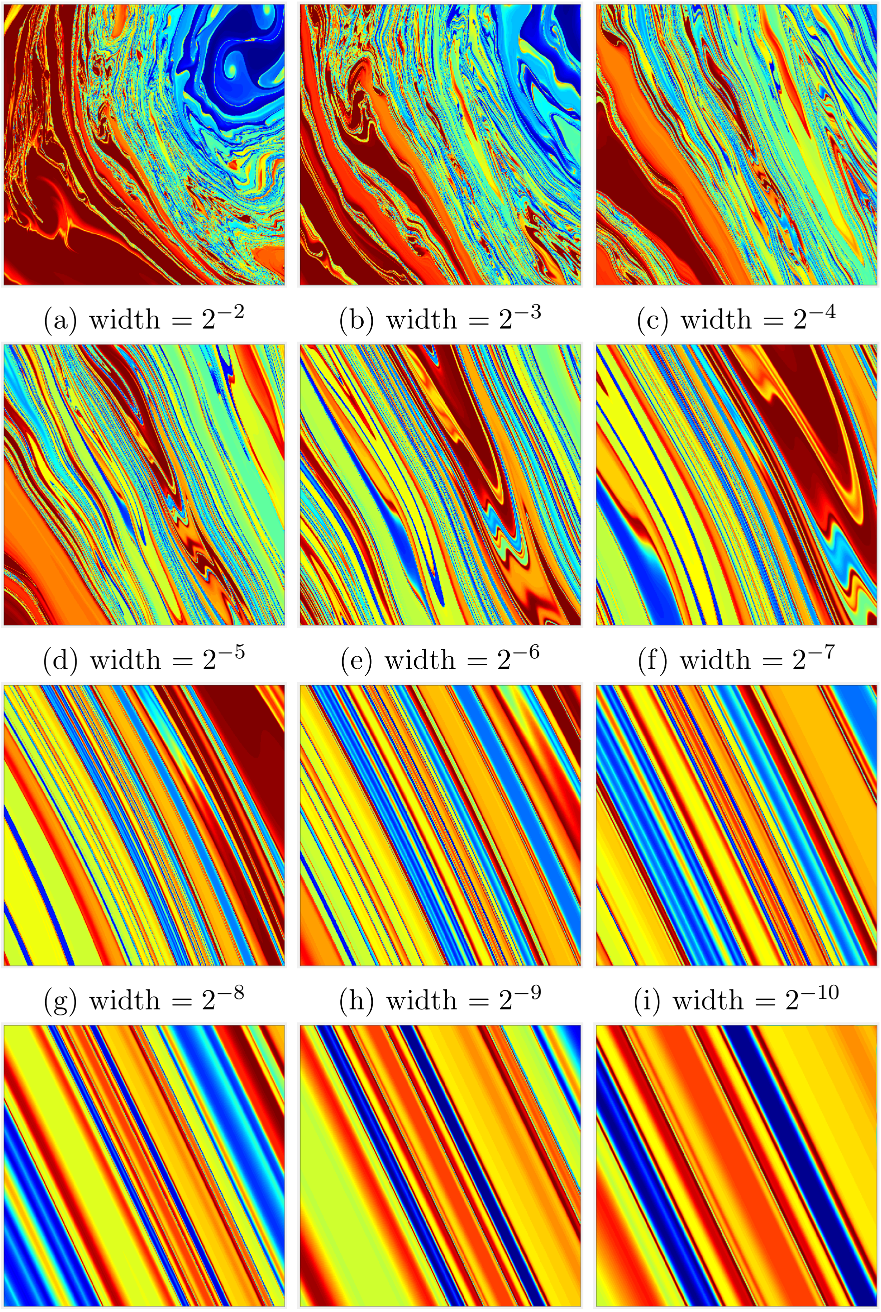


(a) $t = 2$ (b) $t = 4$ (c) $t = 6$

(d) $t = 8$ (e) $t = 10$ (f) $t = 12$

(g) $t = 14$ (h) $t = 16$ (i) $t = 18$

Figure 4–18: Evolution of the vorticity field in the vortex merger simulation.



(a) width = 2^{-2} (b) width = 2^{-3} (c) width = 2^{-4}

(d) width = 2^{-5} (e) width = 2^{-6} (f) width = 2^{-7}

(g) width = 2^{-8} (h) width = 2^{-9} (i) width = 2^{-10}

(j) width = 2^{-11} (k) width = 2^{-12} (l) width = 2^{-13}

Figure 4–19: Gradual zoom on the last frame, each subfigure is a 2× zoom on the previous.

4.4 Concluding Remarks for Chapter 4

In this chapter, we have presented the CM method for solving the 2D incompressible Euler equations. This method is novel in that it solves for the deformation map generated by the fluid flow and captures the geometry of the problem; all evolved quantities of interest can be obtained from this transformation. As a result, this scheme is characterized by the arbitrarily fine subgrid resolution it provides on the solutions and a lack of artificial dissipation. Several key observations have lead to the development of this method. Firstly, the arbitrarily fine scales typically generated by an inviscid flow motivated a functional representation of the vorticity field through pullback by the characteristic map. This approach not only preserves fine scales but more importantly avoids spatial truncations of the vorticity field, hence eliminating artificial dissipation. These properties are demonstrated in the tests in section 4.3, in particular in the zoomed view of the solutions. Secondly, the possible exponential growth in the vorticity gradient lead to the use of the group structure of the flow maps to decompose the characteristic map: exponential growth can be generated by a composition of maps of fixed resolution. Lastly, the assumption that the dynamics of the fluid is mainly governed by the large scale low frequency features of the velocity allowed us to carry out the characteristic map computations on a coarse grid, improving the efficiency of the method. We drew a parallel between the use of coarse scale velocity and the Lagrangian-Averaged Euler equations. The method in this chapter relies heavily on the fact that the vorticity in 2D is a scalar function satisfying a homogeneous advection equation under the velocity field $\boldsymbol{u}$. This does not generalize directly to the 3D case due to the extra vorticity stretching term. In order to keep the numerical properties of the CM method for the 3D case, we need to exploit the geometric structure of the Euler equations.

CHAPTER 5
Three-dimensional incompressible Euler Equations

The study of three-dimensional incompressible perfect fluids through the Euler equations is one of the oldest and most challenging problems in pure and applied mathematics. In contrast to the 2D case where the global existence and uniqueness of the solution have been shown [10], the question of finite-time blow-up in the solutions of the 3D incompressible Euler equations with smooth initial data is a notoriously difficult open problem and is related to the Clay institute Millenium problem on the Navier-Stokes equations [41]. Efficient and accurate numerical schemes are useful for supporting our understanding of perfect fluids, and various methods have been used to provide evidence of finite-time singularity [6, 20, 52, 60, 72, 86, 120] or nonsingular super-exponential growth in the vorticity [19, 61, 98]. However, the high-dimensionality and the rapid growth in the solution and its gradient increase numerical difficulties and requires the use of specialized algorithms. In particular, for Eulerian methods, the Courant-Friedrichs-Lewy (CFL) constraints the size of the time steps with respect to the spatial discretization scales, effectively requiring a computational complexity of no lower that N^4 for simulations with N grid points per spatial dimension. This is further compounded by the need for high-resolution computational grids in order to limit the effects of artificial dissipation. With the current computational technology, pseudo-spectral simulations with up to 12288^3 grid points have been performed [67].

In this chapter, we present a novel geometric method for the 3D incompressible Euler equations in its vorticity form. This is based on the 2D incompressible Euler equations using the Characteristic Mapping method in the previous chapter, which we extend and generalize to the three-dimensional case. Compared to the 2D equations, the 3D Euler equations presents several significant challenges. Firstly, the presence of an additional vorticity stretching term $(\boldsymbol{w} \cdot \nabla)\boldsymbol{u}$ requires a more geometric formulation of the CM method in order to be seamlessly incorporated in the framework: a direct treatment of the vortex stretching term as a source term would not conform to the characteristic structure of the method, nullifying its numerical qualities. Secondly, due to the vortex stretching term, the L^∞ norm of the vorticity is no longer conserved. Indeed, in the 2D case, the scalar vorticity is an advected quantity and small scale features appear from rapid growth in vorticity gradient. In the 3D case, the question of finite-time blow-up of the vorticity is an open problem. From a numerical point of view, the rapid growth in both $|\boldsymbol{w}|$ and $|\nabla \boldsymbol{w}|$ further increase the difficulty in providing sufficient spatial resolution of the solution. For the CM method, the dependence of $\boldsymbol{w}$ in the spatial deformations or rate of strain tensor would then involve the Jacobian of the characteristic map in the computation of the vorticity, thus increasing the regularity requirements on the method. Lastly, the higher dimensionality further emphasizes on the computational efficiency of the method and on high order accuracy.

We follow the same approach as in chapter 4: we evolve numerically the backward characteristic map, i.e. the backward-in-time flow map generated by the fluid velocity. From the Kelvin circulation theorem, the time t vorticity field written as a differential 2-form can be computed as the pullback of the initial vorticity by the characteristic map. A Biot-Savart law on the vorticity is then computed using Fourier spectral methods to provide the velocity field needed to further evolve the map. This allows for a non-dissipative computation of the vorticity field with arbitrary spatial resolution. Compared to the 2D case where vorticity is advected, in the 3D case, it will also be stretched and deformed by the material deformation induced by the flow. A similar approach using the deformation map to evolve the solution can be found in the work of Constantin [29] and a geometric approach to the Euler equations can also be found in the work of Deng et al. [36] where a level-set advection formulation combined with a generalized Clebsch variables expansion was used to show new global existence results for the solutions of the 3D Euler and Lagrangian-Averaged Euler equations under certain assumptions.

The rest of the chapter is organized as follows: in section 5.1 we generalize the CM framework presented in chapter 4 to a geometric reformulation in the context of the 3D incompressible Euler equations. In section 5.2, we provide some details on the numerical implementation of the method together with some formal error estimates supported by convergence tests. Section 5.3 contains some numerical tests and discussions, in particular the anti-parallel axisymmetric perturbed vortex tubes tests similar to the ones appearing in [72] and [61]. Finally, in section 5.4 we make some concluding remarks.

5.1 Mathematical Formulation

We present here the Characteristic Mapping method for the incompressible Euler equations in three-dimensional space. This work is the natural continuation of the framework presented in [74, 84] and extends the Characteristic Mapping method for the 2D Euler equations in chapter 4 to the 3D case. The main challenge in the 3D case is the inclusion of the vortex stretching term in a way that is compatible with the CM method and preserves its arbitrary resolution and low-dissipation properties. For this, we expand on the CM framework by including a more geometric formulation of the problem in terms of differential forms. The basic formulation of the CM method for advection and its numerical discretization follows the same approach as given in chapter 2.

5.1.1 The Euler Equations

We consider the incompressible Euler equations on a three-dimensional domain U, for simplicity, we take U to be the periodic cube $\mathbb{T}^3$.

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \frac{1}{\rho}\nabla p, \tag{5.1a}$$

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{5.1b}$$

where ρ is the scalar density and p is the pressure. For the incompressible equations, the density is assumed to be constant, in which case the continuity equation (5.1b) reduces to the divergence-free condition $\nabla \cdot \boldsymbol{u} = 0$.

We let $\boldsymbol{w} = \nabla \times \boldsymbol{u}$ be the vorticity vector field, which is divergence free by construction. The vorticity equation can be expressed as follows

$$\partial_t \boldsymbol{w} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{w} = (\boldsymbol{w} \cdot \nabla)\boldsymbol{u} - \boldsymbol{w}(\nabla \cdot \boldsymbol{u}), \tag{5.2a}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{5.2b}$$

Since we take a geometric approach, it is convenient to express the vorticity equation in terms of differential forms. Define the vorticity 2-form $\omega = \star(\boldsymbol{w}^\flat)$, where $\flat$ and $\sharp$ denote the lowering and raising of the tensor index and $\star$ is the Hodge-star operator (see Lang [78]). Formally, the 2-form ω performs linear measurements on infinitesimal 2D surfaces by dot product of $\boldsymbol{w}$ with the surface normal; by definition of curl, this yields the total circulation of $\boldsymbol{u}$ along the surface boundary. The vorticity equation (5.2a) is equivalent to the Lie-transport of the vorticity 2-form:

$$\partial_t \omega + \mathcal{L}_{\boldsymbol{u}} \omega = 0, \tag{5.3a}$$

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{5.3b}$$

The Cartan formula for the Lie derivative: $\mathcal{L}_{\boldsymbol{v}} \phi = d(\iota_{\boldsymbol{v}} \phi) + \iota_{\boldsymbol{v}} d\phi$, where $\iota_{\boldsymbol{v}}$ is the interior product, is used to recover the original vorticity vector equation[1] :

$$\begin{aligned}(\star(\partial_t \omega + \mathcal{L}_{\boldsymbol{u}} \omega))^\sharp &= \partial_t \boldsymbol{w} + (\star d(\iota_{\boldsymbol{u}} \omega))^\sharp + (\star \iota_{\boldsymbol{u}} d\omega)^\sharp = \partial_t \boldsymbol{w} + \nabla \times (\boldsymbol{w} \times \boldsymbol{u}) \\ &= \partial_t \boldsymbol{w} + \boldsymbol{w}(\nabla \cdot \boldsymbol{u}) - (\boldsymbol{w} \cdot \nabla)\boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{w} = 0,\end{aligned} \tag{5.4}$$

[1] Using the following identities:

$(\iota_{\boldsymbol{v}} \phi)^\sharp = \boldsymbol{v} \cdot \phi^\sharp$ when ϕ is a 1-form, $(\iota_{\boldsymbol{v}} \phi)^\sharp = (\star\phi)^\sharp \times \boldsymbol{v}$ when ϕ is a 2-form,
$(d\phi)^\sharp = \nabla \phi$ when ϕ is a 0-form, and $(d\phi)^\sharp = \nabla \times \phi^\sharp$ when ϕ is a 1-form.

where $\boldsymbol{w} \times \boldsymbol{u}$ is the Lamb vector. This means that the vorticity 2-form is conserved, i.e. it is Lie-advected or “frozen into” the flow. We note that this uses the general barotropic compressible version of the vorticity equation. In principle, in the incompressible case, the compression term $\boldsymbol{w}(\nabla \cdot \boldsymbol{u})$ vanishes, however, this would mean that the Lie-advected vorticity would be allowed to intensify due to numerical errors on the divergence-free condition.

Remark 5.1.1. One can alternatively check that with $\boldsymbol{w}$ evolving strictly under the incompressible equations, the 2-form $\rho\omega$ is Lie-advected, that is

$$(\star (\partial_t \rho\omega + \mathcal{L}_{\boldsymbol{u}} \rho\omega))^\sharp = \rho(\partial_t \boldsymbol{w} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{w} - (\boldsymbol{w} \cdot \nabla)\boldsymbol{u}) = 0,$$

where ρ is still assumed to satisfy the continuity equation (5.1b) (in case the discretized $\boldsymbol{u}$ is not exactly divergence-free). Then the vorticity field can be obtained from the Lie-advected $\rho\omega$ by scaling by ρ^{-1}, this cancels any stretching of vortices due to artificial volume compression from numerical errors in $(\nabla \cdot \boldsymbol{u})$.

In the context of incompressible fluids, the above simply reduces to the statement that the vorticity 2-form is Lie-advected by the velocity field. This gives us an expression of the vorticity as the pullback of the initial condition by the characteristic map:

$$\omega(\cdot, t) = {\boldsymbol{X}_{[t,0]}}^* \omega_0, \tag{5.5}$$

where the superscript asterisk denotes pullback. For a mapping $F : U \to U$ the pullback F^* it is the dual operator to the pushforward operator denoted by the subscript asterisk F_*. The pullback of a k-form η is defined by $(F^*\eta)(\boldsymbol{v}) = \eta\,(F_*\boldsymbol{v})$ where $\boldsymbol{v}$ is an arbitrary k-vector representing an infinitesimal k-dimensional oriented parallelogram and the pushforward $F_*\boldsymbol{v}$ is its image under the mapping F. Hence, for the 2-form ω, by the generalized Stokes' theorem, equation (5.5) is equivalent to the conservation of circulation along all closed curves transported by the forward flow map.

Equation (5.5) provides many simplifications both numerically and in the analysis mainly due to the fact that pullback commutes with exterior derivatives. For instance, in the study of the Euler equations through Clebsch variables, one makes the simplifying assumption that the initial velocity 1-form is given by $fdg + d\psi$ for some scalar functions f, g and ψ. The initial vorticity is then given by $df \wedge dg$. Applying (5.5) to this initial condition and commuting pullback and d, we get that the vorticity 2-form at time t is given by

$$\omega(\cdot, t) = d\left(f \circ \boldsymbol{X}_{[t,0]}\right) \wedge d\left(g \circ \boldsymbol{X}_{[t,0]}\right), \tag{5.6}$$

that is, it is sufficient to solve the advection equations for f and g and reconstruct the vorticity by a cross product of their gradients. We note that the helicity scalar field is defined as $h = \boldsymbol{u} \cdot \boldsymbol{w}$ which corresponds to the volume form $\boldsymbol{u}^\flat \wedge \omega$. This implies that the Clebsch variable representation is limited to cases where h is exact, i.e. $h = d\phi$ for some 2-form ϕ. It follows that total helicity is 0 for flows admitting Clebsch variables, i.e. non-helical flows. A generalized version of the Clebsch approach has been studied in [36], these Generalized Clebsch variables can be used to represent any initial condition, including helical flows. In fact, the initial velocity expansion (5.10) used in this paper can be seen as a special case of these variables.

For the numerical method described here, we proceed in the following general setting. We assume that there exists closed 1-forms denoted (by abuse of notation) $d\theta_1, d\theta_2, \ldots, d\theta_n$ and scalar functions $u_1, u_2, \ldots, u_n$ such that the initial velocity 1-form $\boldsymbol{u}^\flat$ can be expressed as

$$\boldsymbol{u}^\flat = \sum_{k=1}^{n} u^k d\theta_k. \tag{5.7}$$

Then, the initial vorticity form is given by

$$\omega_0 = \sum_{k=1}^{n} du^k \wedge d\theta_k. \tag{5.8}$$

This gives us a closed expression for the vorticity depending only on $\boldsymbol{X}_{[t,0]}$:

$$\omega(\cdot,t) = \sum_{k=1}^{n} d\left(u^k \circ \boldsymbol{X}_{[t,0]}\right) \wedge {\boldsymbol{X}_{[t,0]}}^{*} d\theta_k. \tag{5.9}$$

We do not require that the $d\theta_k$ 1-forms be exact, as long as the pullback is easy to compute. In fact, for the algorithm implemented in this work, on the 3D torus, the $d\theta_k$ forms, with $n=3$, are simply the coordinate covectors $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$, u_k are the corresponding coordinate values of $\boldsymbol{u}_0$ and the pullback ${\boldsymbol{X}_{[t,0]}}^{*} d\theta_k$ is given by $\partial_k \boldsymbol{X}_{[t,0]}$.

Using the following expansion for the initial velocity

$$\boldsymbol{u}_0^\flat = u^1(1,0,0) + u^2(0,1,0) + u^3(0,0,1), \tag{5.10}$$

we get that the vorticity vector at time t is given by

$$\boldsymbol{w}(\cdot,t) = \sum_{k=1}^{3} \left(\nabla u^k \cdot \nabla \boldsymbol{X}_{[t,0]}\right) \times \nabla \boldsymbol{X}_{[t,0]}, \tag{5.11}$$

which further simplifies to

$$w^i(\cdot,t) = \epsilon_{ijk}\epsilon_{abc} w_0^a \partial_j \boldsymbol{X}_{[t,0]}^b \partial_k \boldsymbol{X}_{[t,0]}^c. \tag{5.12}$$

in summation notation, where ϵ are the Levi-Civita symbols.

