Journal of Computational Physics 398 (2019) 108859

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Advection without compounding errors through flow map composition

Chinmay S. Kulkarni, Pierre F.J. Lermusiaux *

Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA - 02139, United States of America

ARTICLE INFO

Article history:
Received 18 December 2018
Received in revised form 25 June 2019
Accepted 28 July 2019
Available online 6 August 2019

Keywords:
Advection-diffusion-reaction
Transport
Flow map composition
Hyperbolic equation
Finite volume
Super-accuracy

ABSTRACT

We propose a novel numerical methodology to compute the advective transport and diffusion-reaction of tracer quantities. The tracer advection occurs through flow map composition and is super-accurate, yielding numerical solutions almost devoid of compounding numerical errors, while allowing for direct parallelization in the temporal direction. It is computed by implicitly solving the characteristic evolution through a modified transport partial differential equation and domain decomposition in the temporal direction, followed by composition with the known initial condition. This advection scheme allows a rigorous computation of the spatial and temporal error bounds, yields an accuracy comparable to that of Lagrangian methods, and maintains the advantages of Eulerian schemes. We further show that there exists an optimal value of the composition timestep that yields the minimum total numerical error in the computations, and derive the expression for this value. We develop schemes for the addition of tracer diffusion, reaction, and source terms, and for the implementation of boundary conditions. Finally, the methodology is applied in three flow examples, namely an analytical reversible swirl flow, an idealized flow exiting a strait undergoing sudden expansion, and a realistic ocean flow in the Bismarck sea. New benchmark problems for advection-diffusion-reaction schemes are developed and used to compare and contrast results with those of classic schemes. The results highlight the theoretical properties of the methodology as well as its efficiency, super-accuracy with minimal numerical errors, and applicability in realistic simulations.

1. Introduction

Advective fluid transport plays an extremely important role in several disciplines of science and engineering. It is essential in classical fluid mechanics [7,43], atmospheric and ocean dynamics [99,95] as well as in microfluidics [108,105] and biological flows [101,5]. Often times, it occurs as a part of a larger dynamical context involving advection, diffusion and/or reaction processes. Advective transport on its own is governed by the following classic partial differential equation (PDE) subject to certain initial and boundary conditions,

$$\frac{\partial \rho(\boldsymbol{x},t)\alpha(\boldsymbol{x},t)}{\partial t} + \nabla \cdot (\rho(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)\alpha(\boldsymbol{x},t)) = 0\,. \tag{1}$$

* Corresponding author.
E-mail addresses: chinmayk@mit.edu (C.S. Kulkarni), pierrel@mit.edu (P.F.J. Lermusiaux).

https://doi.org/10.1016/j.jcp.2019.108859
0021-9991/

In equation (1), α is the 'tracer', *i.e.* the passive quantity (per unit mass of fluid) whose transport is to be studied, $\boldsymbol{v}$ is the known dynamic velocity field, and ρ is the density of the fluid, with $(\boldsymbol{x}, t)$ denoting the spatial and temporal coordinates. The solutions $\alpha(\boldsymbol{x}, t)$ can be obtained analytically for a few known velocity fields with well-posed initial and boundary conditions. In most realistic cases however, the dynamic velocity fields $\boldsymbol{v}(\boldsymbol{x}, t)$ are derived as the output of some computational fluid dynamics (CFD) simulation or are observed quantities. In such cases, one numerically solves equation (1) to study the evolution of the tracer. The pure hyperbolic nature of equation (1) presents significant numerical challenges and most schemes are either susceptible to a high degree of unwanted numerical diffusion or exhibit excessive nonphysical oscillations [73]. Although the use of extremely fine mesh is an option for high fidelity solutions, the computational cost quickly becomes prohibitive and often rules out this possibility for large, realistic problems [72].

Computing the accurate solution of equation (1) is not trivial and there are several classes of numerical methods for such solutions. First, the more traditional 'Eulerian methods' or 'optimal spatial methods' [9] compute the numerical solution using finite differences [17,60,93], finite volumes [73,47,88,81], finite elements [46,31,44], or other numerical discretization techniques on a spatial grid. It can be proven that central difference approximations for the temporal and spatial derivatives commonly yields excessive oscillations and unstable solutions [73]. Considering an upstream bias ('upwinding') for the spatial gradients, where the upwind direction is decided by the local velocity fields, removes these spurious oscillations to a great extent. Some of the benefits of these methods include rigorous theoretical bounds on the spatial and temporal errors and straightforward use of well-established robust and high-order accurate numerical methods [84,96,113,104,91, 89], many of which are implemented in numerical toolboxes [2,24,53] However, they tend to exhibit significant numerical diffusion when the temporal truncation error dominates [73]. Such spurious diffusion effects can be large enough when they compound to overshadow the physical tracer dispersion [9], and hence produce inaccurate solutions. In order to minimize such errors, one can use high-order accurate schemes that rely on wide computational stencils which increase the cost and can pose challenges near boundaries. The use of hp-adaptivity [51,12] reduces the local truncation errors but does not eliminate their compounding over time. Further, explicit numerical schemes often pose restrictions on the Courant number whereas implicit methods involve a large matrix–vector or a nonlinear system solve. For all such Eulerian methods, the computational cost can thus become high and there are no direct avenues for temporal parallelization.

The second type, often referred to as 'Lagrangian methods' leverage the hyperbolic nature of the system to compute tracer transport by utilizing the characteristic lines of equation (1). The various techniques developed under the umbrella of Lagrangian methods involve the method of characteristics [94], the modified method of characteristics [21,25,98], and transport-diffusion methods [41,45] amongst others. Eulerian-Lagrangian methods (ELM) [4,86,87] and their extensions such as localized adjoint-based methods (ELLAM) [9,119] are used for advection-diffusion equations where the advective transport is computed by a Lagrangian method, the diffusive component is treated on an Eulerian grid, and finally both solutions are combined through operator-splitting methods [120]. Lagrangian methods significantly reduce the temporal truncation error, alleviate the Courant number restrictions often found in Eulerian methods, and can be readily parallelized [80]. They pose problems however in the rigorous treatment of boundary fluxes, especially in domains with multiple inlets and outlets, and in maintaining mass conservation. Further, Lagrangian methods involve some form of particle tracking, which implies that the solutions are seldom computed on uniform spatial grids leading to a loss of resolution in some regions of the domain [70,27,28]. Further, the Lagrangian methods require the velocity field defined not only at the (known) grid points, but also at arbitrary locations in the domain. Although high-order reconstruction or interpolation schemes can be used, they increase the overall computational cost. This also implies that rigorous estimates for the spatial error bounds are often not available.

Another class of advection schemes, called 'jet schemes' [102,85] or 'characteristic mapping methods' [83] propose a semi-Lagrangian 'advect-and-project' approach to numerically solve equation (1). These schemes use Hermite polynomials to achieve local high-order accuracy, and the advection process is simulated by Lagrangian particle tracking. For classical examples of level set function evolutions [90], it is observed that these methods can produce solutions with accuracy comparable to discontinuous Galerkin methods. These methods may be combined with appropriate parallel-in-time methods [32] to achieve some speedup.

It would be ideal to obtain numerical diffusion- and dispersion-free accurate solutions comparable to those of Lagrangian methods, while maintaining the theoretical consistency, soundness and ease of use of Eulerian methods (*i.e.* on a spatially fixed grid). In this work, we develop a new method to numerically solve equation (1), referred to as the *method of composition*, and extend it to advection-diffusion-reaction problems. It generalizes prior results in the area of flow map computation for dynamical systems [122,123,70]. As shall be shown, this is a special case of the method of composition, *i.e.* solving equation (1) but for very specific initial and boundary conditions. A main novelty of the present methodology is its capability to handle the advection of any passive tracer while respecting space and time dependent boundary conditions and to also account for any classic diffusion and reaction operators. We further prove that the order of accuracy of the methodology is at least equal to the minimum amongst the orders of accuracy of the constituent operations, which serves as a generalization of Theorem 1 from You and Leung [122]. We also prove that there exists a particular value of the composition timestep that minimizes the total numerical error. Finally, this method is readily parallelizable in time, which leads to a much lower computational time, given enough resources.

The new method of composition in a nutshell involves three components. We first divide the temporal domain into smaller intervals and compute a modified flow map by solving equation (1) taking into account the boundary conditions over each of these intervals. These flow maps are mutually independent and hence can be computed in parallel. Second, we compose these independent (modified) flow maps together through appropriate composition (interpolation) operators to yield the flow map over the entire time interval, similar to the work of [8,102,83]. Last, we compose this modified flow map with the initial condition for the tracer field to yield the advected tracer field. There are multiple advantages to this method: (i) as the flow map computations are mutually independent, it can be readily parallelized in the temporal direction; (ii) as we simply solve equation (1), the computation can be performed on any spatial discretization with any existing solvers or numerical toolboxes; (iii) due to the Eulerian nature of the computations, we can easily obtain error estimates; and (iv) as the flow maps are independently computed and then composed, the truncation errors are not compounded in time and we obtain an accuracy comparable to that of the particle methods.

The paper is organized as follows. The problem statement and the associated notation are specified in Section 2. The theoretical description of the composition based advection is in Section 3; emphasizing the fundamentals behind the new methodology, the details of the three building blocks introduced above, the addition of tracer diffusion and source terms, and the implementation of boundary conditions. Section 4 derives specific numerical properties of the new composition based numerical advection. We first analyze the error estimates in terms of the errors of the spatial, temporal, and interpolation schemes used. We show that the spatial order of accuracy of this method is at least equal the minimum amongst the orders of the spatial gradient scheme and the interpolation scheme. The temporal order of accuracy is equal to that of the time marching scheme used. We next provide an intuitive argument for the existence of an *optimal composition timestep*, that is the timestep value that results in the minimum total numerical error. We then derive the expression for this optimal timestep in terms of the errors of the specific numerical schemes employed and explain how the optimal composition timestep can be evaluated directly or partly numerically.

In Section 5, we illustrate properties and capabilities of the new methodology and compare and contrast results with those of classic high and low-order schemes. Three flow fields are considered. The first is a forward-backward advection in a classic analytical swirl flow. Due to the temporal symmetry of the flow, the initial and final transported tracer fields are the same. This allows us to accurately capture and illustrate the error bounds, error trends with respect to the simulation parameters, and effectiveness of the optimal timestep value for schemes of varied orders of accuracy. We then add a diffusion term and create a new benchmark test for advection-diffusion in the swirl flow. We observe that the net spurious numerical diffusion is minimal for the method of composition and validate results with quantitative error metric comparisons. The second example is an idealized version of a fluid flow exiting a narrow constriction that creates unsteady eddies and meanders downstream. We include a tracer source to confirm that composition based advection can be easily combined with forcing and reactions. Final tracer fields are compared with those of classic advection schemes and results show again that composition based advection admits minimal numerical diffusion and dispersion. Finally, we consider a realistic ocean flow in the Bismarck sea around a possible deep-sea mining site [13]. We simulate the advection of sediment plumes resulting from such deep-sea mining operations. After 5 days of advection by the underlying dynamic ocean currents, due to the absence of compounding numerical errors, we again see that the composition based advection leads to a more accurate tracer evolution than regular PDE advection schemes. We also simulate this case with trajectory based Lagrangian advection. Results indicate that the composition based advection and the Lagrangian trajectory advection yield advected tracer fields with comparable accuracy, as expected. However, for the trajectory advection case, we observe a loss of spatial accuracy in certain regions of the domain due to the discrete nature of particle advection, whereas this issue does not arise in the method of composition.

2. Problem statement

In this section, we define the governing equations and our notation. Let the spatial domain be denoted by $\Omega \subset \mathbb{R}^n$ and its boundary by $\partial\Omega$. The outward normal over $\partial\Omega$ is defined by $\hat{\mathbf{n}}_{\partial\Omega}$. The (unsteady) velocity field is given by $\boldsymbol{v} : \Omega \times [0, T] \to \mathbb{R}^n$, where $[0, T]$ is the time interval of interest. Further, we assume that the velocity $\boldsymbol{v}$ is Lipschitz continuous in the spatial coordinate, with a Lipschitz constant $\mathcal{L}_{\boldsymbol{v}}$. That is, $|\boldsymbol{v}(\boldsymbol{x}, t) - \boldsymbol{v}(\boldsymbol{y}, t)| \le \mathcal{L}_{\boldsymbol{v}} |\boldsymbol{x} - \boldsymbol{y}|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ and $0 \le t \le T$. The tracer, whose advection is to be studied is denoted by $\alpha : \Omega \times [0, T] \to \mathbb{R}^+ \cup \{0\}$. The density of the fluid flow at $(\boldsymbol{x}, t)$ is $\rho(\boldsymbol{x}, t)$. The initial tracer concentration over the entire domain is known and denoted by $\alpha_0 : \Omega \to \mathbb{R}^+ \cup \{0\}$. We assume that α_0 is also Lipschitz continuous, with Lipschitz constant $\mathcal{L}^0_\alpha$, *i.e.* $|\alpha_0(\boldsymbol{x}) - \alpha_0(\boldsymbol{y})| \le \mathcal{L}^0_\alpha |\boldsymbol{x} - \boldsymbol{y}|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.

The evolution of $\alpha(\boldsymbol{x}, t)$ is governed by a vector-valued advective transport equation (2),

$$\begin{aligned} \frac{\partial \rho(\boldsymbol{x}, t)\alpha(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot (\rho(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t)\alpha(\boldsymbol{x}, t)) = 0\,, \\ \alpha(\boldsymbol{x}, 0) = \alpha_0(\boldsymbol{x})\,. \end{aligned} \tag{2}$$

This equation models the transport of a passive tracer α defined per unit mass of the fluid, under the velocity field $\boldsymbol{v}$, with an initial distribution $\alpha(\boldsymbol{x}, t = 0) = \alpha_0(\boldsymbol{x})$. Further, from mass conservation, we have that,

$$\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot (\rho(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t)) = 0\,, \tag{3}$$

Equation (2) can be simplified into equation (4) using equation (3), with no further assumptions.

$$\begin{aligned} \frac{\partial \alpha(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \nabla\alpha(\boldsymbol{x}, t) = 0\,, \\ \alpha(\boldsymbol{x}, 0) = \alpha_0(\boldsymbol{x})\,. \end{aligned} \tag{4}$$

If the tracer concentration was defined per unit volume, α_v, the equation equivalent to equation (2) is given by equation (5),

$$\begin{aligned}&\frac{\partial \alpha_v(\boldsymbol{x},t)}{\partial t} + \nabla \cdot (\boldsymbol{v}(\boldsymbol{x},t)\alpha_v(\boldsymbol{x},t)) = 0\,,\\ &\alpha_v(\boldsymbol{x},0) = \alpha_{v,0}(\boldsymbol{x})\,.\end{aligned} \tag{5}$$

Equation (5) simplifies to equation (4) if one assumes an incompressible velocity field, *i.e.* $\nabla \cdot \boldsymbol{v} = 0$.

Our goal is to numerically compute the tracer field $\alpha(\boldsymbol{x},t)$ for all $0 \le t \le T$, with minimum practical numerical error, by solving equation (4) over a discretized spatio-temporal domain. Without loss of generality, we assume the domain to be homogeneous in all spatial directions, and discretize it into $N_{\boldsymbol{x}}$ distinct nodes, with a spacing of Δx along each direction. Similarly, the temporal domain is discretized into N_t distinct intervals, each of duration Δt. Finally, the numerically computed value of a quantity $\bullet$ is denoted by $\tilde{\bullet}$.

3. The method of composition for tracer advection

This section derives and discusses the method of composition to compute tracer advection, defined by equation (4), as well as the handling of tracer diffusion, reaction, and source terms, and the imposition of generic boundary conditions. The specific numerical aspects of the method are described in Section 4.

3.1. Tracer advection through flow maps

As commonly done, e.g. [19,73,40,70,28], we now utilize the Lagrangian interpretation of equation (4). We consider the motion of an individual fluid parcel initially located at $\boldsymbol{x}_0$. This parcel is carrying with it the tracer value $\alpha_0(\boldsymbol{x}_0)$. As the parcel is passively transported, the tracer value does not change along its trajectory.

$$\begin{aligned}\frac{D\alpha}{Dt} = \frac{\partial \alpha(\boldsymbol{x},t)}{\partial t} &+ \boldsymbol{v}(\boldsymbol{x},t)\cdot\nabla\alpha(\boldsymbol{x},t) = 0\,,\\ \alpha(\boldsymbol{x},0) &= \alpha_0(\boldsymbol{x})\end{aligned} \tag{6}$$

where $\frac{D\bullet}{Dt}$ represents the material derivative of the quantity $\bullet$. Instead of devising numerical schemes to solve for equation (6), we however take advantage of the flow map. The (forward) flow map defined between a given start time 0 and a given end time T describes the mapping of positions in Ω under the advective action of the velocity field $\boldsymbol{v}$ during that time interval $[0, T]$. It is given by equation (7),

$$\phi_0^T(\boldsymbol{x}_0) = \boldsymbol{x} \quad \text{where } \frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}(\boldsymbol{x}(t),t) \text{ with } \boldsymbol{x}(0) = \boldsymbol{x}_0\,. \tag{7}$$

Similarly, the backward flow map is the inverse of the forward flow map and is given by equation (8),

$$\phi_T^0(\boldsymbol{x}) = \boldsymbol{x}_0 \quad \text{where } \frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}(\boldsymbol{x}(t),t) \text{ with } \boldsymbol{x}(0) = \boldsymbol{x}_0\,. \tag{8}$$

Clearly, the ordinary differential equation (ODE) defining the forward and backward flow maps is the characteristic ODE of the transport PDE 6 [73]. That is, the tracer value at $\boldsymbol{x}_0$ at the initial time travels with the fluid along the characteristic originating at $\boldsymbol{x}_0$, and ends up at a position $\boldsymbol{x}$ at the end time t, where $\boldsymbol{x}_0$ and $\boldsymbol{x}$ are related by equation (7) or equation (8). This yields equation (9),

$$\alpha(\boldsymbol{x},t) = \alpha(\boldsymbol{x}_0,0) = \alpha_0(\boldsymbol{x}_0)\,. \tag{9}$$

Using equation (8), we obtain the expression for the tracer field at time t, given by equation (10),

$$\alpha(\boldsymbol{x},t) = \alpha_0(\phi_t^0(\boldsymbol{x}))\,. \tag{10}$$

That is, the tracer value at any position $\boldsymbol{x}$ at time t is the composition of the initial tracer field with the backward flow map between time 0 and t, evaluated at $\boldsymbol{x}$.

It must be noted that our definition of the flow map is a slight extension of the classical flow maps from dynamical systems theory. This is due to the fact that we look at the flow maps as solutions of advective transport PDEs, and hence have the ability to impose specific boundary conditions. However, we will still use the term ‘flow map’ to refer to such flow maps that include the imposition of boundary conditions (to be discussed in Section 3.5).

