

LAGRANGIAN FLUX CALCULATION THROUGH A FIXED
PLANAR CURVE FOR SCALAR CONSERVATION LAWS*QINGHAI ZHANG[†] AND LINGYUN DING[‡]

Abstract. For a scalar function conserved by an unsteady flow, its flux through a simple curve is usually expressed as an Eulerian flux, a double integral both in time and in space. We show that this Eulerian flux equals a Lagrangian flux, a line integral in space with no time dependence. We further exploit this flux identity to propose a new algorithm for Lagrangian flux calculation (LFC), in which the Eulerian flux through the static curve is estimated only from the given velocity field and the initial distribution of the scalar function. In particular, the time interval of LFC can be arbitrarily long, yet the scalar conservation law is never numerically solved. As such, LFC complements traditional finite volume methods both theoretically and computationally. Compared to previous LFC algorithms, the new LFC algorithm is simple and well-conditioned with second-, fourth-, and sixth-order convergence. Results of numerical tests demonstrate the flexibility, accuracy, and efficiency of the proposed LFC algorithm.

Key words. Lagrangian flux calculation, scalar conservation laws, finite volume methods, donating regions, Reynolds transport theorem, Hopf theorem

AMS subject classifications. 35L65, 65M08, 70H99

DOI. 10.1137/18M1210885

1. Introduction. Given a fixed simple planar curve \widetilde{LN} and a time-dependent vector field $\mathbf{w} : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^2$, the total flux of the *flux density vector* $\mathbf{w}(\mathbf{x}, t)$ through \widetilde{LN} within a time interval $[t_0, t_0 + k]$ can be expressed as

$$(1.1) \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} \mathbf{w}(\gamma(s), t) \cdot \mathbf{n}_{\widetilde{LN}} ds dt,$$

where $\gamma(s)$ and $\mathbf{n}_{\widetilde{LN}}$ are a parametrization and the normal vector of \widetilde{LN} , respectively. We also refer to the total flux in (1.1) as the *Eulerian flux integral* of \mathbf{w} through the curve \widetilde{LN} .

The notion of fluxes is fundamental and ubiquitous in transport, mixing, and other physical processes, with the form of the flux density vector $\mathbf{w}(\mathbf{x}, t)$ varying from application to application. In the context of scalar conservation laws, we often write $\mathbf{w}(\mathbf{x}, t)$ as the product of another vector $\mathbf{u}(\mathbf{x}, t)$ and a scalar function $f(\mathbf{x}, t)$, with \mathbf{u} considered as the velocity field of the underlying flow and f the scalar quantity being conserved. If \mathbf{u} and f are sufficiently continuous, the conservation law is written

$$(1.2) \quad \frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{f}\mathbf{u}) = 0.$$

Flux integrals are also important in numerically solving (1.2). In finite volume (FV) methods, flux calculation is of paramount significance in ensuring stability and

*Submitted to the journal's Methods and Algorithms for Scientific Computing section September 27, 2018; accepted for publication (in revised form) September 3, 2019; published electronically November 14, 2019.

<https://doi.org/10.1137/18M1210885>

Funding: The first author's work was partially supported by the National Natural Science Foundation of China through grant 11621101.

[†]Corresponding author. School of Mathematical Sciences, Zhejiang University, Hangzhou, Zhejiang Province, 310027, China (qinghai@zju.edu.cn).

[‡]School of Mathematical Sciences, Zhejiang University, Hangzhou, Zhejiang Province, 310027, China (21435043@zju.edu.cn).

accuracy [12]. For tracking interface between multiple phases, the accuracy of the advection substep in volume-of-fluid (VOF) methods is solely determined by that of calculating fluxes through cell edges [33]. In traditional FV methods for solving (1.2), the curve \widetilde{LN} is identified with boundary edges of control volumes (or cells). Then Eulerian fluxes through cell edges within the computational domain are discretized as linear combinations of cell-averaged values of f ; this spatial discretization leads to a system of ordinary differential equations (ODEs). Lastly, a time-integration method is employed to solve the ODE system.