Upon further inspection, the above expression is Cramer's rule expansion of

$$\boldsymbol{w}(\cdot,t) = \det\left(\nabla \boldsymbol{X}_{[t,0]}\right)\left(\nabla \boldsymbol{X}_{[t,0]}\right)^{-1} \boldsymbol{w}_0(\boldsymbol{X}_{[t,0]}) = \left(\nabla \boldsymbol{X}_{[t,0]}\right)^{-1} \boldsymbol{w}_0(\boldsymbol{X}_{[t,0]}), \tag{5.13}$$

where the determinant factor can be omitted since the maps are volume preserving. *Remark* 5.1.2. The basis 1-forms $d\theta_k$ are chosen here to express general initial conditions on the torus. In specific cases, for instance in the presence of Clebsch variables, the number of basis 1-forms can be reduced to improve computational performance. That is, the computation of the characteristic map allows for a flexible framework where the vorticity at time t can be constructed by a pullback formula (5.9), not limited to the formula in (5.13).

Remark 5.1.3. An equivalent formulation can be obtained from a Lagrangian perspective by considering the forward map. Following characteristic curves $\boldsymbol{\gamma}$, we see that the vorticity field satisfies

$$\frac{d}{dt}\boldsymbol{w}(\boldsymbol{\gamma}(t), t) = \nabla \boldsymbol{u} \cdot \boldsymbol{w}(\boldsymbol{\gamma}(t), t). \tag{5.14}$$

Noticing that the gradient of the forward map evolves according to $\partial_t \nabla \boldsymbol{X}_F = \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{X}_F$, one can show that the vorticity field at $\boldsymbol{X}_F$ is given by

$$\boldsymbol{w}(\boldsymbol{X}_F, t) = \nabla \boldsymbol{X}_F \cdot \boldsymbol{w}_0(\boldsymbol{x}). \tag{5.15}$$

This is Cauchy's Lagrangian formula used in many Lagrangian particle approaches [11, 29]. Composing the above equation with $\boldsymbol{X}_B$ to return to Eulerian frame and applying the inverse function theorem we get

$$\boldsymbol{w}(\cdot, t) = (\nabla \boldsymbol{X}_B)^{-1} \boldsymbol{w}_0(\boldsymbol{X}_B). \tag{5.16}$$

The factor $\det(\nabla \boldsymbol{X}_B)$ discrepancy with (5.13) does not show up in the incompressible case since the transformations are volume preserving. In fact, using that $\rho\omega$ is Lie-advected and $\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{X}_B) \det(\nabla \boldsymbol{X}_B)$, isolating ω from $\rho\omega$ would remove the determinant factor. Chapter 3 contains some investigations of the CM method for a compressible flow in the context of diffusion-driven density transport.

Remark 5.1.4. The evolution of the vorticity 2-form through pullback by $\boldsymbol{X}_B$ is the infinitesimal expression of the Kelvin circulation theorem which states that the total circulation along a closed curve passively carried by the fluid flow is constant. As a matter of fact, equation (5.11) can be obtained directly by applying the Kelvin circulation theorem to the definition of the curl operator. Formally, considering that $\boldsymbol{w} \cdot \boldsymbol{n} = \lim_{|S| \to 0} \frac{1}{|S|} \int_{\partial S} \boldsymbol{u} \cdot \mathrm{d}\boldsymbol{s}$ for some infinitesimal surface S with unit normal $\boldsymbol{n}$, we apply the Kelvin circulation theorem to ∂S, moving it to its position and shape at time $t = 0$, to obtain $\boldsymbol{w} \cdot \boldsymbol{n} = \lim_{|S| \to 0} \frac{1}{|S|} \int_{\boldsymbol{X}_B(\partial S)} \boldsymbol{u}_0 \cdot \mathrm{d}\boldsymbol{s} = \lim_{|S| \to 0} \frac{1}{|S|} \int_{\partial S} \boldsymbol{u}_0(\boldsymbol{X}_B) \cdot \mathrm{d}\boldsymbol{X}_B(\boldsymbol{s})$ which yields equation (5.11) after taking the limit.

This also relates the CM method to the Kelvin-filtered turbulence models which can roughly be summarized by the vorticity equation

$$\partial_t \boldsymbol{w} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{w} = (\boldsymbol{w} \cdot \nabla)\boldsymbol{v} \tag{5.17}$$

where $\boldsymbol{v}$ is a filtered version of the velocity field $\boldsymbol{u}$, for instance $\boldsymbol{v} = (I - \alpha^2 \Delta)^{-1} \boldsymbol{u}$. We refer to [42] for a review on the theory of these nonlinearly dispersive equations. In the inviscid case, the vorticity is given through pullback by a modified flow map; for the Kelvin filtered equations, the modified flow map is obtained from the filtered velocity field $\boldsymbol{v}$, in the CM framework, the modification on the flow map is a result of a combination of filtering and numerical errors.

Remark 5.1.5. The pullback formulation allows us to quickly check the conservation of total helicity. Indeed, given that the vorticity is the curl of the velocity, i.e. $d\boldsymbol{u}^\flat = \omega$, we have that by Helmholtz, there exists a 1-form η and a scalar ψ such that $\boldsymbol{u}^\flat = \eta + d\psi$ and consequently, the vorticity can be written as $\omega = d\eta$. Furthermore, since $\omega = {\boldsymbol{X}_B}^* \omega_0$, we also have that there exists some η_0 such that $\eta = {\boldsymbol{X}_B}^* \eta_0$ and $d\eta_0 = \omega_0$. The local helicity is given by $\boldsymbol{u} \cdot \boldsymbol{w}$ which corresponds to the 3-form $\boldsymbol{u}^\flat \wedge \omega = \eta \wedge d\eta + d\psi \wedge d\eta$. Noting that $d\psi \wedge d\eta = d(\psi d\eta)$ is exact and so has vanishing total integral, we have

$$\int_U \boldsymbol{u}^\flat \wedge \omega = \int_U \eta \wedge d\eta = \int_U {\boldsymbol{X}_B}^* (\eta_0 \wedge d\eta_0) = \int_{\boldsymbol{X}_B(U)} \eta_0 \wedge d\eta_0 = \int_U \boldsymbol{u}_0^\flat \wedge \omega_0. \tag{5.18}$$

This property relies only on the fact that both η and ω evolve through pullback by the same $\boldsymbol{X}_B$.

The derivations in this section allow us to express the evolution of the vorticity field using the characteristic map. The evolution of the characteristic maps is in turn given by the velocity field, which we compute from the vorticity field using the Biot-Savart law $\boldsymbol{u} = -\Delta^{-1} \nabla \times \boldsymbol{w}$. The fully coupled vorticity-characteristic map equations are as follows

$$(\partial_t + \boldsymbol{u} \cdot \nabla)\boldsymbol{X}_B = 0 \tag{5.19a}$$

$$\boldsymbol{w}(\cdot, t) = (\nabla \boldsymbol{X}_B)^{-1} \boldsymbol{w}_0(\boldsymbol{X}_B) \tag{5.19b}$$

$$\boldsymbol{u} = -\Delta^{-1}\nabla \times \boldsymbol{w}. \tag{5.19c}$$

5.2 Numerical Implementation

The numerical approach of the CM method for the 3D Euler equations largely follows the framework of the Gradient-Augmented Level-Set [89] and Jet-Scheme methods [108]. In this section, we will first present the general numerical framework for the CM method. We will then discuss the specific implementation details used for the numerical experiments in this work.

The computational method is mainly based on the three equations in (5.19). Numerically, this corresponds to evolving a characteristic map $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ in a finite dimensional discretization space $\mathcal{V}$ approximating *Diff*(U), the space of diffeomorphisms on U. Then at every time step, the discretized vorticity $\tilde{\boldsymbol{w}}^n$ and velocity $\tilde{\boldsymbol{u}}^n$ will be reconstructed by pullback using $\boldsymbol{\mathcal{X}}_{[t_n,0]}$. For the rest of this chapter, we will denote by the script letter $\boldsymbol{\mathcal{X}}$ the approximation of the characteristic map $\boldsymbol{X}$ in $\mathcal{V}$, the superscript n on a variable will denote the evaluation of said variable at time t_n and the tilde indicates an approximation or modified equation. The numerical method comprises the three following parts:

1. A discretized velocity field $\tilde{\boldsymbol{u}}^n$ at time t_n, (assuming the characteristic map $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ is known). This is given by the Biot-Savart law

$$\tilde{\boldsymbol{u}}^n = -\Delta^{-1}\nabla \times \left((\nabla\boldsymbol{\mathcal{X}}_{[t_n,0]})^{-1}\boldsymbol{w}_0(\boldsymbol{\mathcal{X}}_{[t_n,0]})\right) \tag{5.20}$$

   computed using spectral Fast-Fourier transform methods.

2. A numerical approximation $\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}$ of the one-step map $\boldsymbol{X}_{[t_{n+1},t_n]}$. For instance, a first order approximation would give

$$\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}) \approx \boldsymbol{x} - \Delta t \tilde{\boldsymbol{u}}^n(\boldsymbol{x}). \tag{5.21}$$

In the current implementation, we use a third-order Runge-Kutta backward in time integrator with a Hermite cubic time-interpolation of the velocity field. A Hermite in time interpolation is used instead of the Lagrange interpolation in chapter 4 to improve accuracy.

3. A time update for the characteristic map based on the group property (2.7a). The map $\boldsymbol{\mathcal{X}}_{[t_{n+1},0]}$ at time t_{n+1} is given by

$$\boldsymbol{\mathcal{X}}_{[t_{n+1},0]} = \mathcal{H}\left[\boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}\right], \tag{5.22}$$

   for some interpolation operator $\mathcal{H} : \mathit{Diff}(U) \to \mathcal{V}$.

We will use the Hermite cubic spatial interpolation method described in chapter 2, the following subsections will examine in more details the velocity interpolation and provide some error estimates.

5.2.1 Velocity Interpolation

Using the Hermite cubic interpolation in the previous section, the numerical characteristic map $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ is defined as a diffeomorphism of the domain U. We define the numerical vorticity $\tilde{\boldsymbol{w}}^n$ through pullback by $\boldsymbol{\mathcal{X}}_{[t_n,0]}$:

$$\tilde{\boldsymbol{w}}^n(\boldsymbol{x}) = \left(\nabla\boldsymbol{\mathcal{X}}_{[t_n,0]}\right)^{-1}\boldsymbol{w}_0(\boldsymbol{\mathcal{X}}_{[t_n,0]}(\boldsymbol{x})), \tag{5.23}$$

where the gradient of $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ is directly evaluated from the interpolant. This defines $\tilde{\boldsymbol{w}}^n$ as a $\mathcal{C}^0$ vector field on U. The numerical velocity $\tilde{\boldsymbol{u}}^n$ is in turn computed from the convolution of the Biot-Savart kernel with $\tilde{\boldsymbol{w}}^n$; we will do this using Fourier spectral methods. We will discretize the velocity field on a grid $\boldsymbol{V}$ and denote by $\mathcal{F}_{\boldsymbol{V}}$ the discrete Fourier transform computed by FFT on the grid $\boldsymbol{V}$. Sampling $\tilde{\boldsymbol{w}}^n$ on $\boldsymbol{V}$ using (5.23) and applying a forward Fourier transform yields a truncated Fourier series for the vorticity field. Computing the Biot-Savart kernel in frequency space then gives us a Fourier series representation for the velocity. Finally, we define the numerical velocity $\tilde{\boldsymbol{u}}^n$ as the Hermite interpolant of this truncated Fourier series; this allows us to evaluate the velocity at arbitrary locations in the domain without having to compute the inverse Fourier transform at non-uniform grid points. The definition of the numerical velocity at t_n can be summarized as follows:

$$\tilde{\boldsymbol{u}}^n(\boldsymbol{x}) = \mathcal{H}_{\boldsymbol{V}}\left[\mathcal{F}_{\boldsymbol{V}}^{-1}\left[-\Delta^{-1}\nabla\times\mathcal{F}_{\boldsymbol{V}}[\tilde{\boldsymbol{w}}^n]\right]\right](\boldsymbol{x}). \tag{5.24}$$

In the above equation, it is understood that in order to define the Hermite cubic interpolant for the velocity, the required mixed partial spatial derivatives are computed directly from the Fourier series.

Similarly, from $\partial_t \boldsymbol{w} = (\boldsymbol{w}\cdot\nabla)\boldsymbol{u} - (\boldsymbol{u}\cdot\nabla)\boldsymbol{w}$, we can also discretize the time derivative of $\boldsymbol{u}$ at t_n:

$$\partial_t\tilde{\boldsymbol{u}}^n(\boldsymbol{x}) = \mathcal{H}_{\boldsymbol{V}}\left[\mathcal{F}_{\boldsymbol{V}}^{-1}\left[-\Delta^{-1}\nabla\times\mathcal{F}_{\boldsymbol{V}}[(\tilde{\boldsymbol{w}}^n\cdot\nabla)\tilde{\boldsymbol{u}}^n - (\tilde{\boldsymbol{u}}^n\cdot\nabla)\tilde{\boldsymbol{w}}^n]\right]\right](\boldsymbol{x}). \tag{5.25}$$

The data $\tilde{\boldsymbol{u}}^n$ and $\partial_t\tilde{\boldsymbol{u}}^n$ at time steps t_n allow us to locally approximate the velocity using a 4-dimensional time-space Hermite cubic interpolant. For the one-step map (5.21), we will define an approximate $\tilde{\boldsymbol{u}}$ in the interval $[t_n, t_{n+1}]$ by extending the interpolant obtained from the velocity data $\boldsymbol{u}^{n-1}$ and $\boldsymbol{u}^n$ in the interval $[t_{n-1}, t_n]$. This gives the following definition for the numerical velocity field:

$$\begin{aligned}\tilde{\boldsymbol{u}}(\boldsymbol{x},t) = {}&\big(q_0(t-t_{n-1})\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x}) + q_0(t-t_n)\tilde{\boldsymbol{u}}^n(\boldsymbol{x})\big)\\ &+\Delta t\left(q_1(t-t_{n-1})\partial_t\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x}) + q_1(t-t_n)\partial_t\tilde{\boldsymbol{u}}^n(\boldsymbol{x})\right) \quad \text{for} \quad t\in[t_n,t_{n+1}),\end{aligned} \tag{5.26}$$

using the Hermite basis functions given in (2.13). We note that since for each n, $\tilde{\boldsymbol{u}}^n$ and $\partial_t\tilde{\boldsymbol{u}}^n$ are Hermite interpolants of divergence-free vector fields, the modified velocity field $\tilde{\boldsymbol{u}}(\boldsymbol{x},t)$ is a linear combination of divergence-free velocity fields and is also divergence-free at all time up to interpolation error. This error can be reduced by refining the velocity interpolation grid which can be achieved by a zero-padding in frequency space before taking the inverse Fourier transform.

The one-step map in the interval $[t_n, t_{n+1}]$ is then obtained from the backward in time flow of the approximate velocity field $\tilde{\boldsymbol{u}}$. We define the numerical one-step map $\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}$ pointwise using a third order Runge-Kutta integration of $\tilde{\boldsymbol{u}}$:

$$\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]} = \boldsymbol{x} + \int_{t_{n+1}}^{t_n} \tilde{\boldsymbol{u}}\left(\tilde{\boldsymbol{X}}_{[\tau,t_n]}, \tau\right) \mathrm{d}\tau. \tag{5.27}$$

The one-step map is used in the time update of the characteristic map (5.22); it is therefore only evaluated at grid points. However, in order to compute the chain rules for the derivatives required to define the Hermite interpolant, we also need the mixed partial derivatives of $\tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}$ at grid points. This is done using a 4^{th} order version of the ϵ-difference scheme described in [26]. The ϵ-difference schemes introduce an L^∞ error of order $\epsilon^4 + \delta \sum_{k=0}^{3} \Delta x^k \epsilon^{-k}$ where δ is the machine precision. For all computations presented in this paper, we used $\epsilon = 2.5 \times 10^{-3}$ which corresponds to an error term of at most 10^{-8} and effectively less than 10^{-11}, if $\Delta x < 0.1$.

Remark 5.2.1. Evaluating the extrapolation formula (5.26) at $t = t_{n+1}$ would generally not give the same velocity as $\tilde{\boldsymbol{u}}^{n+1}$ which is obtained by vorticity pullback followed by the Biot-Savart law. Indeed, the extrapolation of the velocity in the interval $[t_n, t_{n+1})$ is used only to evolve the characteristic map to time t_{n+1}. The velocity at t_{n+1} is then reconstructed using (5.23) and (5.24), similar to a predictor-corrector approach. This also implies that numerical errors in the extrapolation are not directly carried and amplified in the next time step.

5.2.2 Error Estimates

We will examine in this section the numerical error on the characteristic map and its relation to the error on the vorticity field. We will try to characterize the nature of the numerical error and provide some estimates. We use as starting assumption that the numerical map $\boldsymbol{\mathcal{X}}_B$ is consistent with the exact map $\boldsymbol{X}_B$ in the $\mathcal{C}^{1,\alpha}$ norm for some $\alpha \in (0,1)$, that is the error is $o(1)$. This will allow us to estimate the global $\mathcal{C}^{1,\alpha}$ error to third-order in time and space by omitting higher order terms in the error. The consistency assumption is then implied for short-time since the initial numerical map $\boldsymbol{\mathcal{X}}_{[0,0]} = \boldsymbol{x}$ is exact. In order to preserve the advective structure of the error, we define the following error map:

$$\boldsymbol{E}_{[0,t_n]} := \boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \boldsymbol{X}^{-1}_{[t_n,0]} = \boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \boldsymbol{X}_{[0,t_n]} = \boldsymbol{\mathcal{X}}_B \circ \boldsymbol{X}_F, \tag{5.28}$$

which measures by how much the diffeomorphism $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ differs from the inverse of the exact forward flow map $\boldsymbol{X}_{[0,t_n]}$. Indeed, since the composition of the forward and backward maps $\boldsymbol{X}_{[t_n,0]} \circ \boldsymbol{X}_{[0,t_n]} = \boldsymbol{x}$ gives the identity map, we have that

$$\boldsymbol{E}_{[0,t_n]} = \boldsymbol{x} + \left(\boldsymbol{\mathcal{X}}_{[t_n,0]} - \boldsymbol{X}_{[t_n,0]}\right) \circ \boldsymbol{X}_{[0,t_n]}, \tag{5.29}$$

that is, the deviation of $\boldsymbol{E}_{[0,t_n]}$ from the identity map is the numerical error of the map evaluated at the pushforward location. This is a Lagrangian representation of the error since $\boldsymbol{x} - \boldsymbol{E}_{[0,t_n]}(\boldsymbol{x})$ essentially gives the time t_n map error for a particle starting at $\boldsymbol{x}$ at time 0. We also note that since $\boldsymbol{\mathcal{X}}_B$ is a $\mathcal{C}^1$ diffeomorphism by construction, it follows that all maps considered here are also $\mathcal{C}^1$ diffeomorphisms. Thus left and right inverses exist, are equal and are also diffeomorphisms.

We also define the auxiliary "modified map"

$$\tilde{\boldsymbol{X}}_{[t_n,0]} := \tilde{\boldsymbol{X}}_{[t_1,0]} \circ \tilde{\boldsymbol{X}}_{[t_2,t_1]} \circ \cdots \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}, \tag{5.30}$$

where $\tilde{\boldsymbol{X}}_{[t_k,t_{k-1}]}$ is the one-step map given in (5.27) obtained from RK3 integration on the numerical interpolated velocity field.

