Journal of Computational Physics 424 (2021) 109781

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A Characteristic Mapping method for the two-dimensional incompressible Euler equations

Xi-Yuan Yin [a], Olivier Mercier [a], Badal Yadav [a], Kai Schneider [b], Jean-Christophe Nave [a,*]

[a] Department of Mathematics and Statistics, McGill University, Montréal, Québec H3A 0B9, Canada
[b] Institut de Mathématiques de Marseille (I2M), Aix-Marseille Université, CNRS, Centrale Marseille, 13453 Marseille Cedex 13, France

ARTICLE INFO

Article history:
Received 3 December 2019
Received in revised form 11 August 2020
Accepted 11 August 2020
Available online 26 August 2020

Keywords:
Fluid dynamics
Euler equations
Characteristic Mapping method
Gradient-Augmented Level-set method

ABSTRACT

We propose an efficient semi-Lagrangian method for solving the two-dimensional incompressible Euler equations with high precision on a coarse grid. This new approach evolves the flow map using a combination of the Characteristic Mapping (CM) method [1] the gradient-augmented level-set (GALS) method [2]. The flow map possesses a semigroup structure which allows for the decomposition of a long-time deformation into short-time submaps. This leads to a numerical scheme that achieves exponential resolution in linear time. Error estimates are provided and conservation properties are analysed. The computational efficiency and the high precision of the method are illustrated in the vortex merger, four-modes and random flow problems. Comparisons with the Cauchy-Lagrangian method [3] are also presented.

1. Introduction

The numerical simulation of incompressible fluids is an important mathematical problem with many scientific and industrial applications. 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 [4], this bound was proven to be sharp in the case of smooth initial data on a disk [5]. 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 [6]. For a summary on the advances in mathematics for the Euler equations, we refer to a 2013 review by Bardos and Titi [7]. 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. [8] in the case of a conservative Fourier-Galerkin scheme, and a wavelet regularization technique has been proposed by Pereira et al. in [9].

[*] Corresponding author.
E-mail address: jcnave@math.mcgill.ca (J.-C. Nave).

https://doi.org/10.1016/j.jcp.2020.109781
0021-9991/© 2020 Elsevier Inc. All rights reserved.

Existing numerical methods for fluid simulation lie in the spectrum from fully Lagrangian to fully Eulerian formulations. Eulerian methods use fixed spatial meshes to represent evolving quantities. This gives easy access to the relevant simulated quantities. The spatial features generated by large fluid deformations can be represented as long as the grid has sufficient resolution. However, finer scale features are lost resulting in artificial dissipation. Examples of Eulerian methods include the Fluid-in-Cell [10] method and the Marker-in-Cell method [11], where fluid quantities are associated to fixed spatial subdivisions and updated in time using the evolution equations. In Lagrangian methods, particles or grid nodes follow the movement of the flow, resulting in less dissipative schemes. However, with the particle or moving grid frameworks, the evaluation of Eulerian quantities is difficult and less accurate. Furthermore, large fluid deformations can be difficult to resolve and often require frequent remeshing or resampling routines. Examples of purely Lagrangian methods include the Smooth Particle Hydrodynamics [12,13] and the Vortex blob [14] methods, where fluid quantities are transported using a particle method. More recently, Bowman et al. [15] developed a fully Lagrangian scheme for the advection and advection-diffusion equations, exploiting the relabelling symmetry of the incompressible advection and Euler equations. This scheme does not require costly remeshing routines and is void of artificial dissipation; in particular, the algorithm preserves all Casimir invariants of the equations. Many numerical solvers use a mix of Lagrangian and Eulerian descriptions to exploit the advantages of each approach. For instance, Arbitrary Lagrangian-Eulerian [16,17] methods use reference coordinates that are neither fully Lagrangian nor fully Eulerian to describe the fluid configuration. 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 [3] 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. [18]. In these methods, characteristics are traced backwards in time and Eulerian interpolation schemes are used to update the solution. A level-set formulation of the 3D incompressible Euler equations was recently studied by Deng et al. [19] through the use of generalized Clebsch variables. This theoretical framework provided new global existence results for the solutions of the 3D Euler and Lagrangian-Averaged Euler equations under certain assumptions, and existing level-set methods can potentially be applied to generate efficient schemes for fluid simulations in 3D space. Semi-Lagrangian schemes have also been used to generate fast simulations intended for computer graphics [20,21]. 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.

In this paper, we present a method where the evolved numerical quantity is the deformation map of the domain generated by the fluid velocity. This approach is geometric and fully exploits the characteristic structure of the fluid flow. We construct numerically a characteristic map which identifies a point on an arbitrary characteristic curve to its initial position through a spatial transformation. Advected quantities, in particular vorticity, can then be constructed as the function composition of the initial condition with the characteristic map. This method stems from the semi-Lagrangian Gradient-Augmented Level-Set (GALS) [2] and Jet-Scheme methods [22] and constitutes an extension of the Characteristic Mapping (CM) method for linear advection [23,1] to the self-advection in 2D incompressible Euler. The CM method for Euler splits the evolution equations into the advection of the vorticity 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 CM 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 [3] 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 multiresolution mesh refinement to efficiently resolve fine scale features [24–27]. One unique property of the CM method for 2D Euler demonstrated in this paper 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 semigroup 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 the paper is organized as follows: in section 2 we lay out the mathematical framework for the CM method and carry out some heuristic analysis of its properties. In section 3, we present in detail the numerical implementation of the method and provide some error bounds. Section 4 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 5.

2. Mathematical framework

In this section, we present the mathematical framework behind the CM method for 2D incompressible Euler. The section is organized as follows: first, we present the equations with the advection/Biot-Savart splitting used in the method. Then we look at the CM method for the advection equation along with its intrinsic semigroup structure. We will next examine the advection-vorticity coupling of the equation through the Biot-Savart law. Finally, we will put everything together to write the modified equation which naturally arises from a numerical CM method for the Euler equations.

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

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p \qquad (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1b}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), \tag{1c}$$

where U is some fixed star-shaped spatial domain, $\boldsymbol{u}$ is a vector field describing the instantaneous velocity of a fluid element and p is the pressure. For this paper, 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 non-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 paper.

Define $\omega = \nabla \times \boldsymbol{u}$ the scalar vorticity of the fluid, and taking the 2D curl of (1), we obtain the vorticity equations:

$$\partial_t \omega + (\boldsymbol{u} \cdot \nabla)\omega = 0 \qquad (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{2a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2b}$$

$$\omega(\boldsymbol{x}, 0) = \omega_0(\boldsymbol{x}). \tag{2c}$$

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

$$\boldsymbol{u} = -\triangle^{-1} \nabla \times \omega. \tag{3}$$

Given a solution $\boldsymbol{u}(\boldsymbol{x}, t)$ of (1), we make the observation that ω solves the advection equation (2a) under the velocity field $\boldsymbol{u}$. The approach presented in this paper consists in splitting the vorticity equations (2) into the coupling of the advection of the vorticity by (2a) and the Biot-Savart law (3). The advection equation is solved using the Characteristic Mapping method and the Biot-Savart law is applied in Fourier space. We present details on these two methods in the following sections.

2.1. Characteristic mapping framework

The Characteristic Mapping approach consists in finding the solution operator for the advection equation associated to some transport velocity. A more detailed exposition of this approach can be found in [1,2,23].

Consider a linear advection equation with a divergence-free transport velocity $\boldsymbol{u}$

$$\partial_t \phi + (\boldsymbol{u} \cdot \nabla)\phi = 0 \qquad (\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{4a}$$

$$\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}). \tag{4b}$$

The method of characteristics for advection equations implies that ϕ satisfies

$$\frac{d}{dt}\phi(\boldsymbol{\gamma}(t), t) = 0, \tag{5}$$

or equivalently

$$\phi(\boldsymbol{\gamma}(t), t) = \phi_0(\boldsymbol{\gamma}_0), \tag{6}$$

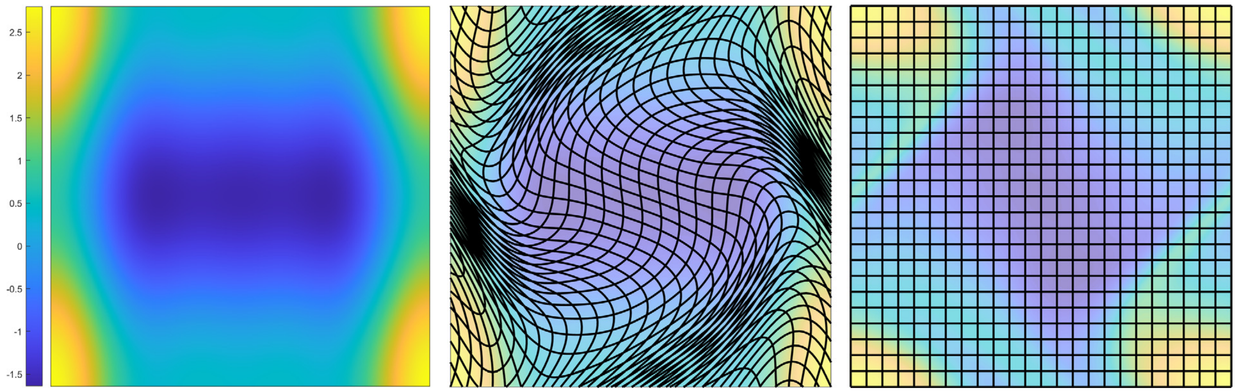
for any characteristic curve $\boldsymbol{\gamma}$ solving

$$\frac{d}{dt}\boldsymbol{\gamma}(t) = \boldsymbol{u}(\boldsymbol{\gamma}(t), t), \tag{7a}$$

$$\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}_0. \tag{7b}$$

The smoothness of these characteristic curves in the case of the Euler equations is proven in [28].

Therefore, we look for a solution operator $\boldsymbol{X} : U \times \mathbb{R}_+ \to U$ such that $\boldsymbol{X}(\boldsymbol{\gamma}(t), t) = \boldsymbol{\gamma}_0$ for all characteristic curves $\boldsymbol{\gamma}(t)$ associated to the velocity $\boldsymbol{u}$. For any fixed t, the map $\boldsymbol{X}(\cdot, t) : U \to U$ is the inverse of the t-time flow map of the velocity $\boldsymbol{u}$. We call $\boldsymbol{X}$ the *backward characteristic map* since it “traces back” a particle at time t to its initial position $\boldsymbol{\gamma}_0$. Assuming



(a) Initial condition.

(b) Domain deformation from characteristic map.

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

Fig. 1. Evolution of the characteristic map acting on the initial condition. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

smooth divergence-free velocity, the flow map is a diffeomorphism, hence, we can write $\boldsymbol{X}$ as the solution to the following vector-valued advection equation:

$$\partial_t \boldsymbol{X} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{X} = 0 \qquad \forall(\boldsymbol{x}, t) \in U \times \mathbb{R}_+, \tag{8a}$$

$$\boldsymbol{X}(\boldsymbol{x}, 0) = \boldsymbol{x}. \tag{8b}$$

One can check that $\frac{d}{dt}\boldsymbol{X}(\boldsymbol{\gamma}(t), t) = 0$ and hence $\boldsymbol{X}(\boldsymbol{\gamma}(t), t) = \boldsymbol{X}(\boldsymbol{\gamma}_0, 0) = \boldsymbol{\gamma}_0$.

Remark 1. Wolibner proved in 1933 [29] that for an L^1 initial vorticity field (with appropriate decay at infinity), the corresponding velocity field (which is log-Lipschitz) solving the Euler equations generates a Hölder continuous forward characteristic map. That is to say, the transformation $\Phi(\cdot, t) : \gamma_0 \mapsto \gamma(t)$ is a Hölder continuous homeomorphism for all t. The existence, continuity and area-preservation of the backward map $\boldsymbol{X}$ follows since $\boldsymbol{X}(\cdot, t) = \Phi^{-1}(\cdot, t)$.