3.2. Flow map computation

A governing PDE for the flow map ϕ_t^0 consists of equation (6) with initial condition given by equation (11),

$$\alpha_0(\boldsymbol{x}) = \boldsymbol{x}\,. \tag{11}$$

Indeed, equation (10) then implies that:

$$\alpha(\boldsymbol{x}, t) = \alpha_0(\phi_t^0(\boldsymbol{x})) = \phi_t^0(\boldsymbol{x})\,. \tag{12}$$

Thus, by simply solving equation (6) with the initial condition given by equation (11), we can compute the backward flow map ϕ_t^0. We refer to this as the *'PDE based flow map computation'* [70,71]. The specifics of the boundary conditions for solving equation (6) are discussed in Section 3.5.

Due to the reversibility of equation (6), without loss of generality, we only deal with the specifics of the backward flow map in this work. The corresponding conclusions hold true also for the forward flow map, and can be derived by simply flipping the temporal index.

3.3. PDE based flow map composition

Computing flow maps over longer time intervals may be expensive, especially without the possibility of parallelization. Further as the governing equation is hyperbolic, the flow map computation is susceptible to compounding diffusive and dispersive numerical errors. To alleviate these problems, we utilize and build upon the idea first proposed by [8] and later used by [123] to efficiently compute the flow map. Instead of computing the flow map over the entire duration, the interval is broken up into multiple smaller intervals of the order of a numerical timestep, and the individual flow maps (over these smaller intervals) are computed independently of each other. Finally, flow maps are composed to obtain the flow map over the larger interval.

Let us assume that the time interval $[0, T]$ is broken up into $k = 1, \cdots, N_c$ distinct flow map intervals (of length $\Delta t_c = T/N_c$ each), and the flow map is independently computed over each of these intervals. Further, Δt_c is assumed an integral multiple of the discrete timestep Δt, *i.e.* $\Delta t_c = M\Delta t$. Then, the flow map over the entire duration is given by,

$$\phi_{N_c}^0 = \phi_1^0 \circ \phi_2^1 \circ \ldots \circ \phi_{N_c-1}^{N_c-2} \circ \phi_{N_c}^{N_c-1}\,, \tag{13}$$

where $\phi_{k+1}^k = \phi_{(k+1)\Delta t_c}^{(k)\Delta t_c}$. Each of these individual flow maps is computed by using equation (6), with the initial condition given by equation (11), restated here for future convenience,

$$\begin{aligned} &\frac{\partial \phi_{k+1}^k(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \nabla \phi_{k+1}^k(\boldsymbol{x}, t) = 0\,, \\ &\phi_{k+1}^k(\boldsymbol{x}, t_k) = \boldsymbol{x}\,. \end{aligned} \tag{14}$$

Once the flow maps over the individual timesteps are computed, interpolation is required to compose the new flow map at the positions given by the previous (total) flow map. We denote the interpolation operator associated with the flow map ϕ_{k+1}^k by $\mathcal{I}_{k+1}^k$. Note that this interpolation operator is specific to the flow map, and does not change with the argument of the flow map: hence it needs to be computed only once (per flow map), see also Section 4.3. With the inclusion of this interpolation operator, equation (13) is transformed into equation (15).

$$\phi_{N_c}^0(\boldsymbol{x}) = \mathcal{I}_1^0 \phi_1^0 \circ \mathcal{I}_2^1 \phi_2^1 \circ \ldots \circ \mathcal{I}_{N_c-1}^{N_c-2} \phi_{N_c-1}^{N_c-2} \circ \phi_{N_c}^{N_c-1}(\boldsymbol{x})\,. \tag{15}$$

While computing the flow maps using this method, each individual computation can be readily parallelized. Further, the method also allows the very efficient computation of $\phi_{t+\tau}^0$ for any τ given ϕ_t^0, as we can use $\phi_{t+\tau}^0 = \mathcal{I}_t^0 \phi_t^0 \circ \phi_{t+\tau}^t$. Finally, the application of general boundary conditions is straightforward, as will be discussed in the following section.

To summarize, the solution of equation (6) is computed by solving equation (16),

$$\alpha(\boldsymbol{x}, T) = \alpha(\boldsymbol{x}, N_c \Delta t_c) = \mathcal{I}_\alpha \alpha_0 \circ \mathcal{I}_1^0 \phi_1^0 \circ \mathcal{I}_2^1 \phi_2^1 \circ \ldots \circ \mathcal{I}_{N_c-1}^{N_c-2} \phi_{N_c-1}^{N_c-2} \circ \phi_{N_c}^{N_c-1}(\boldsymbol{x})\,, \tag{16}$$

where $\mathcal{I}_\alpha$ is the interpolation operator for the initial tracer field $\alpha_0(\boldsymbol{x})$. The individual flow maps are computed according to equation (14), using a timestep of Δt. The methodology is schematized in Fig. 1.

3.4. Inclusion of diffusion, sources, and reactions terms

We now incorporate the effects of tracer diffusion, sources, sinks, and reactions while computing the advective contribution through the method of composition. We utilize the commonly used technique of operator splitting. Splitting methods include Lie splitting [79] and Strang splitting [107], which we shall discuss. As will be seen, the Lie splitting introduces a first-order splitting error while Strang splitting introduces a second-order splitting error. However, in certain situations,

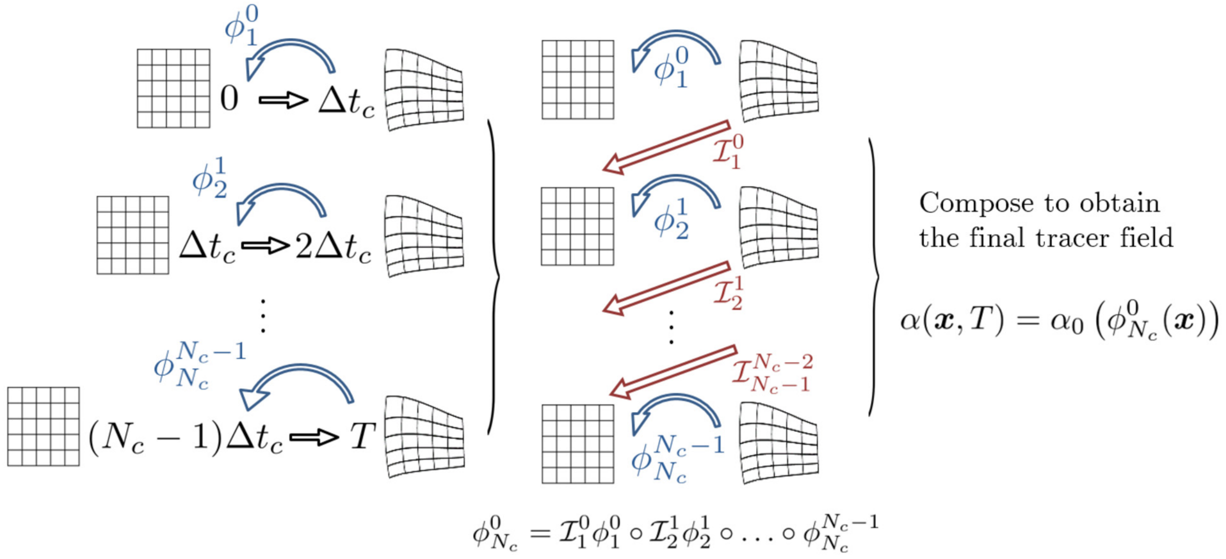

ϕ_1^0
$0 \Rightarrow \Delta t_c$
ϕ_2^1
$\Delta t_c \Rightarrow 2\Delta t_c$
$\phi_{N_c}^{N_c-1}$
$(N_c-1)\Delta t_c \Rightarrow T$
ϕ_1^0
$\mathcal{I}_1^0$
ϕ_2^1
$\mathcal{I}_2^1$
$\mathcal{I}_{N_c-1}^{N_c-2}$
$\phi_{N_c}^{N_c-1}$
Compose to obtain the final tracer field
$\alpha(\boldsymbol{x},T) = \alpha_0\left(\phi_{N_c}^0(\boldsymbol{x})\right)$
$\phi_{N_c}^0 = \mathcal{I}_1^0\phi_1^0 \circ \mathcal{I}_2^1\phi_2^1 \circ \ldots \circ \phi_{N_c}^{N_c-1}$

Fig. 1. Composition based advection methodology. The first step involves computing the individual flow maps (while accounting for boundary conditions), which can be done in parallel. The second step then involves computing the cumulative flow map over the entire time interval by sequentially interpolating and composing the individual flow maps. The final step involves composing this cumulative flow map with the initial tracer field to obtain the final tracer field. We note that the methodology is schematized for $\mathbb{R}^2$ but it generalizes to $\mathbb{R}^n$ as it is agnostic to the spatial dimension.

these operator splitting methods can either yield a higher order of accuracy or even be exact and introduce no splitting error [57].

Let us consider the general form of advection-diffusion-reaction equation (17) which is an extension of equation (4),

$$\frac{\partial\alpha(\boldsymbol{x},t)}{\partial t} + \boldsymbol{v}\cdot\nabla\alpha(\boldsymbol{x},t) = \nabla\cdot(\kappa(\boldsymbol{x},t)\nabla\alpha(\boldsymbol{x},t)) + S_\alpha(\alpha(x,t),\boldsymbol{x},t) \quad \text{with} \quad \alpha(\boldsymbol{x},0) = \alpha_0(\boldsymbol{x})\,, \tag{17}$$

where $\kappa(\boldsymbol{x},t)$ is the diffusivity of the tracer and $S_\alpha(\alpha(x,t),\boldsymbol{x},t)$ serves to model the reaction terms (if any) and the source/sink contributions (if any). We shall refer to $S_\alpha(\alpha(x,t),\boldsymbol{x},t)$ as the 'source term', keeping in mind the fact that it may very well represent any of the aforementioned phenomena.

In our methodology, we solve for the advection contribution using flow map composition. The source contribution is computed explicitly or implicitly for a first-order scheme, or integrated over one time stage or timestep for a higher-order time-marching scheme. This is followed by solving for the diffusion implicitly. An implicit solve for the diffusion operator is often preferred as it is stiff and imposes a strict CFL condition. We will denote this resulting implicit diffusion operator by $\mathcal{D}$.

If using the Lie operator splitting, the split equations are given by equation (18).

$$\begin{aligned}
&\alpha_* = \alpha(\phi_{t+\Delta t}^t(\boldsymbol{x}),t)\,,\\
&\alpha_{**} = S_\alpha(\alpha_*,\boldsymbol{x},t)\Delta t\,,\\
&\alpha(\boldsymbol{x},t+\Delta t) = \mathcal{D}(\alpha_* + \alpha_{**})\;.
\end{aligned} \tag{18}$$

The BCs are incorporated in the advection contribution and the diffusion contribution, according to the corresponding fluxes (see Section 3.5 next). The possible ill-posedness of the diffusion operator (under zero Neumann boundary conditions everywhere, for example) can be eliminated by any of the established singularity removal techniques [78,115]. It can further be shown that the leading order truncation error term in the splitting error is given by equation (19) [106].

$$\mathcal{E}_{\text{lie}} = \frac{\Delta t}{2}\left(\nabla\cdot\boldsymbol{v}\left(S_\alpha - \frac{\partial S_\alpha}{\partial\alpha}\alpha\right) + \nabla\cdot(\kappa\nabla S_\alpha) - \frac{\partial S_\alpha}{\partial\alpha}(\nabla\cdot(\kappa\nabla\alpha))\right) \tag{19}$$

Thus, the Lie splitting is also second-order accurate if the (i) source term is absent (*i.e.* an advection – diffusion equation), (ii) diffusion is absent (*i.e.* an advection – reaction equation) and the velocity field $\boldsymbol{v}$ is divergence free or the source term is linear in α.

If second-order splitting is desired, one can use the Strang splitting method given by equation (20).

$$\begin{aligned}
\alpha_* &= \alpha(\phi^t_{t+\frac{\Delta t}{2}}(\boldsymbol{x}), t)\,,\\
\alpha_{**} &= S_\alpha(\alpha_*, \boldsymbol{x}, t)\frac{\Delta t}{2}\,,\\
\alpha' &= \mathcal{D}(\alpha_* + \alpha_{**})\,,\\
\alpha'_* &= \alpha'(\phi^{t+\frac{\Delta t}{2}}_{t+\Delta t}(\boldsymbol{x}))\,,\\
\alpha &= S_\alpha\left(\alpha'_*, \boldsymbol{x}, t+\frac{\Delta t}{2}\right)\frac{\Delta t}{2}\,.
\end{aligned} \tag{20}$$

Similar to the Lie splitting, the boundary conditions are accounted separately in the advection and the diffusion computation. Note that we chose the order of operations in the splitting to ensure that the implicit diffusion solve is only carried out once for computational efficiency. However, this sequence is not unique, and maybe altered depending on the specific application. For specific forms of the source term, it is also possible to absorb it in the advection / diffusion operators and in such cases the computation would be appropriately altered.

Even though the Strang splitting is second-order accurate in general, there are specific cases under which it is exact and no splitting error is introduced. We borrow the results from Lanser and Verwer [57] who detail the sufficient conditions for the absence of the splitting error. They prove that for equation (17) with Strang splitting, no splitting error exists if (i) S_α is at most linear in α and independent of $\boldsymbol{x}$, and (ii) $\boldsymbol{v}$ and κ are independent of $\boldsymbol{x}$. Note that this is rarely possible in realistic cases, and hence there will always be a splitting error introduced. Lanser and Verwer [57] detail several ingenious approaches to split equation (17) based on the specific nature of various terms involved to minimize the splitting error. Although higher-order splitting methods are possible [49], they are rarely used in practice due to the computational overhead and complexity of implementation. Further, it is typically observed that the temporal/spatial errors in the computation of the individual terms (*i.e.* advection, diffusion, and reaction) often dominate the splitting errors, which further curbs the need to go to higher-order accurate splitting methods [117].

In our examples, we solve an advection – diffusion equation in Section 5.1.2 and an advection – reaction equation in Section 5.2. We use the Lie splitting method for both. Note that, for the first case, the source term is absent, and, for the second, diffusion is absent and the velocity field is divergence free (as it is obtained from an incompressible flow simulation). Thus Lie splitting is second-order accurate in both cases.

3.5. Implementation of boundary conditions

As equation (6) and equation (17) are PDEs, boundary conditions (BCs) are typically required to solve them, especially when inlets and/or outlets are present.

Previous works that compute PDE based flow maps [70,71] solve equation (14) which essentially amounts to equation (6) for a very specific initial condition, *i.e.* equation (11). The examples studied either involve closed domains (such as the analytical double gyre flow) or open domains with a very specific set of BCs, stating that either the flow map field (ϕ) is free to leave the domain or, when new flow map positions are to enter, they bear the value equal to the value at the position at the initial time. These BCs, well-posed for the flow map problem, are summarized in equation (21),

$$\frac{\partial\phi(\boldsymbol{x},t)}{\partial\hat{\mathbf{n}}_{\partial\Omega}} = 0 \quad \text{if} \quad \hat{\mathbf{n}}_{\partial\Omega}\cdot\boldsymbol{v} > 0 \quad \text{and} \quad \phi(\boldsymbol{x},t) = \phi(\boldsymbol{x},0) \quad \text{if} \quad \hat{\mathbf{n}}_{\partial\Omega}\cdot\boldsymbol{v} < 0\,. \tag{21}$$

For equation (17) and general tracer advection-diffusion-reaction problems, these BCs are however not sufficient as one may have time dependent inlets, outlets with varying strengths, specified tracer advection-diffusion fluxes, or more complex conditions. The nature of the BCs may itself also locally change in time. None of these possibilities are accounted for in equation (21). More general BCs are thus needed.

For tracer advection-reaction without diffusion, equation (17) becomes hyperbolic as equation (6) and equation (14). Either a single Dirichlet or Neumann BC is then commonly imposed at boundaries. For open boundaries, due to the hyperbolic nature, if the velocity vector at the boundary is outwards then no BCs are necessary, as the tracer then is simply advected outside the domain. However, BCs are commonly provided when the velocity vector points into the domain. In some cases, BCs may be irrelevant even for open domains and the equations can be exactly solved without BCs [58,29]. This occurs for example when the whole domain of computation, e.g. the whole fluid, is deformed according to the advection process itself.

Next, we discuss the generic implementation of Dirichlet and Neumann type BCs, noting that results extend to other well-posed conditions. We discuss these BCs when the velocity vector points inwards, but the treatment extends to the outward BCs when they are needed.

3.5.1. Dirichlet boundary condition

Let us assume that the BC for equation (6) consists of provided tracer values, given by equation (22),

$$\alpha(\boldsymbol{x},t) = \alpha_{BC}(\boldsymbol{x},t) \quad \text{for } \boldsymbol{x}\in\partial\Omega \text{ such that } \boldsymbol{v}\cdot\hat{\mathbf{n}}_{\partial\Omega} < 0\,. \tag{22}$$

For regular PDE based advection, such BCs are typically imposed by using ghost cells, *i.e.* numerical cells at the domain boundaries with the tracer value set equal to $\alpha_{BC}(\boldsymbol{x},t)$ [73]. Such a setup makes the resulting discretized equation well-posed. The addition of the diffusion operator can also be handled in a similar way. Through operator splitting, we split the advection-diffusion equation into a hyperbolic contribution (advection part) and a parabolic contribution (diffusion part). BCs for the hyperbolic part are handled as described before. As diffusion is a stiff operator, we solve the parabolic part implicitly. Dirichlet boundary conditions are readily incorporated in the RHS of the resulting linear system by again considering ghost cells at the boundary with known tracer value $\alpha_{BC}(\boldsymbol{x},t)$ [58].

For the method of composition, we solve the transport PDE to compute $\phi(\boldsymbol{x},t)$, not $\alpha(\boldsymbol{x},t)$. Thus, we need to reformulate the BCs given by equation (22) into BCs that make equation (14) well posed. This is achieved by the using *marker positions*. The marker positions are defined as fictitious position values that do not lie in the domain, and are exclusively utilized to impose boundary values on the flow map computation, *i.e.* equation (14). These marker positions simply keep track of where a particular position entered the domain from, and what its corresponding tracer value was. Thus, the total number of distinct marker positions required is equal to the number of distinct discrete Dirichlet BC values throughout the entire spatio-temporal domain of interest. Let us denote the marker position corresponding to a location $\boldsymbol{x}$ on $\partial\Omega$ at time t by $\boldsymbol{x}_{BC}(t)$. Note that $\boldsymbol{x}_{BC}(t) \notin \Omega$. We also define composition operations over these marker positions through equation (23),

$$\begin{aligned}
&\alpha_0\left(\boldsymbol{x}_{BC}(t)\right) \triangleq \alpha_{BC}(\boldsymbol{x},t)\,,\\
&\phi_{k+1}^{k}\left(\boldsymbol{x}_{BC}(t)\right) \triangleq \boldsymbol{x}_{BC}(t)\,.
\end{aligned} \tag{23}$$

The numerical solution of equation (6) with Dirichlet BCs starts by solving the following system of PDEs,

$$\begin{aligned}
&\frac{\partial \phi_{k+1}^{k}(\boldsymbol{x},t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x},t)\cdot\nabla\phi_{k+1}^{k}(\boldsymbol{x},t) = 0\,,\\
&\phi_{k+1}^{k}(\boldsymbol{x},t_k) = \boldsymbol{x}\,,\\
&\phi_{k+1}^{k}(\boldsymbol{x},t) = \boldsymbol{x}_{BC}(t) \quad \forall\,\boldsymbol{x}\in\partial\Omega\,,
\end{aligned} \tag{24}$$

which provides the set of individual flow maps. The addition of the diffusion operator is handled similarly, as described above, by reformulating the resulting linear system to include these BCs in the RHS by using ghost cells. The composition step equation (16) is then completed, utilizing the composition operation over the marker positions $\boldsymbol{x}_{BC}$, defined by equation (23).