The full error map is decomposed as follows:

$$\boldsymbol{E}_{[0,t_n]} = \boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}^{-1}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}_{[t_n,0]} \circ \boldsymbol{X}^{-1}_{[t_n,0]}. \tag{5.31}$$

We let $\boldsymbol{\Phi}_{[0,t_n]} = \boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}^{-1}_{[t_n,0]}$ and $\boldsymbol{\Psi}_{[0,t_n]} = \tilde{\boldsymbol{X}}_{[t_n,0]} \circ \boldsymbol{X}^{-1}_{[t_n,0]}$ and compute their time-evolution as follows:

$$\begin{aligned}
\boldsymbol{\Phi}_{[0,t_n]} &= \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right] \circ \left(\tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right)^{-1} \\
&= \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right] \circ \left(\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right)^{-1} \circ \boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}^{-1}_{[t_{n-1},0]} \\
&= \boldsymbol{\xi}_n \circ \boldsymbol{\Phi}_{[0,t_{n-1}]},
\end{aligned} \tag{5.32}$$

where we defined a one-step error $\boldsymbol{\xi}_n := \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right] \circ \left(\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right)^{-1}$. We note that $\boldsymbol{\xi}_n$ is the error due to Hermite interpolation since

$$\begin{aligned}\boldsymbol{\xi}_n - \boldsymbol{x} &= \left(\mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right] - \boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right) \circ \tilde{\boldsymbol{X}}^{-1}_{[t_n,t_{n-1}]} \circ \boldsymbol{\mathcal{X}}^{-1}_{[t_{n-1},0]} \\ &= \left(\mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right] - \boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right) \circ \tilde{\boldsymbol{X}}^{-1}_{[t_n,0]} \circ \boldsymbol{\Phi}^{-1}_{[0,t_{n-1}]}. \qquad (5.33)\end{aligned}$$

We define the following interpolation error

$$\boldsymbol{\varphi}_n := \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right] - \boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}, \qquad (5.34)$$

we then obtain that

$$\boldsymbol{\Phi}_{[0,t_n]} = (\boldsymbol{x} + (\boldsymbol{\xi}_n - \boldsymbol{x})) \circ \boldsymbol{\Phi}_{[0,t_{n-1}]} = \boldsymbol{\Phi}_{[0,t_{n-1}]} + \boldsymbol{\varphi}_n \circ \tilde{\boldsymbol{X}}^{-1}_{[t_n,0]} = \boldsymbol{x} + \sum_{k=1}^{n} \boldsymbol{\varphi}_k \circ \tilde{\boldsymbol{X}}^{-1}_{[t_k,0]}. \qquad (5.35)$$

A similar derivation gives us

$$\boldsymbol{\Psi}_{[0,t_n]} = \boldsymbol{\eta}_n \circ \boldsymbol{\Psi}_{[0,t_{n-1}]}, \qquad (5.36)$$

where $\boldsymbol{\eta}_n := \left(\tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}\right) \circ \left(\tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \boldsymbol{X}_{[t_n,t_{n-1}]}\right)^{-1}$. We note that $\boldsymbol{\eta}$ is the velocity approximation error which gives the discrepancy between the true flow and the flow obtained from the modified velocity $\tilde{\boldsymbol{u}}$.

$$\boldsymbol{\eta}_n - \boldsymbol{x} = \left(\tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \boldsymbol{X}_{[t_n,t_{n-1}]}\right) \circ \boldsymbol{X}^{-1}_{[t_n,0]} \circ \boldsymbol{\Psi}^{-1}_{[0,t_{n-1}]}. \qquad (5.37)$$

We define the following modified flow error

$$\boldsymbol{\psi}_n = \tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_{n-1},0]} \circ \boldsymbol{X}_{[t_n,t_{n-1}]}, \qquad (5.38)$$

i.e. the map evolution error due to errors in the approximated velocity field; we note that this term is approximately $\Delta t \nabla \tilde{\boldsymbol{X}}_{[t_{n-1},0]} \cdot (\tilde{\boldsymbol{u}} - \boldsymbol{u})$. This gives us

$$\boldsymbol{\Psi}_{[0,t_n]} = (\boldsymbol{x} + (\boldsymbol{\eta}_n - \boldsymbol{x})) \circ \boldsymbol{\Psi}_{[0,t_{n-1}]} = \boldsymbol{\Psi}_{[0,t_{n-1}]} + \boldsymbol{\psi}_n \circ \boldsymbol{X}^{-1}_{[t_n,0]} = \boldsymbol{x} + \sum_{k=1}^{n} \boldsymbol{\psi}_k \circ \boldsymbol{X}^{-1}_{[t_k,0]}. \tag{5.39}$$

This allows us to write the error map as

$$\boldsymbol{E}_{[0,t_n]} = \left(\boldsymbol{x} + \sum_{k=1}^{n} \boldsymbol{\varphi}_k \circ \tilde{\boldsymbol{X}}^{-1}_{[t_k,0]} \right) \circ \left(\boldsymbol{x} + \sum_{k=1}^{n} \boldsymbol{\psi}_k \circ \boldsymbol{X}^{-1}_{[t_k,0]} \right). \tag{5.40}$$

We use the fact that for two $\mathcal{C}^{1,\alpha}$ diffeomorphisms, $\boldsymbol{f}$ and $\boldsymbol{g}$, the composition $\boldsymbol{f} \circ \boldsymbol{g}$ is also $\mathcal{C}^{1,\alpha}$ with $\|\boldsymbol{f} \circ \boldsymbol{g}\|_{\mathcal{C}^{1,\alpha}} \leq C \|\boldsymbol{f}\|_{\mathcal{C}^{1,\alpha}} \|\boldsymbol{g}\|^{1+\alpha}_{\mathcal{C}^{1,\alpha}}$, to estimate the norm of the Lagrangian displacement error $\boldsymbol{\epsilon}_{[0,t_n]} := \boldsymbol{x} - \boldsymbol{E}_{[0,t_n]}$:

$$\|\boldsymbol{\epsilon}_{[0,t_n]}\|_{\mathcal{C}^{1,\alpha}} \lesssim \sum_{k=1}^{n} \|\boldsymbol{\psi}_k\|_{\mathcal{C}^{1,\alpha}} + \left(1 + \sum_{k=1}^{n} \|\boldsymbol{\psi}_k\|_{\mathcal{C}^{1,\alpha}} \right)^{1+\alpha} \sum_{k=1}^{n} \|\boldsymbol{\varphi}_k\|_{\mathcal{C}^{1,\alpha}}, \tag{5.41}$$

where the terms dependent on $\boldsymbol{X}_{[t_n,0]}$ have been absorbed in the constants of the inequality; $\tilde{\boldsymbol{X}}_{[t_n,0]}$ is also approximated with $\boldsymbol{X}_{[t_n,0]}$ using the consistency assumption. Here we use the notation $A \lesssim B$ to denote that there exists some constant c such that $A < cB$. In this section, the constant will depend on the dimension, the domain, the constants involved in the norms, and the solution $\boldsymbol{u}$ as well as $\boldsymbol{X}_B$.

The $\boldsymbol{\varphi}_n$ error is an error pertaining the numerical resolution of $\boldsymbol{\mathcal{X}}_B$, it can be controlled as long as we can control the higher derivatives of $\boldsymbol{\mathcal{X}}_{[t_{n-1},0]} \circ \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}$. The stability analysis of a similar methods has been studied in [50].

The $\boldsymbol{\psi}_n$ error is a feedback between the map error and the velocity error (and also numerical integration), with $\boldsymbol{\psi}_n \approx \nabla \tilde{\boldsymbol{X}}_{[t_{n-1},0]} \Delta t (\tilde{\boldsymbol{u}} - \boldsymbol{u})$ and $\|\boldsymbol{\psi}_n\|_{\mathcal{C}^{1,\alpha}} \lesssim \Delta t \|\tilde{\boldsymbol{u}}^{n-1} - \boldsymbol{u}^{n-1}\|^{1+\alpha}_{\mathcal{C}^{1,\alpha}}$. We first bound the error on the vorticity which will require the Eulerian version of the error map. Define

$$\boldsymbol{E}_{[t_n,0]} := \boldsymbol{X}^{-1}_{[t_n,0]} \circ \boldsymbol{\mathcal{X}}_{[t_n,0]} = \boldsymbol{X}_{[0,t_n]} \circ \boldsymbol{E}_{[0,t_n]} \circ \boldsymbol{X}_{[t_n,0]}, \tag{5.42}$$

here the conjugation exactly serves the purpose of changing the error from a Lagrangian frame to an Eulerian frame.

We note that

$$\boldsymbol{w}(\boldsymbol{x}, t_n) = (\nabla \boldsymbol{X}_B)^{-1}\boldsymbol{w}_0(\boldsymbol{X}_B) \quad \text{and} \quad \tilde{\boldsymbol{w}}^n(\boldsymbol{x}) = (\nabla \boldsymbol{\mathcal{X}}_B)^{-1}\boldsymbol{w}_0(\boldsymbol{\mathcal{X}}_B), \tag{5.43}$$

So we have that, using $\boldsymbol{\mathcal{X}}_B = \boldsymbol{X}_B \circ \boldsymbol{E}_B$ and $(\nabla \boldsymbol{\mathcal{X}}_B)^{-1} = (\nabla \boldsymbol{E}_B)^{-1} (\nabla \boldsymbol{X}_B)^{-1}|_{\boldsymbol{E}_B^{-1}}$

$$\tilde{\boldsymbol{w}}^n = (\nabla \boldsymbol{E}_B)^{-1} \boldsymbol{w}^n(\boldsymbol{E}_B) \quad \text{i.e.} \quad \tilde{\omega}^n = \boldsymbol{E}_B{}^*\omega^n. \tag{5.44}$$

The numerical vorticity is the pullback of the exact vorticity by the error map. This was expected since the exact map was decomposed into the composition of the numerical map and the error map.

A first look at the velocity error, letting $\boldsymbol{\epsilon}_B = \boldsymbol{x} - \boldsymbol{E}_B$ assuming $\boldsymbol{\epsilon}_B$ small, we have to leading order terms

$$\begin{aligned} \boldsymbol{w}^n - \tilde{\boldsymbol{w}}^n &\approx (\boldsymbol{I} - (\nabla \boldsymbol{E}_B)^{-1})\boldsymbol{w}^n(\boldsymbol{E}_B) + \nabla \boldsymbol{w}^n \cdot (\boldsymbol{x} - \boldsymbol{E}_B) \\ &\approx \nabla \boldsymbol{\epsilon}_B \boldsymbol{w}^n + \nabla \boldsymbol{w}^n \boldsymbol{\epsilon}_B. \end{aligned} \tag{5.45}$$

up to second order terms of $\mathcal{O}(\|\boldsymbol{\epsilon}_B\|^2)$.

This gives us $\|\boldsymbol{w}^n - \tilde{\boldsymbol{w}}^n\|_{\mathcal{C}^{0,\alpha}} \lesssim \|\boldsymbol{\epsilon}_B\|_{\mathcal{C}^{1,\alpha}}$ for some $\alpha \in (0, 1)$, which after Biot-Savart, yields

$$\|\boldsymbol{u}^n - \tilde{\boldsymbol{u}}^n\|_{\mathcal{C}^{1,\alpha}} \lesssim \|\boldsymbol{\epsilon}_B\|_{\mathcal{C}^{1,\alpha}} \tag{5.46}$$

We use this to control the $\boldsymbol{\psi}_n$ error which is due to the difference between the numerical and the exact velocities as well as the interpolation and integration schemes both of which have 4^{th} order local truncation error.

$$\|\boldsymbol{\psi}_n\|_{\mathcal{C}^{1,\alpha}} \lesssim \Delta t \|\boldsymbol{\epsilon}_{[t_{n-1},0]}\|^{1+\alpha}_{\mathcal{C}^{1,\alpha}} + \mathcal{O}(\Delta t^4). \tag{5.47}$$

Since the Lagrangian and Eulerian error maps are related by conjugation by the map $\boldsymbol{X}_{[t_n,0]}$, we have that $\|\boldsymbol{\epsilon}_{[t_n,0]}\|_{\mathcal{C}^{1,\alpha}} \lesssim \|\boldsymbol{\epsilon}_{[0,t_n]}\|^{1+\alpha}_{\mathcal{C}^{1,\alpha}}$ and vice-versa. Therefore, we can write the following estimate for the Eulerian error, up to leading order terms, using (5.41):

$$\|\boldsymbol{\epsilon}_{[t_n,0]}\|_{\mathcal{C}^{1,\alpha}}^{1/(1+\alpha)} \lesssim \sum_{k=1}^{n} \Delta t \|\boldsymbol{\epsilon}_{[t_{k-1},0]}\|_{\mathcal{C}^{1,\alpha}}^{1+\alpha} + \sum_{k=1}^{n} \|\boldsymbol{\varphi}_k\|_{\mathcal{C}^{1,\alpha}} . \tag{5.48}$$

This can be majorized by the ODE by approximating the discrete sum of order Δt terms with the integral from 0 to $t = n\Delta t$ and taking a time derivative.

$$\dot{\boldsymbol{\epsilon}}_B = (1+\alpha)\boldsymbol{\epsilon}_B^{\alpha/(1+\alpha)}(\boldsymbol{\epsilon}_B^{1+\alpha} + A), \tag{5.49}$$

where $A = \mathcal{O}(\Delta t^3) + \frac{1}{\Delta t}\|\boldsymbol{\varphi}_k\|_{\mathcal{C}^{1,\alpha}}$, which is the CM advection $\mathcal{C}^1$ error and is $\mathcal{O}(\Delta t^3 + \Delta x^2)$.

We note that the Hölder-α norm was introduced artificially to gain the full 2 degrees of regularity from the Poisson equation in the Biot-Savart law. We can therefore pick $\alpha > 0$ arbitrarily small, in which case, the map error estimate $\boldsymbol{\epsilon}_B$ in (5.49) solves a regularly perturbed 1^{st} order linear ODE with a source term of $\mathcal{O}(\Delta t^3 + \Delta x^2)$. We recall that this is built on the assumption that the modified flow map $\tilde{\boldsymbol{X}}_{[t_n,0]}$ is consistent with $\boldsymbol{X}_{[t_n,0]}$ so that higher order error terms can be omitted; this is true since the initial $\tilde{\boldsymbol{X}}_{[t_0,0]}$ is exact and the time-evolution of the error is third-order in Δx and Δt according to the above derivations. It is also assumed that $\boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}$ is well represented by Hermite interpolation, i.e. that the spatial resolution is high enough and that the grid data of $\boldsymbol{\mathcal{X}}_{[t_i,0]}$ do not oscillate unboundedly. This can be in part controlled by having high enough resolution and also by a remapping method discussed in the following section. The stability of the CM method was also discussed in [84, 89, 122] and convergence of similar methods using Hermite interpolation was proven in [50]. Overall, the CM method should have $\mathcal{O}(\Delta x^2 + \Delta t^3)$ error in $\mathcal{C}^1$ norm which would translate to a 3^{rd} order global convergence in L^∞ norm.

We provide here two numerical tests for the error estimates derived above. As a sanity check, we test the method on the stationary Arnold-Beltrami-Childress (ABC) flow so that the numerical solution can be compared against a known exact solution. We will also perform a second test using a standard Taylor-Green vortex initial condition, numerical results will be compared against a high-resolution reference test.

Both tests are performed on a periodic domain $[-2\pi, 2\pi]^3$. The ABC flow initial condition is given by

$$\boldsymbol{w}_0(x, y, z) = \frac{1}{2}\begin{pmatrix} \cos(y) + \sin(z) \\ \cos(z) + \sin(x) \\ \cos(x) + \sin(y) \end{pmatrix}, \tag{5.50}$$

and the Taylor-Green initial condition is given by

$$\boldsymbol{w}_0(x, y, z) = \begin{pmatrix} \cos\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right)\sin\left(z\right) \\ \sin\left(\frac{x}{2}\right)\cos\left(\frac{y}{2}\right)\sin\left(z\right) \\ -\sin\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right)\cos\left(z\right) \end{pmatrix}. \tag{5.51}$$

We run each test on increasingly finer grids of N cells per dimension for both $\boldsymbol{M}$ and $\boldsymbol{V}$ and using $N/12$ time steps to reach the final time. In both cases, the final time is $T_f = 2$, so that $\Delta x = \frac{4\pi}{N}$ and $\Delta t = \frac{24}{N}$. For the ABC test, we measure the vorticity error using the exact solution $\boldsymbol{w}^n = \boldsymbol{w}_0$ and for the Taylor-Green vortex test, we measure the $\boldsymbol{w}$, $\boldsymbol{\mathcal{X}}_B$ and $\nabla\boldsymbol{\mathcal{X}}_B$ errors at grid points by comparing against a reference higher resolution test with $N = 216$. We note that for both tests the maximum velocity throughout the simulation is greater than 0.9 at all times so that Δt exceeds the CFL condition. The maximum vorticity for the ABC test is constant in time, for the Taylor-Green test, a 20% growth is observed over the $[0, 2]$ time interval. Figures 5–1 and 5–2 show the L^∞ errors for both tests at $T_f = 2$; the errors are computed directly from grid values and confirm the expected 3^{rd} order error.

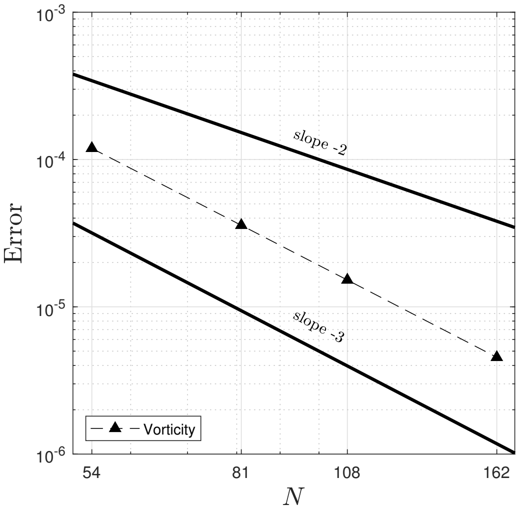


Figure 5–1: L^∞ vorticity error for the ABC test at $T_f = 2$. Numerical solution is directly compared to the exact solution $\boldsymbol{w}(\boldsymbol{x}, t) = \boldsymbol{w}_0(\boldsymbol{x})$.

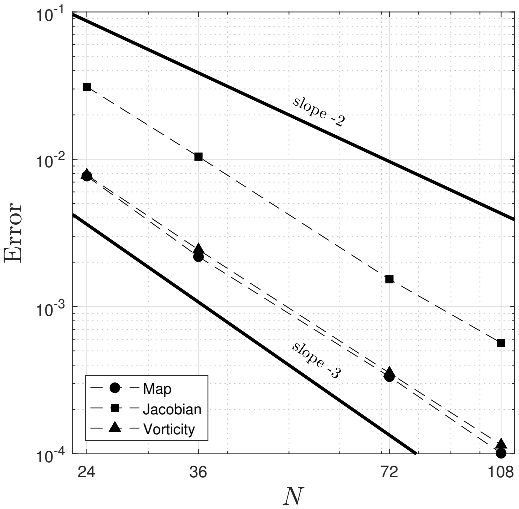


Figure 5–2: L^∞ map, Jacobian and vorticity errors for the Taylor-Green Vortex test at $T_f = 2$. Error is calculated by comparing results against a $N = 216$ test.