The solutions to the advection equation (4) can be then computed as the pullback of the initial condition by this backward map using (4)

$$\phi(\boldsymbol{x}, t) = \phi_0\left(\boldsymbol{X}(\boldsymbol{x}, t)\right). \tag{9}$$

In fact, for any initial condition ϕ_0, the corresponding advection equation has solution $\phi_0 \circ \boldsymbol{X}$ (see Fig. 1). The characteristic map is independent of the advected quantity, it captures the deformation of the space induced by the velocity $\boldsymbol{u}$ and acts as a solution operator of all quantities transported by $\boldsymbol{u}$. In particular, if $\boldsymbol{u}$ is a solution to the Euler equations (1), then its vorticity field must satisfy

$$\omega(\boldsymbol{x}, t) = \omega_0\left(\boldsymbol{X}(\boldsymbol{x}, t)\right). \tag{10}$$

2.1.1. Semigroup 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 initiated by Arnold in 1966 [30]. 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. Hence, they inherit the same semigroup properties.

Definition 1. We denote by $\boldsymbol{X}_{[t_f,t_0]} : U \to U$ the characteristic **submap** of the time interval $[t_0, t_f]$. The endpoints of the interval in the subscript $[t_f, t_0]$ are inverted to reflect the fact that $\boldsymbol{X}_{[t_f,t_0]}$ is the characteristic transformation of the domain mapping U at time t_f back to time t_0. The submap is given by the following equations:

$$\partial_t \boldsymbol{X}^\dagger + (\boldsymbol{u} \cdot \nabla)\boldsymbol{X}^\dagger = 0, \tag{11a}$$

$$\boldsymbol{X}^\dagger(\boldsymbol{x}, t_0) = \boldsymbol{x}, \tag{11b}$$

$$\boldsymbol{X}_{[t_f,t_0]}(\boldsymbol{x}) := \boldsymbol{X}^\dagger(\boldsymbol{x}, t_f). \tag{11c}$$

One can then check that given arbitrary times $t_0 < t_1 < t_2$, the submaps satisfy the semigroup property:

$$\boldsymbol{X}_{[t_2,t_0]} = \boldsymbol{X}_{[t_1,t_0]} \circ \boldsymbol{X}_{[t_2,t_1]}. \tag{12}$$

This is in fact true without the $t_0 < t_1 < t_2$ assumption and would involve forward characteristic maps. This paper will only focus on the use of the backward maps.

This semigroup structure is at the heart of the CM method. The time evolution of the characteristic map is defined though this map composition rather than through the usual PDE. We can see the evolution of $\boldsymbol{X}$ as given by

$$\boldsymbol{X}(\cdot, t + \Delta t) = \boldsymbol{X}(\cdot, t) \circ \boldsymbol{X}_{[t+\Delta t,t]}. \tag{13}$$

The numerical approximation of $\boldsymbol{X}$ relies on discretizing the above using some small time step Δt.

We also make use of this semigroup property to adaptively adjust the numerical resolution of the map. In particular, a backward characteristic map $\boldsymbol{X}(\cdot, t)$ 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 $[\tau_{i-1}, \tau_i]$ with $0 = \tau_0 < \tau_1 < \cdots < \tau_m = t$. One can then check that the following decomposition holds:

$$\boldsymbol{X}(\cdot, t) = \boldsymbol{X}_{[t,0]} = \boldsymbol{X}_{[\tau_1,0]} \circ \boldsymbol{X}_{[\tau_2,\tau_1]} \circ \cdots \circ \boldsymbol{X}_{[\tau_{m-1},\tau_{m-2}]} \circ \boldsymbol{X}_{[t,\tau_{m-1}]} \tag{14}$$

Each of the submaps $\boldsymbol{X}_{[\tau_i,\tau_{i-1}]}$ 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 3. 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

$$\omega(\boldsymbol{x}, t) = \omega_0(\boldsymbol{X}(\boldsymbol{x}, t)). \tag{15}$$

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

$$\nabla\omega(\boldsymbol{x}, t) = \nabla\omega_0 \cdot \nabla\boldsymbol{X}(\boldsymbol{x}, t). \tag{16}$$

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 (14), we have

$$\nabla\boldsymbol{X}_{[t,0]} = \nabla\boldsymbol{X}_{[\tau_1,0]}\nabla\boldsymbol{X}_{[\tau_2,\tau_1]} \cdots \nabla\boldsymbol{X}_{[\tau_{m-1},\tau_{m-2}]}\nabla\boldsymbol{X}_{[t,\tau_{m-1}]}. \tag{17}$$

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

2.2. Advection-vorticity coupling

We use the backward characteristic map to rewrite the vorticity equation (2a) as the following coupling of $\boldsymbol{u}, \omega$ and $\boldsymbol{X}$:

$$\omega(\boldsymbol{x}, t) = \omega_0(\boldsymbol{X}(\boldsymbol{x}, t)), \tag{18a}$$

$$\boldsymbol{u} = -\triangle^{-1}\nabla \times \omega, \tag{18b}$$

$$(\partial_t + \boldsymbol{u} \cdot \nabla)\boldsymbol{X} = 0. \tag{18c}$$

Equation (18b) is known as the Biot-Savart law and can be obtained from $\nabla \cdot \boldsymbol{u} = 0$ and the definition of the vector Laplacian:

$$\triangle\boldsymbol{F} = \nabla(\nabla \cdot \boldsymbol{F}) - \nabla \times (\nabla \times \boldsymbol{F}) \tag{19}$$

for $\boldsymbol{F}$ a $\mathbb{R}^2 \to \mathbb{R}^2$ vector field. We let $\boldsymbol{u} = \boldsymbol{F}$ and commute $\triangle^{-1}$ and $\nabla\times$ in (18b), the velocity is then obtained from the stream function ψ:

$$\psi = -\triangle^{-1}\omega, \qquad \boldsymbol{u} = \nabla \times \psi. \tag{20}$$

The CM method for Euler equations then consists of numerically evolving $\boldsymbol{X}$ in time using equation (18c); the velocity and vorticity are defined using (18a), and (20). Additionally, if we employ the submap decomposition from the semigroup property, we would solve equation (18c) for the submaps $\boldsymbol{X}_{[t,\tau_i]}$ for $t \in [\tau_i, \tau_{i+1}]$ and initial condition $\boldsymbol{X}_{[\tau_i,\tau_i]}(\boldsymbol{x}) = \boldsymbol{x}$; the pullback (18a) would be computed from the composition of all submaps.

2.3. Modified equations

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

Let $\boldsymbol{\mathcal{X}}^n$ 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

$$\partial_t \tilde{\boldsymbol{X}} + (\tilde{\boldsymbol{u}} \cdot \nabla)\tilde{\boldsymbol{X}} = 0. \tag{21}$$

From here on, we will use the notation *tilda* 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 $\boldsymbol{X}$ and the solution to the modified equation $\tilde{\boldsymbol{X}}$. This is a useful strategy as it will allow us to estimate the error of the numerical solution $\boldsymbol{\mathcal{X}}$ using $|\boldsymbol{X} - \boldsymbol{\mathcal{X}}| \leq |\boldsymbol{X} - \tilde{\boldsymbol{X}}| + |\tilde{\boldsymbol{X}} - \boldsymbol{\mathcal{X}}|$.

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

$$\boldsymbol{X}_{[t,t_0]}(\boldsymbol{x}) = \boldsymbol{x} + \int_t^{t_0} \boldsymbol{u}\big(\boldsymbol{X}_{[t,r]}(\boldsymbol{x}), r\big)dr, \tag{22a}$$

$$\tilde{\boldsymbol{X}}_{[t,t_0]}(\boldsymbol{x}) = \boldsymbol{x} + \int_t^{t_0} \tilde{\boldsymbol{u}}\big(\tilde{\boldsymbol{X}}_{[t,r]}(\boldsymbol{x}), r\big)dr. \tag{22b}$$

By Grönwall's lemma, we have that

$$\left|\boldsymbol{X}_{[t,t_0]}(\boldsymbol{x}) - \tilde{\boldsymbol{X}}_{[t,t_0]}(\boldsymbol{x})\right| \lesssim C(t-t_0)e^{C(t-t_0)}\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_\infty, \tag{23}$$

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

Heuristically, we note from this that if a numerical solution $\boldsymbol{\mathcal{X}}$ approximates well the solution $\tilde{\boldsymbol{X}}$ 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:

$$\partial_t(\omega_0 \circ \tilde{\boldsymbol{X}}) = \nabla\omega_0 \cdot \partial_t \tilde{\boldsymbol{X}} = -\nabla\omega_0 \cdot (\tilde{\boldsymbol{u}} \cdot \nabla)\tilde{\boldsymbol{X}} = -(\tilde{\boldsymbol{u}} \cdot \nabla)(\omega_0 \circ \tilde{\boldsymbol{X}}). \tag{24}$$

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 spatially truncation errors dominate, we shall see that this is closely related to the Lagrangian-Averaged Euler equations.

2.3.1. 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 $\boldsymbol{X}$. 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 (14)) 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 $\boldsymbol{X}$, 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 $\omega_0 \circ \boldsymbol{X}$ over the entire domain and evolves in time as a consequence of the changes in the deformation map $\boldsymbol{X}$. 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.

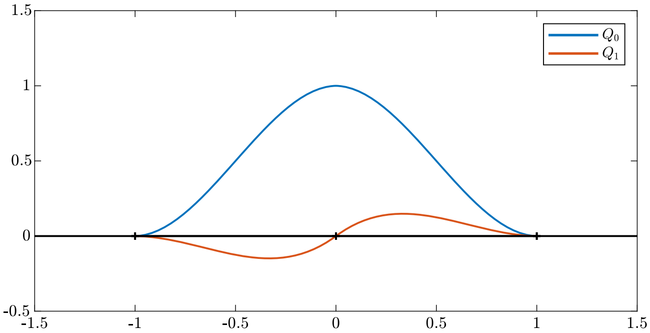


Fig. 2. 1D Hermite cubic basis functions.

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 [31–33]. In its vorticity form, the LAE-α can be written as

$$\partial_t \omega + (\boldsymbol{u} \cdot \nabla)\omega = 0, \tag{25a}$$

$$\boldsymbol{u} = -\nabla \times \triangle^{-1}(1 - \alpha\,\triangle)^{-1}\omega. \tag{25b}$$

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 [14] and the LAE-α equations can be found in [34].

We see from (25) that the vorticity is transported by $\boldsymbol{u}$ and that $\boldsymbol{u}$ is obtained from the Biot-Savart law on a smoothed vorticity field $(1 - \alpha\,\triangle)^{-1}\omega$. 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 2.3). 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 $(1 - \alpha\,\triangle)^{-1}$ smoothing in LAE-α, the flow maps from both formulations are the same and hence generate the same dynamics.

3. Numerical implementation

We present in this section the numerical framework for the CM method. First, we present the Hermite interpolation structure used for spatial discretization as well as the spatial definition of the discretized velocity field. We then provide a time semi-discretization of the equations and make the link to the modified equation in 2.3. The space and time discretizations are combined to generate the CM solver for the Euler equations for which we then give some error estimates. Finally, we describe the remapping method which aims to adaptively resolve increasingly complicated spatial features.

3.1. Spatial discretization

We review in this section the definition of Hermite cubic interpolants which is at the heart of all spatial computations in this method. We then present the spatial discretization of the velocity field.

3.1.1. Hermite cubic interpolation

We assume that the computational domain U is discretized in a rectangular meshgrid $\boldsymbol{G}$. In 2D, $\boldsymbol{G}$ consists of grid points $\boldsymbol{x}_{i,j}$ and has cells $C_{i,j}$ with corners $\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i,j+1}$ and $\boldsymbol{x}_{i+1,j+1}$. We assume for simplicity that U is a flat torus and $\boldsymbol{G}$ is a square grid with uniform cell widths Δx.

The space of Hermite cubic functions on $\boldsymbol{G}$, $\mathcal{V}_{\boldsymbol{G}}$, is a finite dimensional subspace of $\mathcal{C}^1(U)$ consisting of functions that are bicubic in each $C_{i,j}$. To construct the basis functions, we take a tensor product of the 1D cubic basis (see Fig. 2):

$$Q_0(x) = (1 + 2|x|)(1 - |x|)^2 \qquad x \in [-1, 1], \tag{26a}$$

$$Q_1(x) = x(1 - |x|)^2 \qquad x \in [-1, 1], \tag{26b}$$

$$Q_0(x) = Q_1(x) = 0 \qquad x \notin [-1, 1]. \tag{26c}$$

These functions have the property that $\partial^c Q_k|_y = \delta^c_k \delta^0_y$ for $c, k \in \{0, 1\}$ and $y \in \{-1, 0, 1\}$ and are cubic in $[-1, 0]$ and $[0, 1]$ and are everywhere continuously differentiable.