3.5.2. *Neumann boundary condition*

We now discuss the implementation of the Neumann BC, given by equation (25),

$$\frac{\partial\alpha(\boldsymbol{x},t)}{\partial\hat{\mathbf{n}}_{\partial\Omega}} = k_\alpha(\boldsymbol{x},t) \quad \text{for } \boldsymbol{x}\in\partial\Omega \text{ such that } \boldsymbol{v}\cdot\hat{\mathbf{n}}_{\partial\Omega} < 0\,. \tag{25}$$

Although we discuss BCs on the first derivative, results directly extend to higher-order BCs and Robin BCs. Neumann BCs are implemented by suitably modifying the problem such that they can be reduced to special Dirichlet BCs for the individual flow maps. For the advection PDE, we write equation (6) along the boundary as follows:

$$\frac{\partial\alpha(\boldsymbol{x},t)}{\partial t} + \big(\boldsymbol{v}_{\hat{\mathbf{n}}}(\boldsymbol{x},t),\,\boldsymbol{v}_\perp(\boldsymbol{x},t)\big)\cdot\big(\nabla\alpha_{\hat{\mathbf{n}}}(\boldsymbol{x},t),\,\nabla\alpha_\perp(\boldsymbol{x},t)\big) = 0\,, \tag{26}$$

where the subscript $\hat{\mathbf{n}}$ indicates gradients along the normal direction to the boundary and the subscript $c\perp$ gradients along the basis components of the hyperplane orthogonal to $\hat{\mathbf{n}}$. Note that $\hat{\mathbf{n}}$ and the basis of its orthogonal hyperplane can always be uniquely expressed in terms of the grid coordinates. Substituting equation (25) in equation (26), we obtain:

$$\frac{\partial\alpha(\boldsymbol{x},t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x},t)_\perp\cdot\nabla\alpha(\boldsymbol{x},t)_\perp = -\boldsymbol{v}_{\hat{\mathbf{n}}}(\boldsymbol{x},t)k_\alpha(\boldsymbol{x},t)\,. \tag{27}$$

Note that equation (27) is similar to equation (17) with $\kappa = 0$ (see Section 3.4), simply one dimension lower, with Dirichlet BCs over the codimension-one hyperplane locally orthogonal to the boundary. Thus, equation (27) can be solved by the combination of schemes already presented in Section 3.4 and Section 3.5.1.

For the advection-diffusion PDE, the BC is implemented similarly to Section 3.5.1. We first split the PDE into a hyperbolic part (solved explicitly) and a parabolic part (solved implicitly). Neumann BCs can be applied as described above to the hyperbolic part. For the parabolic part, we can accommodate the BCs again through ghost cells, where the ghost cell values are implicitly incorporated by adding local numerical approximations of the derivative to the linear system.

Computationally, another approach to impose Neumann BC is as follows. First, according to the numerical stencil used to discretize $\frac{\partial\alpha(\boldsymbol{x},t)}{\partial\hat{\mathbf{n}}_{\partial\Omega}}$, the boundary values of α can be explicitly computed in terms of the interior values by solving a local linear system resulting from the BC discretization. Once the boundary values to be advected into the domain are computed, they can be imposed as Dirichlet BCs, as described above. Such numerical Neumann BCs is then completed for each of the individual time duration independently.

Imposition of Neumann BCs for the individual flow maps require special attention at the composition/interpolation step. Specifically, the order of the numerical interpolation scheme must be equal to or higher than the order of the derivative involved in the BC. This ensures that the error introduced in the composition of the BC is at least one order higher than the error in the discretization of the BC itself, and the dominant order of accuracy thus remains unchanged.

4. Numerical properties

This section derives numerical properties of the computation of advective transport, *i.e.* equation (6), using the composition method. First, total error bounds are derived. This is followed by a discussion on the existence and derivation of the 'optimal' composition timestep value, for which the net error is minimized.

4.1. Error estimates

We derive error bounds for the computation of the flow map using the method of composition in terms of the spatial discretization (Δx) and temporal discretization (Δt). We assume that the accuracy of the velocity gradient computation is $\mathcal{O}(\Delta x^{\beta})$ and that of all interpolation operations are $\mathcal{O}(\Delta x^{\gamma})$. The time marching scheme for the advection computation is assumed to have a local truncation error of order $\mathcal{O}(\Delta t^{\theta})$. We denote by $|\bullet|$ a well-defined norm to measure error fields.

The proof extends the results of You and Leung [122] who show that the composition of the flow map for a dynamical system is second-order accurate, as their advection scheme, if the interpolation is at least second-order accurate. Inspired by classic initial value problem error analyses [112,10], we generalize this result to the advection of any passive tracer and prove that the advection computation has a temporal accuracy equal to that of the time marching scheme used, and has a spatial accuracy at least equal to the minimum amongst the orders of accuracies of the advection and interpolation operations, regardless of the specific numerical schemes and orders of integration employed.

The exact value of the tracer field at time T is given by $\alpha(\boldsymbol{x},T)=\alpha_0(\phi^0_{N_c}(\boldsymbol{x}))$, whereas the computed solution is given by $\mathcal{I}_\alpha\alpha_0(\tilde{\phi}^0_{N_c}(\boldsymbol{x}))$. The total global error (omitting the $\boldsymbol{x}$ dependency for brevity) is then:

$$E=\left|\mathcal{I}_\alpha\alpha_0(\tilde{\phi}^0_{N_c})-\alpha_0\left(\phi^0_{N_c}\right)\right|=\left|\mathcal{I}_\alpha\alpha_0\left(\mathcal{I}^0_1\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)-\alpha_0\left(\phi^0_1\left(\phi^1_{N_c}\right)\right)\right|. \tag{28}$$

Adding and subtracting $\alpha_0\left(\mathcal{I}^0_1\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)$ and using the triangle inequality, one obtains:

$$E\leq\left|\mathcal{I}_\alpha\alpha_0\left(\mathcal{I}^0_1\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)-\alpha_0\left(\mathcal{I}^0_1\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)\right|+\left|\alpha_0\left(\mathcal{I}^0_1\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)-\alpha_0\left(\phi^0_1\left(\phi^1_{N_c}\right)\right)\right|. \tag{29}$$

Further simplifying the notation, we henceforth omit the temporal indices of interpolation operators (since each interpolation corresponds to the flow map following it). We denote the exact tracer advection over $[0,t_k]$ by $\alpha_k(\bullet)=\alpha_0\left(\phi^0_k(\bullet)\right)$ and the exact backward propagation to $t=0$ of errors made beyond $t=t_k$ by $\mathcal{E}_k=|\alpha_k(\tilde{\bullet})-\alpha_k(\bullet)|$. The equation (29) can then be rewritten,

$$E\leq\left|\mathcal{I}_\alpha\alpha_0\left(\mathcal{I}\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)-\alpha_0\left(\mathcal{I}\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)\right|+\mathcal{E}_0\,, \tag{30}$$

where the first norm is the error due to the interpolation of the tracer field at $t=0$ and the second is the net error $\mathcal{E}_0$ due to the numerical backward flow map integration over $[T,0]$. As the interpolations (including $\mathcal{I}_\alpha$) are of order γ, we can further bound E by:

$$E\leq C_{I_\alpha}\Delta x^{\gamma}+\mathcal{E}_0\,, \tag{31}$$

where C_{I_α} is a constant independent of Δx and Δt. Next, we expand $\mathcal{E}_0$ to link $\mathcal{E}_0$ to $\mathcal{E}_1$, using the fact that, $\forall k=1,\cdots,N_c$, $\alpha_k(\bullet)=\alpha_0\left(\phi^0_k(\bullet)\right)=\alpha_0\left(\phi^0_{k-1}\left(\phi^{k-1}_k(\bullet)\right)\right)=\alpha_{k-1}\left(\phi^{k-1}_k(\bullet)\right)$, and thus, $\mathcal{E}_k=\left|\alpha_{k-1}\left(\phi^{k-1}_k(\tilde{\bullet})\right)-\alpha_{k-1}\left(\phi^{k-1}_k(\bullet)\right)\right|$. We then obtain equation (32),

$$\mathcal{E}_0\leq\left|\alpha_0\left(\mathcal{I}\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)-\alpha_0\left(\mathcal{I}\phi^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)\right|+\left|\alpha_0\left(\mathcal{I}\phi^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)-\alpha_0\left(\phi^0_1\left(\tilde{\phi}^1_{N_c}\right)\right)\right|+\mathcal{E}_1\,, \tag{32}$$

where the first norm is an error due to the numerical flow map integration over $[0,t_1]$, the second is an interpolation error for the flow map over $[0,t_1]$, and the third is the net error $\mathcal{E}_1$ due to the numerical backward flow map integration over $[T,t_1]$. Since α_0 is Lipschitz continuous of constant $\mathcal{L}^0_\alpha$ (see Section 2),

$$\mathcal{E}_0\leq\mathcal{L}^0_\alpha\left|\mathcal{I}\tilde{\phi}^0_1\left(\tilde{\phi}^1_{N_c}\right)-\mathcal{I}\phi^0_1\left(\tilde{\phi}^1_{N_c}\right)\right|+\mathcal{L}^0_\alpha\left|\mathcal{I}\phi^0_1\left(\tilde{\phi}^1_{N_c}\right)-\phi^0_1\left(\tilde{\phi}^1_{N_c}\right)\right|+\mathcal{E}_1\,. \tag{33}$$

Let us assume that the norm of each discrete interpolation operator, $|\mathcal{I}|$, is bounded by C_I independent of Δx. From Section 2, the dominant numerical advection error in the individual flow map computation can be asymptotically written as

$M\left(C_{AX}\Delta x^{\beta}+C_{AT}\Delta t^{\theta}\right)$, where C_{AX} and C_{AT} are non-negative constants independent of Δx and Δt, and of C_I. As a result, we obtain for the first term in equation (33),

$$\left|\mathcal{I}\tilde{\phi}_1^0\left(\tilde{\phi}_{N_c}^1\right)-\mathcal{I}\phi_1^0\left(\tilde{\phi}_{N_c}^1\right)\right|\le C_I\max\left|\tilde{\phi}_1^0-\phi_1^0\right|\le C_I M\left(C_{AX}\Delta x^{\beta}+C_{AT}\Delta t^{\theta}\right). \tag{34}$$

For the second term in equation (33), as the interpolation is of order γ, we obtain:

$$\left|\mathcal{I}\phi_1^0\left(\tilde{\phi}_{N_c}^1\right)-\phi_1^0\left(\tilde{\phi}_{N_c}^1\right)\right|\le\Delta x^{\gamma}\max\left|\partial^{\gamma}\phi_1^0\right|\le C_{I_\phi}\Delta t_c\Delta x^{\gamma}e^{(2\gamma-1)\mathcal{L}_{\boldsymbol{v}}\Delta t_c}, \tag{35}$$

where the latter step is obtained by using *Lemma 5* from [122], and C_{I_ϕ} is a constant independent of Δx and Δt. Finally, substituting equation (34) and equation (35) in equation (33) results in equation (36),

$$\mathcal{E}_0\le\mathcal{L}_\alpha^0\left(C_I M\left(C_{AX}\Delta x^{\beta}+C_{AT}\Delta t^{\theta}\right)+C_{I_\phi}\Delta t_c\Delta x^{\gamma}e^{(2\gamma-1)\mathcal{L}_{\boldsymbol{v}}\Delta t_c}\right)+\mathcal{E}_1=\mathcal{L}_\alpha^0\mathcal{E}+\mathcal{E}_1, \tag{36}$$

where $\mathcal{E}=C_I M\left(C_{AX}\Delta x^{\beta}+C_{AT}\Delta t^{\theta}\right)+C_{I_\phi}\Delta t_c\Delta x^{\gamma}e^{(2\gamma-1)\mathcal{L}_{\boldsymbol{v}}\Delta t_c}$. Finally, to obtain a recursive relation to compute $\mathcal{E}_0$, we need to compute the Lipschitz constant of α_k (denoted by $\mathcal{L}_\alpha^k$) in terms of the known quantities.

Lemma 4 from [122] provides a way to relate these quantities. It states that if the velocity field $\boldsymbol{v}$ has a Lipschitz constant $\mathcal{L}_{\boldsymbol{v}}$, then we have $\left|\phi_N^0(\boldsymbol{x}_1)-\phi_N^0(\boldsymbol{x}_2)\right|\le e^{\mathcal{L}_{\boldsymbol{v}}N\Delta t}\left|\boldsymbol{x}_1-\boldsymbol{x}_2\right|$. (The theorem stated in [122] deals with forward flow maps, however the analogous results hold true for backward flow maps as well). As a result, we have:

$$\left|\alpha_k(\boldsymbol{x}_1)-\alpha_k(\boldsymbol{x}_2)\right|\le\mathcal{L}_\alpha^0\left|\phi_k^0(\boldsymbol{x}_1)-\phi_k^0(\boldsymbol{x}_2)\right|\le\mathcal{L}_\alpha^0e^{\mathcal{L}_{\boldsymbol{v}}k\Delta t_c}\left|\boldsymbol{x}_1-\boldsymbol{x}_2\right|. \tag{37}$$

Equation (37) implies that the Lipschitz constant of α_k is $\mathcal{L}_\alpha^k=\mathcal{L}_\alpha^0e^{\mathcal{L}_{\boldsymbol{v}}k\Delta t_c}$.

Using equation (36), recursively substituting for $\mathcal{E}_k$ and using $\mathcal{L}_\alpha^k$, we obtain equation (38),

$$\mathcal{E}_0\le\mathcal{L}_\alpha^0\mathcal{E}+\mathcal{E}_1\le\mathcal{E}\left(\mathcal{L}_\alpha^0+\ldots+\mathcal{L}_\alpha^{N_c}\right)+\mathcal{E}_{N_c}. \tag{38}$$

However, $\mathcal{E}_{N_c}=\left|\alpha_0\left(\phi_{N_c}^0\right)-\alpha_0\left(\phi_{N_c}^0\right)\right|=0$. Substituting for $\mathcal{L}_\alpha^k$ in equation (38), we obtain equation (39),

$$\mathcal{E}_0\le\mathcal{E}\mathcal{L}_\alpha^0\left(1+\ldots+e^{\mathcal{L}_{\boldsymbol{v}}k\Delta t_c}+\ldots+e^{\mathcal{L}_{\boldsymbol{v}}N_c\Delta t_c}\right)\le\mathcal{E}\mathcal{L}_\alpha^0\frac{e^{\mathcal{L}_{\boldsymbol{v}}T}-1}{e^{\mathcal{L}_{\boldsymbol{v}}\Delta t_c}-1}. \tag{39}$$

Substituting equation (39) in equation (31), we obtain:

$$E\le\mathcal{L}_\alpha^0\frac{e^{\mathcal{L}_{\boldsymbol{v}}T}-1}{e^{\mathcal{L}_{\boldsymbol{v}}\Delta t_c}-1}\left(C_I M\left(C_{AX}\Delta x^{\beta}+C_{AT}\Delta t^{\theta}\right)+C_{I_\phi}\Delta t_c\Delta x^{\gamma}e^{(2\gamma-1)\mathcal{L}_{\boldsymbol{v}}\Delta t_c}\right)+C_{I_\alpha}\Delta x^{\gamma}. \tag{40}$$

Note that, even though the first term on the right hand side has a non-polynomial dependence on Δt, the following relation always holds:

$$1+\ldots+e^{\mathcal{L}_{\boldsymbol{v}}k\Delta t_c}+\ldots+e^{\mathcal{L}_{\boldsymbol{v}}N_c\Delta t_c}\le N_c\max\left(1,e^{\mathcal{L}_{\boldsymbol{v}}T}\right). \tag{41}$$

Substituting this in equation (40), we finally have:

$$E\le\mathcal{L}_\alpha^0\max\left(1,e^{\mathcal{L}_{\boldsymbol{v}}T}\right)\left(C_I\left(N_cMC_{AX}\Delta x^{\beta}+C_{AT}T\Delta t^{\theta-1}\right)+C_{I_\phi}T\Delta x^{\gamma}e^{(2\gamma-1)\mathcal{L}_{\boldsymbol{v}}\Delta t_c}\right)+C_{I_\alpha}\Delta x^{\gamma}. \tag{42}$$

We note that this is the global error summed over all timesteps and not the instantaneous local error [20,112]. At worst, there is an addition of errors. There is indeed no multiplicative compounding of errors. This leads to “super-accurate” properties discussed in Section 4.3.

4.2. Optimal composition timestep

We note that the method of composition involves two basic operations, namely advection and interpolation. We now present an analysis that provides an optimal composition timestep, yielding minimum (or near minimum) total error. As the available velocity field is gridded, we have an already established Δx, and hence we are only concerned with finding an optimal Δt_c, henceforth denoted by Δt_{opt}.

The intuitive argument for the existence of the optimal composition timestep is as follows. First, each individual flow map advection (over M timesteps) introduces errors that depend on both Δx and Δt. Second, when composing this flow map with the flow map for the previous time interval, the interpolation operation introduces an error that depends only on Δx, as this operation is localized in time. Hence for fixed start and end times, as Δt and thus $\Delta t_c=M\Delta t$ decrease, the advection error decreases. However with a decreasing Δt_c more interpolations are performed, and the total interpolation error increases. Conversely, for a large Δt_c, the total advection error is high, but the interpolation error is low (as less number of interpolations are performed). Such an interplay between the advection and interpolation errors suggests the existence of an optimal Δt_c that minimizes the total error.

For the following analysis, we assume that $M = 1$ (*i.e.* $\Delta t_c = \Delta t$, $N_c = N_t$, and $\Delta t = T/N_t$). That is, the number of composition intervals is the same as the number of timesteps. The PDE 6 is thus solved for each timestep independently, and the resulting timestep flow maps composed with each other. The extension of this result is straightforward for $M > 1$.