5.2.3 Submap Decomposition and Vorticity Sampling

The error estimates in section 5.2.2 show that the error is advective in nature since the numerical solution can be written as the pullback of the exact solution by an error map; therefore, viscous and hyper-viscous type dissipation in the solution are avoided. Indeed, by computing the characteristic map, the vorticity field is provided functionally and can be evaluated anywhere on the domain by interpolating $\boldsymbol{\mathcal{X}}_B$. Instead of directly evolving the vorticity on some grid, in which case $\tilde{\boldsymbol{w}}^n$ depends on the grid values of $\tilde{\boldsymbol{w}}^{n-1}$, the vorticity in the CM method is in principle "reconstructed" at every step using $\tilde{\omega}^n = \boldsymbol{\mathcal{X}}_{[t_n,0]}{}^*\omega_0$. For traditional grid-based methods, $\tilde{\boldsymbol{w}}^n$ is obtained from grid values of $\tilde{\boldsymbol{w}}^{n-1}$ and typically carries a viscous or hyper-viscous error of the form $\epsilon\Delta^p\tilde{\boldsymbol{w}}^{n-1}$, in the CM method, the error is instead advective with $\tilde{\boldsymbol{w}}^n = \boldsymbol{E}_B{}^*\boldsymbol{w}^n$, thus preventing the loss of subgrid scales due to artificial viscosity. One implication is that the CM method can better avoid artificial merging of vortices at the subgrid scale in particular in the study of vortex tube reconnection problems. The error in this case is of elastic type. Indeed, since $\boldsymbol{\mathcal{X}}_B$ is evolved using the GALS method, the leading order spatial errors on the map are of the form $\epsilon\Delta^p\boldsymbol{\mathcal{X}}_B$. This can be seen as an elasticity term which dampens extensive deformation of the domain under the $\boldsymbol{\mathcal{X}}_B$ mapping. According to the error estimates in section 5.2.2, the map error can be decomposed into two parts: the velocity error $\boldsymbol{\psi}$ and the map interpolation error $\boldsymbol{\varphi}$.

The $\boldsymbol{\psi}$ velocity representation error depend in part on the resolution of the $\boldsymbol{V}$ grid used for the vorticity sampling. This grid needs to be fine enough to avoid Fourier aliasing from lack of resolution. One possible solution would be to use a dynamic sampling of the vorticity field such as through the use of oct-tree structured meshgrids. Here we will consider another adaptive sampling method: we directly evaluate a mollified version of the vorticity, that is defining $\tilde{\boldsymbol{w}} = \mu_h * \boldsymbol{w}^n$ where μ_h is a mollifier supported in a neighborhood of size h, this approach was studied to some extent in [122]. We pick h to be smaller than the cellwidth of $\boldsymbol{V}$, the evaluation of the mollified vorticity at grid points $\boldsymbol{x_i}$ can then be expressed as the sum of the convolution integrals in all 8 cells $C_{\boldsymbol{i}+\boldsymbol{r}}$ adjacent to $\boldsymbol{x_i}$:

$$(\mu_h * \boldsymbol{w}^n)(\boldsymbol{x_i}) = \sum_{\boldsymbol{r}\in\{-1,0\}^3} \int_{C_{\boldsymbol{i}+\boldsymbol{r}}} \mu_h(\boldsymbol{x_i} - \boldsymbol{y})\boldsymbol{w}^n(\boldsymbol{y})\mathrm{d}\boldsymbol{y}. \tag{5.52}$$

For instance, in the tests in section 5.3.2, the mollifier μ_h is chosen to be a 3D tensor of $\cos^2(x/h)$ supported in the cells adjacent to $\boldsymbol{x_i}$. The integral in each cell is computed using numerical quadrature and the number of quadrature points is then adapted to the local oscillations of $\boldsymbol{w}^n$ to ensure the accuracy of the mollification. The resulting algorithm effectively computes a mollified vorticity field where subgrid oscillations are filtered out, by choosing an appropriate mollification scale, the $\boldsymbol{V}$ grid can resolve the mollified vorticity without aliasing errors and the pointwise evaluation of $\mu_h * \boldsymbol{w}^n$ on $\boldsymbol{V}$ is accurate as long as the sampling in each cell is sufficiently dense. This approach is also related to the Kelvin-filtered Euler equations. From the Kelvin circulation theorem, total circulation along a closed curve evolving under the flow is conserved. Both in the Kelvin-filtered equations and the CM method, circulation is conserved for closed curves evolving under a modified flow; in the Kelvin-filtered case, the modified flow arises from a filtered velocity field and in the CM case, the numerical flow map is modified by the $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ errors, of which $\boldsymbol{\psi}$ contains the velocity filtering. In any case, the mollification of the transport velocity still deteriorates the accuracy of the method and, although the vorticity evolves by pulllback, dispersion is introduced in the form of a less energetic transport flow. The proper sampling of the vorticity is still subject of our current work, and future directions may include the use of adaptive meshes and wavelet transforms to reduce sampling errors.

The second main source of error is the map representation error $\boldsymbol{\varphi}$, which depends on the $\boldsymbol{M}$ grid and the regularity of the characteristic map $\boldsymbol{X}_B$. For Hermite cubic interpolation, this error roughly scales with the 4^{th} spatial derivative of $\boldsymbol{X}_B$. At time $t = 0$, this error is 0 since $\boldsymbol{X}_{[0,0]}$ is the identity map, then the error increases with time as the characteristic map develops more complicated spatial features. We limit the growth of this error by periodically reinitializing the characteristic map using the group property of the flow maps. Indeed, a time t characteristic map can be decomposed as follows using (2.7a):

$$\boldsymbol{X}_{[t,0]} = \boldsymbol{X}_{[T_1,0]} \circ \boldsymbol{X}_{[T_2,T_1]} \circ \cdots \circ \boldsymbol{X}_{[T_{m-1},T_{m-2}]} \circ \boldsymbol{X}_{[t,T_{m-1}]}, \tag{5.53}$$

for $0 = T_0 < T_1 < \cdots < T_{m-1} < T_m = t$. The T_i are remapping times, in each interval $[T_i, T_{i+1}]$, the evolution of the characteristic map is given by the $T_{i+1} - T_i$ time flow of equations (5.19) with $\boldsymbol{w}(\boldsymbol{x}, T_i)$ as initial condition.

Remark 5.2.2. Note that the T_i refer to the remapping times and t_n are the time steps of the scheme with $t_n - t_{n-1} = \Delta t$ small, approximating the limit to 0. On the other hand $T_i - T_{i-1}$ is $\mathcal{O}(1)$, its purpose is to subdivide the time interval $[0, T_{final}]$ into shorter subintervals where the characteristic maps are better behaved.

At any t, the vorticity is given by the following pullback using submap decomposition:

$$\boldsymbol{w}(\boldsymbol{x}, t) = \left(\nabla \boldsymbol{X}_{[t,T_{m-1}]}\right)^{-1} \cdots \left(\nabla \boldsymbol{X}_{[T_2,T_1]}\right)^{-1} \left(\nabla \boldsymbol{X}_{[T_1,0]}\right)^{-1} \left(\boldsymbol{w}_0 \circ \boldsymbol{X}_{[T_1,0]} \circ \cdots \circ \boldsymbol{X}_{[t,T_{m-1}]}\right). \tag{5.54}$$

Numerically, this means that we can compute each submap individually and use

$$\tilde{\boldsymbol{w}}(\cdot, t) = \left(\prod_{i=0}^{m-1} \left(\nabla \boldsymbol{\mathcal{X}}_{[T_{m-i},T_{m-i-1}]}\right)^{-1} \right) \boldsymbol{w}_0 \left(\boldsymbol{\mathcal{X}}_{[T_1,0]} \circ \cdots \circ \boldsymbol{\mathcal{X}}_{[t,T_{m-1}]}\right) \tag{5.55}$$

to compute the pullback. Each submap will only perform the mapping in the time subinterval $[T_i, T_{i+1}]$ and remapping times can be either fixed or chosen dynamically so that each submap can be well represented using the grid $\boldsymbol{M}$. Indeed, after each remapping, we essentially solve a separate Euler equation in the time interval $[T_i, T_{i+1}]$, the characteristic map is reinitialized with the identity map. This means that the spatial representation of the current submap is again exact and will accumulate error over $[T_i, T_{i+1}]$, until an error threshold is exceeded and remapping is triggered. In terms of the error estimates in section 5.2.2, the remapping resets the $\boldsymbol{\varphi}$ error of the i^{th} submap to 0, which prevents further accumulation of spatial interpolation errors due to the fixed resolution of the $\boldsymbol{M}$ grid. One can therefore control the accumulation of the elasticity type error by changing the frequency of the remapping. More frequent remapping reduces the effect that the $\boldsymbol{M}$ grid has on the spatial features of the map; in the extreme case where remapping is done at every time step, the $\boldsymbol{\varphi}$ error can no longer accumulate and numerical error is reduced to $\boldsymbol{\psi}$ only. For the numerical experiments presented in this paper, we use the volume-preservation error of the $\boldsymbol{\mathcal{X}}_B$ map as remapping criterion, that is, the remapping time T_i is chosen to be the first time t such that the error $|\det \nabla \boldsymbol{\mathcal{X}}_{[t,T_{i-1}]} - 1|$ is greater than some chosen tolerance. On one hand, this serves as an *a posteriori* estimate of the $\mathcal{C}^1$ error of the map and on the other hand, this allows us to guarantee that the composition of all submaps yields a diffeomorphism and provides some control on the overall volume-preserving property of the characteristic map.

5.2.4 Implementation Summary

The previous subsections contain the numerical tools for implementing the CM method for the 3D incompressible Euler equations, we give here a short summary of the method in pseudocode format. We note that the method uses two discretization grids, a grid $\boldsymbol{M}$ for representing the numerical map $\boldsymbol{\mathcal{X}}_{[t_n,0]}$ and a grid $\boldsymbol{V}$ for sampling the numerical vorticity $\boldsymbol{w}^n$ and computing the Biot-Savart law using Fourier spectral methods. These two grids do not need to have the same resolution, in fact, using the submap decomposition method described in the previous subsection, a short time characteristic map can be represented on a coarse grid $\boldsymbol{M}$. The $\boldsymbol{V}$ grid used to represent the vorticity needs to be fine enough to avoid sampling errors. Indeed, as the flow evolves, the vorticity can develop small scale features and high gradients. If $\boldsymbol{V}$ is not fine enough to resolve $\boldsymbol{w}^n$, this can cause aliasing errors in the Fourier transform. One way to reduce the effect of undersampling is to mollify $\boldsymbol{w}^n$ in Fourier space, however this also reduces the accuracy of the scheme. Other possibilities for future investigation include the use adaptive mesh for the vorticity sampling and wavelet methods for the computation of the Biot-Savart law [105, 107]. The CM method for 3D incompressible Euler equations can be summarized with the following pseudocode algorithm.

Algorithm 3 CM method 3D incompressible Euler equations.

```
Input: Initial vorticity $\boldsymbol{w}_0$, grids $\boldsymbol{M}$ and $\boldsymbol{V}$, time step $\Delta t$, final time $T_f$
 1: Initialize $n \leftarrow 0$, $t_n \leftarrow 0$, $m \leftarrow 0$, $T_m \leftarrow 0$.
 2: while $T_m < T_f$ do
 3:     $\boldsymbol{\mathcal{X}}_{[t_n,T_m]} \leftarrow \boldsymbol{x}$ (identity map)
 4:     while $\| \det \nabla \boldsymbol{\mathcal{X}}_{[t_n,T_m]} - 1 \|_\infty < TOL$ do
 5:         Sample $\boldsymbol{w}^n$ on grid points of $\boldsymbol{V}$.                              ▷ using (5.55)
 6:         $\mathcal{F}_{\boldsymbol{V}}[\tilde{\boldsymbol{u}}^n] = -\Delta^{-1} \nabla \times \mathcal{F}_{\boldsymbol{V}}[\tilde{\boldsymbol{w}}^n]$.
 7:         Compute $\partial^{\boldsymbol{b}} \tilde{\boldsymbol{u}}^n$ for $\boldsymbol{b} \in \{0,1\}^3$ in Fourier space. Define $\tilde{\boldsymbol{u}}^n(\boldsymbol{x})$.   ▷ using (5.24)
 8:         Compute $(\tilde{\boldsymbol{w}}^n \cdot \nabla)\tilde{\boldsymbol{u}}^n - (\tilde{\boldsymbol{u}}^n \cdot \nabla)\tilde{\boldsymbol{w}}^n$ on $\boldsymbol{V}$ and define $\partial_t \tilde{\boldsymbol{u}}^n(\boldsymbol{x})$.   ▷ using (5.25)
 9:         Define $\tilde{\boldsymbol{u}}(\boldsymbol{x}, t)$, by linear combination of spatial interpolants.   ▷ using (5.26)
10:         Compute $\partial^{\boldsymbol{b}} \tilde{\boldsymbol{X}}_{[t_n+\Delta t, t_n]}$ on $\boldsymbol{M}$ using RK3 integration of $\tilde{\boldsymbol{u}}$.   ▷ using (5.27)
11:         Update characteristic map $\boldsymbol{\mathcal{X}}_{[t_{n+1},T_m]} \leftarrow \mathcal{H}_{\boldsymbol{M}} \left[ \boldsymbol{\mathcal{X}}_{[t_n,T_m]} \circ \tilde{\boldsymbol{X}}_{[t_n+\Delta t, t_n]} \right]$.
12:         $t_{n+1} \leftarrow t_n + \Delta t$, $n \leftarrow n + 1$.
13:     end while
14:     $T_{m+1} \leftarrow t_n$, $m \leftarrow m + 1$.
15: end while
```

Remark 5.2.3. Since the discrete vorticity evaluation at line 5 in algorithm 3 will eventually produce aliasing errors from undersampling, we will usually introduce a low-pass filter or a Fourier truncation at lines 6 and 8. This is the effective scale cut-off of the velocity field governing the discrete flow. As a rule of thumb, we pick this low-pass filter to be the coarsest scale in the discretization and the grid $\boldsymbol{V}$ to be the finest scale with $\boldsymbol{M}$ at an intermediate scale. This is to ensure that the map grid has enough resolution to represent a short-time deformation generated by the filtered velocity field, and also that the $\boldsymbol{V}$ grid is fine enough so that the Hermite interpolation of the filtered velocity is accurate and preserves the divergence-free property.

5.3 Numerical Tests

In this section, we present several numerical tests computed using the CM method for the 3D incompressible Euler equations. The algorithm is implemented in C using OpenMP parallelization and Discrete Fourier transforms are performed using the FFTW library [45]. The tests in this section are performed on a computer with an AMD Ryzen 7 4800H CPU with 8 cores (16 threads) and 16GB of RAM; for these tests, a wallclock computation time is recorded. The larger FFT computations for the spectrum plots are performed on a cluster computer. The application of spatial adaptivity was not studied in this work, however, the general formulation allows for the use of adaptive grids; this has been studied for the Gradient-Augmented Level-Set methods in [75] which should be straightforwardly extendable to the CM methods.

5.3.1 Perturbed Antiparallel Vortex Tubes

The question of finite-time blow-up in the solution of the 3D incompressible Euler equations is an important open problem in mathematics. One extensively studied initial condition for potentially generating a finite-time blow-up are the perturbed antiparallel vortex tubes studied by Kerr in 1993 [72]. In the viscous case, for the Navier-Stokes equations, this initial condition evolves into a vortex reconnection in the process of which a topological change of the vortex cores occurs.

The initial condition can be constructed as the pullback of two antiparallel vortex tubes by a shear-deformation of the $[-2\pi, 2\pi]^3$ periodic domain. The initial vorticity field is antisymmetric across the $z = 0$ plane with each half-space containing a vortex tube of opposite orientation. We construct the initial condition $\boldsymbol{w}_0$ as follows. Consider the unperturbed vortex tube in the $z > 0$ half given by

$$\boldsymbol{\varphi}_+(x, y, z) = \exp\left(\frac{-r^2}{1-r^2} + r^4(1 + r^2 + r^4)\right)\begin{pmatrix}0\\1\\0\end{pmatrix} \quad \text{if} \quad r < 1, \tag{5.56}$$

and is the zero vector if $r \geq 1$. Here, r is the scaled distance from the vortex core given by

$$r(x, y, z) = R^{-1}\sqrt{(x - x_0)^2 + (z - z_0)^2}. \tag{5.57}$$

This forms a vortex tube oriented in the y-direction centered at $x = x_0$, $z = z_0$ and supported in a tube of radius R. One can check that this initial condition is divergence-free and therefore a valid vorticity field on the flat 3-torus. The pair of antiparallel vortex tubes is given by

$$\boldsymbol{\varphi}(x,y,z) = \boldsymbol{\varphi}_+(x,y,z) - \boldsymbol{\varphi}_+(x,y,-z). \tag{5.58}$$

The vortex tubes are perturbed by the following domain deformation:

$$\boldsymbol{T} : \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x + \delta_x \cos\left(\frac{\pi}{L_x} s(y)\right) \\ y \\ z + \delta_z \cos\left(\frac{\pi}{L_z} s(y)\right) \end{pmatrix}, \tag{5.59}$$

where

$$s(y) = y + L_y\delta_{y2} \sin\left(\pi y/L_y\right) + L_y\delta_{y1} \sin\left(y + L_y\delta_{y2}\sin\left(\pi y/L_y\right)\right). \tag{5.60}$$

The perturbed vortex tubes are then defined as the pullback $(\boldsymbol{T}^{-1})^*\boldsymbol{\varphi}$. A closed form expression can be obtained using $(\boldsymbol{T}^{-1})^*\boldsymbol{\varphi} = \nabla\boldsymbol{T}|_{\boldsymbol{T}^{-1}}\boldsymbol{\varphi}(\boldsymbol{T}^{-1})$ and the fact that $\boldsymbol{T}$ is a shear-deformation, and for a fixed y, $\boldsymbol{T}$ is simply a translation on the x-z plane.

The initial vorticity $\boldsymbol{w}_0$ used in [61, 72, 120] is defined as a filtered and rescaled version of the above perturbed vortex tubes given by

$$\boldsymbol{w}_0 = 8K * (\boldsymbol{T}^{-1})^*\boldsymbol{\varphi}. \tag{5.61}$$

The exact expression for the filter K might have been slightly different in references [72] and [61], here we use the filter K defined in Fourier space by $\hat{K}(\boldsymbol{\xi}) = \exp(-0.05(\xi_1^4 + \xi_2^4 + \xi_3^4))$ where ξ_i are the integer wave numbers. The specific parameters for the initial condition, taken from [61], are $R = 0.75$, $\delta_{y1} = 0.5$, $\delta_{y2} = 0.4$, $\delta_x = -1.6$, $\delta_z = 0$, $x_0 = 0$, $z_0 = 1.57$, $L_x = L_y = 4\pi$, $L_z = 2\pi$. Figure 5–3 shows a level-set surface of the initial condition.