The aforementioned FV approach with the method-of-lines (MOL) discretization have been very successful in the past several decades [20, 21]. However, it is by no means perfect. For example, when the scalar f is not continuous yet still integrable, it is difficult to achieve high convergence rates. Even when f is indeed smooth, the time step size of an FV-MOL solver is subject to the constraint of the well-known Courant–Friedrichs–Lewy (CFL) condition, which may be too restrictive for certain problems.

To analytically and numerically complement the FV-MOL approach, it is desirable to find the so-called *donating region* (DR) at the initial time so that the flux of a scalar $f(\mathbf{x}, t)$ through a static simple smooth curve \widetilde{LN} satisfies

$$(1.3) \quad \exists k_M > 0 \text{ s.t. } \forall k < k_M, \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f(s, t) \mathbf{u}(s, t) \cdot \mathbf{n}_{\widetilde{LN}} ds dt = \int_{\mathcal{D}_{\widetilde{LN}}(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x},$$

where $\mathbf{n}_{\widetilde{LN}}$ is the unit normal vector of \widetilde{LN} ; see Definition 2.5. The DR $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ is a point set at the initial time t_0 consisting of passively advected particles that will go across \widetilde{LN} once within the time interval $[t_0, t_0 + k]$ and contribute to the total flux. The right-hand side (RHS) of (1.3) is called a *Lagrangian flux integral*, as opposed to the Eulerian flux integral in the left-hand side (LHS) of (1.3). Hereafter the term *Lagrangian flux calculation* (LFC) refers to the estimation of an Eulerian flux integral by evaluating a corresponding Lagrangian flux integral. For steady flows, it is trivial to construct a DR from streamlines; thus the main difficulty in finding such a DR in (1.3) lies in the time dependence of the velocity field.

We list as follows a number of prominent utilities of LFC and the flux identity.

- (UFI.1) The time dependence of the scalar f is removed so that the spatial-temporal integral reduces to a spatial integral at the initial time.
- (UFI.2) The flux identity (1.3) is useful in determining local truncation errors of unsplit multidimensional FV algorithms, even when f is discontinuous in space; see [33] for such an analysis on a family of VOF advection methods.
- (UFI.3) Sometimes our primary interest is not the evolution of the scalar over the entire computational domain, but the dynamics in *local* regions. In this case, LFC would be much more flexible and efficient than FV-MOL.
- (UFI.4) LFC can also be adapted to solve conservation laws (1.2) over the *entire* computational domain, with the advantage that the time step size is not subject to the CFL condition. In this sense, LFC is closely related to the so-called semi-Lagrangian methods [4, 27], also known as Eulerian–Lagrangian methods [15] or the modified method of characteristics [9]; see [32] and references therein for more discussions.

Previously, we have analytically determined the DR of \widetilde{LN} in (1.3), with the assumption that the length of the time interval k is short enough to prevent intersections or self-intersections of the streaklines seeded at endpoints of \widetilde{LN} [34, 35].

Based on (1.3), we also proposed an LFC algorithm [32] (hereafter referred to as LFC-2013), which consists of constructing the DR $\mathcal{D}_{\widetilde{LN}}$ from the given velocity $\mathbf{u}(\mathbf{x}, t)$ and then calculating the Lagrangian flux integral over $\mathcal{D}_{\widetilde{LN}}$ via Green's theorem and one-dimensional (1D) quadrature formulas [25, 24]. However, due to the restrictive assumption of nonintersecting streaklines, sometimes we *have to* divide the whole time interval $[t_0, t_0 + k]$ into a number of sufficiently small subintervals to ensure the applicability of LFC-2013, after which we perform LFC-2013 for all subintervals and sum up the results. Consequently, values of the scalar f at the starting time of each subinterval must be known, and this undermines the leverage of LFC as discussed in (UFI.1) and (UFI.3). After all, one advantage of LFC is to compute the Eulerian flux integral with no knowledge of the scalar f except at the very initial time t_0 .