We define the 2D basis functions on the grid $\boldsymbol{G}$ for $\boldsymbol{x} = (x^1, x^2)$ using tensor products:

$$H_{k,l}^{i,j}(\boldsymbol{x}) = Q_k\left(\frac{x^1 - x^1_{i,j}}{\Delta x}\right) Q_l\left(\frac{x^2 - x^2_{i,j}}{\Delta x}\right) \Delta x^{k+l}. \tag{27}$$

These basis functions satisfy

$$\partial^{(c,d)} H_{k,l}^{i,j}(\boldsymbol{x}_{a,b}) = \delta_i^a \delta_j^b \delta_k^c \delta_l^d. \tag{28}$$

Each $H_{k,l}^{i,j}$ is supported on 4 cells $C_{i-1,j-1}, C_{i,j-1}, C_{i-1,j}$ and $C_{i,j}$. It is bicubic in each cell, everywhere $\mathcal{C}^1$ with continuous mixed derivative $\partial^{(1,1)}$. It has continuous normal derivatives and $\mathcal{C}^\infty$ tangential derivatives on cell boundaries.

The interpolation operator is the projection operator $\mathcal{H}_{\boldsymbol{G}} : \mathcal{C}^1(U) \to \mathcal{V}_{\boldsymbol{G}}$

$$\mathcal{H}_{\boldsymbol{G}}[f](\boldsymbol{x}) = \sum_{i,j} \sum_{k,l} f_{i,j}^{k,l} H_{k,l}^{i,j}(\boldsymbol{x}), \tag{29}$$

where $f_{i,j}^{k,l} = \partial^{(k,l)} f(\boldsymbol{x}_{i,j})$ for $f \in \mathcal{C}^1(U)$.

It is well known that for $f \in \mathcal{C}^4(U)$ the interpolation is order 4:

$$\|f - \mathcal{H}_{\boldsymbol{G}}[f]\|_\infty < c\|D^4 f\|_\infty \Delta x^4, \tag{30}$$

for some constant c.

Furthermore, we see that if the grid data $f_{i,j}^{k,l}$ are perturbed by ϵ, then the perturbation of the interpolant is

$$\|\mathcal{H}_{\boldsymbol{G}}[f] - \mathcal{H}_{\boldsymbol{G}}[f_\epsilon]\|_\infty = \mathcal{O}(\epsilon \Delta x^{k+l}). \tag{31}$$

Therefore, in order to obtain a 4th order accurate interpolant of a function f, it is sufficient to know the (k,l) derivative of f to order $\mathcal{O}(\Delta x^{4-(k+l)})$.

3.1.2. *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, $\boldsymbol{X}(\cdot, t)$, and that the initial condition for the vorticity ω_0 is given analytically, we have that the vorticity at time t is given by

$$\omega(\boldsymbol{x}, t) = \omega_0(\boldsymbol{X}(\boldsymbol{x}, t)). \tag{32}$$

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 $\boldsymbol{u}$. Instead of filtering by a coarse grid sampling (undersampling), we define

$$\boldsymbol{u}_\epsilon = \eta_\epsilon * \boldsymbol{u}, \tag{33}$$

where η_ϵ is a mollifier supported in a ball of radius ϵ.

Commuting mollification with derivatives, we have that

$$\boldsymbol{u}_\epsilon = \nabla \times \psi_\epsilon \quad \text{for } \psi_\epsilon = \eta_\epsilon * \psi, \tag{34a}$$

$$\psi_\epsilon = -\triangle^{-1} \omega_\epsilon \quad \text{for } \omega_\epsilon = \eta_\epsilon * \omega. \tag{34b}$$

Finally, using $\omega = \omega_0 \circ \boldsymbol{X}$, we can write the pullback of the convolution

$$\begin{aligned} \omega_\epsilon(\boldsymbol{x}, t_n) &= \int\limits_{B_\epsilon(\boldsymbol{x})} \eta_\epsilon(\boldsymbol{x} - \boldsymbol{y}) \omega_0\left(\boldsymbol{X}(\boldsymbol{y}, t_n)\right) d\boldsymbol{y} \\ &= \int\limits_{\boldsymbol{X}(B_\epsilon(\boldsymbol{x}), t_n)} \eta_\epsilon(\boldsymbol{x} - \boldsymbol{X}^{-1}(\boldsymbol{y}, t_n)) \omega_0(\boldsymbol{y}) |\det(\nabla \boldsymbol{X})|^{-1} d\boldsymbol{y} \\ &= \int\limits_{\boldsymbol{X}(B_\epsilon(\boldsymbol{x}), t_n)} \tilde{\eta}_\epsilon(\boldsymbol{X}(\boldsymbol{x}, t_n), \boldsymbol{y}) \omega_0(\boldsymbol{y}) d\boldsymbol{y} = [\tilde{\eta}_\epsilon * \omega_0]_{\boldsymbol{X}(\boldsymbol{x}, t_n)}, \end{aligned} \tag{35}$$

where $\tilde{\eta}_\epsilon(\boldsymbol{x}, \boldsymbol{y}) = \eta_\epsilon(\boldsymbol{X}^{-1}(\boldsymbol{x}, t_n) - \boldsymbol{X}^{-1}(\boldsymbol{y}, t_n)) |\det(\nabla \boldsymbol{X}(\boldsymbol{x}, t_n))|^{-1}$. The last line above can be verified by plugging in $\boldsymbol{x} = \boldsymbol{X}^{-1}(\cdot, t_n)$.

During simulations, when sampling the vorticity at discrete locations $\boldsymbol{x}_i$, we pick a mollifier such that $\eta_{\epsilon,\boldsymbol{x}_i}$ form a partition of unity of the domain. This further guarantees that the numerical average vorticity is equal to the true exact average up to quadrature and Jacobian determinant errors. Given a numerical approximation $\boldsymbol{\mathcal{X}}^n$ of $\boldsymbol{X}(\cdot, t_n)$, we define the mollified numerical vorticity at time step t_n to be

$$\omega^n_\epsilon(\boldsymbol{x}) := \left[\eta^n_\epsilon * \omega_0\right]_{\boldsymbol{\mathcal{X}}^n(\boldsymbol{x})}, \tag{36}$$

for $\eta^n_\epsilon(\boldsymbol{x}, \boldsymbol{y}) = \eta_\epsilon((\boldsymbol{\mathcal{X}}^n)^{-1}(\boldsymbol{x}) - (\boldsymbol{\mathcal{X}}^n)^{-1}(\boldsymbol{y}))|\det(\nabla\boldsymbol{\mathcal{X}}^n(\boldsymbol{x}))|^{-1}$. Note that the numerical evaluation of ω^n_ϵ is performed by quadrature on the deformed cells $\boldsymbol{\mathcal{X}}^n(C_{ij})$ and requires no computations of the inverse or Jacobian determinant of $\boldsymbol{\mathcal{X}}^n$.

Remark 2. We only use this mollified vorticity to generate a mollified velocity, the true numerical solution is still defined as $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}^n$.

The velocity field can be obtained from ω^n_ϵ through the stream function as in equation (20). Numerically we will solve this using a spectral method with Fast Fourier transforms. Let $\mathcal{F}$ denote the Discrete Fourier transform operator, we have:

$$\psi^n_\epsilon := -\mathcal{F}^{-1}\left[\triangle^{-1}\mathcal{F}\left[\omega^n_\epsilon\right]\right], \tag{37}$$

where the above $\triangle^{-1}$ 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 $\boldsymbol{U}$ for the representation of the velocity field and define

$$\boldsymbol{u}^n_\epsilon := \nabla \times \mathcal{H}_{\boldsymbol{U}}[\psi^n_\epsilon]. \tag{38}$$

This definition guarantees that $\boldsymbol{u}^n_\epsilon \in \mathrm{curl}(\mathcal{V}_{\boldsymbol{U}}) \subset \{f \in \mathcal{C}^0(U) \mid \nabla \cdot f \equiv 0\}$. 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 $\partial^{(1,1)}\mathcal{H}_{\boldsymbol{U}}[\psi^n_\epsilon]$. 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 3. The grid $\boldsymbol{U}$ and the parameter ϵ used to represent the velocity field are independent of the grid used for the evolution of $\boldsymbol{\mathcal{X}}$, the interpolant is also not restricted to Hermite cubics. Consequently, $\boldsymbol{U}$ can be made arbitrarily fine and ϵ arbitrarily small such that $\boldsymbol{u}^n_\epsilon$ approach the exact Biot-Savart velocity associated to the numerical vorticity $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}$. Furthermore, it is important to have a fine enough $\boldsymbol{U}$ so that the interpolation of ψ^n_ϵ 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 $\hat{\psi}^n_\epsilon$.

3.2. Time discretization

Section 3.1.2 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 $\boldsymbol{u}^n_\epsilon$.

The characteristic map can be evolved using the semigroup property (13). Let t_n be the discrete time steps, with $\Delta t = t_{n+1} - t_n$, we have

$$\boldsymbol{X}_{[t_{n+1},0]} = \boldsymbol{X}_{[t_n,0]} \circ \boldsymbol{X}_{[t_{n+1},t_n]}. \tag{39}$$

In order to approximate the one-step map $\boldsymbol{X}_{[t_{n+1},t_n]}$, we extend the velocity $\boldsymbol{u}^n_\epsilon$ 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

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}, t) := \sum_{i=0}^{p-1} l_i(t)\boldsymbol{u}^{n-i}_\epsilon(\boldsymbol{x}), \tag{40}$$

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 $\boldsymbol{u}^{n-i}_\epsilon$. In particular, this implies that if $\nabla \cdot \boldsymbol{u}^{n-i}_\epsilon = 0$ then $\nabla \cdot \tilde{\boldsymbol{u}} = 0$.

The one-step map $\boldsymbol{X}_{[t_{n+1},t_n]}$ is then approximated from (22b) by a k-stage Runge-Kutta integration of the velocity $\tilde{\boldsymbol{u}}$ along characteristic curves, for Δt backward in time:

$$\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}(\boldsymbol{x}) = \boldsymbol{x} - \Delta t \sum_{j=1}^{s} b_j k_j, \tag{41}$$

with

$$k_j = \tilde{\boldsymbol{u}}\left(\boldsymbol{x} - \Delta t \sum_{m=1}^{j-1} a_{jm} k_m, t_{n+1} - c_j \Delta t\right), \tag{42}$$

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

Remark 4. 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 3.1.2 is $\mathcal{C}^0$. Nonetheless, this does not pose a problem since $\mathcal{H}_{\boldsymbol{U}}[\psi_\epsilon^n]$ 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 $\boldsymbol{U}$ (in the limit of infinitely fine grid $\boldsymbol{U}$, $\boldsymbol{u}_\epsilon^n$ is smooth). Therefore, by appropriately scaling $\boldsymbol{U}$ with Δt the lack of smoothness does not affect the convergence.

3.3. 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 (39). We construct the numerical approximation of $\boldsymbol{X}$ 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 $\boldsymbol{M}$. We use the same evolve-project strategy as the CM method for linear advection to advance the characteristic map [1]:

$$\boldsymbol{\mathcal{X}}^{n+1} = \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}^n \circ \boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}\right], \tag{43}$$

$$\boldsymbol{\mathcal{X}}^0(\boldsymbol{x}) = \boldsymbol{x}, \tag{44}$$

where the one step map $\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}$ is given in (40) and (41). The velocities $\boldsymbol{u}_\epsilon^n$ required to define the one step map are given by (36), (37) and (38). The velocities $\boldsymbol{u}_\epsilon^{n-i}$ at previous time steps used in (40) are stored until no longer needed (for p steps where p is the order of the Lagrange polynomial chosen to represent $\tilde{\boldsymbol{u}}$).