We first perform a low-resolution simulation using a laptop computer. The simulation is carried out on a $\boldsymbol{M}$ grid of $64 \times 48 \times 32$ points and vorticity sampling is

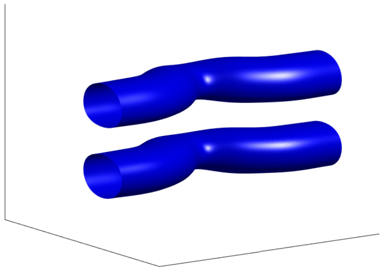
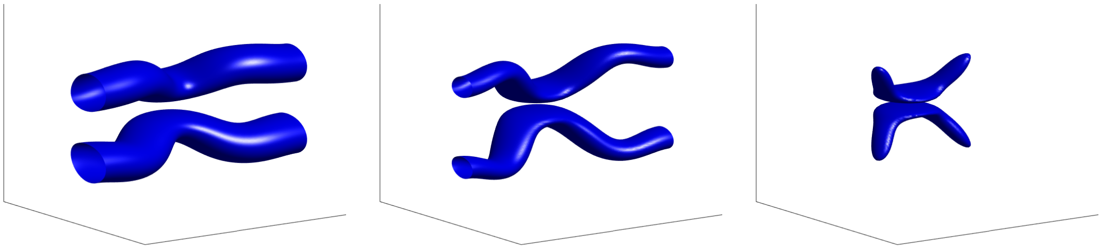


Figure 5–3: Initial vortex tubes for the Kerr initial condition [72]. The figure shows the level-set surface of $|\boldsymbol{w}_0|$ at 60% of the maximum value.

computed on a $\boldsymbol{V}$ grid of $96 \times 72 \times 48$ points; here the z-direction is most finely sampled as important higher frequencies are expected to be produced in that direction, similarly, the y-direction has the coarsest representation. The resulting Fourier series is truncated at a radius of 32, the time steps are fixed at 1/50 and the Jacobian determinant error tolerance $|\det \nabla \boldsymbol{\mathcal{X}}_B - 1|$ for the remapping is set at 10^{-3}. The initial condition is defined on a 128^3 grid, we then compute its mixed-partial derivatives in Fourier space in order to define a Hermite cubic interpolant. The simulation is run until time $t = 17$ requiring a total of 79 submaps, taking under an hour of wallclock computation time. Figure 5–4 shows the evolution of the vortex cores throughout the simulation and table 5–1 contains the energy and helicity conservation errors as well as total enstrophy and maximum vorticity and velocity evaluated at regular time intervals. The energy is defined as the squared L^2 norm of the velocity in the $[-2\pi, 2\pi]^3$ domain $\|\boldsymbol{u}\|^2_{L^2}$, the total enstrophy is $\|\boldsymbol{w}\|^2_{L^2}$ and the helicity is defined as the L^2 inner-product of the velocity and the vorticity, $H := (\boldsymbol{u}, \boldsymbol{w})_{L^2}$.

Remark 5.3.1. We note here that the Fourier truncation or filtering used in the computation of the velocity field from the sampled vorticity is not related to the 2/3 rule typically used in Fourier pseudo-spectral methods, whose purpose is to dealias spurious modes generated by the frequency convolutions when computing the nonlinear term in physical space in each time step. In the CM method, there is no direct time-stepping of the velocity and vorticity fields, at each step, the vorticity is reconstructed by direct sampling of the functional expression in equation (5.55).



(a) $t = 3$ (b) $t = 6$ (c) $t = 9$

Figure 5–4: Evolution of the vortex cores for the Kerr initial condition. We show level-set surfaces of $|\boldsymbol{w}|$ at 60% of the maximum value. Figures are generated using a 128^3 grid.

Since all map computations are carried out on a coarse grid $\boldsymbol{M}$, the purpose of the filtering is to ensure that the velocity field defined from the sampled vorticity is sufficiently band-limited so that the backward flow map it generates is regular enough to be accurately represented on $\boldsymbol{M}$. In the extreme cases where the vorticity field exhibits important subgrid scales, essentially discontinuous from a numerical point of view, this filtering can help prevent the Gibbs phenomenon from generating spurious oscillations in the entire domain. Ultimately, the size of the truncation would scale with the resolution of $\boldsymbol{M}$ to maintain consistency. This filtering is not always necessary, without filtering, the effective truncation of the Fourier series for the velocity will be the grid size of $\boldsymbol{M}$ as higher frequencies in the flow cannot be represented on $\boldsymbol{M}$. However, our numerical experiments suggest that a small amount of smoothing generates better results with more accurate energy and helicity conservation.

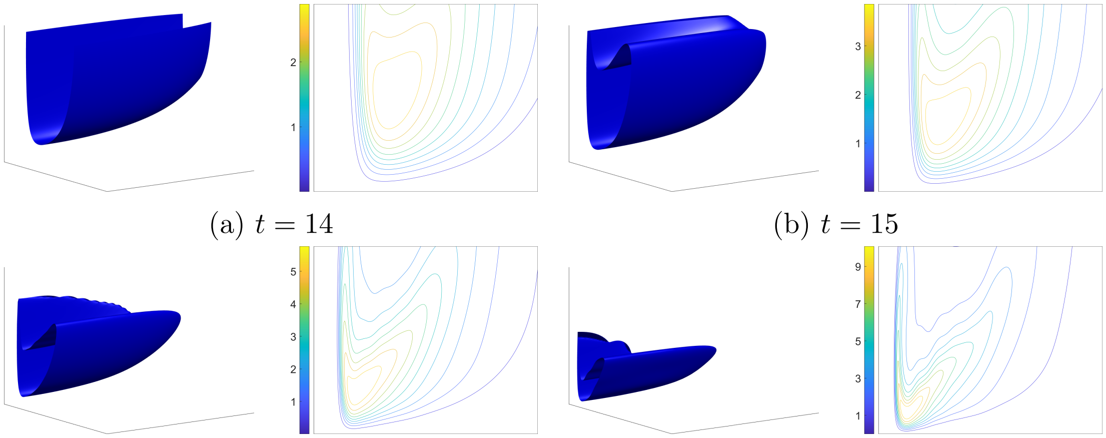
Remark 5.3.2. We note that the vorticity maxima $\|\boldsymbol{w}\|_{L^\infty}$ shown in table 5–1 are lower bounds since they are evaluated using a 256^3 grid. The actual maximum vorticity of the numerical solution is higher. Using the arbitrary resolution property of the method, we can refine this computation by recursively refining the vorticity sampling around the maxima. For instance, on a grid of N^3 points, we can locate the vorticity maximum on the grid and resample the vorticity in a region of size $3\Delta x$,

t	$\|\boldsymbol{w}\|_{L^2}^2$	$\|\boldsymbol{u}\|_{L^2}^2/\|\boldsymbol{u}_0\|_{L^2}^2 - 1$	$H - H_0$	$\|\boldsymbol{w}\|_{L^\infty}$	$\|\boldsymbol{u}\|_{L^\infty}$	n_{maps}	time (s)
0	67.2181	0.000	0.000	0.6691	0.7393	1	0
1	67.0990	1.341×10^{-6}	-5.619×10^{-14}	0.6666	0.7272	1	102
2	67.3909	1.744×10^{-6}	-1.717×10^{-13}	0.6715	0.7131	1	206
3	68.0925	1.989×10^{-7}	-2.849×10^{-13}	0.6841	0.7019	1	310
4	69.1954	-1.406×10^{-6}	3.333×10^{-13}	0.7387	0.7171	2	416
5	70.6844	-1.663×10^{-6}	5.680×10^{-13}	0.8181	0.7350	2	524
6	72.5405	-2.369×10^{-6}	8.913×10^{-13}	0.9059	0.7501	3	633
7	74.7460	-3.141×10^{-6}	1.129×10^{-12}	1.0046	0.7622	4	746
8	77.2912	-4.061×10^{-6}	1.218×10^{-12}	1.1179	0.7715	5	861
9	80.1845	-5.020×10^{-6}	-2.656×10^{-12}	1.2532	0.7785	6	981
10	83.4644	-6.245×10^{-6}	-7.347×10^{-12}	1.4181	0.7833	8	1108
11	87.2154	-7.793×10^{-6}	-4.307×10^{-12}	1.6296	0.7864	10	1240
12	91.5883	-1.012×10^{-5}	-3.304×10^{-12}	1.9121	0.7885	13	1382
13	96.8361	-1.246×10^{-5}	8.511×10^{-12}	2.2935	0.7920	18	1538
14	103.3978	-8.680×10^{-6}	3.923×10^{-11}	2.8495	0.8013	25	1719
15	112.1115	1.893×10^{-5}	4.295×10^{-11}	3.8549	0.8223	37	1935
16	124.5780	9.829×10^{-5}	-2.643×10^{-11}	5.6541	0.8545	56	2208
17	143.8254	2.633×10^{-4}	1.910×10^{-10}	9.1819	0.9052	79	2558

Table 5–1: Evolution of total enstrophy, energy conservation relative error (divided by initial energy), helicity conservation error (initial helicity is 0), maximum vorticity and velocity, number of remaps and wallclock computation time of the Kerr initial condition using CM method for 3D Euler. Grid resolutions: $64 \times 48 \times 32$ for $\boldsymbol{M}$, $96 \times 72 \times 48$ for $\boldsymbol{V}$, $\Delta t = 1/50$, Fourier truncation at radius 32, remapping Jacobian determinant tolerance at 10^{-3}. All data in this table are evaluated using a grid of resolution 256^3.

again using a grid of N^3 points. With $N = 256$, 3 iterations of the above procedure allows us to estimate the vorticity maximum to 3.8618, 5.7674 and 9.5185 for times 15, 16 and 17 respectively. This shows reasonable agreement with the high resolution reference computations performed in [61].

The functional definition of the vorticity field through the pullback by $\boldsymbol{\mathcal{X}}_B$ provides arbitrary resolution of the solution independently of the discretization grids. This allows us to zoom in on the solution, in particular for larger times t where the vorticity starts developing significant small scale features. Figure 5–5 shows zoomed views of the vortex tubes and contour plots of the vorticity intensity across the symmetry plane $y = 0$. We note that the vertical length of the viewed domain is 0.25,



(a) $t = 14$

(b) $t = 15$

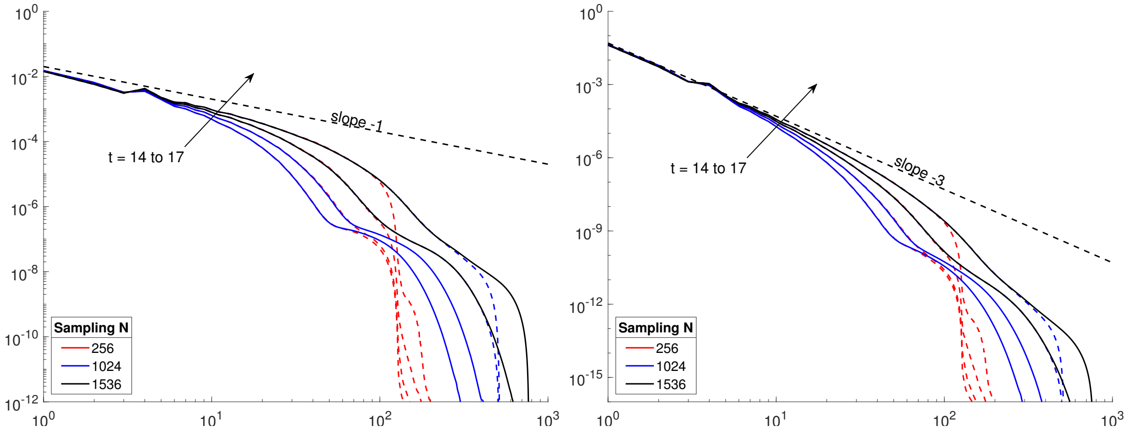
(c) $t = 16$

(d) $t = 17$

Figure 5–5: Zoomed view of the vortex core from $t = 14$ to 17 and contour plot of the vorticity strength $|\boldsymbol{w}|$ across the symmetry plane $y = 0$; 10 evenly distributed isoline values are shown for times 14, 15, 16, and 17, respectively. Viewed domain: $[-5.5, 0] \times [0, 1] \times [0, 0.25]$. Contour plots are generated using 512^2 2D grids.

a bit more than the width of a single cell of the map grid $\boldsymbol{M}$. The domain deformation at time $t = 17$ cannot be represented properly using a single map on the $\boldsymbol{M}$ grid, however, through the dynamic remapping method, the full deformation at time $t = 17$ can be represented using the composition of 79 short-time submaps defined on a coarse $64 \times 48 \times 32$ grid. The vorticity field $\tilde{\boldsymbol{w}}$ defined by pullback through the 79 submaps is therefore able to exhibit small scale features and high gradients as shown in figure 5–5.

As seen from table 5–1, the remapping routine is triggered more frequently as the simulation approaches $t = 17$, meaning that the Jacobian determinant error accumulates at an increasing rate in part due to a lack of spatial resolution. One often used measurement of the smoothness of the solution is the isotropic spectrum



(a) Enstrophy

(b) Energy

Figure 5–6: Enstrophy and energy spectra at times $t = 14, 15, 16$ and 17. Dotted black lines are the k^{-1} and k^{-3} curves, respectively. Successive finer sampling are shown, red curves are obtained from 256^3 sampling, blue curves from 1024^3 and black curves from 1536^3 sampling. The highest resolution sampling for each time t is drawn in full-line, coarser samplings for the same t are shown as dotted-lines.

of the Fourier series given by

$$E(k) = \frac{1}{2} \sum_{|\boldsymbol{\xi}| \in [k-\frac{1}{2}, k+\frac{1}{2})} \left|\mathcal{F}[\tilde{\boldsymbol{w}}](\boldsymbol{\xi})\right|^2 \tag{5.62a}$$

$$Z(k) = \frac{1}{2} \sum_{|\boldsymbol{\xi}| \in [k-\frac{1}{2}, k+\frac{1}{2})} \left|\mathcal{F}[\tilde{\boldsymbol{u}}](\boldsymbol{\xi})\right|^2 \tag{5.62b}$$

We plot the enstrophy and energy spectra of the solution at times $t = 14$ to 17 in figure 5–6.

The spectrum plots in figure 5–6 demonstrate a key property of the CM method: that the resolution scale of the map is not the dissipation scale of the vorticity solution, the vorticity field itself is not being dissipated in a viscous manner, and therefore can be resolved to arbitrary resolution. Indeed, although all computations were carried out on coarse grids and the Fourier support of the velocity field evolving the map is only a ball of radius 32, we can reconstruct the vorticity field by computing a fine grid pullback by evaluating the maps and applying equation (5.55). Figure 5–6 shows that the vorticity and velocity fields obtained this way have the expected decay in their Fourier transforms. These can be compared with Figs. 17 and 18 in [61] and general agreement is found. The vorticity field $\boldsymbol{w}^n = \left(\nabla \boldsymbol{\mathcal{X}}_{[t_n,0]}\right)^{-1} \boldsymbol{w}_0(\boldsymbol{\mathcal{X}}_{[t_n,0]}(\boldsymbol{x}))$, defined functionally (i.e. $\boldsymbol{\mathcal{X}}_B$ evaluated by interpolation and $\boldsymbol{w}_0$ by direct function evaluation), contains arbitrary fine scales until round-off errors dominate. The reason for this arbitrary resolution is twofold: Firstly, since the discretized quantity is the characteristic map, the pointwise definition of the vorticity field can in fact be viewed as Lagrangian. Indeed, consider a particle starting at position $\boldsymbol{y}$ at time $t = 0$, its position at time t is given by the forward map $\boldsymbol{X}_F(\boldsymbol{y}, t)$ and the vorticity field at the particle location is given by $\boldsymbol{w}(\boldsymbol{X}_F(\boldsymbol{y}, t), t) = \nabla \boldsymbol{X}_F(\boldsymbol{y}, t) \cdot \boldsymbol{w}_0(\boldsymbol{y})$. In order to obtain the vorticity at an Eulerian point $\boldsymbol{x}$ at time t, we plug in $\boldsymbol{y} = \boldsymbol{X}_B(\boldsymbol{x}, t)$ and use the inverse property (2.7b) to get $\boldsymbol{w}(\boldsymbol{x}, t) = (\nabla \boldsymbol{X}_B)^{-1} \boldsymbol{w}_0(\boldsymbol{X}_B)$. This means that by interpolating the discrete map $\boldsymbol{\mathcal{X}}_B$ at an Eulerian point $\boldsymbol{x}$, we are approximating a particle path and the associated local material deformation of a Lagrangian particle whose time t position is $\boldsymbol{x}$; the time t vorticity is then directly constructed from the initial condition by applying the material deformation. This can also be expressed using the error map studied in 5.2.2, where the numerical vorticity field can be written as the pullback of the exact vorticity by an error map $\boldsymbol{E}_B$: $\tilde{\boldsymbol{w}}^n = {\boldsymbol{E}_B}^* \boldsymbol{w}^n$. Therefore, since $\boldsymbol{E}_B$ is a $\mathcal{C}^1$ diffeomorphism, as long as the error is controlled, the pullback ${\boldsymbol{E}_B}^* \boldsymbol{w}^n$ will not destroy small scales. The functional definition of $\tilde{\boldsymbol{w}}^n$ by pullback therefore allows us to oversample the vorticity on a 1536^3 grid even though all computations were carried out on much coarser grids. For traditional Eulerian methods, in order to preserve these small scales and prevent large artificial dissipation, the vorticity field will have to be discretized and evolved on a 1536^3 grid throughout the entire computation. With the CM method, these scales are not lost to dissipation and can be obtained by a fine grid sampling. Secondly, the map error can be controlled using the submap decomposition method made possible by the group structure of the characteristic maps. Indeed, the map error $\boldsymbol{E}_B$ arises in part from the error in approximating $\mathit{SDiff}(U)$ by a finite-dimensional interpolation space

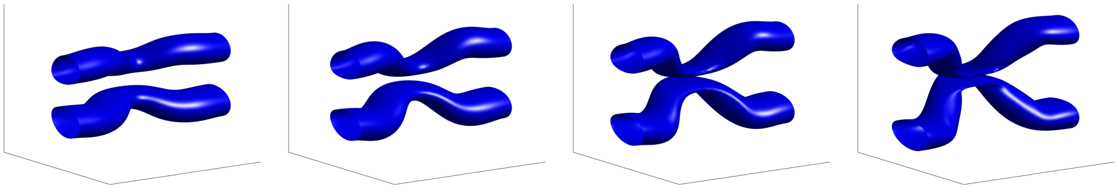
$\mathcal{V}$. As the flow develops, the small scale features in $\boldsymbol{X}_B$ not resolved in $\mathcal{V}$ increase, adding to the $\boldsymbol{E}_B$ error. Through the remapping method, this representation error is reset to 0 for each submap since the initial condition for each new submap is the identity map, which is represented exactly in $\mathcal{V}$. Appropriate remapping therefore guarantees that each submap $\boldsymbol{X}_{[T_{i+1},T_i]}$ can be well represented in $\mathcal{V}$ and that its numerical error remains in the asymptotic regime, i.e. the omitted small scales are not significant enough to pollute the large scale, lower frequency features which carry most of the energy. The resulting global-time map $\boldsymbol{\mathcal{X}}_B$ is obtained as the composition of n_{maps} submaps; $\boldsymbol{\mathcal{X}}_B$ can be seen as an element of $\mathcal{V}^{n_{maps}}$ and therefore is able to represent the small scales features generated by the long-time flow through the composition of coarse grid maps.

Another feature of the CM method is that we have access to the solution operator $\boldsymbol{X}_B$ of the advection under the velocity field $\boldsymbol{u}$, this means that we can evolve passively advected quantities at no additional computational cost. This has several applications such as tracking passively transported fluid quantities or solute densities or visualization of the fluid flow. As example we solve the following scalar advection problem using the initial vorticity strength $|\boldsymbol{w}_0|$ as initial condition:

$$(\partial_t + \boldsymbol{u} \cdot \nabla)\phi = 0 \tag{5.63a}$$

$$\phi(\boldsymbol{x}, 0) = |\boldsymbol{w}_0(\boldsymbol{x})|. \tag{5.63b}$$

From chapter 2, the solution to this advection equation is given by $\phi(\boldsymbol{x}, t) = |\boldsymbol{w}_0 \circ \boldsymbol{X}_{[t,0]}|$. This gives us the evolution of the initial vortex strength as a passively advected quantity. In figure 5–7, we show a level-set surface of ϕ at 60% the maximum value; this allows us the track the motion of the initial vortex core transported under the fluid flow. We note that this does not correspond to the evolution of the actual vortex core as the vortex stretching can play an important role in moving the location of the vortex core.