Naturally, the next step along this line of research is to remove the restrictive assumption of nonintersecting streaklines so that the flux identity (1.3) holds for an arbitrary $k > 0$. Consequently, fluid parcels may cross the curve \widetilde{LN} more than once and thus contribute to the total flux multiple times. Then we have

$$(1.4) \quad \forall k > 0, \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt = \sum_{n \in \mathbb{Z}} n \int_{\mathcal{D}_{\widetilde{LN}}^n(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x},$$

where the *fluxing index* $n \in \mathbb{Z}$ on the RHS denotes the number of times of crossing \widetilde{LN} in the net sense of any Lagrangian particle whose initial position is in $\mathcal{D}_{\widetilde{LN}}^n(t_0, k)$.

This fluxing index n induces the equivalence classes of points at the initial time called *flux sets*, which are shown in our previous work to be index-by-index equivalent to DRs [35] in the context of scalar conservation laws. This partition of the phase space via flux sets is more fundamental than the proof of the flux identity (1.4). After all, the flux sets are well defined from the given curve \widetilde{LN} , the velocity field \mathbf{u} , and the induced ODE $\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t)$. However, for the flux identity to hold, another relation between f and \mathbf{u} , namely (1.2), is needed in addition to those for the flux sets. Therefore, the DRs induced by the flux identity (1.4) cannot replace flux sets because they holds only in the more restricted context of scalar conservation laws. In this paper, however, we do not emphasize the difference between flux sets and DRs (since we are working with scalar conservation laws) but only mention the utility of partitioning the phase space in touting the usefulness of the proposed algorithms.

(UFI.5) An explicit construction of the DRs $\mathcal{D}_{\widetilde{LN}}^n(t_0, k)$ is essentially a partition of the phase space that determines regions inside which particles cross \widetilde{LN} in different times and different directions.

Recently, Karrasch and colleagues [10, 7] gave a proof of the flux identity (1.4), which holds not only in two dimensions but also in higher dimensions and for moving surfaces. In addition, Hofherr and Karrasch proposed an LFC algorithm [7, Algorithm 1], hereafter referred to as LFC-HK, which consists of

- (i) constructing the bounding curve $\partial\mathcal{D}$ of the DR,
- (ii) dividing $\partial\mathcal{D}$ into simple loops,
- (iii) calculating the index n or the winding number for each sub-DR $\mathcal{D}_{\widetilde{LN}}^n(t_0, k)$,
- (iv) evaluating integrals on these sub-DRs and their weighted sum according to (1.4).

As a major advantage of LFC-HK over LFC-2013, values of the scalar f are needed only at the initial time. This work by Hofherr and Karrasch [7] sheds valuable insights on the theoretical side and is an excellent contribution to the DR theory and LFC algorithms. On the other hand, in LFC-HK the DRs are approximated by linear polygons, which limits the algorithm to second-order accuracy. Also, it is potentially ill-conditioned to calculate intersections of line segments in step (ii).

The flux identities (1.3) and (1.4) have been utilized by other researchers to design new algorithms for tracking moving interface of multiphase flows [3], to quantify transport in a single divergence-free fluid [10], and to study transport between two fluids across their mutual flow interface [7]. In the wider context of transport and mixing, there is a long history and a large body of work on Lagrangian fluxes through *moving* curves and surfaces. As a notable example, Rom-Kedar and colleagues [17, 18] describe the time-periodic lobe dynamics as an effort to quantify the Lagrangian flux through a moving vortex. In a recent work by Balasuriya [2], the notion of streaklines is utilized to study the transport between two fluids across their interface.

So far we have reviewed the successes and limitations of the current DR theory and LFC algorithms. This work is motivated by questions on further augmenting the utilities (UFI.1)–(UFI.4).