3.3.1. Error estimates

We give some estimates on the characteristic map error:

$$\mathcal{E}^n := \|\boldsymbol{X}(\boldsymbol{x}, t_n) - \boldsymbol{\mathcal{X}}^n(\boldsymbol{x})\|_\infty. \tag{45}$$

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 (40) to be the velocity field of the modified equation (21). 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 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*

$$\|D^\alpha(\tilde{\boldsymbol{X}}(\boldsymbol{x}, t_n) - \boldsymbol{\mathcal{X}}^n(\boldsymbol{x}))\|_\infty = \mathcal{O}(t_n(\Delta x^{2-|\alpha|} \min(\Delta t, \Delta x^2 \Delta t^{-1}) + \Delta t^s)), \tag{46}$$

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 [1]. 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), 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 5. 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 define $\tilde{\boldsymbol{u}}$ 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 $U = \tilde{\boldsymbol{X}}(U)$,

$$\int\limits_U f(\omega_0(\boldsymbol{x}))dx = \int\limits_{\tilde{\boldsymbol{X}}(U)} f(\omega_0(\tilde{\boldsymbol{X}}(\boldsymbol{x},t_n)))\det(\nabla\tilde{\boldsymbol{X}}(\boldsymbol{x},t_n))dx = \int\limits_U f(\omega_0(\tilde{\boldsymbol{X}}(\boldsymbol{x},t_n)))dx, \tag{47}$$

for any measurable f. Therefore, we have

$$\begin{aligned}\int\limits_U f(\omega_0(\boldsymbol{\mathcal{X}}^n(\boldsymbol{x})))dx - \int\limits_U f(\omega_0(\boldsymbol{x}))dx &\approx \int\limits_U \nabla(f\circ\omega_0)(\boldsymbol{\mathcal{X}}^n(\boldsymbol{x}) - \tilde{\boldsymbol{X}}(\boldsymbol{x},t_n))dx \\ &\leq \|\nabla(f\circ\omega_0)\|_{L^2}\|\boldsymbol{\mathcal{X}}^n - \tilde{\boldsymbol{X}}(\cdot,t_n)\|_{L^2} = \mathcal{O}(t_n(\Delta x^2\min(\Delta t,\Delta x^2\Delta t^{-1}) + \Delta t^s))\end{aligned} \tag{48}$$

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. □

Remark 6. In other words, Corollary 1 shows that the discretized map reproduces with controlled error the relabelling symmetry in the solutions of 2D Euler equations, and hence the method can achieve accurate conservation of the Casimir invariants. This could be of interest for the study of 2D turbulence in the zero-viscosity limit of the Navier-Stokes equations. It is conjectured that higher-order Casimir invariants also exhibit turbulent cascades in their energy spectrum. However, due to larger aliasing error incurred in spectral methods for these higher order terms, it is generally difficult to obtain numerical evidence. One computation of the cascades in the fourth-order Casimir invariant appeared in [35], where the author used a specialized implementation of the FFT convolution routine to tack the scale transfers and provided numerical evidence for the presence of a direct transfer to small scales.

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

$$\tilde{\mathcal{E}}^n := \|\boldsymbol{X}(\boldsymbol{x},t_n) - \tilde{\boldsymbol{X}}(\boldsymbol{x},t_n)\|_\infty. \tag{49}$$

From Theorem 1, we then have that

$$\mathcal{E}^n \leq \tilde{\mathcal{E}}^n + \mathcal{O}(t_n(\Delta x^2\min(\Delta t,\Delta x^2\Delta t^{-1}) + \Delta t^s)). \tag{50}$$

Theorem 2. *From the above error decomposition we can deduce that the global truncation error for the characteristic map is*

$$\mathcal{E}^n = \mathcal{O}\left(\Delta t^s + \Delta x^2\min(\Delta t,\Delta x^2\Delta t^{-1}) + \Delta t^p\right). \tag{51}$$

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

$$\boldsymbol{X}_{[t_n,0]} - \tilde{\boldsymbol{X}}_{[t_n,0]} = (\boldsymbol{X}_{[t_{n-1},0]} - \tilde{\boldsymbol{X}}_{[t_{n-1},0]})\circ\boldsymbol{X}_{[t_n,t_{n-1}]} + \mathcal{O}(\boldsymbol{X}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]}). \tag{52}$$

From our estimates in (23), 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

$$\boldsymbol{X}_{[t_n,t_{n-1}]} - \tilde{\boldsymbol{X}}_{[t_n,t_{n-1}]} = \mathcal{O}(\Delta t\mathcal{E}^{n-1} + \Delta t^{p+1}), \tag{53}$$

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

Therefore,

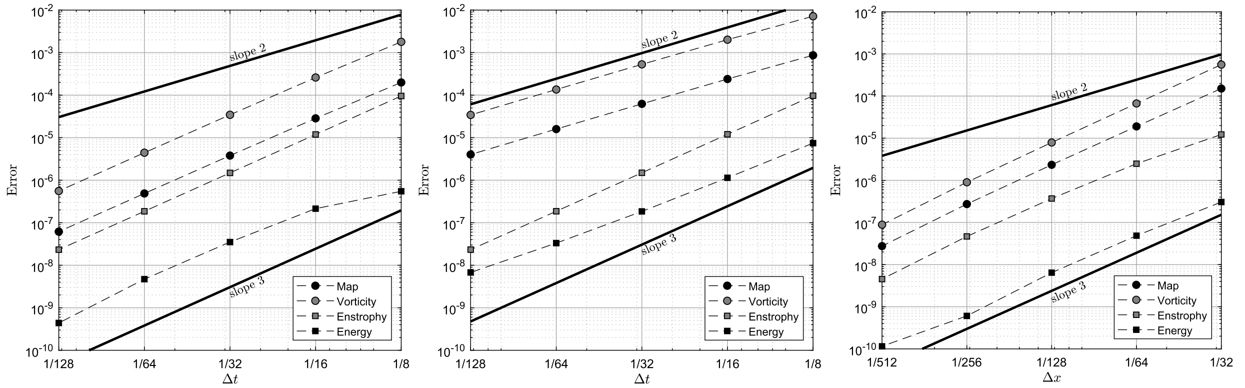
$$\tilde{\mathcal{E}}^n \leq \tilde{\mathcal{E}}^{n-1} + \mathcal{O}(\Delta t\mathcal{E}^{n-1} + \Delta t^{p+1}) \tag{54}$$

$$= \tilde{\mathcal{E}}^{n-1} + \mathcal{O}\left(\Delta t\tilde{\mathcal{E}}^{n-1} + \Delta t(t_n(\Delta x^2\min(\Delta t,\Delta x^2\Delta t^{-1}) + \Delta t^s)) + \Delta t^{p+1}\right), \tag{55}$$

which implies that

$$\tilde{\mathcal{E}}^n = \mathcal{O}\left(\Delta x^2\min(\Delta t,\Delta x^2\Delta t^{-1}) + \Delta t^s + \Delta t^p\right). \tag{56}$$

Together with (50), we obtain the desired error estimate. □



(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. $\Delta t = 1/512$.

Fig. 3. Map and vorticity error and conservation errors of enstrophy and energy for the Characteristic Mapping method without remapping.

Remark 7. Equation (53) in fact omits a sampling error incurred when defining $\boldsymbol{u}_\epsilon^n$, due to sampling the vorticity at discrete points. In this method, we define ω_ϵ^n 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_ϵ^n is independent of the computational grid for the map, hence the sampling error can be controlled separately.

3.3.2. 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.

We use a standard four-modes initial condition (66) 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:

$$\text{Map error } := \|\boldsymbol{\mathcal{X}}^n - \boldsymbol{X}(\cdot, t_n)\|_\infty, \tag{57a}$$

$$\text{Vorticity error } := \|\omega^n - \omega(\cdot, t_n)\|_\infty, \tag{57b}$$

$$\text{Enstrophy conservation error } := \|\omega^n\|_{L^2}^2 - \|\omega_0\|_{L^2}^2, \tag{57c}$$

$$\text{Energy conservation error } := \|\boldsymbol{u}^n\|_{L^2}^2 - \|\boldsymbol{u}_0\|_{L^2}^2. \tag{57d}$$

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 Fig. 3.

Figs. 3a and 3b 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 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.

Fig. 3c 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 time stepping in GALS methods. 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 Fig. 3c.

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 3.3.1. In practice, since a faithful 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. 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 semigroup property of the characteristic map mentioned in section 2.1.1. Numerically, the time t characteristic map can be constructed as the composition of several submaps of time subintervals:

$$\boldsymbol{\mathcal{X}}(\cdot,t) := \boldsymbol{\mathcal{X}}_{[\tau_1,0]} \circ \boldsymbol{\mathcal{X}}_{[\tau_2,\tau_1]} \circ \cdots \circ \boldsymbol{\mathcal{X}}_{[\tau_{m-1},\tau_{m-2}]} \circ \boldsymbol{\mathcal{X}}_{[t,\tau_{m-1}]}, \tag{58}$$

for some subdivision $0 < \tau_1 < \tau_2 < \cdots < \tau_{m-1} < t$. Each of these submaps are computed using the CM method described in previous sections. We initialize $\boldsymbol{\mathcal{X}}_{[\tau,\tau_i]}$ with the identity map at $\tau = \tau_i$ and evolved until a remapping time $\tau = \tau_{i+1}$ which can be determined dynamically. Once the remapping time is reached, we store $\boldsymbol{\mathcal{X}}_{[\tau_{i+1},\tau_i]}$ in memory and start computing the map for the next subinterval.

Heuristically speaking, each of the subintervals $[\tau_i, \tau_{i+1}]$ should be short enough such that the grid used to discretize $\boldsymbol{\mathcal{X}}_{[\tau_{i+1},\tau_i]}$ can correctly represent the deformation generated by the velocity $\tilde{\boldsymbol{u}}$ in this interval. Here we should point out that the intervals $[\tau_i, \tau_{i+1}]$ are different from the $[t_n, t_{n+1}]$ from previous sections. The latter has length Δt and is used to discretize the one-step maps $\boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}$; these are immediately composed and projected through $\boldsymbol{\mathcal{X}}_{[t_{n+1},t^\dagger]} = \mathcal{H}_{\boldsymbol{M}}\left[\boldsymbol{\mathcal{X}}_{[t_n,t^\dagger]} \circ \boldsymbol{\mathcal{X}}_{[t_{n+1},t_n]}\right]$, they are used to evolve the maps. The intervals $[\tau_i, \tau_{i+1}]$ on the other hand are longer, each $\boldsymbol{\mathcal{X}}_{[\tau_i,\tau_{i+1}]}$ comprise of several Δt steps. Once computed, they will be stored in memory.

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

$$e^n_{\text{det}} := \| \det \nabla \boldsymbol{\mathcal{X}}^n - 1 \|_\infty \tag{59}$$

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 ith submap is initialized with the identity map. After each time step, we compute the Jacobian determinant of $\boldsymbol{\mathcal{X}}_{[\tau_i+n\Delta t,\tau_i]}$ at off-grid sample points. If for some n, the Jacobian error exceeds δ_{det} for the first time, we define $\tau_{i+1} := \tau_i + n\Delta t$ and store $\boldsymbol{\mathcal{X}}_{[\tau_{i+1},\tau_i]}$. The same process is repeated for the $i+1$st submap using τ_{i+1} as initial time.

Although $\| \det \nabla \boldsymbol{\mathcal{X}}^n - 1 \|_\infty < \delta_{\text{det}}$ does not constitute a proper bound on the error in $\boldsymbol{\mathcal{X}}$, 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 $\|\boldsymbol{\mathcal{X}} - \tilde{\boldsymbol{X}}\|_\infty$ 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 $\nabla\boldsymbol{\mathcal{X}} - \nabla\tilde{\boldsymbol{X}}$ is small, we can justify the following first order expansion

$$e_{\text{det}} = \| \det \nabla \boldsymbol{\mathcal{X}} - \det \nabla \tilde{\boldsymbol{X}} \|_\infty \approx \left\| \operatorname{tr}\left(\nabla \tilde{\boldsymbol{X}}^{-1} (\nabla \boldsymbol{\mathcal{X}} - \nabla \tilde{\boldsymbol{X}}) \right) \right\|_\infty = \mathcal{O}(\|\nabla \boldsymbol{\mathcal{X}} - \nabla \tilde{\boldsymbol{X}}\|_\infty) \tag{60}$$

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 $\tilde{\boldsymbol{X}}$ 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 [8].