(a) $t = 4$ (b) $t = 8$ (c) $t = 12$ (d) $t = 16$

Figure 5–7: Transport of the initial vortex core. We show level-set surfaces of ϕ at 60% of the maximum value. Figures are generated using a 128^3 grid.

5.3.2 Perturbed Perpendicular Vortex Tubes

Another test we performed is the merging of two perturbed perpendicular vortex tubes inspired by the tests in [94]. The initial condition is constructed in a similar fashion as in section 5.3.1. The unperturbed vortex tube is given by (5.56) with $R = 0.5$, $x_0 = 0$, $z_0 = -1$. We apply the perturbation transformations $\boldsymbol{T} : [-2\pi, 2\pi]^3 \to [-2\pi, 2\pi]^3$ given by a sinusoidal shear deformation $\boldsymbol{T} : x \mapsto x - 0.5\sin(0.5y)$ and a reflection and translation map $\boldsymbol{R} : (x, y, z) \mapsto (y, x, z + 2)$. The two vortex tubes are then defined as

$$\boldsymbol{\varphi} = (\boldsymbol{T}^{-1})^* \boldsymbol{\varphi}_+ + (\boldsymbol{R}^{-1})^* (\boldsymbol{T}^{-1})^* \boldsymbol{\varphi}_+, \tag{5.64}$$

which corresponds to a sinusoidal vortex tube in the y-direction through $(x, z) = (0, -1)$ combined with a reflected tube in the x-direction through $(y, z) = (0, 1)$. The initial vorticity field is given as a scaled and filtered version of $\boldsymbol{\varphi}$ where the filter is the same as the one used in section 5.3.1,

$$\boldsymbol{w}_0 = 24K * \boldsymbol{\varphi}. \tag{5.65}$$

Figure 5–8 shows a level-set surface of this initial condition.

These two vortex tubes will start rotating around each other, creating high vorticity gradients and significant small scale features. In the viscous case, the vortices are expected to collide at the intersection. Figure 5–9 shows the evolution of the vortex tubes computed from a simulation using a 48^3 grid for both $\boldsymbol{M}$ and $\boldsymbol{V}$, the

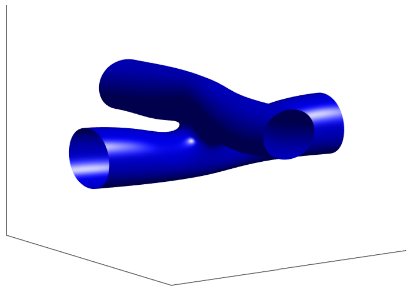


Figure 5–8: Initial condition for the perpendicular vortex tubes test. The figure shows the level-set surface of $|\boldsymbol{w}_0|$ at 0.5402, which is 60% of the maximum value.

Hermite cubic interpolation of the velocity field uses a 96^3 grid to ensure sufficient smoothness. The time step Δt is fixed at $1/50$, Fourier truncation radius at 32 and the Jacobian determinant error tolerance at 10^{-3}. Due to the significant localized small scale features, the vorticity computation uses the adaptive sampling described in section 5.2.3. The mollifier μ_h is given by a tensor product of a 1D function $\cos^2\left(\frac{\pi s}{2h}\right)$ in each cell. The cell integral (5.52) is computed by numerical quadrature using equidistributed sample points in each cell. The number of sample points per cell is at minimum 2 points per dimension and for each cell, this number is allowed to increases adaptively depending on the range and total variation of $\boldsymbol{w}$ in each cell. The total number of sample points is capped at 192^3 at which point all cells have their sample number rescaled down proportionally. The results of this test until time $t = 12$ are presented in table 5–2. Figure 5–9 shows the evolution of the vortex cores until time $t = 9$. To better visualize the flow, we also include in figure 5–10 the scalar advection of the initial vorticity strength $\phi_0 = |\boldsymbol{w}_0|$ as given by equation (5.63).

The small scale features that develop in the flow after $t = 9$ become very fine and make 3D visualization difficult. We present instead three 2D contour plots on the xy-, xz- and yz- planes, respectively. Figures 5–11 and 5–12 show zoomed views on the lateral cuts of the vorticity magnitude and the advected initial vorticity strength.

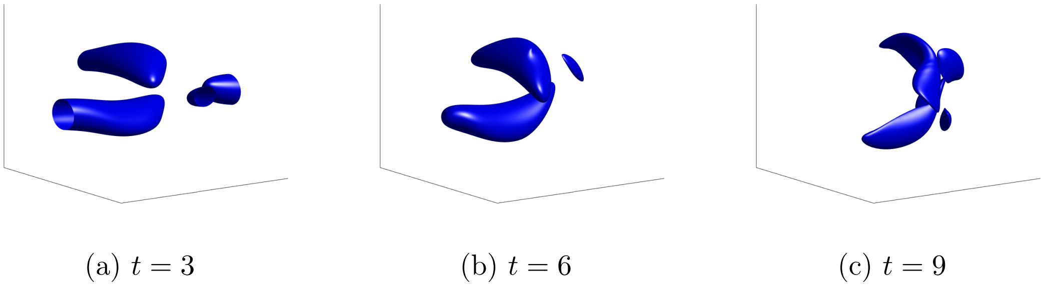

(a) $t = 3$ (b) $t = 6$ (c) $t = 9$

Figure 5–9: Evolution of the vortex cores for the perpendicular vortex tubes test. We show the level-set surfaces of $|\boldsymbol{w}|$ at 60% of the maximum value. Figures are generated using a 324^3 grid.

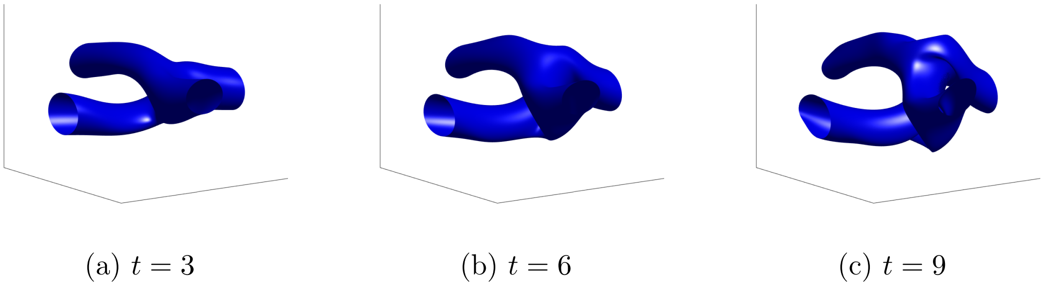
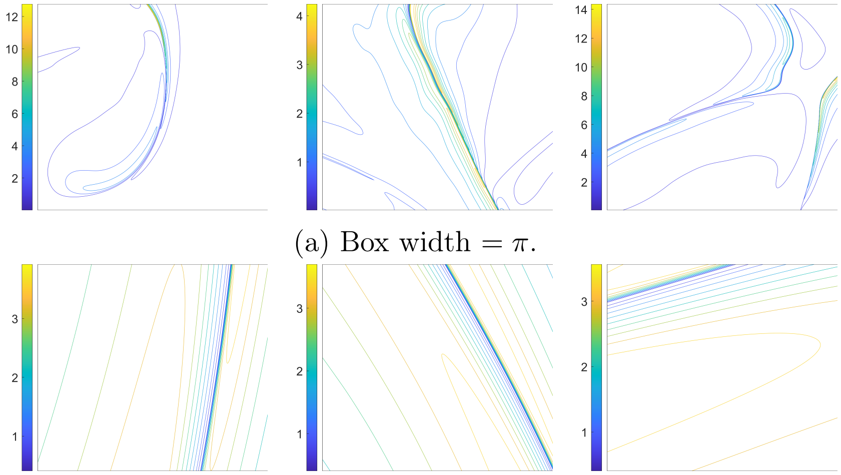

(a) $t = 3$ (b) $t = 6$ (c) $t = 9$

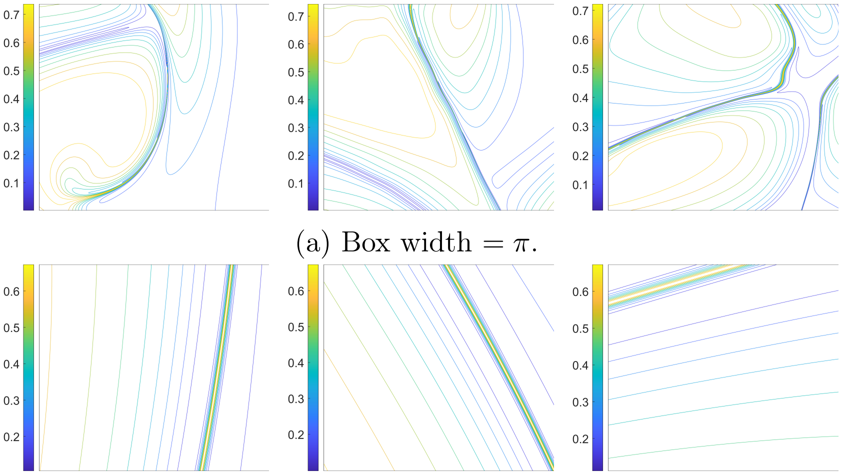
Figure 5–10: Transport of the initial vortex core. We show level-set surfaces of ϕ at 60% of the maximum value. Figures are generated using a 324^3 grid.



(a) Box width $= \pi$.

(b) Box width $= \frac{\pi}{16}$.

Figure 5–11: Contour plot of vorticity strength $|\boldsymbol{w}|$ at time $t = 12$, centered at $(x, y, z) = (2.15, 1.5, 1.425)$; 10 isoline values evenly distributed in the range of each frame are shown. From left to right, contours on: xy- plane, xz-plane and yz-plane. Figures are produced using a 1024^2 2D grid.



(a) Box width $= \pi$.

(b) Box width $= \frac{\pi}{16}$.

Figure 5–12: Contour plot of the transported initial vorticity ϕ at time $t = 12$, centered at $(x, y, z) = (2.15, 1.5, 1.42)$, 10 isoline values evenly distributed in the range of each frame are shown. From left to right, contours on: xy- plane, xz-plane and yz-plane. Figures are produced using a 1024^2 2D grid.

t	$\|\boldsymbol{w}\|_{L^2}^2$	$\|\boldsymbol{u}\|_{L^2}^2/\|\boldsymbol{u}_0\|_{L^2}^2 - 1$	$H/H_0 - 1$	$\|\boldsymbol{w}\|_{L^\infty}$	$\|\boldsymbol{u}\|_{L^\infty}$	n_{maps}	time (s)
0	125.7910	0.000	0.000	0.9004	0.9684	1	0
1	126.9290	-7.816×10^{-4}	2.296×10^{-3}	0.9183	0.9644	1	187
2	130.3522	-3.043×10^{-3}	8.889×10^{-3}	1.0317	0.9478	1	381
3	136.0808	-6.604×10^{-3}	1.901×10^{-2}	1.1361	0.9331	2	586
4	144.1831	-1.117×10^{-2}	3.160×10^{-2}	1.2187	0.9431	3	801
5	154.8535	-1.636×10^{-2}	4.557×10^{-2}	1.2885	0.9794	4	1027
6	168.4914	-2.178×10^{-2}	5.988×10^{-2}	1.3653	1.0208	8	1568
7	185.7193	-2.707×10^{-2}	7.354×10^{-2}	1.4806	1.0495	14	2386
8	207.4032	-3.201×10^{-2}	8.561×10^{-2}	1.6472	1.0636	22	3380
9	234.9474	-3.648×10^{-2}	9.518×10^{-2}	2.0750	1.0697	30	4691
10	271.2813	-4.048×10^{-2}	1.015×10^{-1}	3.3802	1.0602	40	6442
11	322.7244	-4.403×10^{-2}	1.038×10^{-1}	6.7721	1.0433	51	8489
12	400.7262	-4.733×10^{-2}	1.024×10^{-1}	14.1254	1.0929	64	10391

Table 5–2: Evolution of total enstrophy, energy conservation relative error (divided by initial energy), helicity conservation relative error (divided by initial helicity), maximum vorticity and velocity, number of remaps and wallclock computation time of the perturbed perpendicular vortex tubes initial condition using CM method for 3D Euler. Grid resolutions: 48^3 for $\boldsymbol{M}$, 48^3 for $\boldsymbol{V}$, $\Delta t = 1/50$, adaptive sampling with mollifier convolution is used for the vorticity, Fourier truncation at radius 32, remapping Jacobian determinant tolerance at 10^{-3}. All data in this table are evaluated using a grid of resolution 256^3.

The highest zoom shows a domain of width $\pi/16$ corresponding to 1/64 of the computational domain, i.e. smaller than a single cell of the advection grid. In figure 5–11, we can see a presence of a high vorticity gradient at time $t = 12$ and the formation of a vortex sheet. Figure 5–12 shows more clearly the material deformations which lead to this high gradient. Indeed, the advected initial vorticity shows that two separate level-sets of the vorticity were pushed close together by the flow. The yellow region squeezed between the blue curves in 5–12b was formed from the flattening of the initial vortex tubes. However, the absence of a highly concentrated vorticity peak in 5–11 in contrast to 5–12 suggests that the vorticity direction was not fully aligned with the strain tensor eigenvector of the largest eigenvalue indicating that the vortex stretching term attenuated the vorticity gradient in this region. This nicely illustrates that vortex stretching can be quantified locally.

5.4 Concluding Remarks for Chapter 5

In this chapter, we extended the CM framework for the 2D Euler equations in chapter 4 to the 3D case. For this purpose, we presented a geometric reformulation of the framework based on the conservation of the vorticity 2-form. In doing so, we were able to seamlessly incorporate the vorticity stretching term to compute the evolving vorticity through pullback by the map, thus preserving the unique numerical properties of the CM method for the linear advection and 2D Euler equations. Indeed, we've shown in this chapter that the generalized method maintains an arbitrary resolution of the solution, does not introduce artificial viscous dissipation and can improve computational efficiency using the submap decomposition approach based on the group structure of the characteristic maps. The error estimates based on expressing the true characteristic map as a composition of the numerical map with an error map further clarify the advective nature of the vorticity error: the numerical error does not arise from an artificial viscous or hyperviscous term, but rather from modified characteristic trajectories. The numerical results in section 5.3 show that due to the advective nature of the error, the fine scale features of the solution are indeed preserved, since by successive fine sampling of the solution, the enstrophy and energy spectra from the reconstructed velocity and vorticity fields contain the expected higher frequency modes rather than a sharp decay symptomatic of grid-scale viscous dissipation. This property, illustrated in figure 5–6, is a remarkable and unique feature of the method; this not only sets the CM approach apart from others but also warrants further investigation for a deeper understanding of this framework in more general advection dominated problems.

CHAPTER 6
Conclusion and Outlook

In this thesis, we studied the mathematical framework of the Characteristic Mapping method and its application to various transport dominated equations. We used the linear advection equation as starting point of our investigation and developed a more general framework through a geometric formulation of advection dominated evolution equations. In particular, the use of the characteristic map as the solution operator for the Lie-advection of differential forms allowed for solutions to be expressed as the pullback of the initial condition by the map; this resulted in a numerical method which provides a non-dissipative evolution of the solution at arbitrary spatial resolution. The proof of concept in this thesis showed the novelty and uniqueness of this method and revealed many interesting directions for future research. Here, we make some preliminary investigations on two directions of interest and conclude with some final remarks.

6.1 Outlook

The CM framework opens many interesting directions of future research. From a numerical analysis point of view, the CM framework employs a new solution method where the advected quantity is in fact never discretized, but defined in functional form and can be evaluated at arbitrary locations through pullback by the map. This approach may require new methods of analysis since numerical solutions do not stay in the same finite-dimensional vector space at all times. From a scientific computing point of view, the work presented here provides a proof of concept of the CM method for the general transport and for the Euler equations. However, many applications require the inclusion of source terms and boundary conditions; this will require further extensions to the current framework. We present here a short overview of these future outlooks for the CM method.

6.1.1 Inhomogeneous Equations

The CM methods used for the Lie-advections in chapters 3 to 5 were all applied to homogeneous equations of the form $(\partial_t + \mathcal{L}_{\boldsymbol{u}})\phi = 0$. Here, we make here a few comments on the inclusion of source terms within the CM framework. Consider the inhomogeneous linear advection equations with velocity $\boldsymbol{u}$ given:

$$(\partial_t + \mathcal{L}_{\boldsymbol{u}})\phi = f_t, \tag{6.1a}$$

$$\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}), \tag{6.1b}$$

for some time-dependent differential form f_t of the same degree as ϕ. Here, the subscript t on dependent variables denotes evaluation at time t. Using the solution operator $\boldsymbol{X}_{[t,0]}$ and Duhamel's principle for linear equations, we have that solutions are given by

$$\phi(\cdot, t) = {\boldsymbol{X}_{[t,0]}}^* \phi_0 + \int_0^t {\boldsymbol{X}_{[t,s]}}^* f_s ds. \tag{6.2}$$

For general forcing terms f_t, this breaks the characteristic mapping structure to some degree as the solution is no longer obtained directly from pullback. Instead, the presence of the time integral implies the need to discretize the "accumulated source term" which we denote by $F_{[t,0]}$:

$$F_{[t,0]} := \int_0^t {\boldsymbol{X}_{[t,s]}}^* f_s ds. \tag{6.3}$$

Numerically, ${\boldsymbol{X}_{[t,0]}}^* \phi_0$ and $F_{[t,0]}$ do not arise from the same operations and errors in $F_{[t,0]}$ depend on the source term as well as the velocity integration. Therefore, incoherence between $\boldsymbol{X}_{[t,0]}$ and $F_{[t,0]}$ may arise due to numerical errors reducing the self-consistency of the method. In other words, without a source term, the modified equation is obtained by modifying only the advection velocity $\boldsymbol{u}$, whereas with the above source term integration, error terms depending on f_t will now emerge.

Nonetheless, the submap decomposition method can still be applied to limit this error. Indeed, we note that

$$\int_{T_1}^{T_2} \boldsymbol{X}_{[t,s]}{}^* f_s ds = \boldsymbol{X}_{[t,T_2]}{}^* \int_{T_1}^{T_2} \boldsymbol{X}_{[T_2,s]}{}^* f_s ds. \tag{6.4}$$

Following the previous notation, we denote the last integral above by $F_{[T_2,T_1]}$, this gives us the following submap decomposition for the accumulated source term:

$$F_{[t,0]} = \sum_{i=1}^{m} \boldsymbol{X}_{[T_m,T_{m-1}]}{}^* \cdots \boldsymbol{X}_{[T_{i+1},T_i]}{}^* F_{[T_i,T_{i-1}]}, \tag{6.5}$$

for $0 = T_0 < T_1 < \cdots < T_m = t$. Numerically, this decomposition limits the interpolation of the integrated source term to short-time subintervals, doing so prevents to some extent the accumulation of numerical errors.