- (Q.1) Is it possible to prove (1.4) via our approach [34, 35] that reveals much of the geometrical and topological details of DRs? This requires the removal of the restrictive assumption of non-intersecting streaklines.
- (Q.2) Are there other forms of flux identities more suitable for LFC than (1.4)?
- (Q.3) Can we design a simple and well-conditioned LFC algorithm that is free of calculating intersections of curve segments? Ideally, such an LFC algorithm only entails weighted sum of function values of the scalar at the initial time.
- (Q.4) Can we achieve fourth- and higher-order accuracy in LFC?

In this paper, we provide positive answers to all of the above questions.

In answering (Q.1), we prove (1.4) by coupling the Hopf theorem with fundamental results in ODE and scalar conservation laws. After the removal of the aforementioned restrictive assumption, we bring our DR theory to the same generality as that of Hofherr and Karrasch [7] in two dimensions. One distinguishing feature of our approach from that of Hofherr and Karrasch [10, 7] is the revealing of many geometric and topological details, and these details, in our opinion, are essential in the design of efficient and accurate LFC algorithms. In addition, our approach includes an explicit and intuitive partitioning of the phase space for (UFI.5) [35], which is not afforded by the approach of Hofherr and Karrasch [10, 7].

Although (1.4) seems intuitively clear, it is necessary to formally prove it. As a major uncertainty, why must the scalar conservation law (1.2) be given a priori in order for these flux identities to hold? Why not the advection equation $\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = 0$? In fact, (1.4) would not hold for compressible flows if the advection equation were assumed as the relation between f and \mathbf{u} .

Our answer to (Q.2) is the following form of flux identity,

$$(1.5) \quad \forall k > 0, \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt = \oint_{\gamma_D} F(x, y, t_0) dy,$$

where $F(x, y, t_0) = \int_{\varsigma}^x f(v, y, t_0) dv$, ς is an arbitrary fixed real number and the oriented closed curve γ_D generates sub-DRs of various indices; see Definitions 2.1 and 2.4. We emphasize that, although γ_D can be regarded as the “boundary” of the DR in (1.4), (1.5) is not a simple deduction from (1.4) and Green’s theorem: γ_D may be self-intersecting while the boundary of the integral domain in Green’s theorem must be simple closed.

Thanks to the RHS of (1.5) being a *line integral*, our new LFC algorithm only consists of constructing the closed curve γ_D and integrating F along γ_D . In particular, it is free of steps (ii) and (iii) of LFC-HK. Consequently, our LFC algorithm reduces to numerically solving ODEs and calculating weighted sums of scalar function values at

the initial time. This is our answer to (Q.3). The proposed LFC algorithm gives a user the flexibility to choose a length scale h and a time integrator of order $\kappa = 2, 4$, or 6 so that the error in estimating the Eulerian flux is asymptotically $O(h^\kappa)$. This answers (Q.4). In summary, the proposed LFC algorithm is simple, flexible, well-conditioned, accurate, and free of all disadvantages of LFC-2013 and LFC-HK.

The rest of this paper is organized as follows. In section 2, we introduce fundamental building blocks that naturally lead to the formalization of the proposed LFC algorithm. In section 3, we show the correctness of this LFC algorithm by proving the flux identity (1.5) and then deriving (1.4) as a consequence. Sections 2 and 3 are largely self-contained in that they should be accessible to anyone with a good understanding of basic real analysis. On the other hand, a reader only interested in algorithm implementation can safely skip section 3. In section 4, we perform numerical tests to demonstrate the accuracy, efficiency, and utility of the proposed LFC algorithm. Section 5 concludes this paper with several research prospects.

2. Algorithm. The ultimate goal of this section is to explain and formalize our new LFC algorithm. Along the way to fulfilling this goal in the last subsection, we introduce crucial concepts such as flow maps, streaklines, flux-generating curves, and DRs. Modular procedures that implement these basic concepts serve as building blocks of the proposed LFC algorithm. In particular, the procedure of streakline approximation are utilized to create counterexamples to two common beliefs on streakline intersections; see section 2.3.