We could start from the bounds provided by equations 34 and 35. However, we can obtain an estimate of the optimal timestep in closed form from a simpler analysis. We consider the errors in the computation of the tracer field during a single timestep. They are due to the numerical advection and interpolation errors, but for $\Delta t_c = \Delta t$, without compounding of errors within the flow-map computation. We assume that we are given an exact backward flow map $\phi_{N_t}^1$ and consider the remaining timestep $[0, t_1]$. The error then is:

$$E_{\Delta t} = \left| \mathcal{I}_\alpha \alpha_0 \left(\mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) \right) - \alpha_0 \left(\phi_1^0 \left(\phi_{N_t}^1 \right) \right) \right| . \tag{43}$$

Separating the interpolation error from the flow-map error and using triangle inequality leads to the bound:

$$E_{\Delta t} \leq \left| \mathcal{I}_\alpha \alpha_0 \left(\mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) \right) - \alpha_0 \left(\mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) \right) \right| + \left| \alpha_0 \left(\mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) \right) - \alpha_0 \left(\phi_1^0 \left(\phi_{N_t}^1 \right) \right) \right| .$$

As the order of interpolation is of order γ and α_0 is Lipschitz continuous, we obtain,

$$E_{\Delta t} \leq C_{I_\alpha} \Delta x^\gamma + \mathcal{L}_\alpha^0 \left| \mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) - \phi_1^0 \left(\phi_{N_t}^1 \right) \right| .$$

Next, as for equation (34), since the numerical advection is of order β in space and θ in time, the error in the computation of $\mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right)$ can be written asymptotically as Δx and $\Delta t \to 0$,

$$\left| \mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) - \phi_1^0 \left(\phi_{N_t}^1 \right) \right| \approx \left| \mathcal{I}\tilde{\phi}_1^0 \left(\phi_{N_t}^1 \right) - \mathcal{I}\phi_1^0 \left(\phi_{N_t}^1 \right) \right| \approx C_{AX} \Delta x^\beta + C_{AT} \Delta t^\theta .$$

Hence, if we now only retain the lowest-order terms for Δx and Δt in the estimate for E, we obtain,

$$E_{\Delta t} \approx C_X \Delta x^\lambda + \mathcal{L}_\alpha^0 C_{AT} \Delta t^\theta , \tag{44}$$

where $\lambda = \min\{\gamma, \beta\}$ and $C_X = C_{I_\alpha X}$ if $\gamma < \beta$, $C_X = \mathcal{L}_\alpha^0 C_{AX}$ if $\gamma > \beta$, and $C_X = C_{I_\alpha X} + \mathcal{L}_\alpha^0 C_{AX}$ if $\gamma = \beta$. Equation (44) yields the error bound for the computation of a *single timestep*, independent of the errors committed in the other timesteps. With the present method of composition, the order of the largest possible error is only the sum of these errors from the individual timesteps (individual composition steps when $\Delta t_c > \Delta t$), unlike the faster compounding of errors seen in common time integration schemes of ODEs and PDEs (e.g. exponential or polynomial growth). Hence, using equation (44), we write the net error as:

$$E = N_t E_{\Delta t} \approx N_t \left(C_X \Delta x^\lambda + \mathcal{L}_\alpha C_{AT} \Delta t^\theta \right) ,$$
$$\Leftrightarrow \quad E \approx N_t \left(C_X \Delta x^\lambda + \frac{\mathcal{L}_\alpha^0 C_{AT} T^\theta}{N_t^\theta} \right) .$$

Denoting $C_T = \mathcal{L}_\alpha^0 C_{AT} T^\theta$, we then have:

$$E \approx C_X \Delta x^\lambda N_t + \frac{C_T}{N_t^{\theta - 1}} . \tag{45}$$

Our goal is to compute the optimal timestep, *i.e.* optimal N_t, given the time interval. Hence we enforce:

$$\left. \frac{\partial E}{\partial N_t} \right|_{N_t^{opt}} = 0 .$$

We then obtain:

$$C_X \Delta x^\gamma - \frac{(\theta - 1) C_T}{\left(N_t^{opt} \right)^\theta} = 0 ,$$
$$\Leftrightarrow \quad N_t^{opt} = \left(\frac{(\theta - 1) C_T}{C_X \Delta x^\gamma} \right)^{\frac{1}{\theta}} .$$

Therefore, the optimal timestep $\Delta t = T / N_t^{opt}$ is given by equation (46):

$$\Delta t^{opt} = T\left(\frac{C_X \Delta x^{\lambda}}{(\theta - 1)\, C_T}\right)^{\frac{1}{\theta}} . \tag{46}$$

The computation of the optimal timestep requires prior knowledge of C_X and C_T, which is typically well-known for low-order schemes. However, for higher-order schemes, these estimates may not be available in closed form, or the bounds may be coarse. In such cases, the optimal timestep may still be computed numerically. As the existence of the optimal timestep is known, we compute the total error while decreasing the timestep. The value of the timestep at which the total error just starts increasing is the optimal timestep.

Another possible approach is as follows. The value of $\mathcal{L}_{\alpha}^{0}$ can be easily computed as the initial tracer field is known. For example, if the initial tracer field is differentiable, $\mathcal{L}_{\alpha}^{0} = \max_{x\in\Omega} \frac{\partial \alpha_0}{\partial x}$. To estimate the constants C_{IX}, C_{AX}, and C_{AT}, we complete a set of simulations with different Δx and Δt. The y intercepts of the appropriate $\log - \log$ plots yield the asymptotic values of these constants [62]. Using these values along with equation (46), we can compute the optimal timestep value for higher-order schemes, as will be illustrated in Section 5.

4.3. Remarks

We now provide a few remarks and observations.

Super-accuracy

If the numerical PDE advection involves explicit schemes, stability (e.g. CFL) conditions would be commonly imposed. However, each composed advection is here carried out separately (if $M = 1$, for each Δt), and are thus independent and decoupled from the other advections, without multiplication of errors. This is because the numerical advections of past composition timesteps are not directly utilized in the new numerical advection. Within $\Delta t_c = M\,\Delta t$, past numerical advections are only used $M - 1$ times, with $M = 1$ or small. After Δt_c, there is thus no multiplicative compounding of past errors, neither of spatial and temporal truncation errors, nor of round-off errors. For M small, there is thus no possibility for numerical instability, even for classically unstable advection schemes. Without multiplicative error growth, spurious numerical diffusion or dispersion are also not possible. Hence, in all of our simulations (not all shown), we observed neither growth nor propagation of numerical errors and instabilities, even for larger timesteps. Combining all of these properties together, the result is what we refer to in short as "super-accuracy".

To ensure stability in the case of M large and large optimal composition timestep $\Delta t_c = \Delta t_{opt}$, one needs to ensure that Δt is small enough. This is because each individual flow map (of duration Δt_{opt}) is computed by solving equation (14) using the small timestep Δt, ensuring that the advection scheme is stable.

Of course, in all cases, the global truncation error is in general larger if Δt is larger. The super-accuracy however ensures that this global error is not affected by compounding of the local truncation errors.

Optimal grid spacing

If the velocity field is analytically known or if spatial interpolations can be performed, then the optimal grid spacing may also be computed, by using a desired CFL number or desired local accuracy. Considering the former, let us assume a characteristic velocity $\mathcal{U}$ and a desired CFL value of $\mathcal{C}$. Then we have:

$$\mathcal{C} = \mathcal{U}\frac{\Delta t}{\Delta x} . \tag{47}$$

Inserting the optimal timestep of equation (46), we obtain,

$$\mathcal{C} = \mathcal{U} T\left(\frac{C_X \Delta x^{\lambda-\theta}}{(\theta - 1)\, C_T}\right)^{\frac{1}{\theta}} . \tag{48}$$

As a result, for the desired $\mathcal{C}$, the optimal grid spacing is given by equation (49):

$$\Delta x_{opt} = \left(\frac{(\theta - 1)\, C_T}{C_X}\right)^{\frac{1}{\lambda-\theta}} \left(\frac{\mathcal{C}}{\mathcal{U}T}\right)^{\frac{\theta}{\lambda-\theta}} . \tag{49}$$

The corresponding optimal timestep is given by equation (50):

$$\Delta t_{opt} = \frac{\mathcal{C}}{\mathcal{U}} \left(\frac{\mathcal{C}}{\mathcal{U}T}\right)^{\frac{\theta}{\lambda-\theta}} \left(\frac{(\theta - 1)\, C_T}{C_X}\right)^{\frac{1}{\lambda-\theta}} . \tag{50}$$

Parallelization and memory

It is clear from Section 3.3 that the individual flow map computations and the corresponding interpolation operator computations are completely independent and hence can be parallelized. Thus, if one has access to large number of processors (larger than the number of flow map computations *i.e.* N_c), the entire computation would simply require a computational time comparable to the time needed for a single advection timestep computation. However, where we gain in computational time, we lose in storage. The more the parallelization, the more the storage (if all the N_c processes are run in parallel, we need to store N_c fields instead of just one). Often, one has access to only a small number of processors. In such cases, efficient parallelization strategies can be designed to minimize the net computational time and memory requirement. For example, if we have access to four processors, than an efficient strategy can be to compute the flow map $\phi^{0}_{0.25N_c}$ on processor 1, $\phi^{0.25N_c}_{0.5N_c}$ on processor 2, $\phi^{0.5N_c}_{0.75N_c}$ on processor 3, and $\phi^{0.75N_c}_{N_c}$ on processor 4 in parallel. Each of these four computations would still use composition of constituent flow maps (each over a single optimal composition timestep duration), but now without storing the intermediate fields. Once these computations are complete, they can be composed together and with the tracer initial condition to yield the final tracer field. In general, if N_{proc} processors are available, then the computational time is reduced by a factor of N_{proc} while the memory usage is increased by the same factor (as compared to a completely serial solve). Depending on the specifications of the system (FLOP count, I/O speeds etc.) an optimal choice about the degree of parallelization can be made. Note that this discussion is only pertinent to the case of the advection equation, and if one is to solve for a case with diffusion and/or sources, no time parallelization is possible. Of course, space parallelization can still be used.

Storage vs. re-computation of the flow map

For multiple tracers, it may be desirable to store the computed intermediate flow maps (and the associated interpolation operators) in memory if the storage access is efficient, or they can be recomputed whenever required if the machine has a better FLOP count. In general, there is no preferred choice, and the decision may vary from case-to-case. If the flow map computation is completed online, just after the new advection velocity is itself computed, the storage needs are limited. These needs then depend on the ratio of the optimal flow map composition timestep to the velocity advection timestep.

Unavailability of the velocity field at all intermediate times

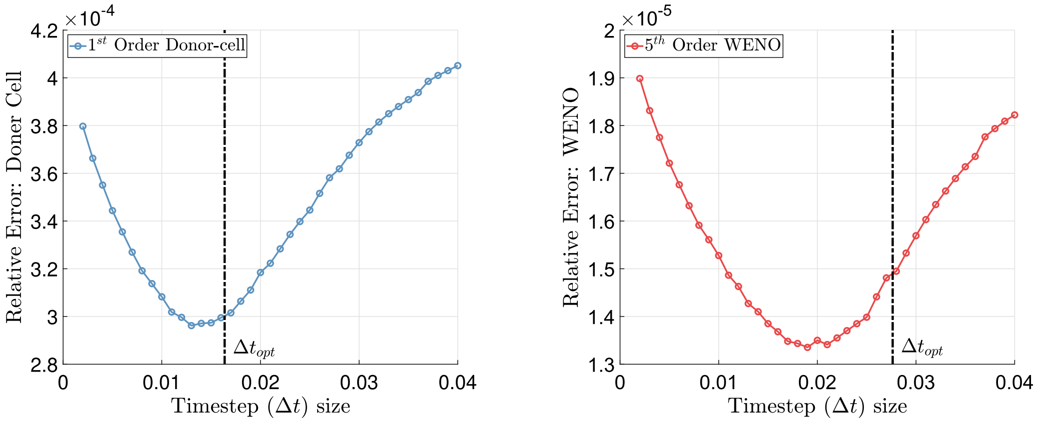
It is common for velocity fields to be available as snapshots only at certain fixed times, and not at all the required times. This occurs when one is trying to study the advection of a tracer field in experimentally observed velocity fields, either in a lab or in nature (ocean, atmosphere, etc.). In such cases, the snapshots of the velocity field need to be interpolated at the required times. This introduces further errors in the numerical advection, however it does not affect the interpolation as this operation simply requires only the computed flow maps. Even though a rigorous treatment as in Section 4.2 is possible, here we present a qualitative description of the effect of velocity interpolations. As the timestep size decreases, the number of advection computations increase. Even though the truncation error in each of the advection computations is small, the error due to the interpolation of the velocity fields required for these computations is larger (along with the interpolation error in the flow maps). However, for larger timesteps, even though the (truncation) error from the individual advection computations is large, the error due to the interpolation of the velocity fields at intermediate times is low (as the velocity field is needed at less intermediate times). This implies that knowing the velocity field only as snapshots in time has the same effect on the optimal composition timestep size as the flow map interpolation. Thus, if this fact is not taken into account while predicting the optimal composition timestep, the computed optimal timestep value would be smaller than the actual optimal timestep (*i.e.* when the velocity interpolation is also taken into account).

Effect of round-off error on the optimal composition timestep

The current analysis to compute the optimal numerical composition timestep only accounts for the truncation errors in the flow map computation, interpolation operation, and the propagated error. However, during the actual implementation, one might also want to consider the effect of round-off errors due to finite precision arithmetic. Specifically, let us assume that round-off errors in a single advection computation (to compute ϕ^{i}_{i+1}) is $\mathcal{R}_A$ and that in the interpolation computation is $\mathcal{R}_I$. The value of $\mathcal{R}_A$ would not be significantly affected by the size of the timestep (*i.e.* even though the truncation error will be large for large timesteps, the magnitude of the round-off errors may not change much). Assuming that N_t is large enough, and as all the individual advection computations are decoupled, we can assume that the total round-off error in the advection computations is of the order of the maximum amongst the individual $\mathcal{R}_A$'s. We can further assume that the round-off error in the interpolation operations ($\mathcal{R}_I$'s) would all be of the same order. Thus, similar to $\mathcal{R}_A$, the net round-off error in interpolation would be of the order of the maximum amongst the individual interpolation round-off errors. Once these are estimated, we can account for these in the error balance equation (equation (45)) that now yields the modified optimal composition timestep. Often times, the round-off errors will be negligibly small as compared to the truncation errors and can be neglected. However, for machines with lower precision, and stiff flow fields, the round-off errors may be significant.

5. Results and applications

We now illustrate the methodology and schemes developed in Sections 3 and 4 in both ideal and realistic examples. We first show the results related to the optimal composition timestep estimates. This is followed by the applications of the composition based advection methodology to idealized and realistic fluid flows, and by comparisons with common numerical advection schemes.



(a) Optimal timestep for first-order donor-cell scheme

(b) Optimal timestep for 5^{th}-order WENO scheme

Fig. 2. Relative error in the final tracer field against the timestep size for the donor-cell – forward-Euler and WENO5 – TVD-RK3 combinations, respectively. For the donor-cell – forward-Euler (Fig. 2a), the analytically evaluated 'optimal timestep' is $\Delta t_{opt} = 0.01637$ (dash-dotted line), which is close to the observed minimum error timestep. For the WENO5 – TVD-RK3 (Fig. 2b), the predicted optimal timestep is at $\Delta t_{opt} = 0.027633$ (dash-dotted line), which is also close to its observed minimum error timestep at $\Delta t = 0.019$.

5.1. Optimal timestep results and advection-diffusion in a reversible analytical swirl flow

In order to compare the resulting numerical errors accurately, we require a test case where the analytical solution is exactly known. We therefore choose the classic test case of an analytical swirl flow in a closed square domain $(x, y) \in [0,1] \times [0,1]$ [22,76], where the non-dimensional velocity is given by equation (51),

$$\boldsymbol{v}(x, y, t) = \left(C(t)\sin^2(\pi x)\sin(2\pi y), -C(t)\sin^2(\pi y)\cdot\sin(2\pi x)\right) . \tag{51}$$

Here $0 \le t < 1$, and $C(t) = 1 \ \forall\ t < 0.5$ and $C(t) = -1 \ \forall\ t \ge 0.5$. The direction of the flow thus reverses at $t = 0.5$. Due to symmetry, the final tracer field is exactly equal to the initial condition. This allows computing the errors incurred in the various methods to a high degree of precision. The non-dimensional initial condition for the tracer field is given by equation (52),

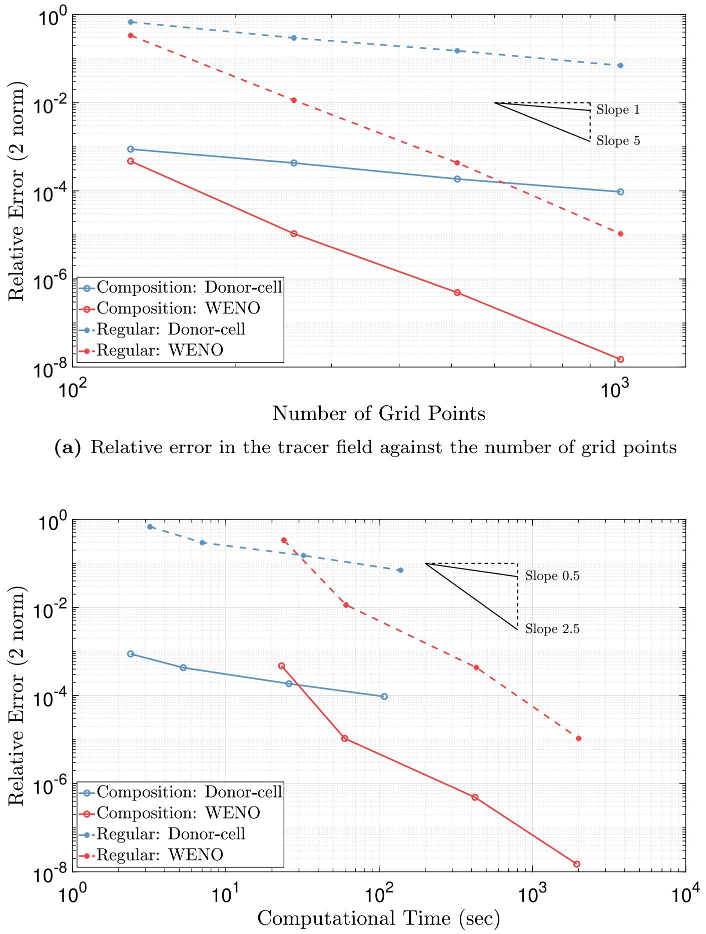
$$\alpha_0(x, y) = \exp\left(-100\left((x-0.25)^2 + (y-0.25)^2\right)\right) . \tag{52}$$

5.1.1. Optimal composition timesteps

To illustrate the results related to the optimal composition timestep, we use the first-order upwind scheme (also called the donor-cell method) for the advection flux, forward Euler time marching, and bilinear interpolation. The composition operation is here performed after every timestep, *i.e.* $M = 1$. The domain is discretized in 256 × 256 control volumes. As explicit error bounds are available for these schemes, the optimal timestep Δt_{opt} is evaluated using equation (46). Fig. 2a shows this optimal timestep value and compares it to the actual total error (numerically computed) for a wide range of timestep sizes. This actual total error is observed to be minimum at a timestep very close to the optimal one.