Due to the discrete nature of numerical computations, truncation errors are somewhat inevitable. In the present method, the evolution of $\boldsymbol{\mathcal{X}}^n$ (and only $\boldsymbol{\mathcal{X}}^n$) contains a diffusive type error since during 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 $\omega^n = \omega_0 \circ \boldsymbol{\mathcal{X}}^n$, 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 $\boldsymbol{X}$ and $\boldsymbol{\mathcal{X}}^n$ are both diffeomorphisms of Ω, there exists a diffeomorphism $\boldsymbol{\Psi}^n = (\boldsymbol{\mathcal{X}}^n)^{-1} \circ \boldsymbol{X}_{[t_n,0]}$ such that

$$\boldsymbol{X}_{[t_n,0]} = \boldsymbol{\mathcal{X}}^n \circ \boldsymbol{\Psi}^n. \tag{61}$$

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

$$\omega(\boldsymbol{x}, t_n) = \omega^n(\boldsymbol{\Psi}^n(\boldsymbol{x})), \tag{62}$$

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 $1 \le p < \infty$ are controlled by $\|\boldsymbol{\mathcal{X}} - \tilde{\boldsymbol{X}}\|_\infty$. 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 $\boldsymbol{\mathcal{X}}$ 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 $[\tau_i, \tau_{i+1}]$ 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

$$\omega^{n+1} = \omega^n + \Delta t \partial_t \omega^n, \tag{63}$$

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:

$$\nabla\omega(\cdot, t_n) = \nabla\omega_0 \nabla\boldsymbol{\mathcal{X}}_{[t_n,0]}. \tag{64}$$

Here, the exponential quantity is $\nabla\boldsymbol{\mathcal{X}}^n$. 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 (17) is the intrinsic generating process of the gradient just as multiplication is the generating process of the exponential function. We have

$$\nabla\boldsymbol{\mathcal{X}}_{[t,0]} = \prod_{j=1}^{m} \nabla\boldsymbol{\mathcal{X}}_{[\tau_j,\tau_{j-1}]}, \tag{65}$$

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 $1 + \mathcal{O}(\Delta\tau)$, 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. Numerical tests

In this section we present some numerical tests using the CM method for 2D Euler to simulate incompressible flows starting from some given initial condition. All computations are done on Matlab using double precision. However, the GALS

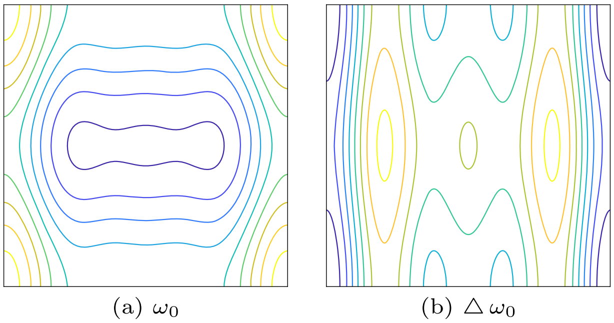


Fig. 4. Contour plot of the 4-modes initial vorticity and its Laplacian. Isolines are shown for 9 values uniformly distributed in the range of each function.

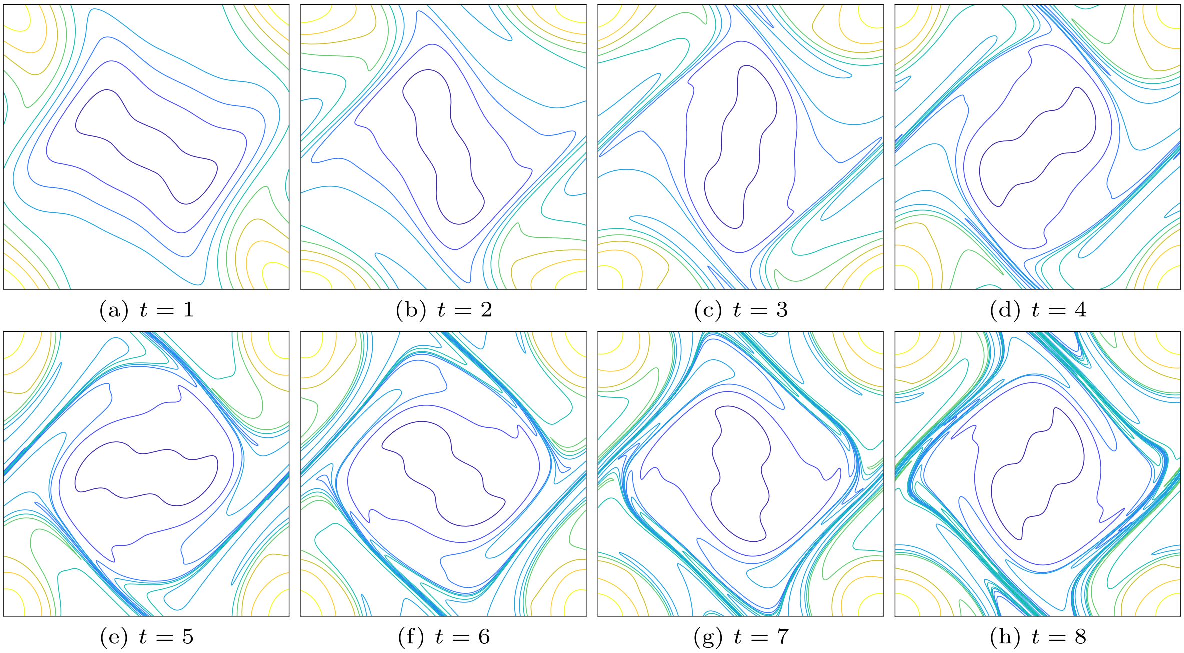


Fig. 5. Contour plot of the vorticity using 128^2 grid for $\boldsymbol{\mathcal{X}}^n$, 512^2 grid for representing ψ^n, $\Delta t = 1/32$ and $\delta_{\text{det}} = 10^{-4}$.

method for evolving the characteristic map uses a 4th order ϵ-difference method to replace the analytic chain rules involved in map updates. This simplifies implementation, but with our choice of $\epsilon = 5 \times 10^{-4}$, it effectively limits the machine precision for the characteristic map to around 10^{-13}.

4.1. "4-modes" test

We test the CM method for 2D incompressible Euler using the "4-modes" initial condition. We use the tests performed in [3] as reference. The initial vorticity is given by:

$$\omega_0(x, y) = \cos(x) + \cos(y) + 0.6\cos(2x) + 0.2\cos(3x). \tag{66}$$

This flow can roughly be characterised as two vortices of opposite signs partitioning a flat torus. The contour plot of ω_0 and $\triangle\,\omega_0$ are shown in Fig. 4, 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 Fig. 16, and for several times until $t = 8$ for the long time simulation in Figs. 5 an 6.

The same tests were performed in [3] 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 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

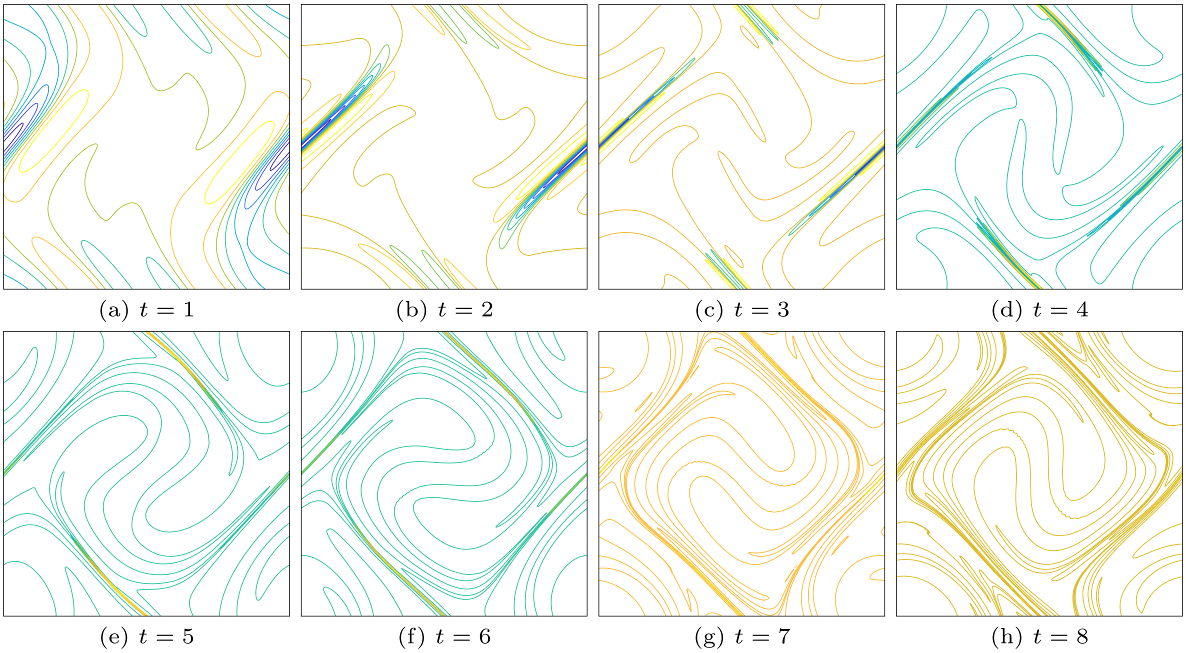


Fig. 6. Contour plot of the Laplacian of the vorticity using 128^2 grid for $\boldsymbol{\mathcal{X}}^n$, 512^2 grid for representing ψ^n, $\Delta t = 1/32$ and $\delta_{\text{det}} = 10^{-4}$.

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

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

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.

The simulations in this section were carried out on Matlab on a computer with an Intel Core i5-2320 3.00 GHz 4 cores processor and 8 GB 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.

Figs. 5 and 6 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 1 shows the number of remaps and total computational times required to reach the various plotting times.

From Figs. 5 and 6, 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 [3], 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 Figs. 5 and 6 where we 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

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

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	$-3.51 \cdot 10^{-7}$	$-1.64 \cdot 10^{-6}$	$-3.98 \cdot 10^{-6}$	$-7.51 \cdot 10^{-6}$

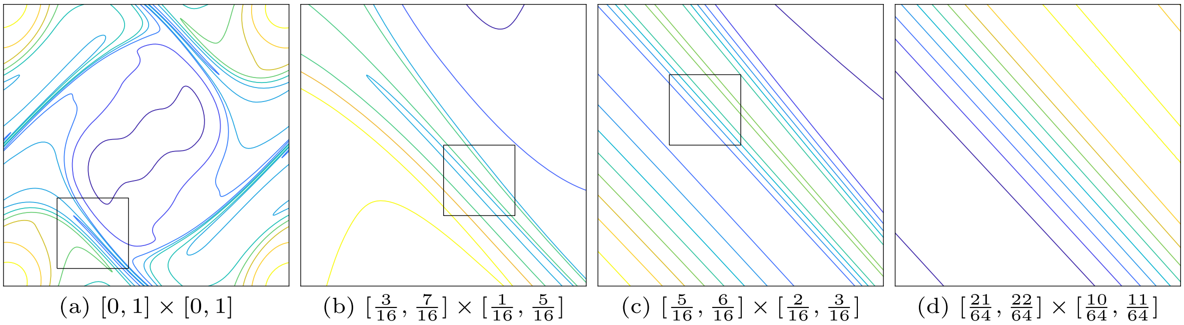

(a) $[0, 1] \times [0, 1]$ (b) $[\frac{3}{16}, \frac{7}{16}] \times [\frac{1}{16}, \frac{5}{16}]$ (c) $[\frac{5}{16}, \frac{6}{16}] \times [\frac{2}{16}, \frac{3}{16}]$ (d) $[\frac{21}{64}, \frac{22}{64}] \times [\frac{10}{64}, \frac{11}{64}]$

Fig. 7. Gradual 64× zoom on the vorticity at $t = 4$.