2.1. The flow map of an ODE. A nonautonomous ODE

$$(2.1) \quad \frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t)$$

admits a unique solution for any given initial time t_0 and initial position $p_0 \in \mathbb{R}^D$ if the time-dependent velocity field $\mathbf{u}(\mathbf{x}, t)$ is continuous in time and Lipschitz continuous in space. This uniqueness gives rise to a flow map $\phi : \mathbb{R}^D \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^D$ that takes the initial position p_0 of a Lagrangian particle p , the initial time t_0 , and the time increment $\pm k$ and returns $p(t_0 \pm k)$, the position of p at the final time $t_0 \pm k$:

$$(2.2) \quad \begin{cases} \phi_{t_0}^{+k}(p) := p(t_0 + k) = p(t_0) + \int_{t_0}^{t_0+k} \mathbf{u}(p(t), t) dt, \\ \phi_{t_0}^{-k}(p) := p(t_0 - k) = p(t_0) + \int_{t_0-k}^{t_0} \mathbf{u}(p(t), t) dt. \end{cases}$$

The flow map also generalizes to point sets in a straightforward way:

$$\phi_{t_0}^{\pm k}(\mathcal{M}) = \{\phi_{t_0}^{\pm k}(p) : p \in \mathcal{M}\}.$$

We will also use the shorthand notation $\overleftarrow{M} := \phi_{t_0+k}^{-k}(M)$ when t_0 and k are clear from the context. In Figure 2.1, we detail a procedure for numerically approximating the action of the flow map upon a finite set \mathcal{P} of Lagrangian particles. For efficiency, all particles in \mathcal{P} are advected simultaneously.

2.2. Characteristic curves of the flow map. An *open curve* is (the image of) a continuous map $\gamma : (0, 1) \rightarrow \mathbb{R}^2$; it is *simple* if γ is injective; otherwise it is *self-intersecting*. A *closed curve* is (the image of) a continuous map $\gamma : [0, 1] \rightarrow \mathbb{R}^2$ with $\gamma(0) = \gamma(1)$; it is *simple closed* or *Jordan* if the restriction of γ to $[0, 1]$ is injective. An isolated intersection point p of two curves is a *proper intersection* or a *crossing* if at a sufficiently small local neighborhood of p one curve lies at both sides of the other

Procedure flowMap(\mathbf{u} , $\{p_i\}_{i=1}^n$, t_0 , t_e , κ , Δt)

Input: A velocity field $\mathbf{u}(\mathbf{x}, t)$, a sequence of points $\{p_i\}_{i=1}^n$, the initial time t_0 , the ending time t_e , a κ th-order time integrator **ODEsolve**, and a tentative time stepping increment Δt

Preconditions : $\Delta t(t_e - t_0) > 0$

Output: a finite sequence of points $\{q_i\}_{i=1}^n$

Postconditions: $\forall i = 1, \dots, n, \|q_i - \phi_{t_0}^{t_e-t_0}(p_i)\|_2 = O((\Delta t)^\kappa)$

```

1  $m \leftarrow \lceil \frac{t_e-t_0}{\Delta t} \rceil; \Delta t \leftarrow \frac{t_e-t_0}{m}$ 
2  $\{q_i\}_{i=1}^n \leftarrow \{p_i\}_{i=1}^n$ 
3 for  $i = 0 : m - 1$  do
4    $\{q_i\}_{i=1}^n \leftarrow \text{ODEsolve}(\mathbf{u}, \{q_i\}_{i=1}^n, t_0 + i\Delta t, \Delta t)$ 
5 end
6 return  $\{q_i\}_{i=1}^n$ 
```

FIG. 2.1. A procedure that approximate the flow map ϕ in (2.2). The time integrator **ODEsolve** has the signature $\{q_i(t + \Delta t)\}_{i=1}^n = \text{ODEsolve}(\mathbf{u}, \{q_i(t)\}_{i=1}^n, t, \Delta t)$, where $|\Delta t|$ is a stable time stepping size, and the finite set $\{q_i\}_{i=1}^n$ of particles is advected from their position at t to $t + \Delta t$. The postconditions hold from line 1 and the κ th-order accuracy of **ODEsolve**.

curve; otherwise it is an *improper intersection*. An *orientation* of a curve γ is the assignment of a direction in which γ is traversed. We follow the common convention that counterclockwise orientation is positive while clockwise orientation is negative.