Fig. 2b shows the actual total error against the size of the timestep for the same test case, but when the 5th-order accurate WENO scheme (WENO5) [91,50] along with third-order accurate TVD Runge Kutta (TVD-RK3) time marching [36] are used for advection and a WENO based interpolation scheme (also 5th-order accuracy) is used for interpolation [100]. It is observed that the actual total error is minimum for a certain timestep value. It can be challenging to compute the optimal timestep value accurately for higher-order schemes, as the analytical error bounds may not be available and/or may be loose. To estimate the optimal timestep in this case, we follow the approach outlined in Section 4. We know the values of γ, β and θ through our choices of numerical schemes, along with the value of $\mathcal{L}^0_\alpha$. We estimate the constants C_{IX}, C_{AX}, and C_{AT} through multiple simulations with different Δx and Δt. Once the values for these constants are obtained, we compute the optimal timestep value for higher-order schemes using equation (46). Fig. 2 confirms that this optimal value is close to the observed minimum of the actual total error. Lastly, as the following results describe, when using the method of composition for advection, higher-order schemes are typically not needed, as lower-order schemes already provide accurate solutions without compounding errors. Hence, even though the analytical optimal timestep may not be available, or might be difficult to estimate for certain higher-order schemes, it is of little impact in practice.

In Fig. 2, we note that the two predicted optimal timesteps are very close to, but slightly larger than, the two observed minima. Overall, a larger analytical optimal timestep will be observed when interpolation errors are overestimated or PDE errors are underestimated. This can occur for several reasons. First, interpolation errors are estimated using the global bound through $\partial^\gamma \phi$. Since this value varies in space, the expected average interpolation error could be low but the maximum could



(a) Relative error in the tracer field against the number of grid points

(b) Relative error in the tracer field against computational time

Fig. 3. Relative error in the tracer field at final time $t = 1$ for the forward-backward advection in a reversible analytical swirl flow. Fig. 3a shows the relative error against the number of grid points, where we observe the expected orders of convergence for the two numerical schemes used. Fig. 3b plots the relative error against the computational time, for the same spatial grid resolutions as Fig. 3a. It can be seen that composition based advection requires an order of magnitude lower computational time for the same accuracy. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

be high (in which case we could overestimate the true averaged error). Second, the PDE error can be underestimated since we only consider the dominant error component and truncate the rest of the error expression. This truncated expression can thus underestimate the net error in the PDE solve. We also note that the magnitude of the truncated terms relative to the actual total error will vary with the order of the schemes, and so the relative accuracy of the optimal timestep will also vary, as is shown in Fig. 2. Finally, in deriving equation (46), the integrated effects of the interpolation and advection errors over one timestep were separated into to a sum of errors.

5.1.2. Advection by the reversible swirl flow

Building on the above validation of the optimal timestep derivations, in all the studies henceforth, we utilize $\Delta t = \Delta t_{opt}$ and select $M = 1$. Fig. 3a shows the relative total error in the final tracer field at $t = 1$ against the grid spacing. The relative error for the method of composition is found to be much smaller than that for the regular PDE based approach even if the order of convergence of the latter is significantly higher. One observes that even a first-order composition based method yields smaller errors than a 5th-order regular PDE based method, for up to about 600 × 600 grid points. This is mainly because the composition based method computes an independent flow map for each timestep, thus the diffusive or dispersive errors in the numerical scheme for advection are not compounded in time. The computational time required for a specific accuracy is orders of magnitude lower for the method of composition. This is clearly observed in Fig. 3b, which plots the relative error in the solution against the computational time (in seconds). The spatial grid resolutions considered in this study are the same as those from the study of relative error against grid spacing, and the computational time increases with decreasing grid spacing (increased resolution). These simulations were performed in MATLAB on a single quad-core Intel i7-4790 CPU clocked at 3.60 GHz. It was ensured that no parallelization was used between the different flow map computations for the composition based advection to maintain a fair comparison between the methods. However, MATLAB's internal vectorization/parallelization for matrix-vector operations was not disabled. As both the approaches used the same internal routines and sub-routines, these internal operations impacts both the methods similarly. These results suggest that even as low as first-order accurate composition based advection should yield good accuracy, and practically there is little need to go to higher-order schemes.

Finally, Fig. 4 shows the initial tracer field followed by the tracer fields at $t = 0.5$ and at $t = 1$. As mentioned above, ideally the tracer field at $t = 1$ should be identical to the initial condition. We find that regular advection with low-order numerical schemes (Fig. 4a) suffers heavily due to numerical diffusion, whereas regular advection with high-order numerical schemes (Fig. 4b) is minimally prone to such errors, and recovers the initial tracer field well. However, even when using low-order numerical schemes, the method of composition is able to almost exactly recover the initial tracer field (Fig. 4c) due to the lack of compounding numerical errors. This further reinforces the result that low-order composition based advection produces results comparable to much higher-order regular advection schemes.

5.1.3. Advection and diffusion in the reversible swirl flow

To showcase the developments of Section 3.4, we now add tracer diffusion, specifically a non-dimensional diffusivity $\kappa = 3.5 \times 10^{-4}$ in equation (17). We create a new benchmark problem that extends the classic swirl flow advection test to a new advection-diffusion test. The underlying velocity field is still given by equation (51), but the tracer is now advected and diffused, with an initial condition given by equation (53),

$$\alpha_0(x, y) = \mathbb{I}\left(0.185 \le \sqrt{(x-0.25)^2 + (y-0.25)^2} \le 0.20\right), \tag{53}$$

where $\mathbb{I}$ is the indicator function, that is equal to unity if the condition in the parentheses is satisfied, and is zero otherwise. That is, the initial tracer is unity inside a ring of outer radius 0.2 and thickness 0.015, centered at $(0.25, 0.25)$. This initial condition provides high tracer gradients along the inner and outer circumferences of the ring for diffusion to work, during both the forward and backward swirl advections. The results thus highlight the effects of spurious numerical diffusion and dispersion.

We solve this benchmark using Lie operator splitting (see Section 3.4) with, for all advection schemes, a second-order accurate implicit diffusion solve. We compare and contrast classic advections (first-order donor-cell and 5th-order WENO) with the composition based advection (first-order donor-cell) using the optimal numerical timestep of equation (46). The grid resolution is 256×256.

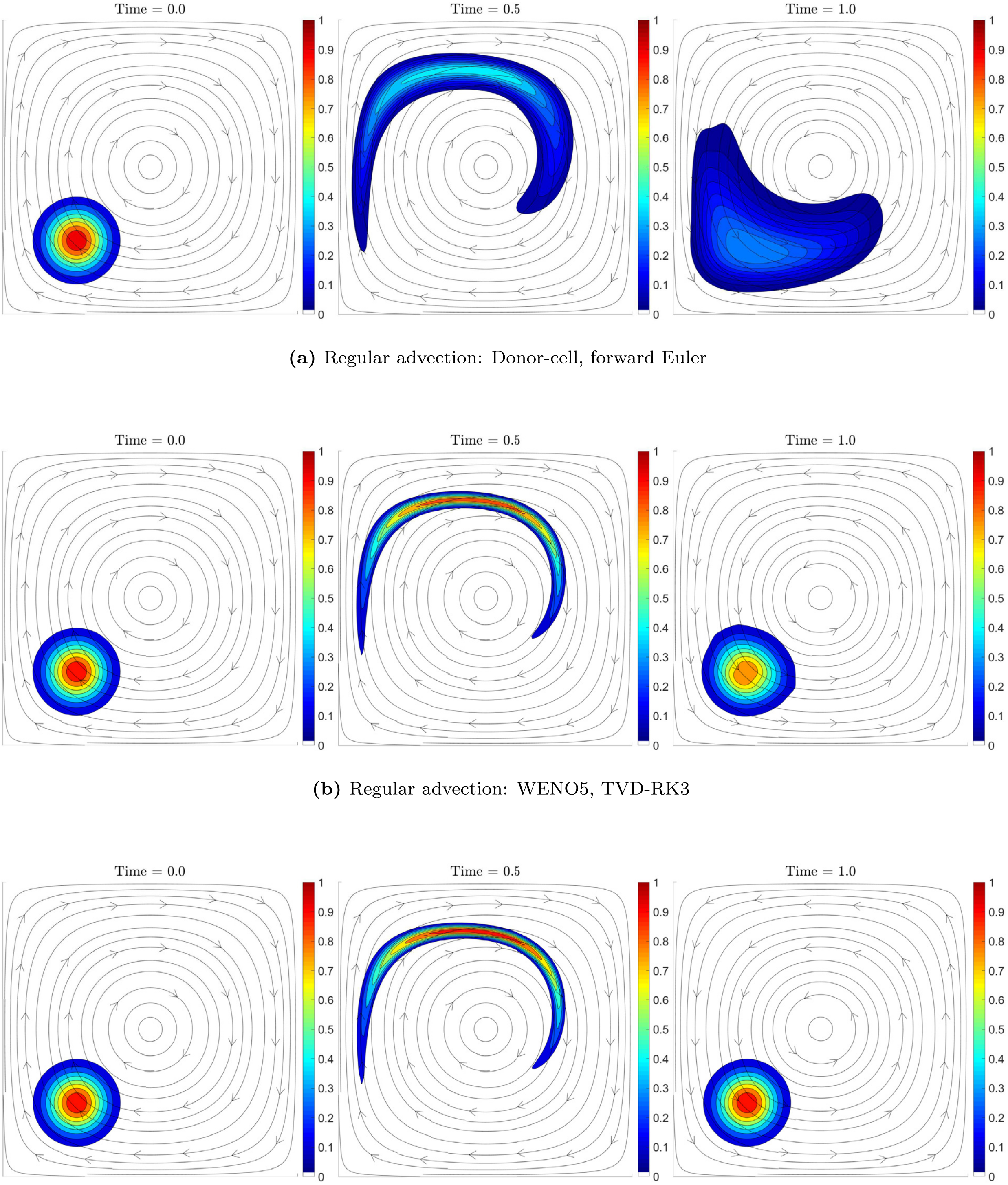
Fig. 5 shows the initial tracer field followed by the tracer fields at $t = 0.5$ and $t = 1$ for the regular advection – diffusion solve (with low-order and high-order advection schemes) and the composition based advection – diffusion solve. Once can clearly see that the numerical diffusion in Fig. 5a causes the inner hole in the ring to close up by $t = 0.5$, whereas the hole is not closed up for the higher-order regular advection and composition based advection (even though this composition scheme is only first-order). Further, after backward advection and diffusion ($t = 1$), contrasting the last panels of Fig. 5b and Fig. 5c, we find that the hole is closed up for the WENO based regular advection, but still persists for composition based advection, indicating that the former has larger spurious numerical diffusion than the latter. This fact is corroborated by the relative error in these fields (where the exact solution is assumed to be that on a 1024×1024 grid, using 5th-order WENO regular advection), as illustrated in Table 1. This showcases the usability and the superiority of composition based advection in solving advection-diffusion PDEs.

5.2. Composition based advection: advection-reaction in an idealized flow exiting a strait

We now examine tracer advection in flows encountered in the coastal oceans and in urban environments, and include a source term and variable boundary conditions. A barotropic jet exits a strait or an estuary into a wider channel and creates unsteady eddies and meanders downstream [75]. In urban settings, wind blowing through narrow constrictions between buildings into an open area also creates similar dynamics. Such flows can be idealized as sudden expansion flows that have been studied extensively [11,23,26].

Fig. 6a shows the schematic of the idealized 2D test case. We consider a 20 m × 1 m channel with an inlet narrow constriction of width 1/3 m and length 4 m. A uniform jet with a velocity 1 m/s enters the inlet constriction at $x = 0$. The other boundary conditions for the velocity are no slip at all the walls and an open (radiation) boundary condition at the $x = 20$ outlet. The velocity field is first developed for 500 s. Recirculation zones and breaks form, either to the north or south of the centerline, depending on the initial uncertain perturbations. The Navier-Stokes equations are solved using a finite volume framework with second-order spatial and temporal schemes and a projection method for pressure-velocity coupling [115]. Fig. 6 shows the advection flow field, at three discrete times $t = 10$, $t = 30$, and $t = 50$ s, after the initial development period of 500 s.

For the non-dimensional tracer field, we assume that a unit value enters the domain at the $x = 0$ inlet along the centerline, with a width of 0.1 m, and that there exists another Gaussian-shaped tracer source of intensity 0.05 s^{-1} centered at $x = 2$ m, $y = 0.25$ m. The boundary conditions are Dirichlet at the inlet, no flux at all the walls, and open (radiation) at the outlet. Further, we assume that there is no tracer diffusion, *i.e.* $\kappa = 0$. The tracer advection-reaction equation (17) is solved

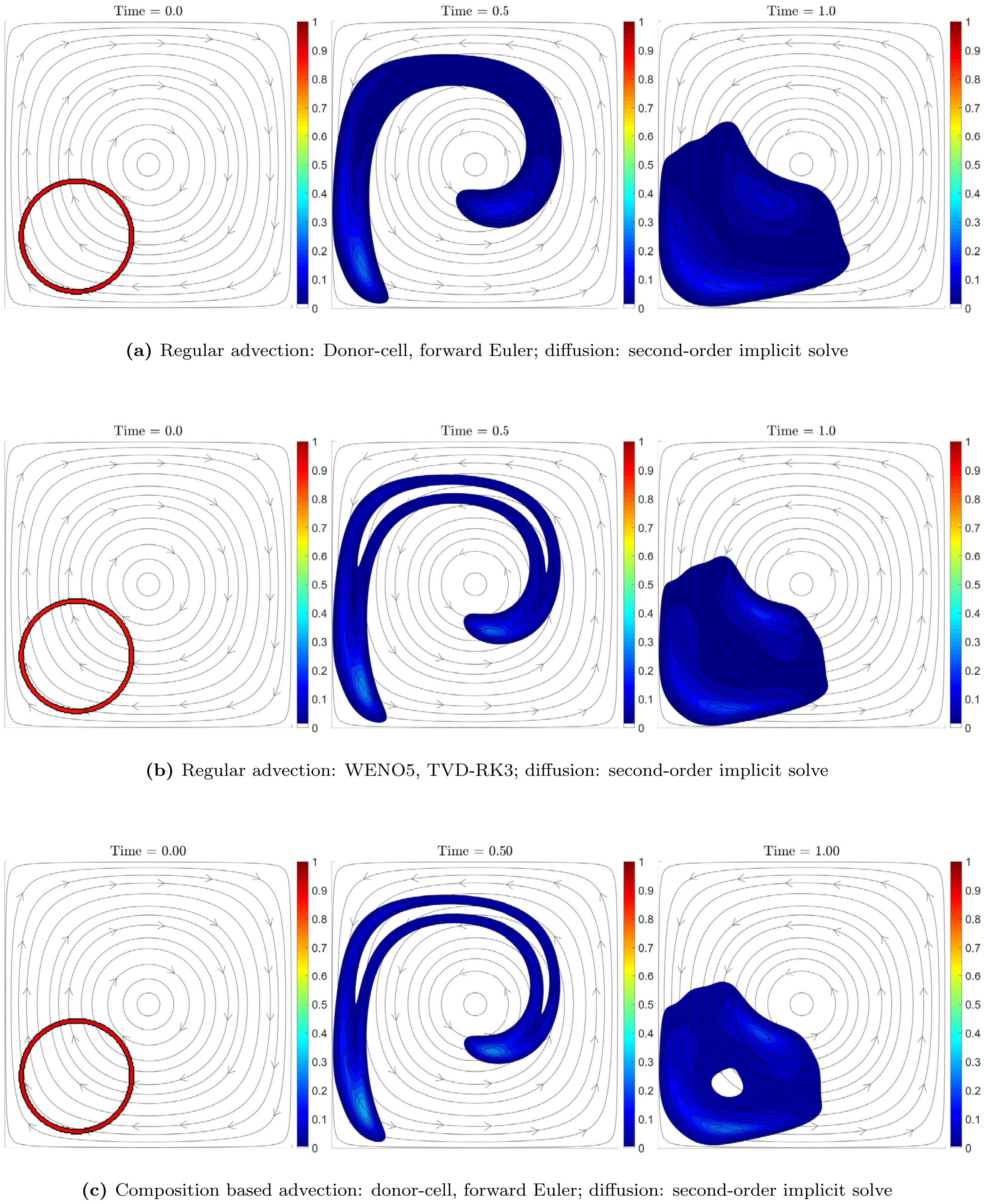


(a) Regular advection: Donor-cell, forward Euler

(b) Regular advection: WENO5, TVD-RK3

(c) Composition based advection: donor-cell, forward Euler

Fig. 4. Forward-backward advection in the analytical swirl flow: the three panels in each sub-figure show the initial tracer condition, the tracer field after the forward advection is complete, and the tracer field after the backward advection is complete, respectively. Regular advection with low-order numerical schemes (Fig. 4a) suffers heavily from spurious numerical diffusion, whereas the method of composition can almost exactly recover the initial tracer field (Fig. 4c), due to the lack of compounding numerical errors.