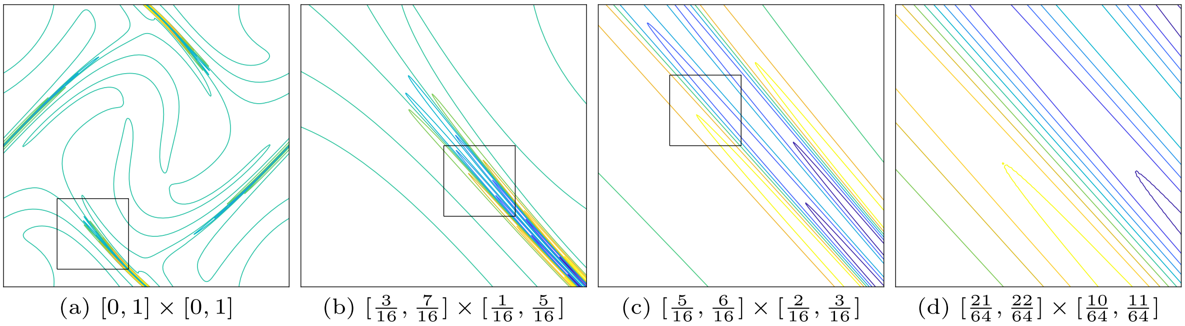

(a) $[0, 1] \times [0, 1]$ (b) $[\frac{3}{16}, \frac{7}{16}] \times [\frac{1}{16}, \frac{5}{16}]$ (c) $[\frac{5}{16}, \frac{6}{16}] \times [\frac{2}{16}, \frac{3}{16}]$ (d) $[\frac{21}{64}, \frac{22}{64}] \times [\frac{10}{64}, \frac{11}{64}]$

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

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 2.

Compared to a direct grid based representation of the vorticity, this growth is much slower. Indeed, in [3], using the 8th 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.

Figs. 7, 8, 9 and 10 demonstrate the arbitrary spatial resolution provided by the characteristic map. Each zoomed plot is sampled with the same number 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 Figs. 9a v.s. 9d). 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

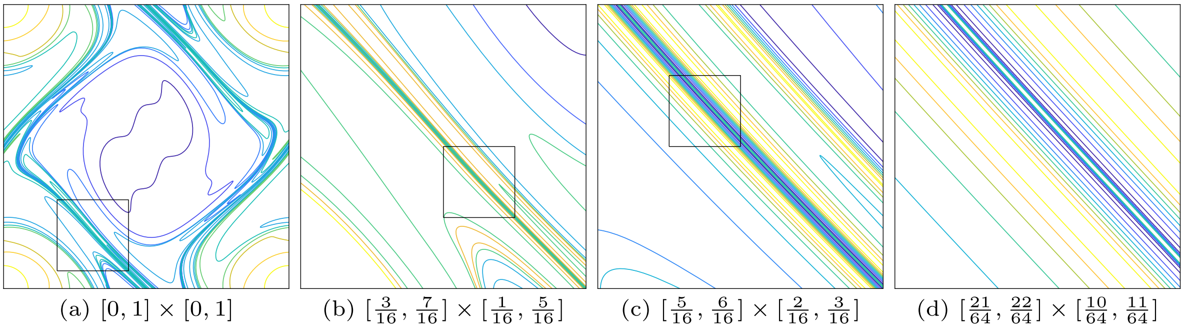


Fig. 9. Gradual 64× zoom on the vorticity at $t = 8$.

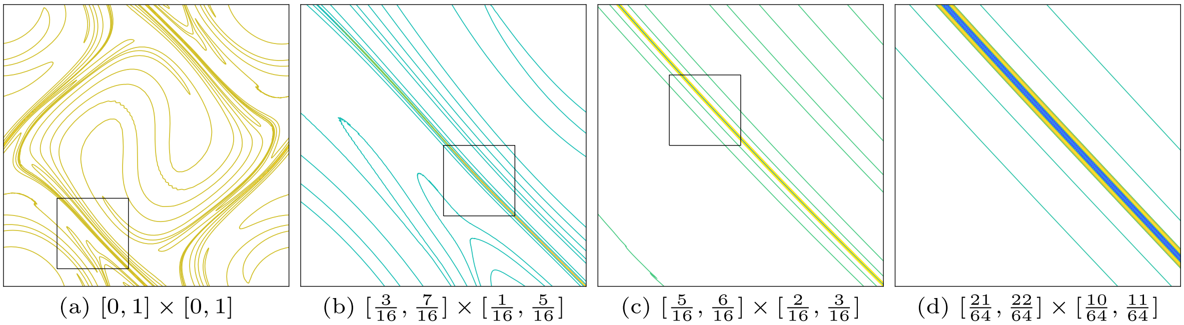


Fig. 10. Gradual 64× zoom on the Laplacian of the vorticity at $t = 8$.

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.

4.2. Random initial conditions

In this section, we perform the same tests as in section 4.1 on a randomly generated initial condition. The procedure to generate the random initial condition is given in [8]. 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 Kth 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 Kth 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 [3] until $t = 1$ using the Cauchy-Lagrange methods with 2048^2 spatial Fourier modes.

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. $K = 0, 1, \dots 32$; 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 Fig. 11. We ran the CM method on this initial condition using a 256^2 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 its Laplacian are shown at 0.5 time intervals in Figs. 12 and 13. The number of remaps and CPU times required are shown in Table 3. We also include a zoom of the vorticity and its Laplacian in Figs. 14 and 15.

As in the 4-modes test, we also observe slow growth in conservation errors for the random initial data in Table 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.

4.3. 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

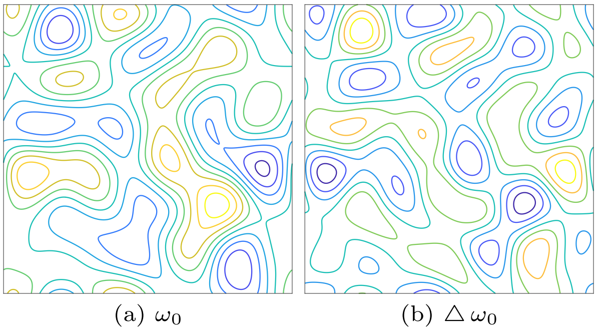


Fig. 11. Contour plot of the random initial vorticity and its Laplacian.

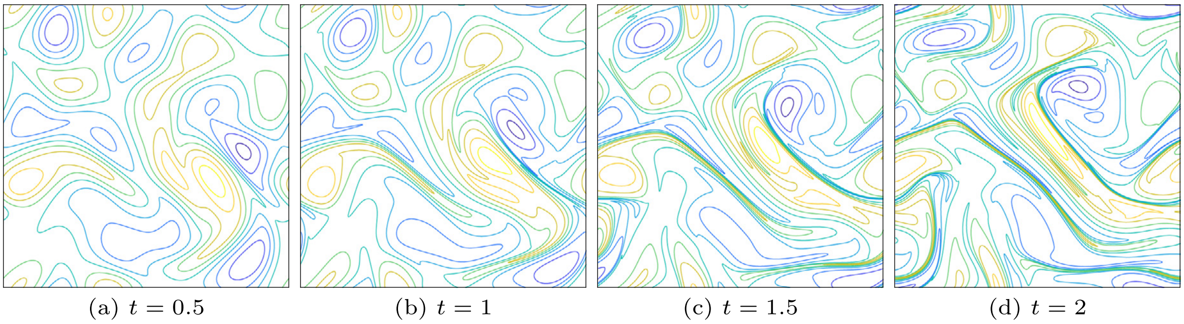


Fig. 12. Contour plot of ω^n for the random initial condition test using 256^2 grid for $\boldsymbol{\mathcal{X}}^n$, 1024^2 grid for representing ψ^n, $\Delta t = 1/128$ and $\delta_{\text{det}} = 10^{-4}$.

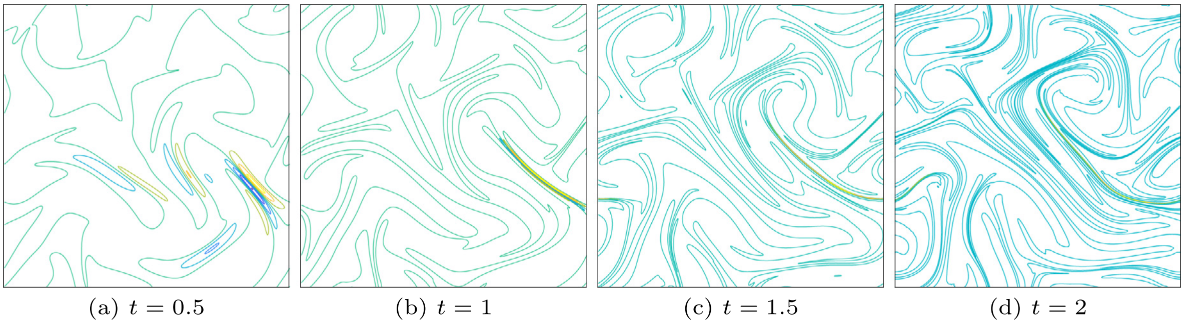


Fig. 13. Contour plot of $\triangle\,\omega^n$ for the random initial condition test using 256^2 grid for $\boldsymbol{\mathcal{X}}^n$, 1024^2 grid for representing ψ^n, $\Delta t = 1/128$ and $\delta_{\text{det}} = 10^{-4}$.

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

$$E_\omega(K) := \frac{1}{2} \sum_{K \le |\boldsymbol{k}| < K+1} |\hat{\omega}_{\boldsymbol{k}}|^2. \tag{67}$$

We compare the enstrophy spectrum from the CM method to that of the Cauchy-Lagrangian (CL8) method presented in [3]. Fig. 16 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 steps in the CL8 method, the times presented in [3] did not land exactly on $t = 3.5$ and 4).[1]

We see that in Fig. 16, the enstrophy spectra obtained from the CM method match 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

[1] The Cauchy-Lagrangian method employs variable length time steps. The enstrophy spectrum presented in [3] is 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.

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

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
Conservation errors for 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	$-7.74 \cdot 10^{-7}$	$-4.71 \cdot 10^{-6}$	$-3.79 \cdot 10^{-5}$	$-9.36 \cdot 10^{-5}$

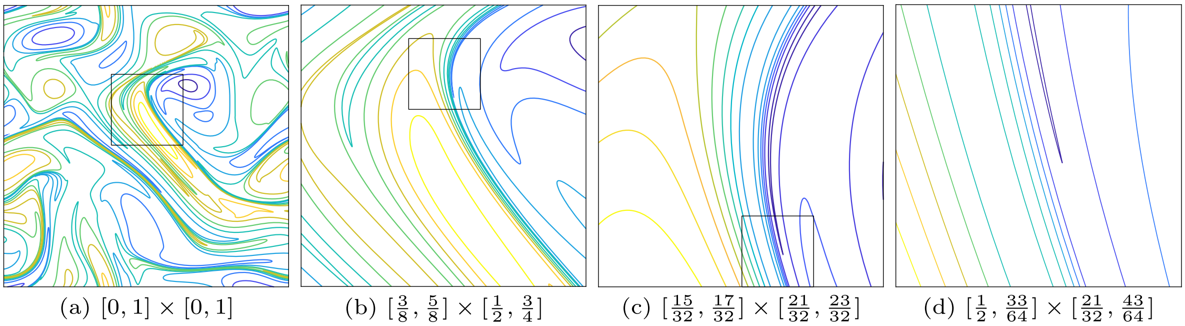

(a) $[0,1] \times [0,1]$ (b) $[\frac{3}{8}, \frac{5}{8}] \times [\frac{1}{2}, \frac{3}{4}]$ (c) $[\frac{15}{32}, \frac{17}{32}] \times [\frac{21}{32}, \frac{23}{32}]$ (d) $[\frac{1}{2}, \frac{33}{64}] \times [\frac{21}{32}, \frac{43}{64}]$

Fig. 14. Gradual 64× zoom on the vorticity at $t = 2$.

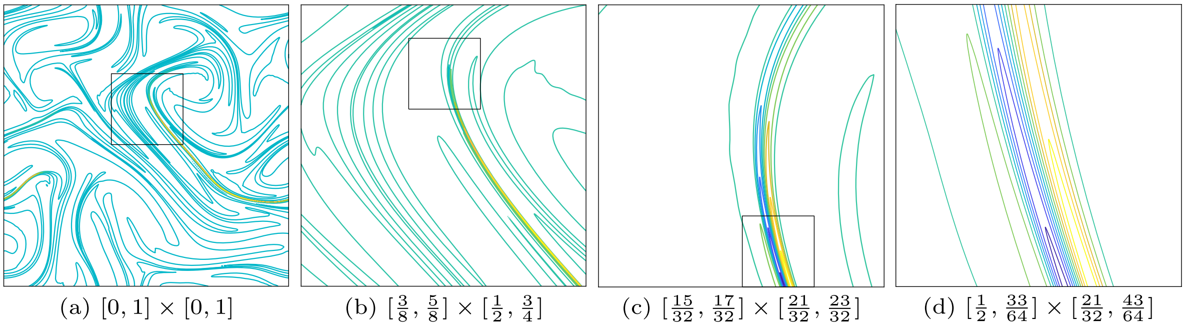

(a) $[0,1] \times [0,1]$ (b) $[\frac{3}{8}, \frac{5}{8}] \times [\frac{1}{2}, \frac{3}{4}]$ (c) $[\frac{15}{32}, \frac{17}{32}] \times [\frac{21}{32}, \frac{23}{32}]$ (d) $[\frac{1}{2}, \frac{33}{64}] \times [\frac{21}{32}, \frac{43}{64}]$

Fig. 15. Gradual 64× zoom on the Laplacian of the vorticity at $t = 2$.