One common characteristic curve of the flow map is the *pathline*, a curve generated by following a single particle in a time interval:

$$(2.3) \quad \Phi_{t_0}^{\pm k}(p) = \{\phi_{t_0}^{\pm \tau}(p) : \tau \in (0, k)\}.$$

A *backward streakline* is the loci of all particles that will pass continuously through a fixed seeding location M ,

$$(2.4) \quad \Psi_{t_0+k}^{-k}(M) := \{\phi_{t_0+\tau}^-(M) : \tau \in (0, k)\},$$

and a *forward streakline* is the loci of all particles that have passed M ,

$$(2.5) \quad \Psi_{t_0-k}^{+k}(M) := \{\phi_{t_0-\tau}^+(M) : \tau \in (0, k)\},$$

where the time increment $k > 0$.

Backward streaklines and forward streaklines are distinguished by the sign of the superscript of Ψ . A streakline in the above definition is a snapshot: all particles in (2.4) and in (2.5) are at $t = t_0$. In comparison, a pathline is the history of a single Lagrangian particle. By (2.3), (2.4), and (2.5), pathlines and streaklines are indeed curves because they are continuous maps from an interval to the plane.

In Figure 2.2, we detail a procedure for generating a sequence of points that discretely approximate the backward streakline. Note that forward streaklines are only useful for analysis and are not needed in our LFC algorithm.

2.3. Intersecting streaklines. By the uniqueness of the solution of the ODE (2.1), two different particles cannot be present at the same location and the same time. This observation is sometimes used to justify two widespread common beliefs on streaklines [31]: “a streakline does not intersect itself” and “two different streaklines do not intersect.” Unfortunately, neither of these two beliefs is correct.

Procedure backwardStreak(\mathbf{u} , M , t_0 , k , $\{\tau_i\}_{i=1}^{\iota+1}$, κ , Δt).

Input: A velocity field $\mathbf{u}(\mathbf{x}, t)$, a seeding position M , the initial time t_0 , the time increment k , a sequence $\{\tau_i\}_{i=1}^{\iota+1}$, a κ th-order time integrator **ODEsolve**, and a tentative time stepping increment Δt .

Preconditions : $\Delta t < 0$; $k > 0$; $\tau_1 = 0$, $\tau_{\iota+1} = k$, $\forall i = 1, \dots, \iota$, $\tau_i < \tau_{i+1}$.

Output: A sequence of points $\{p_i\}_{i=1}^{\iota+1}$.

Postconditions: $\forall i = 1, \dots, \iota + 1$, $\|p_i - \phi_{t_0 + \tau_i}^{-k}(M)\|_2 = ((\Delta t)^\kappa)$.

```

1  $p_1 \leftarrow M$ 
2 for  $i = 2 : \iota + 1$  do
3   |  $p_i \leftarrow \text{flowMap}(\mathbf{u}, M, t_0 + \tau_i, t_0, \kappa, \Delta t)$ 
4 end
5 return  $\{p_i\}_{i=1}^{\iota+1}$ 

```

FIG. 2.2. A procedure that generates a sequence of points so that connecting these points in order yields a consistent approximation of a backward streakline. The postconditions follow from those of the procedure **flowMap**.