(a) Regular advection: Donor-cell, forward Euler; diffusion: second-order implicit solve

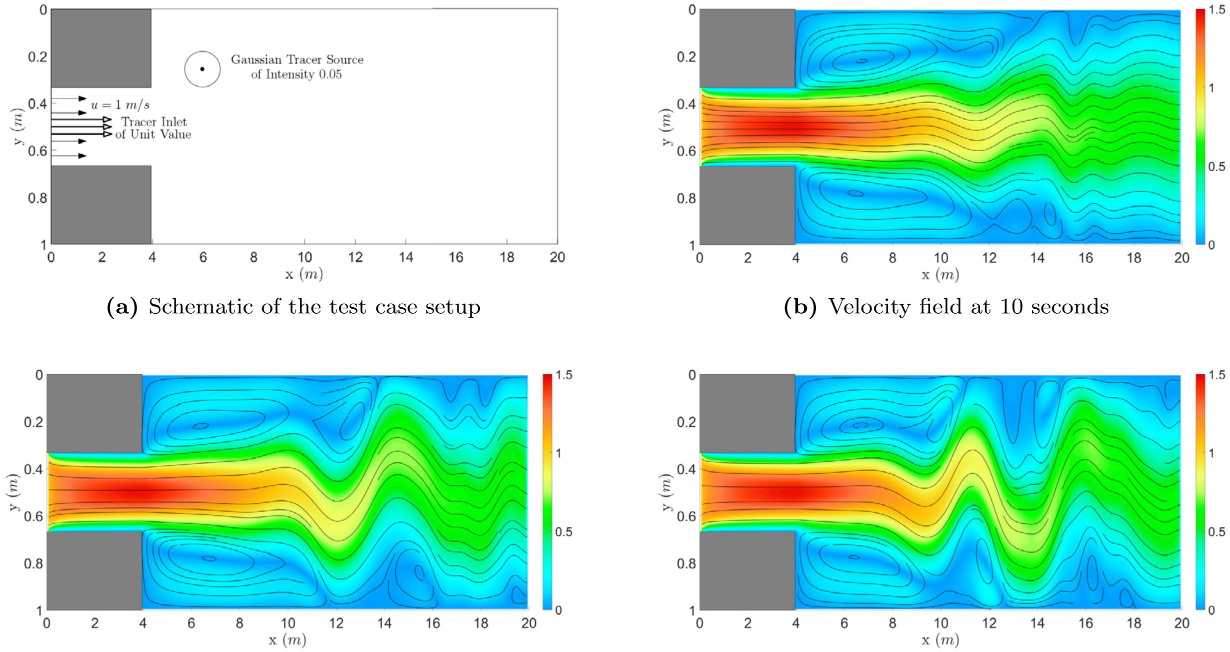
(b) Regular advection: WENO5, TVD-RK3; diffusion: second-order implicit solve

(c) Composition based advection: donor-cell, forward Euler; diffusion: second-order implicit solve

Fig. 5. Forward-backward advection and diffusion in the analytical swirl flow: the three panels in each sub-figure show the initial tracer condition, the tracer field after the forward advection and diffusion part is complete (*i.e.* at $t = 0.5$), and the tracer field after the backward advection and diffusion is complete (*i.e.* at $t = 1$), respectively.

Table 1
Relative errors in solving the advection-diffusion equation for the reversible swirl flow (all methods use Lie operator splitting with a second-order implicit solve for diffusion).

Advection method	Relative error (%)
Regular advection: Donor-cell, forward Euler	7.32%
Regular advection: WENO5, TVD-RK3	1.02%
Composition based advection: Donor-cell, forward Euler	0.76%



(a) Schematic of the test case setup

(b) Velocity field at 10 seconds

(c) Velocity field at 30 seconds

(d) Velocity field at 50 seconds

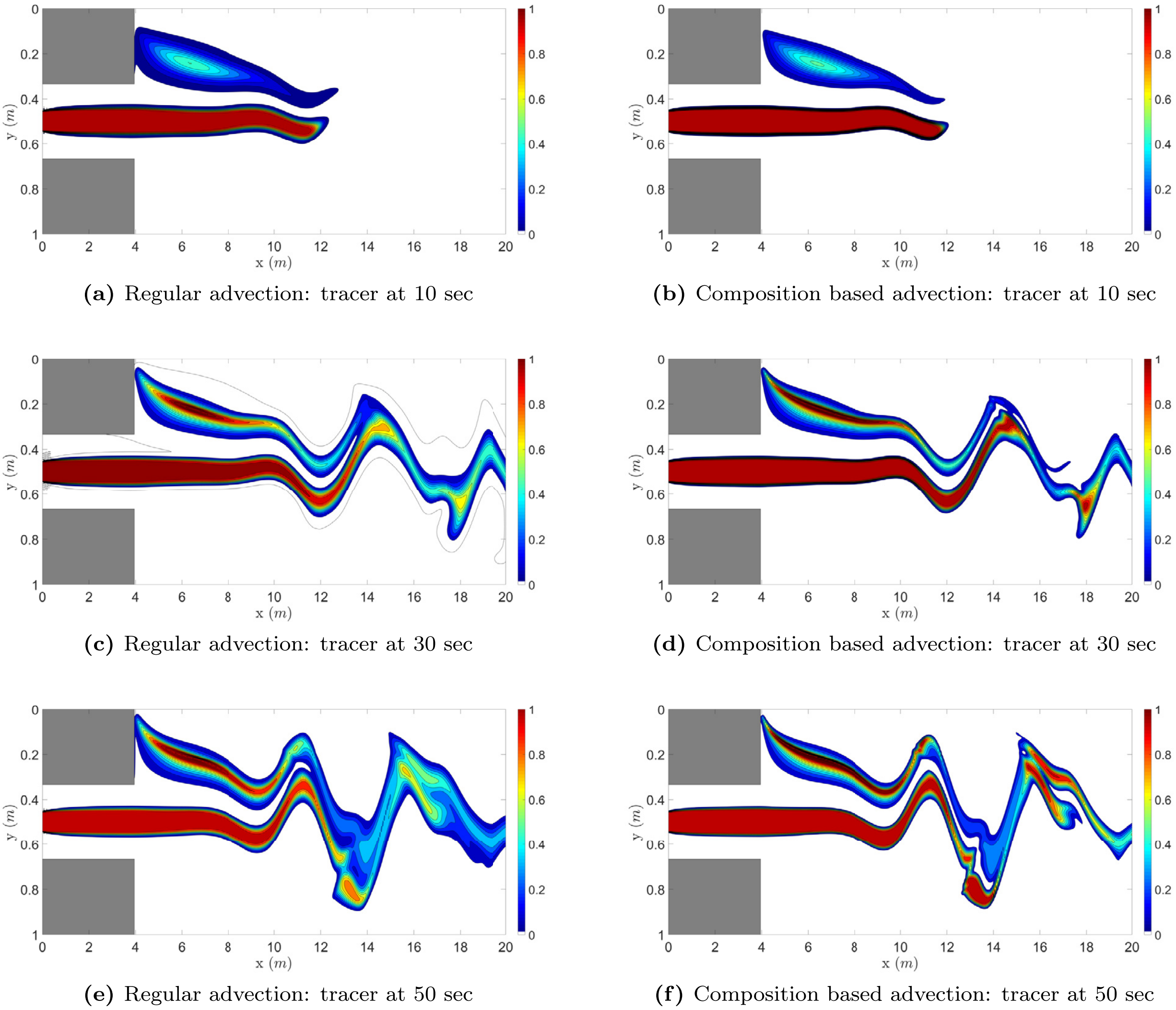
Fig. 6. Flow exiting a strait: Test case setup and the velocity fields. Panel (a) shows the setup and the latter three panels show the velocity field at the specified times after the development period. The velocity field is computed by solving the Navier-Stokes equations in a finite volume framework with second-order spatial and temporal schemes and a projection method for pressure-velocity coupling [115].

for $t = 0, \cdots, 50$ s after the initial velocity development period. The tracer computational setup involves a 160×120 grid, and both the method of composition and the regular advection computations utilize the WENO5 scheme for the spatial gradients and TVD-RK3 for time marching. The timestep value is the optimal timestep for the method of composition. The contributions from the source and advective transport are computed independently of each other using Lie splitting, as outlined in Section 3.4. This example showcases the ability of composition based advection to correctly handle tracer sources and multiple types of tracer boundary conditions, as the inlet has a Dirichlet boundary condition whereas all the walls have a Neumann boundary condition.

Fig. 7 shows the tracer fields at $t = 10$, $t = 30$, and $t = 50$ s after the initial development period. It is clear that the method of composition results in minimal numerical diffusion in the advection process, whereas the regular advection results in a much higher amount of numerical errors. To compute the accuracy of our solution, we compare both the results with a more refined simulation (of grid size 480×360). We obtain that the relative error in the method of composition is 1.27%, whereas that in the regular advection is 4.18%. This showcases the advantages of the method of composition to simulate advection-reaction problems.

5.3. Composition based advection: realistic ocean flows and sediment plumes in the Bismarck sea

For a final example, we simulate the advection of passive sediment plumes in the Bismarck sea. This example is inspired by the need to study the impacts of proposed activities to mine the seabed for rare metal resources [42,13]. Extracting these metal ores from the seabed can create a plume of fine particles. These sediment plumes may be released at various depths within the ocean, and they can prove to be extremely harmful to the local marine ecosystems and its components [118]. Further, they are advected by the dynamic ocean currents and hence can end up far from the original release location. It is thus of utmost importance to quantify and mitigate the impact of such activities on the surroundings. High-accuracy sediment transport forecasts without compounding numerical errors are thus needed.



(a) Regular advection: tracer at 10 sec

(b) Composition based advection: tracer at 10 sec

(c) Regular advection: tracer at 30 sec

(d) Composition based advection: tracer at 30 sec

(e) Regular advection: tracer at 50 sec

(f) Composition based advection: tracer at 50 sec

Fig. 7. Flow exiting a strait: tracer advection-reaction. Panels on the left-hand-side show results using regular tracer advection (WENO5 in space and TVD-RK3 in time), specifically the tracer field at 10, 30 and 50 s (after the flow development period). Panels on the right-hand-side show the tracer fields at the same times but computed using the method of composition (WENO5 in space and TVD-RK3 in time, along with flow map composition). It is clear, especially at 30 and 50 s, that the method of composition is less prone to spurious numerical errors.

To advect the sediment plumes, realistic dynamic ocean fields (currents, temperature, salinity, and ocean free-surface) are simulated by our MIT Multidisciplinary Simulation, Estimation and Assimilation System (MSEAS). MSEAS has been utilized for varied data-assimilative transports studies, including the Prestige oil spill [65], biogeochemical transports [6,15], advection of drifting objects [61,64], and tracer plumes in the Bismark sea [16] and Southern California Bight [54]. The MSEAS numerical schemes include second-order finite-volumes with leapfrog time-stepping and implicit two-way nesting for multi-resolution hydrostatic primitive equation dynamics with a nonlinear free-surface [e.g.,38,68,37], as well as a high-order finite element code on unstructured grids for non-hydrostatic processes [114,116,30]. The MSEAS software has provided realistic data-assimilative simulations and predictions in varied regions of the world's ocean and for varied purposes [e.g.,69,121,39,67,33,97,63,14,52,110,111,66].

The present MSEAS simulation domain covers a 855.36 km × 650.43 km region around the Bismarck sea. The initial ocean conditions were downscaled from the coarse analysis fields of January 15, 2016 of the 1/12° HYCOM (Hybrid Coordinate Ocean Model) [18]. The higher-resolution MSEAS simulations were then run up to January 31, 2016, with full tidal forcing [74]. For the dynamical and numerical parameters, as well as the validation of the MSEAS simulations, we refer to [16].

Using these MSEAS four-dimensional dynamic ocean currents, we simulate the advection of passive plumes from two sites in the Bismarck Sea. One site is the Solwara 1 proposed mining site [13]. The other is to the southeast, inside the St. Georges channel. Both the release sites are at a depth of 1275 *m* below the sea level. For the present example, we neglect the vertical movements of the material, and consider only its advection in the horizontal plane at 1275 *m* depth.

Table 2
Relative errors in predicting sediment advection from two possible deep sea mining sites over 5 days.

Advection method	Relative error (%)
Regular advection: WENO5, TVD-RK3	3.22%
Composition based advection: Donor-cell, forward Euler	1.04%
Method of characteristics: RK4	1.09%

We assume that the material is released in a square box around the two sites (shown in panel (a) of Fig. 8, Fig. 9, and Fig. 10), on January 22 2016, 00:00:00Z. The sediment plumes are then advected by the dynamic ocean currents for 5 days. Without loss of generality for the present numerical study, it is assumed that there is no sediment diffusion. For the sediment advection, the horizontal computational domain is discretized in rotated Cartesian grid of size 1152 × 876, hence each control volume is of size 742 $m \times 742$ m. The timestep is chosen to be the optimal timestep for the method of composition (see Section 4.2), and the resulting mean CFL number is 0.8. The net duration of advection is 5 days. For the method of composition, we use a first-order donor-cell (upwind) scheme for the spatial gradients and forward Euler time marching. For the regular advection (*i.e.* without composition), we use the 5th-order WENO scheme for the spatial gradients and TVD-RK3 time marching. Hence, we compare the first-order composition scheme to a much higher-order regular advection scheme.

For contrast, we also advect the passive tracer plumes using the method of characteristics, *i.e.* by seeding Lagrangian particles in the regions of interest and advecting them under the influence of the unsteady velocity field. This allows the comparison between the method of composition and the method of characteristics/Lagrangian advection. For this purpose, we seed each of the sites of interest with 900 uniformly spaced particles. These particles are advected using 4th-order Runge-Kutta time marching.

Fig. 8, Fig. 9, and Fig. 10 show the advection of the two initial sediment plumes under the effect of the MSEAS dynamic ocean currents over 5 days. We clearly observe that for the method of composition, even though it is only first-order, the numerical diffusion is minimal and the hyperbolic nature of the advection process is well maintained. However, even when using much higher 5th-order schemes, classic advection suffers from large numerical diffusion due to the compounding of errors.

In the method of characteristics, as the particles are advected individually, no diffusion is expected, which is clearly seen in Fig. 10. Comparing this with Fig. 9, as shown in Fig. 11, confirms the lack of numerical diffusion in the method of composition, even though we solve a PDE in an Eulerian setting. The drawback of using the method of characteristics when compared to the method of composition is however the loss of spatial accuracy in certain regions of the domain as computations are carried out in a Lagrangian frame. This is especially clear from the zoomed plots in Fig. 11, where one can observe this loss of spatial accuracy and coverage in certain regions, e.g. near the submesoscale eddies in the bottom left plot and near the mouth of St. Georges channel in the top right plot.

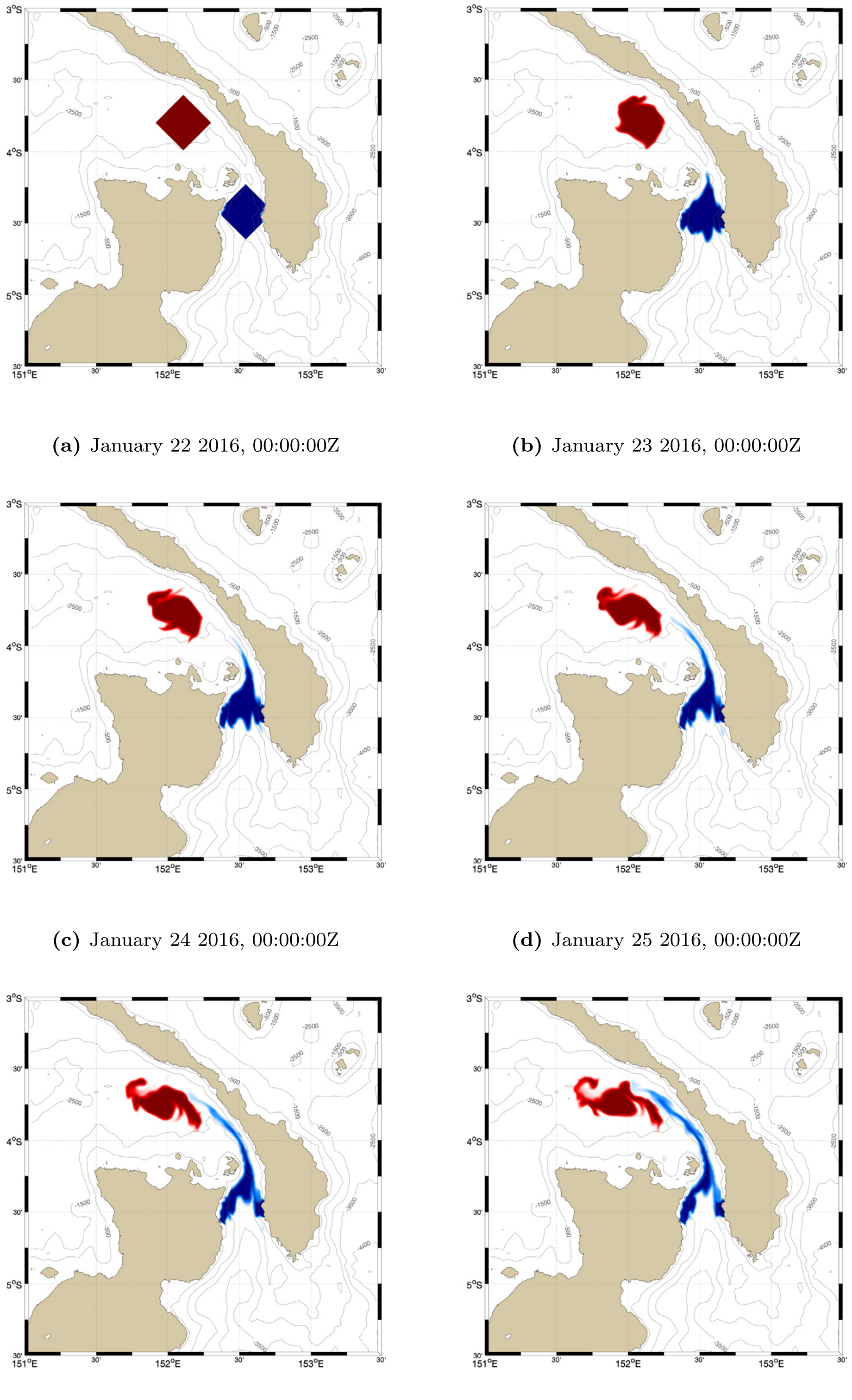
For a more quantitative comparison of the three used methods (regular advection, the method of composition, and the method of characteristics), we compute their relative errors with respect to a field advected using the method of characteristics but using 16 times as many particles. The results are summarized in Table 2, where we see that the method of composition yields more accurate solution than the method of characteristics. If the norm focuses only on the regions where final errors are largest (e.g. where the particle coverage is low), the comparison is much more favorable for the composition.

Further, as the composition based computation uses simple first-order schemes, it was about 30 times faster than the regular high-order advection. The general advection patterns for the two schemes are logically similar (the velocity fields used are identical), but the method of composition maintains sharp gradients and detailed features (effects of submesoscale eddies, fronts, etc.), whereas sharp edges are smeared out by the regular advection. The method of characteristics takes the least amount of time (half that of the method of composition) as its computations were parallelized over the individual particles. However, the Lagrangian particles scheme looses accuracy in regions with low particle density (e.g. repulsive regions), as shown above.

Finally, in operational simulations, turbulent sediment plume diffusion would be modeled, but for accurate sediment forecasts, it would remain crucial for the spurious compounded numerical tracer diffusion to be eliminated. The present composition based advection is an efficient solution.