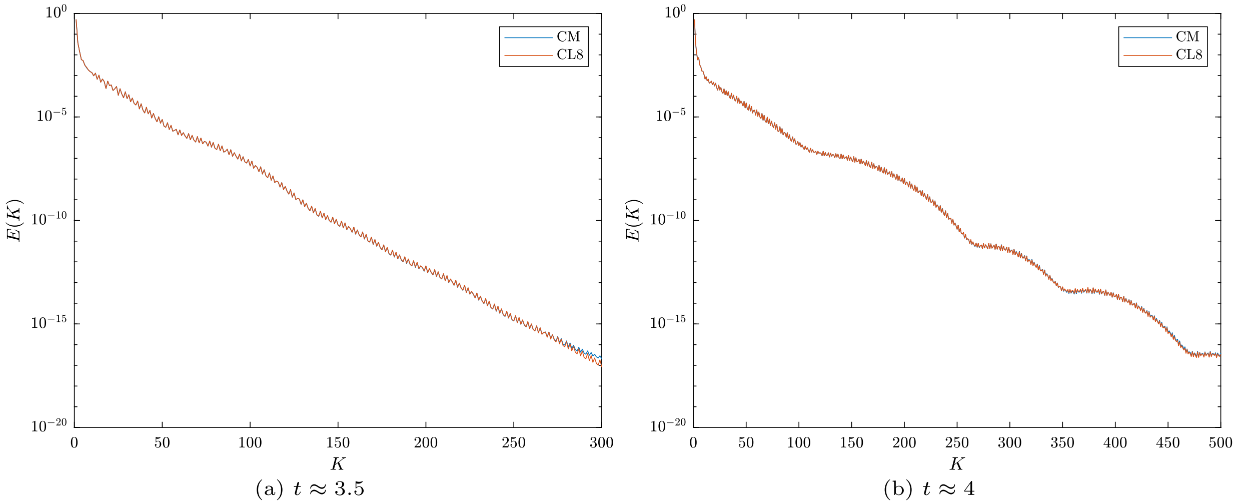
other. The plots are provided in vector-graphics format, zooming in the tail of the spectrum shows some discrepancies between the two curves. This is in large part due to the two simulations not having exactly the same final times). This suggests that small time deformations can be accurately represented 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 enstrophy spectrum is typically

$$E_\omega(K) \sim K^\alpha e^{-2\delta K}. \tag{68}$$

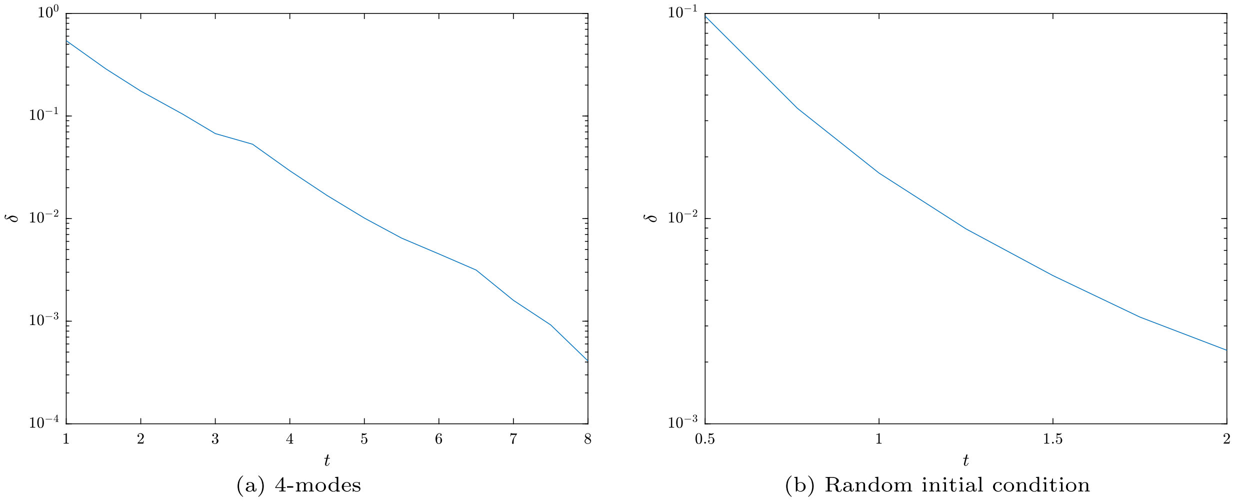
The rate of the exponential, δ, is the radius of analyticity and governs the spatial truncation error. For a grid which resolves a maximum frequency of $k_{\max}$, the spatial truncation error scales like $e^{-\delta k_{\max}}$.

Fig. 17 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 grid size of order 10^4 would be needed to properly resolve the solution. We see that the 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,



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

Fig. 16. Decay in the enstrophy spectrum at times 3.5 and 4.



(a) 4-modes (b) Random initial condition

Fig. 17. Radius of analyticity $\delta(t)$ vs. time.

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 grow as the spatial features in the solution increase.

4.4. 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 Fig. 18). 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.

The fine scale features produced by this flow require 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 Figs. 19 and 20.

The final frame (Fig. 19) 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. Fig. 21 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

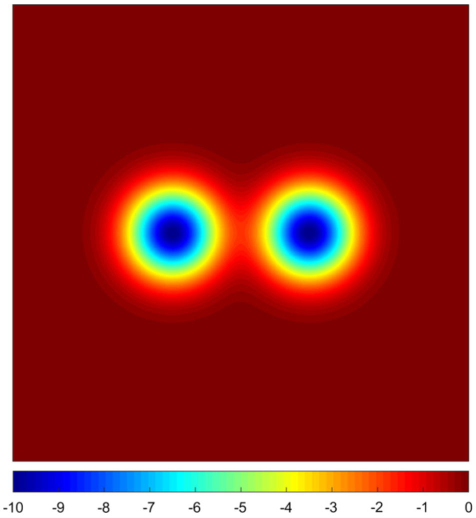


Fig. 18. Initial vorticity for the vortex merger simulation.

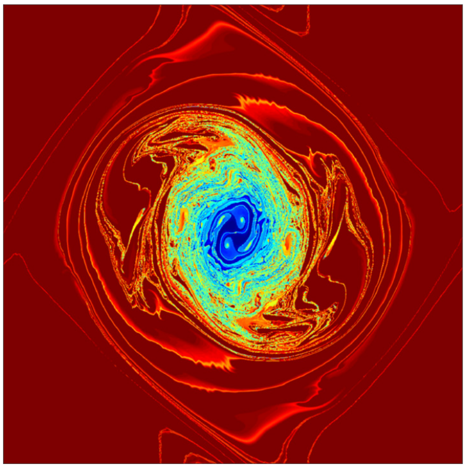
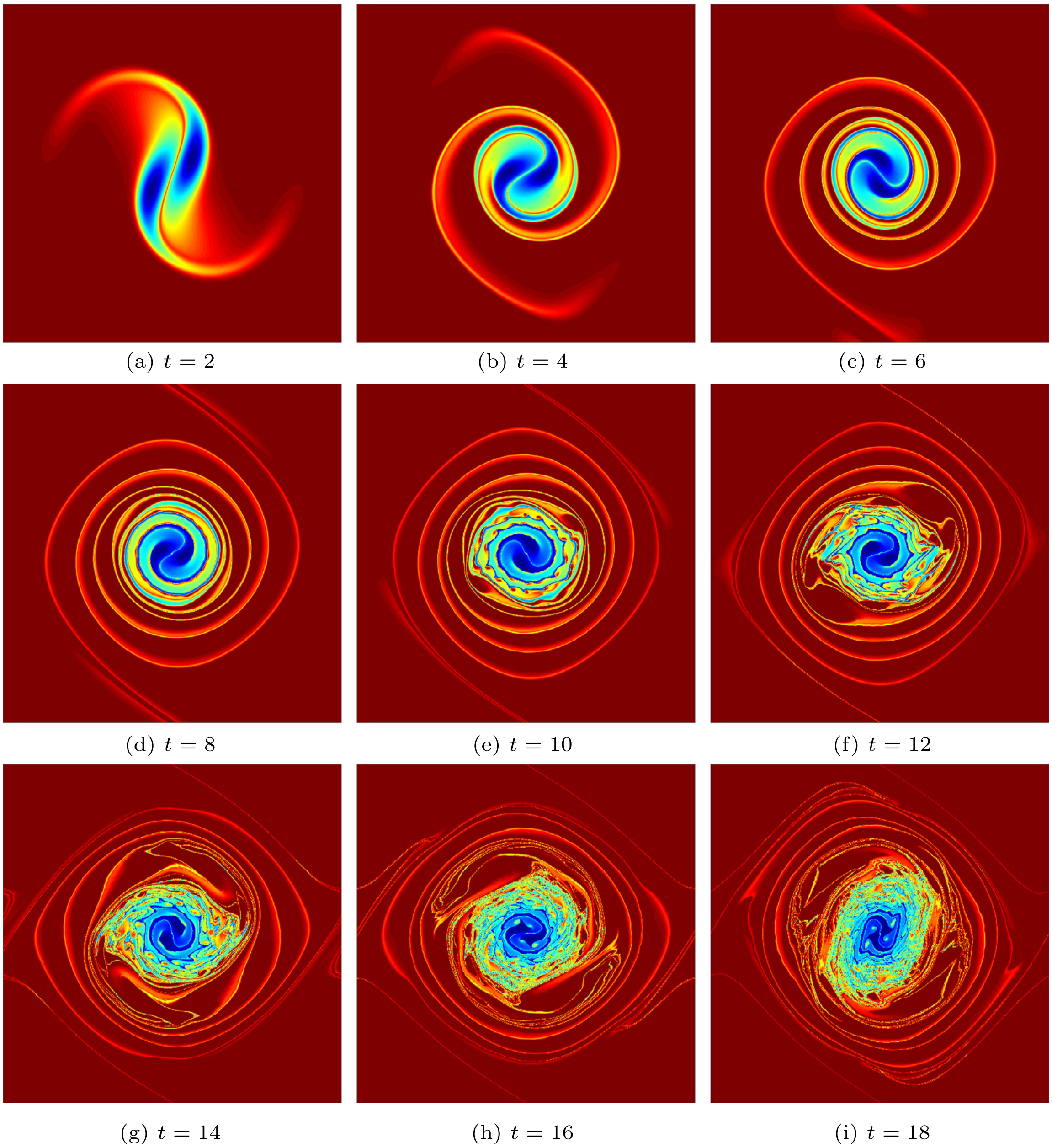


Fig. 19. Vorticity field at time $t = 20$.

obtaining high resolution images of arbitrarily small regions in the domain. For instance, Fig. 21l 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 showcase several advantageous properties of the CM method for the 2D incompressible Euler equations. Firstly, the submap decomposition using the semigroup 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 achieve the same enstrophy spectrum as an 8192^2 grid direct vorticity solver (see Fig. 16). 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 become 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 Figs. 7, 8, 9, 10 and more thoroughly in 21 in section 4.4, 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) $t = 18$

Fig. 20. Evolution of the vorticity field in the vortex merger simulation.