In each subinterval $[T_{i-1}, T_i]$, $F_{[t,T_{i-1}]}$ obeys the inhomogeneous equation

$$(\partial_t + \mathcal{L}_{\boldsymbol{u}}) F_{[t,T_{i-1}]} = f_t, \tag{6.6}$$

and the corresponding Gradient-Augmented Level-Set scheme is given by

$$F_{[t+\Delta t,T_{i-1}]} = \mathcal{H}\left[F_{[t,T_{i-1}]} \circ \tilde{\boldsymbol{X}}_{[t+\Delta t,t]}\right] + \mathcal{H}\left[\int_t^{t+\Delta t} \tilde{\boldsymbol{X}}^*_{[t+\Delta t,s]} f_s ds\right]. \tag{6.7}$$

The above describes a CM method for linear advection with an arbitrary source term f_t. For certain classes of source terms, in particular when f_t can be in part expressed as an advected quantity, the approach can be further improved. We consider the following Euler equations with a forcing term of fixed vertical direction dz scaled by an advected 0-form f_t as is in the case of the Boussinesq approximation with zero viscosity and thermal conductivity.

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + f_t \nabla z, \tag{6.8a}$$

$$\partial_t f_t + \boldsymbol{u} \cdot \nabla f_t = 0, \tag{6.8b}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{6.8c}$$

The corresponding vorticity equation written in differential forms is

$$(\partial_t + \mathcal{L}_{\boldsymbol{u}})\omega = df_t \wedge dz, \tag{6.9a}$$

$$(\partial_t + \mathcal{L}_{\boldsymbol{u}})f_t = 0, \tag{6.9b}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{6.9c}$$

The Duhamel expansion of the solution gives the following vorticity at time t:

$$\omega_t = \boldsymbol{X}_{[t,0]}{}^*\omega_0 + \int_0^t \boldsymbol{X}_{[t,s]}{}^*(df_t \wedge dz)ds. \tag{6.10}$$

Using the fact that f is Lie-advected, i.e. $f_s = \boldsymbol{X}_{[s,0]}{}^* f_0$, after commuting the pullback with d and grouping terms, we get

$$\omega_t = \boldsymbol{X}_{[t,0]}{}^*\omega_0 + d\left(\boldsymbol{X}_{[t,0]}{}^* f_0\right) \wedge \int_0^t d\left(\boldsymbol{X}_{[t,s]}{}^* z\right) ds, \tag{6.11}$$

where $\boldsymbol{X}_{[t,s]}{}^* z$ is just the z component of the map and can be integrated using the same method as the map. This allows us to create a solution operator for the accumulated forcing since the initial condition f_0 has been factored out of the time-integration; again, the numerical error results in a modified equation where only the velocity field is changed. Hence the characteristic structure and arbitrary resolution properties can be preserved by discretizing the "accumulated source term operator" $G_{[t,0]}$:

$$G_{[t,0]} := \int_0^t \boldsymbol{X}_{[t,s]}{}^* z ds, \tag{6.12}$$

and the solution can be reconstructed as $\omega_t = \boldsymbol{X}_{[t,0]}{}^*\omega_0 + d\left(\boldsymbol{X}_{[t,0]}{}^* f_0\right) \wedge dG_{[t,0]}$.

The extension to more general forcing terms and other equations requires further investigation. The inclusion of a viscosity term remains a challenge since the Laplacian operator involves a metric and does not commute with pullbacks; the Eulerian to Lagrangian change of coordinates produces instead a Laplace-Beltrami operator bringing additional numerical difficulties. The diffusion-driven CM method in chapter 3 could be used as an alternative for including viscosity, however, its implementation in three-dimensional space is likely to become impractical. The formulation of the CM method for terms that are not Lie-advected and the involvement of a metric remain important directions for future investigations.

6.1.2 Kelvin-Filtered Turbulence Models and the Euler-α Equations

The CM method is closely related to the Kelvin-Filtered turbulence models and the Euler-α equations, also known as the three-dimensional Camassa-Holm equations. We refer to the review [42] on the Navier-Stokes-α equations for the theory and analysis of this type of nonlinearly dispersive equations. The main idea behind this dispersive approximation can be summarized as applying a regularization only on the velocity field $\boldsymbol{u}$ which Lie-advects the vorticity field. We consider two velocity fields: $\boldsymbol{u}$, the true solution to the Euler-α equations, along with a regularized version $\boldsymbol{u}_\epsilon$, often given by a filtering of the solution $\boldsymbol{u}_\epsilon = \mu_\epsilon * \boldsymbol{u}$, where μ_ϵ is some smooth spatial filter. The equations can then be derived by requiring that the vorticity $\boldsymbol{w} = \nabla \times \boldsymbol{u}$ satisfies the Kelvin circulation theorem under the regularized velocity $\boldsymbol{u}_\epsilon$, that is

$$(\partial_t + \mathcal{L}_{\boldsymbol{u}_\epsilon})\omega = 0. \tag{6.13}$$

Plugging in $\omega = d\boldsymbol{u}^\flat$ and commuting d with Lie-derivative, we get

$$d\left(\partial_t \boldsymbol{u}^\flat + \mathcal{L}_{\boldsymbol{u}_\epsilon}\boldsymbol{u}^\flat\right) = d\left(\partial_t \boldsymbol{u}^\flat + \iota_{\boldsymbol{u}_\epsilon} d\boldsymbol{u}^\flat + d\iota_{\boldsymbol{u}_\epsilon}\boldsymbol{u}^\flat\right) = 0. \tag{6.14}$$

This means that $\partial_t \boldsymbol{u}^\flat + \iota_{\boldsymbol{u}_\epsilon} d\boldsymbol{u}^\flat + d\iota_{\boldsymbol{u}_\epsilon}\boldsymbol{u}^\flat$ is a closed 1-form, or in coordinates

$$\partial_t \boldsymbol{u} + (\nabla \times \boldsymbol{u}) \times \boldsymbol{u}_\epsilon + \nabla(\boldsymbol{u}^T \boldsymbol{u}_\epsilon) = -\nabla p, \tag{6.15}$$

for some scalar pressure p.

We note that $(\iota_{\boldsymbol{u}_\epsilon} d\boldsymbol{u}^\flat)^\sharp = (\nabla \times \boldsymbol{u}) \times \boldsymbol{u}_\epsilon = (\boldsymbol{u}_\epsilon \cdot \nabla)\boldsymbol{u} - (\nabla \boldsymbol{u}^T)\boldsymbol{u}_\epsilon$. Therefore, using $\nabla(\boldsymbol{u}^T \boldsymbol{u}_\epsilon) = (\nabla \boldsymbol{u}^T) \cdot \boldsymbol{u}_\epsilon + (\nabla \boldsymbol{u}_\epsilon^T) \cdot \boldsymbol{u}$, the above simplifies to

$$\partial_t \boldsymbol{u} + (\boldsymbol{u}_\epsilon \cdot \nabla)\boldsymbol{u} + (\nabla \boldsymbol{u}_\epsilon^T) \cdot \boldsymbol{u} = -\nabla p. \tag{6.16}$$

We note that when $\boldsymbol{u}_\epsilon = \boldsymbol{u}$, the term $(\nabla \boldsymbol{u}_\epsilon^T) \cdot \boldsymbol{u}$ becomes $\frac{1}{2}\nabla(\boldsymbol{u}^T \boldsymbol{u})$ which is closed and can be absorbed into the pressure term; in the Euler-α equations, this term is required to correct errors in the conservation of circulation due to modifying the advection velocity.

This advection modification also changes the energy conservation of the equation as $\|\boldsymbol{u}\|_{L^2}$ is no longer conserved. Instead, assuming $\boldsymbol{u}_\epsilon$ is divergence-free, we take an L^2 inner-product of the equation with $\boldsymbol{u}_\epsilon$:

$$(\partial_t \boldsymbol{u}, \boldsymbol{u}_\epsilon)_{L^2} + \big((\boldsymbol{u}_\epsilon \cdot \nabla)\boldsymbol{u} + (\nabla \boldsymbol{u}_\epsilon^T) \cdot \boldsymbol{u}, \boldsymbol{u}_\epsilon\big)_{L^2} = (\nabla p, \boldsymbol{u}_\epsilon)_{L^2} = 0. \tag{6.17}$$

We note that integrating by parts and assuming divergence-free, the trilinear form

$$B(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) := \big((\boldsymbol{v}_2 \cdot \nabla)\boldsymbol{v}_1 + (\nabla \boldsymbol{v}_2^T) \cdot \boldsymbol{v}_1, \boldsymbol{v}_3\big)_{L^2} \tag{6.18}$$

can be rewritten as $B(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) = (\nabla \boldsymbol{v}_1, \boldsymbol{v}_2 \otimes \boldsymbol{v}_3 - \boldsymbol{v}_3 \otimes \boldsymbol{v}_2)_{L^2}$, i.e. B is antisymmetric in its last two entries. Therefore, we obtain

$$(\partial_t \boldsymbol{u}, \boldsymbol{u}_\epsilon)_{L^2} + B(\boldsymbol{u}, \boldsymbol{u}_\epsilon, \boldsymbol{u}_\epsilon) = (\partial_t \boldsymbol{u}, \boldsymbol{u}_\epsilon)_{L^2} = 0. \tag{6.19}$$

Lastly, assuming that $\boldsymbol{u}_\epsilon$ is obtained from $\boldsymbol{u}$ through a linear, self-adjoint filtering operator K_ϵ, i.e. that $(\partial_t \boldsymbol{u}, \boldsymbol{u}_\epsilon)_{L^2} = (\partial_t \boldsymbol{u}, K_\epsilon \boldsymbol{u})_{L^2} = (\partial_t K_\epsilon \boldsymbol{u}, \boldsymbol{u})_{L^2} = \frac{1}{2}\partial_t(\boldsymbol{u}, \boldsymbol{u}_\epsilon)_{L^2}$, we get that the conserved "energy" is the L^2 inner-product $(\boldsymbol{u}, \boldsymbol{u}_\epsilon)_{L^2}$.

This type of nonlinear dissipation is relevant to the CM method since the numerical vorticity solution in the CM method is constructed by pullback, $\tilde{\omega} = \boldsymbol{\mathcal{X}}_{[t,0]}{}^* \omega_0$. If the numerical characteristic map $\boldsymbol{\mathcal{X}}_B$ is indeed achieved by the backward-in-time flow of some velocity field $\tilde{\boldsymbol{u}}_\epsilon$, then we have that the numerical vorticity $\tilde{\omega}$ is Lie-advected by $\tilde{\boldsymbol{u}}_\epsilon$. Equivalently, the corresponding velocity solution $\tilde{\boldsymbol{u}}$ with $\tilde{\omega} = d\tilde{\boldsymbol{u}}^\flat$

satisfies

$$\partial_t \tilde{\boldsymbol{u}} + (\tilde{\boldsymbol{u}}_\epsilon \cdot \nabla)\tilde{\boldsymbol{u}} + (\nabla \tilde{\boldsymbol{u}}_\epsilon^T) \cdot \tilde{\boldsymbol{u}} = -\nabla p. \tag{6.20}$$

It remains to see if the transporting velocity $\tilde{\boldsymbol{u}}_\epsilon$ is divergence-free and is related to the transported solution $\tilde{\boldsymbol{u}}$ by a linear, self-adjoint filtering operator K_ϵ. If this is the case, numerically, this implies a more accurate energy conservation in the solution since

$$\begin{aligned} \|\tilde{\boldsymbol{u}}\|_{L^2} &= (\tilde{\boldsymbol{u}}_\epsilon, \tilde{\boldsymbol{u}})_{L^2} + (\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{u}}_\epsilon, \tilde{\boldsymbol{u}})_{L^2} \\ &= \|\boldsymbol{u}_0\|_{L^2} + \mathcal{O}(\|(I - K_\epsilon)\tilde{\boldsymbol{u}}\|_{L^2}\|\tilde{\boldsymbol{u}}\|_{L^2} + \|(I - K_\epsilon)\tilde{\boldsymbol{u}}_0\|_{L^2}\|\tilde{\boldsymbol{u}}_0\|_{L^2}). \end{aligned} \tag{6.21}$$

Compared to the typical exponential decay of energy for linearly dissipative methods, the energy conservation error here is *a priori* independent of time. Heuristically, this indicates that loss of energy comes from the truncation of the small scales by the operator K_ϵ, i.e. from the inability to spatially resolve the fine scales of the advection. The key assumption for equation (6.1.2) is that the advection velocity $\tilde{\boldsymbol{u}}_\epsilon$ is divergence-free and is obtained from the solution $\tilde{\boldsymbol{u}}$ through an operator K_ϵ which is linear and self-adjoint. For of the CM method, this is not entirely the case. Although the backward map is indeed evolved by a filtered version of the solution $\tilde{\boldsymbol{u}}$, the time-stepping for the map $\boldsymbol{\mathcal{X}}_B$ involves the projection $\mathcal{H}_{\boldsymbol{G}}[\boldsymbol{\mathcal{X}}_{[t_n,0]} \circ \tilde{\boldsymbol{X}}_{[t_{n+1},t_n]}]$ which adds additional error terms to the effective advection velocity. Ultimately, the numerical maps $\boldsymbol{\mathcal{X}}_B^n$ are only computed at discrete time steps t_n; it is unclear whether there exists some velocity $\tilde{\boldsymbol{u}}_\epsilon(\boldsymbol{x}, t)$ whose time t_n backward flow map corresponds to $\boldsymbol{\mathcal{X}}_B^n$ and whether the relation $\tilde{\boldsymbol{u}}_\epsilon = K_\epsilon \tilde{\boldsymbol{u}}$ can hold. However, we think that these properties can be closely approximated and the error can be kept small if the spatial resolution is high enough; therefore the CM method should also have controlled error in energy conservation. The non-dissipative nature of the CM method is an important property for future investigation and further analysis is needed to accurately evaluate the energy conservation qualities of the method.

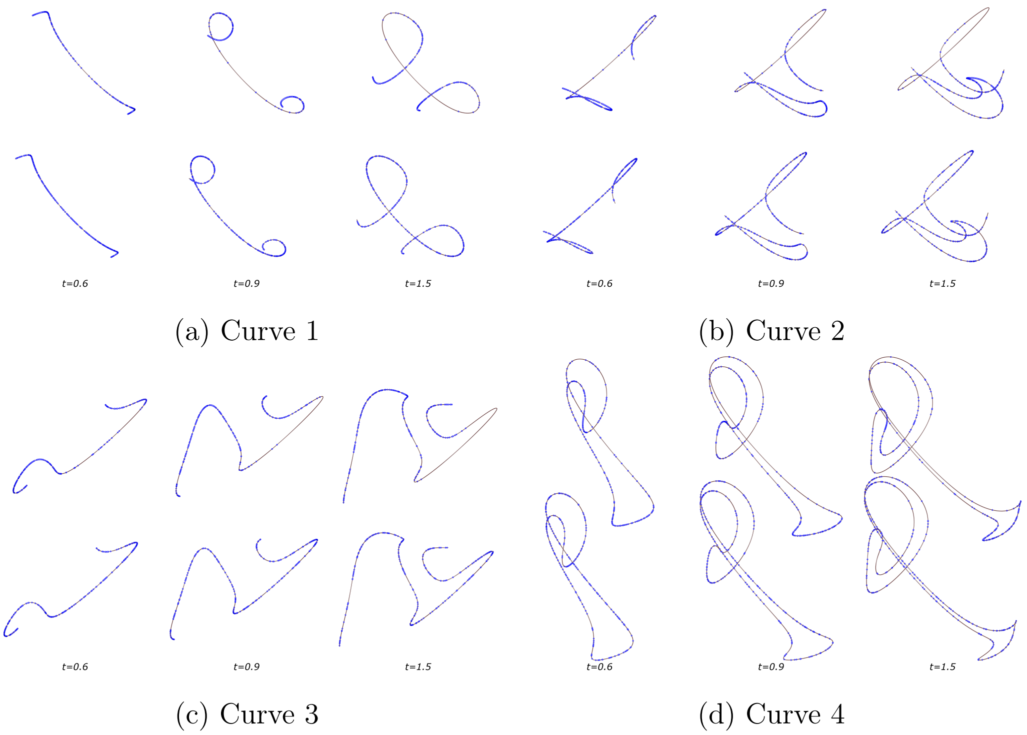
6.2 Concluding Remarks

In this thesis, we studied the mathematical framework of the Characteristic Mapping method and its application to various problems in scientific computing, ranging from particle management and reparametrization of moving surfaces to the solution of the incompressible Euler equations in two and three-dimensional space. In chapter 2, we examined some preliminary notions required for the formulation of the CM method and gave a brief outline of the method for the linear advection equation. We also reviewed some differential geometry tools that we used for the design and analysis of the CM methods presented in later chapters. In particular, the relation between Lie-advection and conservation laws through Cartan's homotopy formula allowed us to generalize the CM method to a wide family of evolution equations. In chapter 3, we applied the CM method to a density redistribution problem; by writing the diffusion equation on probability densities as a conservation law, we obtained a Lie-advection equation. The characteristic map associated with this transport equation then serves as a density redistribution map. We applied this method to the evolution of parametrized surfaces; by composing the time-dependent parametrization with the parametric space redistribution map, the area-element of the surface is uniformized, yielding an equiareal parametrization of an evolving surface. In chapter 4, we applied the CM method to the incompressible Euler equations in two-dimensional space. In the 2D case, the scalar vorticity is an advected quantity, the vorticity field at time t is obtained by function composition of the initial condition with the characteristic map. The velocity field is obtained from the Biot-Savart law and extended in time using a Lagrange interpolant. We also derived error estimates for the method and drew a link between the corresponding modified equations and the Euler-α models. These error estimates were supported by numerical convergence tests. Furthermore, comparisons were made with the Cauchy-Lagrangian method in [97] and a vortex merger test was also included to show the arbitrary spatial resolution provided by the functional representation and the submap decomposition. This method was generalized to the three-dimensional case in chapter 5. In 3D, we took a geometric approach by representing the vorticity field as a differential 2-form. According to the Kelvin circulation theorem, the vorticity 2-form is Lie-advected by the velocity field and therefore can be computed as the pullback of the initial vorticity by the characteristic map. This allowed us to include the vorticity stretching term as an effect of material deformation rather than as a separate source term, thus preserving the characteristic structure of the method. We performed error estimates based on the group structure of the maps which further underlines the advective rather than dissipative nature of the numerical error. The numerical tests in chapter 5 provide further evidence to the low-dissipation nature of the error as spectrum plots of the reconstructed vorticity and velocity show that higher frequency modes are preserved.

In summary, the CM framework brings a novel geometric approach to various transport problems in scientific computing. This approach provides several unique numerical properties to the method: The computation of a solution operator allows for the coherent and simultaneous evolution of multiple transported quantities. The functional discretization approach provides arbitrary resolution of the solution and preserves the back-to-label symmetry. The submap decomposition, made possible by the map's group structure, allows for a non-dissipative evolution of all transported scales using efficient coarse-grid computations. The resulting method is accurate and efficient and shows significant potential for generalization and application to other advection dominated problems. We believe that, with further research and expansion, these unique features can make the CM method an indispensable new tool in scientific computing.

APPENDIX A
Time Evolution of 1D Curves

The curves used in section 3.4.3 are shown below; these figures show the time evolution of the curve sampling. The initial curves are shown in figure 3–3, the curves at times $t = 0.6$, 0.9 and 1.5 are shown in figure A–1. We sample each parametrization 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.



(a) Curve 1

(b) Curve 2

(c) Curve 3

(d) Curve 4

Figure A–1: Comparison of the curves sampling using the original parametrization $\boldsymbol{P}$ (top images) and the redistributed parametrization $\boldsymbol{Q}$ (bottom images).

References

[1] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*, Springer Science & Business Media, 2008.

[2] D. Appelö and T. Hagstrom, *On advection by Hermite methods*, Pacific Journal Of Applied Mathematics, 4 (2012), pp. 125–139.