6. Summary and conclusions

A novel methodology was derived for the numerical computation of advective transport and diffusion-reaction of tracer quantities through flow map composition. The method of composition for advection is “super-accurate”, yielding numerical solutions almost devoid of compounding numerical errors with an accuracy as that of particle based methods but with the advantage of Eulerian resolution in space. It is also readily parallelizable in the temporal direction and can utilize existing PDE toolboxes and computational frameworks. Instead of advecting a tracer field, we advect the spatial positions in the domain and then map them with their corresponding initial (or diffused and forced) tracer value. This advection of positions is carried out *independently* over smaller time intervals, and the advected fields are then composed together. Any



(a) January 22 2016, 00:00:00Z

(b) January 23 2016, 00:00:00Z

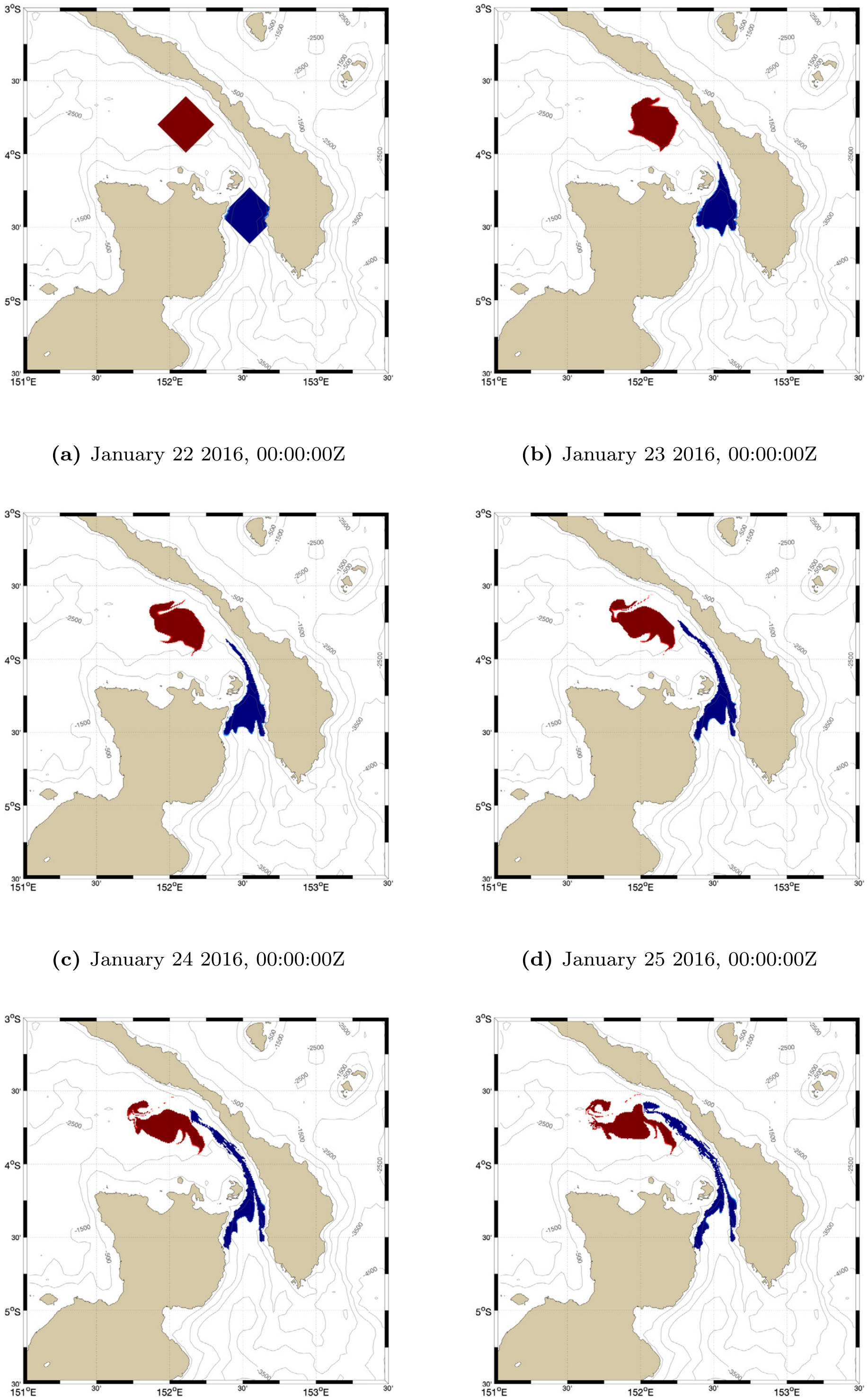
(c) January 24 2016, 00:00:00Z

(d) January 25 2016, 00:00:00Z

(e) January 26 2016, 00:00:00Z

(f) January 27 2016, 00:00:00Z

Fig. 8. Sediment advection over 5 days from two possible deep sea mining sites, using regular advection with high-order WENO5 for the spatial gradients and TVD-RK3 time marching.



(a) January 22 2016, 00:00:00Z

(b) January 23 2016, 00:00:00Z

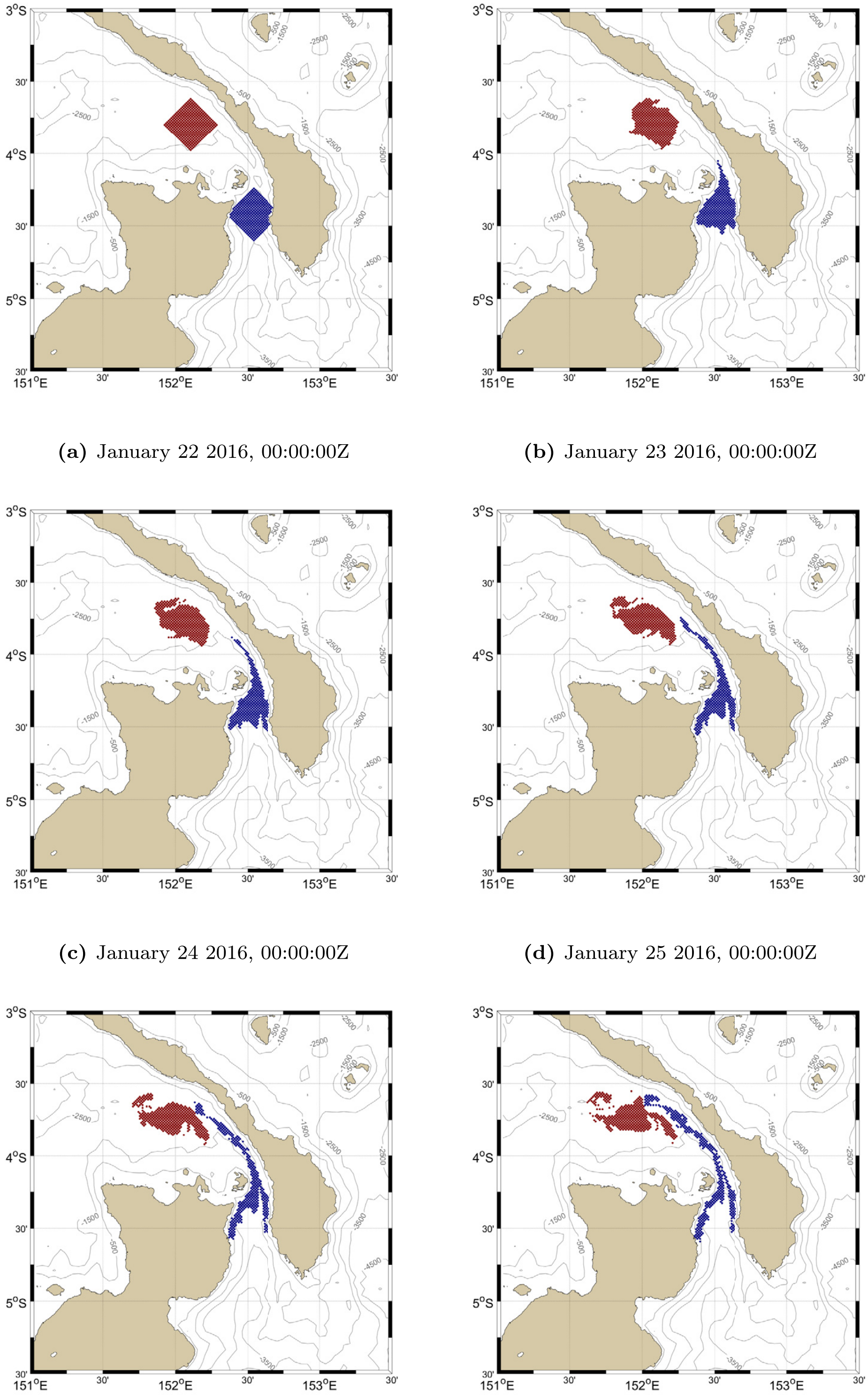
(c) January 24 2016, 00:00:00Z

(d) January 25 2016, 00:00:00Z

(e) January 26 2016, 00:00:00Z

(f) January 27 2016, 00:00:00Z

Fig. 9. As in Fig. 8, but using the method of composition with the first-order donor-cell scheme for the spatial gradients and forward Euler time marching.



(a) January 22 2016, 00:00:00Z

(b) January 23 2016, 00:00:00Z

(c) January 24 2016, 00:00:00Z

(d) January 25 2016, 00:00:00Z

(e) January 26 2016, 00:00:00Z

(f) January 27 2016, 00:00:00Z

Fig. 10. As in Fig. 8, but using Lagrangian trajectory advection using the Runge-Kutta 4 advection scheme.

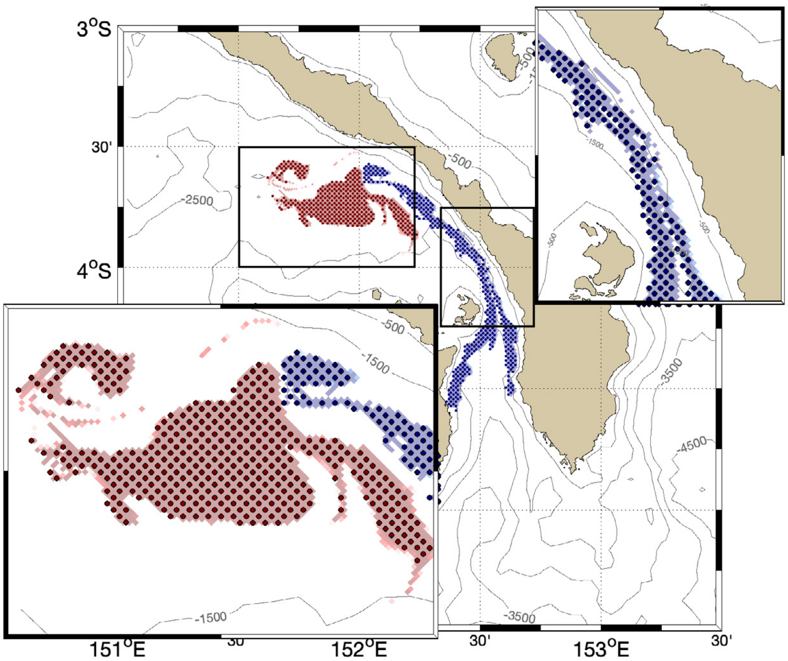


Fig. 11. Comparison of the final sediment fields (on January 27 2016, 00:00:00Z), when advected using the method of composition (Fig. 9f) and trajectory advection (Fig. 10f). The advected fields are extremely close to each other. However, the loss of spatial accuracy for trajectory advection can be seen in the zoomed sections.

established numerical schemes (adaptive, high-order, etc.) may be used in conjunction with adequate interpolations for this advection. There are several benefits to the new methodology: (i) as advection computations over each time intervals are independent, the numerical errors are not compounded in time, which results in much higher accuracy and much lower computational expense for comparable accuracy, (ii) these independent advection computations are parallelizable, (iii) only a single advection computation is required for multiple tracers (when governed by the same underlying velocity field), and (iv) any PDE toolbox can be used to compute the individual position advections, hence minimal programming is required. Theoretically, we derive rigorous expressions for numerical errors and for the *optimal composition timestep*, *i.e.* the timestep value that results in the minimum total numerical error. We also develop schemes for the addition of tracer diffusion, reaction, and source terms, and for the implementation of boundary conditions.

The new methodology and its capabilities are thoroughly illustrated through a wide set of examples, including new benchmark problems for advection-diffusion-reaction schemes. The forward-backward advection and advection-diffusion in an analytical reversible swirl flow verify the behavior of errors for diverse numerical parameters, illustrate the effectiveness of the optimal timestep for schemes of varied orders of accuracy, and show that the new methodology eliminates the compounding of numerical errors and is faster than regular methods. The flow undergoing sudden expansion is then used as an idealization of the exit of river or narrow strait into a wider channel or of a flow through constricted spaces within urban structures. In such a challenging environment with unsteady meanders and eddies downstream, a tracer source and multiple types of tracer boundary conditions are also included. By comparison with highly-resolved tracer fields, we confirm that the novel methodology is more accurate and efficient than the regular advection schemes. Finally, the advection of sediment plumes by ocean currents resulting from potential deep sea mining operations in the Bismarck sea are simulated for two possible mining sites. We find that, even for low-order schemes, advection using the method of composition yields minimal total numerical errors, much smaller than the errors of high-order Eulerian schemes and with a spatial coverage much better than particle based methods.

Portability and the ease of implementation of our PDE based method of flow map composition make it attractive for applications in various domains. The methodology can be extended with minimal effort to the advection of stochastic tracers: even if the uncertainties in the tracer and/or velocity fields are large, an accurate numerical advection is still needed for accurate probabilistic prediction, either in a Monte-Carlo sense or using dynamic reduced-order models [28]. For the latter, it would also decrease the computational cost dependence in the number of stochastic modes from quadratic to linear, which is a substantial gain. Our method of composition is also related to the reinitialization ideas from level set methods [90,1], and it can be used effectively in optimal path planning computations in two and three dimensions [77,110,109,64,66,53,56]. Composition based advection would also be effective in pressure correction methods for Navier-Stokes' equations [29,3], by iteratively composing the intermediate velocity fields. Dynamical systems analyses of Lagrangian transports and coherent structures [40,27,55] and their marine applications [16,54] also depend strongly on the accuracy of the advection computation. Finally, other broad disciplines that rely on accurate simulations of material transport such as atmospheric emissions and plume dispersal [34,35], marine pollution [48,65,59,103], and biogeochemical prediction [82,92] could substantially benefit from our composition based advection.

Acknowledgements

We thank the members of the MSEAS group at MIT for insightful discussions. We also thank the anonymous reviewers and the editor Dr. Piotr Smolarkiewicz for their useful comments. We are grateful to the Office of Naval Research (ONR) for support under grants N00014-14-1-0725 (Bays-DA) and N00014-18-1-2781 (DRI-CALYPSO), the National Oceanographic Partnership Program (NOPP) for research support under grant N00014-15-1-2597 (Seamless Multiscale Forecasting), and to the National Science Foundation for support under grant EAR-1520825 (Hazards SEES – ALPHA), each to the Massachusetts Institute of Technology. We also thank the MIT Environmental Solutions Initiative (MIT-ESI) for Seed Grant research support.

References

[1] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (2) (1995) 269–277.
[2] ANSYS Inc., ANSYS Fluent Theory Guide, 2011.
[3] J. Aoussou, J. Lin, P.F.J. Lermusiaux, Iterated pressure-correction projection methods for the unsteady incompressible Navier–Stokes equations, J. Comput. Phys. 373 (Nov. 2018) 940–974.
[4] A.E. Baptista, Solution of Advection-Dominated Transport by Eulerian-Lagrangian Methods Using the Backwards Method of Characteristics, Ph.D. Thesis, Massachusetts Institute of Technology, 1987.
[5] D.A. Beard, J.B. Bassingthwaighte, Advection and diffusion of substances in biological tissues with complex vascular networks, Ann. Biomed. Eng. 28 (3) (2000) 253–268.
[6] Ş.T. Beşiktepe, P.F.J. Lermusiaux, A.R. Robinson, Coupled physical and biogeochemical data-driven simulations of Massachusetts Bay in late summer: real-time and post-cruise data assimilation, J. Mar. Syst. 40–41 (2003) 171–212.
[7] A. Bennett, Lagrangian Fluid Dynamics, Cambridge University Press, 2006.
[8] S.L. Brunton, C.W. Rowley, Fast computation of finite-time Lyapunov exponent fields for unsteady flows, Chaos 20 (1) (2010) 017503.
[9] M.A. Celia, T.F. Russell, I. Herrera, R.E. Ewing, An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation, Adv. Water Resour. 13 (4) (1990) 187–206.
[10] S.C. Chapra, R.P. Canale, Numerical Methods for Engineers, McGraw Hill Eduction, 2015.
[11] W. Cherdron, F. Durst, J.H. Whitelaw, Asymmetric flows and instabilities in symmetric ducts with sudden expansions, J. Fluid Mech. 84 (1) (1978) 13–31.
[12] B. Cockburn, Discontinuous Galerkin methods for computational fluid dynamics, in: Encyclopedia of Computational Mechanics, second edition, 2018, pp. 1–63.
[13] Coffey Natural Systems Pvt. Ltd., Environmental impact statement, Solwara 1 project, vol. A, http://www.cares.nautilusminerals.com/, 2008.
[14] M.E.G.D. Colin, T.F. Duda, L.A. te Raa, T. van Zon, P.J. Haley Jr., P.F.J. Lermusiaux, W.G. Leslie, C. Mirabito, F.P.A. Lam, A.E. Newhall, Y.-T. Lin, J.F. Lynch, Time-evolving acoustic propagation modeling in a complex ocean environment, in: OCEANS - Bergen, 2013 MTS/IEEE, 2013, pp. 1–9.
[15] G. Cossarini, P.F.J. Lermusiaux, C. Solidoro, Lagoon of Venice ecosystem: seasonal dynamics and environmental guidance with uncertainty analyses and error subspace data assimilation, J. Geophys. Res., Oceans 114 (Jun. 2009), C6.
[16] J. Coulin, P.J. Haley Jr., S. Jana, C.S. Kulkarni, P.F.J. Lermusiaux, T. Peacock, Environmental ocean and plume modeling for deep sea mining in the Bismarck Sea, in: Oceans 2017, Anchorage, AK, Sep. 2017.
[17] R. Courant, E. Isaacson, M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, Commun. Pure Appl. Math. 5 (3) (1952) 243–255.
[18] J.A. Cummings, O.M. Smedstad, Variational data assimilation for the global ocean, in: Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications, Vol. II, Springer, 2013, pp. 303–343.
[19] R.J. DiPerna, P.-L. Lions, Ordinary differential equations, transport theory and Sobolev spaces, Invent. Math. 98 (3) (1989) 511–547.
[20] J.R. Dormand, Numerical Methods for Differential Equations: A Computational Approach, vol. 3, CRC Press, 1996.
[21] J.J. Douglas, T.F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM J. Numer. Anal. 19 (5) (1982) 871–885.
[22] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, vol. 32, Springer Science & Business Media, 2013.
[23] F. Durst, A. Melling, J.H. Whitelaw, Low Reynolds number flow over a plane symmetric sudden expansion, J. Fluid Mech. 64 (1) (1974) 111–128.
[24] Ali A. Eftekhari, FVTool, http://fvt.simulkade.com/, 2017.
[25] R.E. Ewing, T.F. Russell, M.F. Wheeler, Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics, Comput. Methods Appl. Mech. Eng. 47 (1–2) (1984) 73–92.
[26] R.M. Fearn, T. Mullin, K.A. Cliffe, Nonlinear flow phenomena in a symmetric sudden expansion, J. Fluid Mech. 211 (1990) 595–608.
[27] F. Feppon, Riemannian Geometry of Matrix Manifolds for Lagrangian Uncertainty Quantification of Stochastic Fluid Flows, Master's Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, Feb. 2017.
[28] F. Feppon, P.F.J. Lermusiaux, Dynamically orthogonal numerical schemes for efficient stochastic advection and Lagrangian transport, SIAM Rev. 60 (3) (2018) 595–625.
[29] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer Science & Business Media, 2012.
[30] C. Foucart, C. Mirabito, P.J. Haley Jr., P.F.J. Lermusiaux, Distributed implementation and verification of hybridizable discontinuous Galerkin methods for nonhydrostatic ocean processes, in: OCEANS Conference 2018, IEEE, Charleston, SC, Oct. 2018.
[31] L.P. Franca, S.L. Frey, T.J. Hughes, Stabilized finite element methods, I: application to the advective-diffusive model, Comput. Methods Appl. Mech. Eng. 95 (2) (1992) 253–276.
[32] M.J. Gander, 50 years of time parallel time integration, in: Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 69–113.
[33] A. Gangopadhyay, P.F. Lermusiaux, L. Rosenfeld, A.R. Robinson, L. Calado, H.S. Kim, W.G. Leslie, P.J. Haley Jr., The California Current system: a multiscale overview and the development of a feature-oriented regional modeling system (FORMS), in: Special in Honor of Prof. A. R. Robinson, Dyn. Atmos. Ocean. 52 (1–2) (Sep. 2011) 131–169.
[34] E. Gerwing, M. Hort, J. Behrens, B. Langmann, An adaptive semi-Lagrangian advection model for transport of volcanic emissions in the atmosphere, Nat. Hazards Earth Syst. Sci. 18 (5) (2018) 1517–1534.
[35] T. Gerz, T. Dürbeck, P. Konopka, Transport and effective diffusion of aircraft emissions, J. Geophys. Res., Atmos. 103 (D20) (1998) 25905–25913.
[36] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Math. Comput. 67 (221) (1998) 73–85, American Mathematical Society.