5. Conclusion

In this paper, we have presented a novel numerical method for solving the 2D incompressible Euler equations. The work in this paper is an extension of the CM method for linear advection and is based on the GALS and Jet-Scheme frameworks. This method is unique 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 led to the development of this method. Firstly, the arbitrarily fine scales typically generated by an inviscid flow lead to the functional representation of the vorticity field though the 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, in particular in the zoomed view of the solutions. Secondly, the possible exponential growth in the vorticity gradient lead to the use of the semigroup structure of the flow maps to decompose the characteristic map: exponential growth can be generated by a composition of

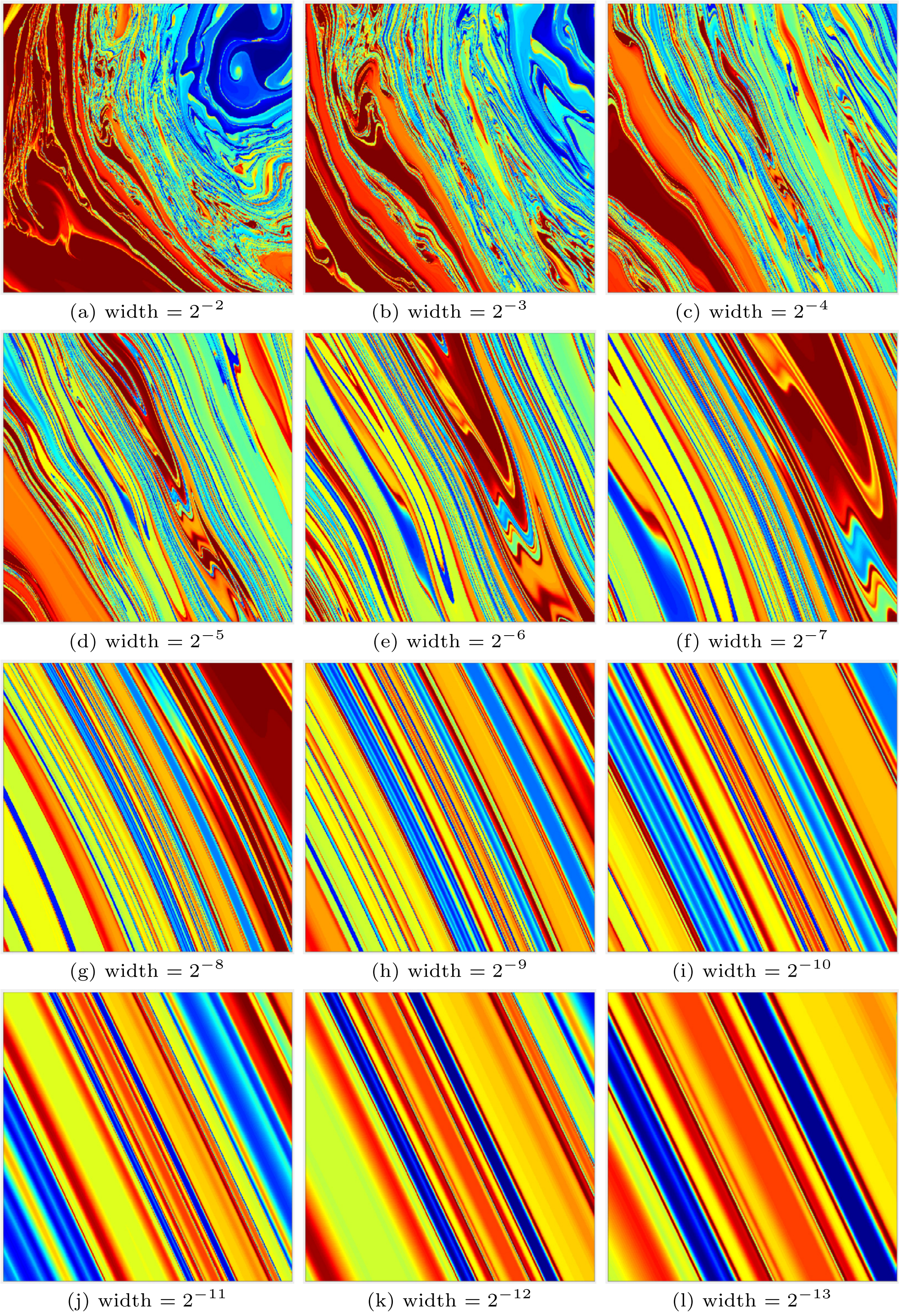


Fig. 21. Gradual zoom on the $t = 20$ frame (Fig. 19), each subfigure is a 2× zoom on the previous.

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 use of the characteristic map for the simulation of inviscid fluids opens many new possibilities for future research. The next step is to include the vorticity-stretching term in order to generalize this 2D method to three-dimensional problems. Another possible direction would be to use the level-set formulation by Deng et al. [19] to reduce the 3D Euler equations to a system of scalar advection equations: the solutions of all advection equations would then be provided by the same characteristic map. There is also a possibility of including forcing terms by tracking an integral quantity along characteristics and potentially extensions to the method which deals with forcing terms while maintaining the characteristic structure. In the CM method, the assumption of a transport structure does not apply to equations with a diffusive source term such as the incompressible Navier-Stokes equations. In cases where solutions do not exhibit arbitrary fine scales, the source terms integrated along forward characteristics need to counteract the fine scales created by the advection, resulting in additional numerical difficulties. Direct grid-based integration of such source terms is not desirable with the current framework as it undermines the advantages the CM method. We are currently examining various approaches to the advective structure of the deformation maps to better manage the viscous term in the Navier-Stokes equations. Lastly, the implementation of boundary conditions and the application of the method to complex geometries is also an important problem to study. These are the current directions of our research as we believe that the CM method provides a unique and appropriate framework for solving more general problems in computational fluid dynamics.

CRediT authorship contribution statement

The research was conducted by the following students: Xi-Yuan Yin, Olivier Mercier, and Badal Yadav.
The research was initiated and directed by the following faculties: Kai Schneider and Jean-Christophe Nave.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work by X.-Y. Y. was partially supported by FRQNT B2X (Fonds de recherche du Québec – Nature et technologies) and by Hydro-Québec. The work by O.M. was partially supported by NSERC CGS-M (Natural Sciences and Engineering Research Council of Canada). K. S. acknowledges support by the French Research Federation for Fusion Studies carried out within the framework of the EFDA (European Fusion Development Agreement). The work by J.-C. N. was partially supported by the NSERC Discovery and Discovery Accelerator programs. K. S. also thankfully acknowledges the kind hospitality at McGill University in the fall of 2012. We also wish to thank the authors of [3] for providing the energy spectrum data for comparison in Fig. 16.

References

[1] O. Mercier, X.-Y. Yin, J.-C. Nave, The Characteristic Mapping method for the linear advection of arbitrary sets, SIAM J. Sci. Comput. 42 (3) (2020) A1663–A1685.
[2] J.-C. Nave, R.R. Rosales, B. Seibold, A Gradient-Augmented Level Set method with an optimally local, coherent advection scheme, J. Comput. Phys. 229 (10) (2010) 3802–3827.
[3] O. Podvigina, V. Zheligovsky, U. Frisch, The Cauchy-Lagrangian method for numerical analysis of Euler flow, J. Comput. Phys. 306 (2016) 320–342.
[4] V. Yudovich, The flow of a perfect, incompressible liquid through a given region, in: Soviet Physics Doklady, vol. 7, 1963, p. 789.
[5] A. Kiselev, V. Šverák, Small scale creation for solutions of the incompressible two-dimensional Euler equation, Ann. Math. 180 (3) (2014) 1205–1220.
[6] S.A. Denisov, Infinite superlinear growth of the gradient for the two-dimensional Euler equation, Dyn. Syst. 23 (3) (2009) 755–764.
[7] C.W. Bardos, E.S. Titi, Mathematics and turbulence: where do we stand?, J. Turbul. 14 (3) (2013) 42–76.
[8] S.S. Ray, U. Frisch, S. Nazarenko, T. Matsumoto, Resonance phenomenon for the Galerkin-truncated Burgers and Euler equations, Phys. Rev. E 84 (1) (2011) 016301.
[9] R. Pereira, R. Nguyen van yen, M. Farge, K. Schneider, Wavelet methods to eliminate resonances in the Galerkin-truncated Burgers and Euler equations, Phys. Rev. E 87 (3) (2013) 033017.
[10] R.A. Gentry, R.E. Martin, B.J. Daly, An Eulerian differencing method for unsteady compressible flow problems, J. Comput. Phys. 1 (1) (1966) 87–118.
[11] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
[12] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. R. Astron. Soc. 181 (3) (1977) 375–389.
[13] J.J. Monaghan, Smoothed particle hydrodynamics, Rep. Prog. Phys. 68 (8) (2005) 1703.
[14] A.J. Chorin, Numerical study of slightly viscous flow, J. Fluid Mech. 57 (4) (1973) 785–796.
[15] J.C. Bowman, M.A. Yassaei, A. Basu, A fully Lagrangian advection scheme, J. Sci. Comput. 64 (1) (2015) 151–177.
[16] C.W. Hirt, A.A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.
[17] J. Donea, S. Giuliani, J.-P. Halleux, An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions, Comput. Methods Appl. Mech. Eng. 33 (1–3) (1982) 689–723.
[18] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, vol. 153, Springer Science & Business Media, 2006.
[19] J. Deng, T.Y. Hou, X. Yu, A level set formulation for the 3D incompressible Euler equations, Methods Appl. Anal. 12 (4) (2005) 427–440.

[20] J. Stam, Stable fluids, in: Siggraph, vol. 99, 1999, pp. 121–128.

[21] N. Foster, R. Fedkiw, Practical animation of liquids, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, ACM, 2001, pp. 23–30.

[22] B. Seibold, J.-C. Nave, R.R. Rosales, Jet schemes for advection problems, Discrete Contin. Dyn. Syst., Ser. B 17 (4) (2012) 1229–1259.

[23] H. Kohno, J.-C. Nave, A new method for the level set equation using a hierarchical-gradient truncation and remapping technique, Comput. Phys. Commun. 184 (6) (2013) 1547–1554.

[24] R. Deiterding, M.O. Domingues, S.M. Gomes, K. Schneider, Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible Euler equations, SIAM J. Sci. Comput. 38 (5) (2016) S173–S193.

[25] D. Kolomenskiy, J.-C. Nave, K. Schneider, Adaptive Gradient-Augmented Level Set method with multiresolution error estimation, J. Sci. Comput. 66 (1) (2016) 116–140.

[26] A. Guittet, M. Theillard, F. Gibou, A stable projection method for the incompressible Navier-Stokes equations on arbitrary geometries and adaptive Quad/Octrees, J. Comput. Phys. 292 (2015) 215–238.

[27] M. Theillard, F. Gibou, D. Saintillan, Sharp numerical simulation of incompressible two-phase flows, J. Comput. Phys. 391 (2019) 91–118.

[28] P. Constantin, V. Vicol, J. Wu, Analyticity of Lagrangian trajectories for well posed inviscid incompressible fluid models, Adv. Math. 285 (2015) 352–393.

[29] W. Wolibner, Un théorème sur l'existence du mouvement plan d'un fluide parfait, homogène, incompressible, pendant un temps infiniment long, Math. Z. 37 (1) (1933) 698–726.

[30] V. Arnold, Sur la géométrie différentielle des groupes de lie de dimension infinie et ses applications à l'hydrodynamique des fluides parfaits, Ann. Inst. Fourier 16 (1) (1966) 319–361.

[31] J.E. Marsden, S. Shkoller, The anisotropic Lagrangian averaged Euler and Navier-Stokes equations, Arch. Ration. Mech. Anal. 166 (1) (2003) 27–46.

[32] K. Mohseni, B. Kosović, S. Shkoller, J.E. Marsden, Numerical simulations of the Lagrangian averaged Navier-Stokes equations for homogeneous isotropic turbulence, Phys. Fluids 15 (2) (2003) 524–544.

[33] J.E. Marsden, S. Shkoller, Global well-posedness for the Lagrangian averaged Navier-Stokes (LANS-α) equations on bounded domains, Philos. Trans. R. Soc. Lond. Ser. A: Math. Phys. Eng. Sci. 359 (1784) (2001) 1449–1468.

[34] M. Oliver, S. Shkoller, The Vortex Blob method as a second-grade non-Newtonian fluid, Commun. Partial Differ. Equ. 26 (2001) 295–314.

[35] J.C. Bowman, Casimir cascades in two-dimensional turbulence, in: Advances in Turbulence XII, Springer, 2009, pp. 685–688.