[3] D. Appelö, T. Hagstrom, and A. Vargas, *Hermite methods for the scalar wave equation*, SIAM Journal on Scientific Computing, 40 (2018), pp. A3902–A3927.

[4] V. I. Arnold, *Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications à l'hydrodynamique des fluides parfaits*, in Annales de l'institut Fourier, vol. 16, 1966, pp. 319–361.

[5] V. I. Arnold, *Mathematical methods of classical mechanics*, vol. 60, Springer Science & Business Media, 2013.

[6] W. T. Ashurst and D. I. Meiron, *Numerical study of vortex reconnection*, Physical review letters, 58 (1987), p. 1632.

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

[8] C. W. Bardos and E. S. Titi, *Mathematics and turbulence: where do we stand?*, Journal of Turbulence, 14 (2013), pp. 42–76.

[9] T. J. Barth and H. Deconinck, *High-order methods for computational physics*, vol. 9, Springer Science & Business Media, 2013.

[10] J. T. Beale, T. Kato, and A. Majda, *Remarks on the breakdown of smooth solutions for the 3-D Euler equations*, Communications in Mathematical Physics, 94 (1984), pp. 61–66.

[11] J. T. Beale and A. Majda, *Vortex methods. I. convergence in three dimensions*, Mathematics of computation, 39 (1982), pp. 1–27.

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

[13] J.-D. Benamou, Y. Brenier, and K. Guittet, *The Monge-Kantorovitch mass transfer and its computational fluid mechanics formulation*, International Journal for Numerical methods in fluids, 40 (2002), pp. 21–30.

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

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

[16] M. Bergdorf and P. Koumoutsakos, *A Lagrangian particle-wavelet method*, Multiscale Modeling & Simulation, 5 (2006), pp. 980–995.

[17] A. Bøckmann and M. Vartdal, *A gradient augmented level set method for unstructured grids*, Journal of Computational Physics, 258 (2014), pp. 47–72.

[18] J. C. Bowman, M. A. Yassaei, and A. Basu, *A fully Lagrangian advection scheme*, Journal of Scientific Computing, 64 (2015), pp. 151–177.

[19] M. Brachet, M. Meneguzzi, A. Vincent, H. Politano, and P. Sulem, *Numerical evidence of smooth self-similar dynamics and possibility of subsequent collapse for three-dimensional ideal flows*, Physics of Fluids A: Fluid Dynamics, 4 (1992), pp. 2845–2854.

[20] M. D. Bustamante and R. M. Kerr, *3D Euler about a 2D symmetry plane*, Physica D: Nonlinear Phenomena, 237 (2008), pp. 1912–1920.

[21] W. Cai, D. Gottlieb, and C.-W. Shu, *Essentially nonoscillatory spectral Fourier methods for shock wave calculations*, Mathematics of Computation, 52 (1989), pp. 389–410.

[22] C. Canuto, M. Y. Hussaini, A. Quarteroni, A. Thomas Jr, et al., *Spectral methods in fluid dynamics*, Springer Science & Business Media, 2012.

[23] G. Carlier, V. Duval, G. Peyré, and B. Schmitzer, *Convergence of entropic schemes for optimal transport and gradient flows*, SIAM Journal on Mathematical Analysis, 49 (2017), pp. 1385–1418.

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

[25] J. A. Carrillo, K. Craig, L. Wang, and C. Wei, *Primal dual methods for Wasserstein gradient flows*, Foundations of Computational Mathematics, (2021), pp. 1–55.

[26] P. Chidyagwai, J.-C. Nave, R. R. Rosales, and B. Seibold, *A comparative study of the efficiency of jet schemes*, International Journal of Numerical Analysis and Modeling - Series B, 3 (2012), pp. 297–306.

[27] A. J. Chorin, *Numerical study of slightly viscous flow*, Journal of Fluid Mechanics, 57 (1973), pp. 785–796.

[28] I. Christiansen, *Numerical simulation of hydrodynamics by the method of point vortices*, Journal of Computational Physics, 13 (1973), pp. 363–379.

[29] P. Constantin, *An Eulerian–Lagrangian approach to the Navier–Stokes equations*, Communications in Mathematical Physics, 216 (2001), pp. 663–686.

[30] G.-H. Cottet, J. Goodman, and T. Y. Hou, *Convergence of the grid-free point vortex method for the three-dimensional Euler equations*, SIAM journal on numerical analysis, 28 (1991), pp. 291–307.

[31] G.-H. Cottet and P. Poncet, *Advances in direct numerical simulations of 3D wall-bounded flows by vortex-in-cell methods*, Journal of Computational Physics, 193 (2004), pp. 136–158.

[32] B. Couet, O. Buneman, and A. Leonard, *Simulation of three-dimensional incompressible flows with a vortex-in-cell method*, Journal of Computational Physics, 39 (1981), pp. 305–328.

[33] R. Courant, E. Isaacson, and M. Rees, *On the solution of nonlinear hyperbolic differential equations by finite differences*, Communications on pure and applied mathematics, 5 (1952), pp. 243–255.

[34] B. Dacorogna and J. Moser, *On a partial differential equation involving the Jacobian determinant*, in Annales de l'Institut Henri Poincare (C) Non Linear Analysis, vol. 7, Elsevier, 1990, pp. 1–26.

[35] R. Deiterding, M. O. Domingues, S. M. Gomes, and K. Schneider, *Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible Euler equations*, SIAM Journal on Scientific Computing, 38 (2016), pp. S173–S193.

[36] J. Deng, T. Y. Hou, and X. Yu, *A level set formulation for the 3D incompressible Euler equations*, Methods and Applications of Analysis, 12 (2005), pp. 427–440.

[37] S. A. Denisov, *Infinite superlinear growth of the gradient for the two-dimensional Euler equation*, Discrete & Continuous Dynamical Systems, 23 (2009), pp. 755–764.

[38] J. Donea, S. Giuliani, and J.-P. Halleux, *An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions*, Computer Methods in Applied Mechanics and Engineering, 33 (1982), pp. 689–723.

[39] E. Eliasen, B. Machenhauer, and E. Rasmussen, *On a numerical method for integration of the hydrodynamical equations with a spectral representation of the horizontal fields*, Kobenhavns Universitet, Institut for Teoretisk Meteorologi, 1970.

[40] M. Farge, N. Okamoto, K. Schneider, and K. Yoshimatsu, *Wavelet-based regularization of the Galerkin truncated three-dimensional incompressible Euler flows*, Physical Review E, 96 (2017), p. 063119.

[41] C. L. Fefferman, *Existence and smoothness of the Navier–Stokes equations*, The Millennium Prize problems, (2006), pp. 57–67.

[42] C. Foias, D. D. Holm, and E. S. Titi, *The Navier–Stokes-alpha model of fluid turbulence*, Physica D: Nonlinear Phenomena, 152 (2001), pp. 505–519.

[43] N. Foster and R. Fedkiw, *Practical animation of liquids*, in Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, 2001, pp. 23–30.

[44] O. Friedrich, *Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids*, Journal of computational physics, 144 (1998), pp. 194–212.

[45] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, Proceedings of the IEEE, 93 (2005), pp. 216–231. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[46] L. Fu, X. Y. Hu, and N. A. Adams, *A family of high-order targeted ENO schemes for compressible-fluid simulations*, Journal of Computational Physics, 305 (2016), pp. 333–359.

[47] P. Gera and D. Salac, *Cahn-Hilliard on surfaces: A numerical study*, Applied Mathematics Letters, 73 (2017), pp. 56–61.

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

[49] F. Gibou, R. Fedkiw, and S. Osher, *A review of level-set methods and some recent applications*, Journal of Computational Physics, 353 (2018), pp. 82–109.

[50] J. Goodrich, T. Hagstrom, and J. Lorenz, *Hermite methods for hyperbolic initial-boundary value problems*, Mathematics of computation, 75 (2006), pp. 595–630.

[51] T. Grafke, H. Homann, J. Dreher, and R. Grauer, *Numerical simulations of possible finite time singularities in the incompressible Euler equations: comparison of numerical methods*, Physica D: Nonlinear Phenomena, 237 (2008), pp. 1932–1936.

[52] R. Grauer, C. Marliani, and K. Germaschewski, *Adaptive mesh refinement for singular solutions of the incompressible Euler equations*, Physical review letters, 80 (1998), p. 4177.

[53] 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 transactions on medical imaging, 23 (2004), pp. 949–958.

[54] 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.

[55] A. Guittet, M. Theillard, and F. Gibou, *A stable projection method for the incompressible Navier-Stokes equations on arbitrary geometries and adaptive Quad/Octrees*, Journal of Computational Physics, 292 (2015), pp. 215–238.

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

[57] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes, III*, in Upwind and high-resolution schemes, Springer, 1987, pp. 218–290.

[58] S. E. Hieber and P. Koumoutsakos, *A Lagrangian particle level set method*, Journal of Computational Physics, 210 (2005), pp. 342–367.

[59] C. W. Hirt, A. A. Amsden, and J. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, Journal of Computational Physics, 14 (1974), pp. 227–253.

[60] T. Y. Hou, T. Jin, and P. Liu, *Potential singularity for a family of models of the axisymmetric incompressible flow*, Journal of Nonlinear Science, 28 (2018), pp. 2217–2247.

[61] T. Y. Hou and R. Li, *Dynamic depletion of vortex stretching and non-blowup of the 3-D incompressible Euler equations*, Journal of Nonlinear Science, 16 (2006), pp. 639–664.

[62] T. Y. Hou and J. Lowengrub, *Convergence of the point vortex method for the 3-D Euler equations*, Communications on pure and applied mathematics, 43 (1990), pp. 965–981.

[63] C. Hu and C.-W. Shu, *Weighted essentially non-oscillatory schemes on triangular meshes*, Journal of Computational Physics, 150 (1999), pp. 97–127.

[64] W. Huang and R. D. Russell, *Adaptive moving mesh methods*, vol. 174, Springer Science & Business Media, 2010.

[65] M. Y. Hussaini and T. A. Zang, *Spectral methods in fluid dynamics*, Annual review of fluid mechanics, 19 (1987), pp. 339–367.

[66] T. Ishihara, T. Gotoh, and Y. Kaneda, *Study of high–Reynolds number isotropic turbulence by direct numerical simulation*, Annual Review Fluid Mechanics, 41 (2009), pp. 165–180.

[67] T. Ishihara, Y. Kaneda, K. Morishita, M. Yokokawa, and A. Uno, *Second-order velocity structure functions in direct numerical simulations of turbulence with R_λ up to 2250*, Physical Review Fluids, 5 (2020), p. 104608.

[68] G.-S. Jiang and D. Peng, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM Journal on Scientific computing, 21 (2000), pp. 2126–2143.

[69] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of computational physics, 126 (1996), pp. 202–228.

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

[71] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the Fokker-Planck equation*, SIAM journal on mathematical analysis, 29 (1998), pp. 1–17.

[72] R. M. Kerr, *Evidence for a singularity of the three-dimensional, incompressible Euler equations*, Physics of Fluids A: Fluid Dynamics, 5 (1993), pp. 1725–1746.

[73] A. Kiselev and V. Šverák, *Small scale creation for solutions of the incompressible two-dimensional Euler equation*, Annals of Mathematics, 180 (2014), pp. 1205–1220.

[74] H. Kohno and J.-C. Nave, *A new method for the level set equation using a hierarchical-gradient truncation and remapping technique*, Computer Physics Communications, 184 (2013), pp. 1547–1554.

[75] D. Kolomenskiy, J.-C. Nave, and K. Schneider, *Adaptive Gradient-Augmented Level Set method with multiresolution error estimation*, Journal of Scientific Computing, 66 (2016), pp. 116–140.

[76] D. Kolomenskiy and K. Schneider, *A Fourier spectral method for the Navier–Stokes equations with volume penalization for moving solid obstacles*, Journal of Computational Physics, 228 (2009), pp. 5687–5709.

[77] G. Lacombe and S. Mas-Gallic, *Presentation and analysis of a diffusion-velocity method*, in ESAIM: Proceedings, vol. 7, EDP Sciences, 1999, pp. 225–233.

[78] S. Lang, *Fundamentals of differential geometry*, vol. 191, Springer Science & Business Media, 2012.

[79] R. J. LeVeque, *High-resolution conservative algorithms for advection in incompressible flow*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 627–665.

[80] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), pp. 200–212.

[81] M. R. Malik, T. A. Zang, and M. Y. Hussaini, *A spectral collocation method for the Navier–Stokes equations*, Journal of Computational Physics, 61 (1985), pp. 64–88.

[82] J. E. Marsden and S. Shkoller, *Global well-posedness for the Lagrangian averaged Navier-Stokes (LANS-α) equations on bounded domains*, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 359 (2001), pp. 1449–1468.

[83] J. E. Marsden and S. Shkoller, *The anisotropic Lagrangian averaged Euler and Navier-Stokes equations*, Archive for Rational Mechanics and Analysis, 166 (2003), pp. 27–46.

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

[85] M. Mirzadeh, A. Guittet, C. Burstedde, and F. Gibou, *Parallel level-set methods on adaptive tree-based grids*, Journal of Computational Physics, 322 (2016), pp. 345–364.

[86] H. Moffatt and Y. Kimura, *Towards a finite-time singularity of the Navier–Stokes equations. part 2. vortex reconnection and singularity evasion*, Journal of Fluid Mechanics, 870 (2019), p. R1.

[87] K. Mohseni, B. Kosović, S. Shkoller, and J. E. Marsden, *Numerical simulations of the Lagrangian averaged Navier-Stokes equations for homogeneous isotropic turbulence*, Physics of Fluids, 15 (2003), pp. 524–544.

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

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

[90] M. Oliver and S. Shkoller, *The Vortex Blob method as a second-grade non-Newtonian fluid*, Communications in Partial Differential Equations, 26 (2001), pp. 295–314.

[91] S. A. Orszag, *Numerical methods for the simulation of turbulence*, The Physics of Fluids, 12 (1969), pp. II–250.

[92] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153, Springer Science & Business Media, 2006.

[93] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.

[94] R. Pelz, *Locally self-similar, finite-time collapse in a high-symmetry vortex filament model*, Physical Review E, 55 (1997), p. 1617.

[95] R. M. Pereira, M. Farge, K. Schneider, et al., *Wavelet methods to eliminate resonances in the Galerkin-truncated Burgers and Euler equations*, Physical Review E, 87 (2013), p. 033017.

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

[97] O. Podvigina, V. Zheligovsky, and U. Frisch, *The Cauchy-Lagrangian method for numerical analysis of Euler flow*, Journal of Computational Physics, 306 (2016), pp. 320–342.

[98] A. Pumir and E. Siggia, *Collapsing solutions to the 3-D Euler equations*, Physics of Fluids A: Fluid Dynamics, 2 (1990), pp. 220–241.

[99] S. S. Ray, U. Frisch, S. Nazarenko, and T. Matsumoto, *Resonance phenomenon for the Galerkin-truncated Burgers and Euler equations*, Physical Review E, 84 (2011), p. 016301.

[100] L. Rosenhead, *The point vortex approximation of a vortex sheet*, Proc. Roy. Soc. London Ser. A, 134 (1932), pp. 170–192.

[101] D. Rossinelli, M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos, *Gpu accelerated simulations of bluff body flows using vortex particle methods*, Journal of Computational Physics, 229 (2010), pp. 3316–3333.

[102] F. Santambrogio, *Optimal transport for applied mathematicians*, Birkäuser/Springer, 2015.

[103] F. Santambrogio, *{Euclidean, metric, and Wasserstein} gradient flows: an overview*, Bulletin of Mathematical Sciences, 7 (2017), pp. 87–154.

[104] I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos, *Ppm–a highly efficient parallel particle–mesh library for the simulation of continuum systems*, Journal of Computational Physics, 215 (2006), pp. 566–588.

[105] K. Schneider, N.-R. Kevlahan, and M. Farge, *Comparison of an adaptive wavelet method and nonlinearly filtered pseudospectral methods for two-dimensional turbulence*, Theoretical and computational fluid dynamics, 9 (1997), pp. 191–206.

[106] K. Schneider, D. Kolomenskiy, and E. Deriaz, *Is the CFL condition sufficient? some remarks*, in The Courant–Friedrichs–Lewy (CFL) Condition, C.A. de Moura, C.S. Kubrusly (eds.), Birkhäuser, 2013, pp. 139–146.

[107] K. Schneider and O. V. Vasilyev, *Wavelet methods in computational fluid dynamics*, Annual review of fluid mechanics, 42 (2010), pp. 473–503.

[108] B. Seibold, J.-C. Nave, and R. R. Rosales, *Jet schemes for advection problems*, Discrete and Continuous Dynamical Systems - Series B, 17 (2012), pp. 1229–1259.

[109] J. Shi, Y.-T. Zhang, and C.-W. Shu, *Resolution of high order WENO schemes for complicated flow structures*, Journal of Computational Physics, 186 (2003), pp. 690–696.

[110] C.-W. Shu, T. A. Zang, G. Erlebacher, D. Whitaker, and S. Osher, *High-order ENO schemes applied to two-and three-dimensional compressible flow*, Applied Numerical Mathematics, 9 (1992), pp. 45–71.

[111] J. Stam, *Stable fluids.*, in Siggraph, vol. 99, 1999, pp. 121–128.

[112] A. Staniforth and J. Côté, *Semi-Lagrangian integration schemes for atmospheric models—A review*, Monthly Weather Review, 119 (1991), pp. 2206–2223.

[113] 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*, Computer Aided Geometric Design, 46 (2016), pp. 76–91.

[114] 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.

[115] M. Theillard, F. Gibou, and D. Saintillan, *Sharp numerical simulation of incompressible two-phase flows*, Journal of Computational Physics, 391 (2019), pp. 91–118.

[116] C. Villani, *Optimal transport: old and new*, vol. 338, Springer Science & Business Media, 2008.

[117] E. Weinan and C.-W. Shu, *A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow*, Journal of Computational Physics, 110 (1994), pp. 39–46.

[118] G. Winckelmans and A. Leonard, *Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows*, Journal of Computational Physics, 109 (1993), pp. 247–273.

[119] A. M. Winslow, *Adaptive-mesh zoning by the equipotential method*, tech. report, Lawrence Livermore National Lab., CA (USA), 1981.

[120] J. Yao and F. Hussain, *A physical model of turbulence cascade via vortex reconnection sequence and avalanche*, Journal of Fluid Mechanics, 883 (2020), p. A51.

[121] X.-Y. Yin, L. Chen, and J.-C. Nave, *A diffusion-driven characteristic mapping method for particle management*, SIAM Journal on Scientific Computing, 43 (2021), p. A3155–A3183.

[122] X.-Y. Yin, O. Mercier, B. Yadav, K. Schneider, and J.-C. Nave, *A characteristic mapping method for the two-dimensional incompressible Euler equations*, Journal of Computational Physics, 424 (2021), p. 109781.

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

[124] V. Yudovich, *The flow of a perfect, incompressible liquid through a given region*, in Soviet Physics Doklady, vol. 7, 1963, p. 789.

[125] J. Zhang and T. L. Jackson, *A high-order incompressible flow solver with WENO*, Journal of Computational Physics, 228 (2009), pp. 2426–2442.

[126] X. Zhang and C.-W. Shu, *Positivity-preserving high order finite difference WENO schemes for compressible Euler equations*, Journal of Computational Physics, 231 (2012), pp. 2245–2258.

[127] X. Zhao, Z. Su, X. D. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo, *Area-preservation mapping using optimal mass transport*, IEEE Transactions on Visualization and Computer Graphics, 19 (2013), pp. 2838–2847.