[37] P.J. Haley Jr., A. Agarwal, P.F.J. Lermusiaux, Optimizing velocities and transports for complex coastal regions and archipelagos, Ocean Model. 89 (2015) 1–28.
[38] P.J. Haley Jr., P.F.J. Lermusiaux, Multiscale two-way embedding schemes for free-surface primitive equations in the "Multidisciplinary Simulation, Estimation and Assimilation System", Ocean Dyn. 60 (6) (Dec. 2010) 1497–1537.
[39] P.J. Haley Jr., P.F.J. Lermusiaux, A.R. Robinson, W.G. Leslie, O. Logoutov, G. Cossarini, X.S. Liang, P. Moreno, S.R. Ramp, J.D. Doyle, J. Bellingham, F. Chavez, S. Johnston, Forecasting and reanalysis in the Monterey Bay/California Current region for the Autonomous Ocean Sampling Network, II: experiment, Deep-Sea Res., Part 2, Top. Stud. Oceanogr. 56 (3–5) (Feb. 2009) 127–148.
[40] G. Haller, Lagrangian coherent structures, Annu. Rev. Fluid Mech. 47 (2015) 137–162.
[41] J. Hervouet, Application of the method of characteristics in their weak formulation to solving two-dimensional advection equations on mesh grids, Recent Adv. Numer. Methods Fluids 5 (1986) 149–185.
[42] P. Hoagland, S. Beaulieu, M.A. Tivey, R.G. Eggert, C. German, L. Glowka, J. Lin, Deep-sea mining of seafloor massive sulfides, Mar. Policy 34 (3) (2010) 728–732.
[43] P. Holmes, J.L. Lumley, G. Berkooz, C.W. Rowley, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, 2012.
[44] P. Houston, C. Schwab, E. Süli, Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems, SIAM J. Numer. Anal. 39 (6) (2002) 2133–2163.
[45] T.J. Hughes, A simple scheme for developing 'upwind' finite elements, Int. J. Numer. Methods Eng. 12 (9) (1978) 1359–1365.
[46] T.J. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics, VIII: the Galerkin/least-squares method for advective-diffusive equations, Comput. Methods Appl. Mech. Eng. 73 (2) (1989) 173–189.
[47] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, vol. 33, Springer Science & Business Media, 2013.
[48] T. Ilyina, T. Pohlmann, G. Lammel, J. Sündermann, A fate and transport ocean model for persistent organic pollutants and its application to the North Sea, J. Mar. Syst. 63 (1–2) (2006) 1–19.
[49] H. Jia, K. Li, A third accurate operator splitting method, Math. Comput. Model. 53 (1–2) (2011) 387–396.
[50] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.
[51] G. Karniadakis, S. Sherwin, Spectral/*hp* Element Methods for Computational Fluid Dynamics, Oxford University Press, 2013.
[52] S.M. Kelly, P.F.J. Lermusiaux, Internal-tide interactions with Gulf Stream and Middle Atlantic Bight shelfbreak front, J. Geophys. Res., Oceans 121 (2016) 6271–6294.
[53] C.S. Kulkarni, Three-Dimensional Time-Optimal Path Planning in Dynamic and Realistic Environments, Master's Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, Jun. 2017.
[54] C.S. Kulkarni, P.J. Haley Jr., P.F.J. Lermusiaux, A. Dutt, A. Gupta, C. Mirabito, D.N. Subramani, S. Jana, W.H. Ali, T. Peacock, C.M. Royo, A. Rzeznik, R. Supekar, Real-time sediment plume modeling in the Southern California Bight, in: OCEANS Conference 2018, IEEE, Charleston, SC, Oct. 2018.
[55] C.S. Kulkarni, P.F.J. Lermusiaux, Persistent rigid sets in realistic fluid flows using flow map composition, Ocean Model. (2019), in preparation.
[56] C.S. Kulkarni, P.F.J. Lermusiaux, Three-dimensional time-optimal path planning in the ocean, Ocean Model. (2019), sub-judice.
[57] D. Lanser, J.G. Verwer, Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling, J. Comput. Appl. Math. 111 (1–2) (1999) 201–216.
[58] L. Lapidus, G.F. Pinder, Numerical Solution of Partial Differential Equations in Science and Engineering, John Wiley & Sons, 2011.
[59] L. Lebreton, B. Slat, F. Ferrari, B. Sainte-Rose, J. Aitken, R. Marthouse, S. Hajbane, S. Cunsolo, A. Schwarz, A. Levivier, et al., Evidence that the great Pacific garbage patch is rapidly accumulating plastic, Sci. Rep. 8 (1) (2018) 4666.
[60] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1) (1992) 16–42.
[61] P.F.J. Lermusiaux, Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts Bay, J. Mar. Syst. 29 (1) (2001) 385–422.
[62] P.F.J. Lermusiaux, Numerical fluid mechanics. MIT OpenCourseWare, https://ocw.mit.edu/courses/mechanical-engineering/2-29-numerical-fluid-mechanics-spring-2015/lecture-notes-and-references/, May 2015.
[63] P.F.J. Lermusiaux, P.J. Haley, W.G. Leslie, A. Agarwal, O. Logutov, L.J. Burton, Multiscale physical and biological dynamics in the Philippine Archipelago: predictions and processes, Oceanography 24 (1) (2011) 70–89.
[64] P.F.J. Lermusiaux, P.J. Haley Jr., S. Jana, A. Gupta, C.S. Kulkarni, C. Mirabito, W.H. Ali, D.N. Subramani, A. Dutt, J. Lin, A. Shcherbina, C. Lee, A. Gangopadhyay, Optimal planning and sampling predictions for autonomous and Lagrangian platforms and sensors in the northern Arabian Sea, in: Special Issue on Autonomous and Lagrangian Platforms and Sensors (ALPS), Oceanography 30 (2) (2017) 172–185.
[65] P.F.J. Lermusiaux, P.J. Haley Jr, N.K. Yilmaz, Environmental prediction, path planning and adaptive sampling: sensing and modeling for efficient ocean monitoring, management and pollution control, Sea Technol. 48 (9) (2007) 35–38.
[66] P.F.J. Lermusiaux, D.N. Subramani, J. Lin, C.S. Kulkarni, A. Gupta, A. Dutt, T. Lolla, P.J. Haley Jr., W.H. Ali, C. Mirabito, S. Jana, A future for intelligent autonomous ocean observing systems, J. Mar. Res. 75 (6) (2017) 765–813, The Sea, Vol. 17, The Science of Ocean Prediction, Part 2.
[67] P.F.J. Lermusiaux, J. Xu, C.-F. Chen, S. Jan, L. Chiu, Y.-J. Yang, Coupled ocean–acoustic prediction of transmission loss in a continental shelfbreak region: Predictive skill, uncertainty quantification, and dynamical sensitivities, IEEE J. Ocean. Eng. 35 (4) (Oct. 2010) 895–916.
[68] W.G. Leslie, P.J. Haley Jr., P.F.J. Lermusiaux, M.P. Ueckermann, O. Logutov, J. Xu, MSEAS Manual, MSEAS Report 06, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2010, http://mseas.mit.edu/?p=2237.
[69] W.G. Leslie, A.R. Robinson, P.J. Haley Jr, O. Logutov, P.A. Moreno, P.F.J. Lermusiaux, E. Coelho, Verification and training of real-time forecasting of multi-scale ocean dynamics for maritime rapid environmental assessment, J. Mar. Syst. 69 (1) (2008) 3–16.
[70] S. Leung, An Eulerian approach for computing the finite time Lyapunov exponent, J. Comput. Phys. 230 (9) (2011) 3500–3524.
[71] S. Leung, The backward phase flow method for the Eulerian finite time Lyapunov exponent computations, Chaos 23 (4) (2013) 043132.
[72] R.J. LeVeque, Nonlinear conservation laws and finite volume methods, in: Computational Methods for Astrophysical Fluid Flow, Springer, 1998, pp. 1–159.
[73] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, vol. 31, Cambridge University Press, 2002.
[74] O.G. Logutov, P.F.J. Lermusiaux, Inverse barotropic tidal estimation for regional ocean applications, Ocean Model. 25 (1–2) (2008) 17–34, http://www.sciencedirect.com/science/article/pii/S1463500308000851.
[75] T. Lolla, P.J. Haley Jr., P.F.J. Lermusiaux, Path planning in multiscale ocean flows: coordination and dynamic obstacles, Ocean Model. 94 (2015) 46–66.
[76] T. Lolla, P.F.J. Lermusiaux, A Gaussian mixture model smoother for continuous nonlinear stochastic dynamical systems: application, Mon. Weather Rev. 145 (Jul. 2017) 2763–2790.
[77] T. Lolla, P.F.J. Lermusiaux, M.P. Ueckermann, P.J. Haley Jr., Time-optimal path planning in dynamic flows using level set equations: theory and schemes, Ocean Dyn. 64 (10) (2014) 1373–1397.
[78] Z.-H. Ma, W.C. Chew, L. Jiang, A novel fast solver for Poisson equation with the Neumann boundary condition, preprint, arXiv:1207.4260, 2012.
[79] S. MacNamara, G. Strang, Operator splitting, in: Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, 2016, pp. 95–114.
[80] A.V. Malevsky, S.J. Thomas, Parallel algorithms for semi-Lagrangian advection, Int. J. Numer. Methods Fluids 25 (4) (1997) 455–473.
[81] G. Manzini, A. Russo, A finite volume method for advection–diffusion problems in convection-dominated regimes, Comput. Methods Appl. Mech. Eng. 197 (13–16) (2008) 1242–1261.

[82] D.J. McGillicuddy, Mechanisms of physical-biological-biogeochemical interaction at the oceanic mesoscale, 2016.
[83] O. Mercier, J.-C. Nave, The characteristic mapping method for the linear advection of arbitrary sets, preprint, arXiv:1309.2731, 2013.
[84] A. Mohebbi, M. Dehghan, High-order compact solution of the one-dimensional heat and advection–diffusion equations, Appl. Math. Model. 34 (10) (2010) 3071–3084.
[85] 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.
[86] S.P. Neuman, S. Sorek, Eulerian-Lagrangian methods for advection-dispersion, in: Finite Elements in Water Resources, Springer, 1982, pp. 849–876.
[87] Oliveira, A.P. d, A comparison of Eulerian-Lagrangian methods for the solution of the transport equation, 1994.
[88] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, J. Comput. Phys. 181 (2) (2002) 729–752.
[89] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, vol. 153, Springer Science & Business Media, 2006.
[90] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.
[91] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Numer. Anal. 28 (4) (1991) 907–922.
[92] C. Pasquero, A. Bracco, A. Provenzale, Coherent vortices, Lagrangian particles and the marine ecosystem, Shallow Flows (2004) 399–412.
[93] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere, Washington, DC, 1980, pp. 115–120.
[94] G.F. Pinder, H.H. Cooper, A numerical technique for calculating the transient position of the saltwater front, Water Resour. Res. 6 (3) (1970) 875–882.
[95] S.V. Prants, Chaotic Lagrangian transport and mixing in the ocean, Eur. Phys. J. Spec. Top. 223 (13) (2014) 2723–2743.
[96] M.J. Prather, Numerical advection by conservation of second-order moments, J. Geophys. Res., Atmos. 91 (D6) (1986) 6671–6681.
[97] S.R. Ramp, P.F.J. Lermusiaux, I. Shulman, Y. Chao, R.E. Wolf, F.L. Bahr, Oceanographic and atmospheric conditions on the continental shelf north of the Monterey Bay during August 2006, in: Special Issue of Dynamics of Atmospheres and Oceans in honor of Prof. A. R. Robinson, Dyn. Atmos. Ocean. 52 (1–2) (Sep. 2011) 192–223.
[98] T.F. Russell, Time stepping along characteristics with incomplete iteration for a Galerkin approximation of miscible displacement in porous media, SIAM J. Numer. Anal. 22 (5) (1985) 970–1013.
[99] R.M. Samelson, S. Wiggins, Lagrangian Transport in Geophysical Jets and Waves: The Dynamical Systems Approach, vol. 31, Springer Science & Business Media, 2006.
[100] C. Santaera, R.M. Pidatella, F. Stanco, ENO/WENO Interpolation Methods for the Zooming of Digital Images, Communications to SIMAI Congress., vol. 1, 2007.
[101] A. Schelin, G. Károlyi, A. De Moura, N. Booth, C. Grebogi, Chaotic advection in blood flow, Phys. Rev. E 80 (1) (2009) 016213.
[102] B. Seibold, J.-C. Nave, R.R. Rosales, Jet schemes for advection problems, preprint, arXiv:1101.5374, 2011.
[103] P. Sherman, E. Van Sebille, Modeling marine surface microplastic transport to assess optimal removal locations, Environ. Res. Lett. 11 (1) (2016) 014006.
[104] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer, 1998, pp. 325–432.
[105] H. Song, M.R. Bringer, J.D. Tice, C.J. Gerdts, R.F. Ismagilov, Experimental test of scaling of mixing by chaotic advection in droplets moving through microfluidic channels, Appl. Phys. Lett. 83 (22) (2003) 4664–4666.
[106] S. Srivastava, Operator Splitting Methods for the Advection-Diffusion-Reaction Equation, Centre for Analysis, Scientific Computing and Applications, TU, Eindhoven, 2008.
[107] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (3) (1968) 506–517.
[108] M.A. Stremler, F.R. Haselton, H. Aref, Designing for chaos: applications of chaotic advection at the microscale, Philos. Trans. R. Soc., Math. Phys. Eng. Sci. 362 (1818) (2004) 1019–1036.
[109] D.N. Subramani, P.J. Haley Jr., P.F.J. Lermusiaux, Energy-optimal path planning in the coastal ocean, J. Geophys. Res., Oceans 122 (2017) 3981–4003.
[110] D.N. Subramani, P.F.J. Lermusiaux, Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization, Ocean Model. 100 (2016) 57–77.
[111] D.N. Subramani, P.F.J. Lermusiaux, P.J. Haley Jr., C. Mirabito, S. Jana, C.S. Kulkarni, A. Girard, D. Wickman, J. Edwards, J. Smith, Time-optimal path planning: real-time sea exercises, in: Oceans '17 MTS/IEEE Conference, Aberdeen, 2017.
[112] E. Süli, D.F. Mayers, An Introduction to Numerical Analysis, Cambridge University Press, 2003.
[113] V.A. Titarev, E.F. Toro, ADER: arbitrary high order Godunov approach, J. Sci. Comput. 17 (1–4) (2002) 609–618.
[114] M.P. Ueckermann, P.F.J. Lermusiaux, High order schemes for 2D unsteady biogeochemical ocean models, Ocean Dyn. 60 (6) (Dec. 2010) 1415–1445.
[115] M.P. Ueckermann, P.F.J. Lermusiaux, 2.29 Finite Volume MATLAB Framework Documentation, MSEAS Report 14, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2012, http://mseas.mit.edu/?p=2567.
[116] M.P. Ueckermann, P.F.J. Lermusiaux, Hybridizable discontinuous Galerkin projection methods for Navier–Stokes and Boussinesq equations, J. Comput. Phys. 306 (2016) 390–421.
[117] A.J. Valocchi, M. Malmstead, Accuracy of operator splitting for advection-dispersion-reaction problems, Water Resour. Res. 28 (5) (1992) 1471–1476.
[118] A. Vanreusel, A. Hiliario, P. Ribeiro, L. Menot, P. Arbizu Martinez, Managing impacts of deep sea resource exploitation, 2016.
[119] H. Wang, H.K. Dahle, R.E. Ewing, M.S. Espedal, R.C. Sharpley, S. Man, An ELLAM scheme for advection-diffusion equations in two dimensions, SIAM J. Sci. Comput. 20 (6) (1999) 2160–2194.
[120] M.F. Wheeler, C.N. Dawson, An operator-splitting method for advection-diffusion-reaction problems, in: The Mathematics of Finite Elements and Applications VI, 1987, pp. 463–482.
[121] J. Xu, P.F.J. Lermusiaux, P.J. Haley Jr., W.G. Leslie, O.G. Logutov, Spatial and temporal variations in acoustic propagation during the PLUSNet-07 exercise in Dabob Bay, in: Proceedings of Meetings on Acoustics (POMA). Vol. 4, Acoustical Society of America 155th Meeting, 2008, p. 11.
[122] G. You, S. Leung, Eulerian based interpolation schemes for flow map construction and line integral computation with applications to Lagrangian coherent structures extraction, J. Sci. Comput. (2017) 1–27.
[123] G. You, T. Wong, S. Leung, Eulerian methods for visualizing continuous dynamical systems using Lyapunov exponents, SIAM J. Sci. Comput. 39 (2) (2017) A415–A437.