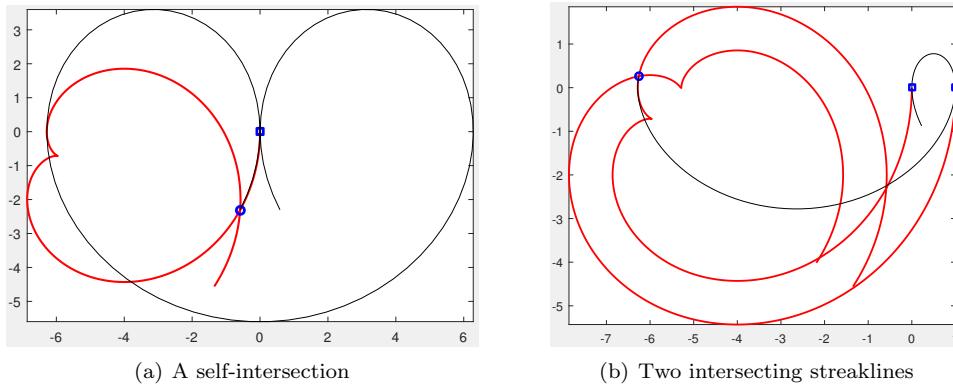


FIG. 2.3. Examples of streakline intersections. The velocity field is $\mathbf{u} = (-y, x - t + 5)^T$, with $t_0 = 0$ and $k = 10$. The seeding locations of streaklines, $L = (0, 0)$ and $N = (1, 0)$, are represented by squares, with the backward streaklines $\Psi_{t_0+k}^{-k}(L)$ and $\Psi_{t_0+k}^{-k}(N)$ represented by red thick lines. Both streaklines contain cusps where tangent vectors of streaklines are undefined. A circle represents a self-intersection p or an intersection p of two different streaklines. A thin solid line represents a pathline $\Phi_{t_0}^{+k'}(p)$ with $k' = 10$ for subplot (a) and $k' = 7$ for subplot (b). Note that in subplot (a) the pathline and streakline almost overlap between L and p .

Starting from the same initial position \mathbf{x} , two particles with different initial times may reach the seeding location M at two different instants within the time interval $[t_0, t_0 + k]$. Hence \mathbf{x} could be a self-intersection of the backward streakline $\Psi_{t_0+k}^{-k}(M)$. For two different seeding locations L and N , although no particle can be simultaneously at L and N for any given time, a particle p may arrive at L and N at two different time instants; hence p might as well be an intersection point of $\Psi_{t_0+k}^{-k}(L)$ and $\Psi_{t_0+k}^{-k}(N)$. In Figure 2.3, we use the procedure **backwardStreak** to generate intersections and self-intersections of streaklines for a simple velocity field.

Two more identities on streaklines can be found in section 3.4; the reader is also referred to [30] and [10, Figure 2] for additional illustrations.

2.4. Flux-generating curves. The central concept for our LFC algorithm is the following algebraically closed curve.

DEFINITION 2.1 (the flux-generating curve). *For a continuously differentiable velocity field $\mathbf{u}(\mathbf{x}, t)$ and a static simple curve \widetilde{LN} , the flux-generating curve for \widetilde{LN} over time interval $(t_0, t_0 + k)$ is the oriented closed curve*

$$(2.6) \quad \gamma_{\mathcal{D}} := L \cup \widetilde{LN} \cup N \cup \Psi_{t_0+k}^{-k}(N) \cup \overleftarrow{N} \cup \phi_{t_0+k}^{-k}(\widetilde{NL}) \cup \overleftarrow{L} \cup \Psi_{t_0+k}^{-k}(L),$$

where $\overleftarrow{\bullet} := \phi_{t_0+k}^{-k}(\bullet)$ and the orientation of $\gamma_{\mathcal{D}}$ is determined by the closed vertex sequence $L \rightarrow N \rightarrow \overleftarrow{N} \rightarrow \overleftarrow{L} \rightarrow L$.

In Figure 2.4, we detail a procedure for producing a closed spline as an approximation to the flux-generating curve. Typical examples of flux-generating curves are shown in Figure 2.5.

2.5. Donating regions. Let γ be a closed curve and $p(t)$ a particle moving continuously on the plane. Suppose p crosses γ at time t_c . A proper intersection

Procedure generatingCurve($\mathbf{u}, \beta, t_0, k, h, \kappa$).

Input: A velocity field $\mathbf{u}(\mathbf{x}, t)$, a simple curve $\widetilde{LN} = \beta([0, 1])$, the time interval $[t_0, t_0 + k]$, a length scale h , and the convergence rate κ .

Preconditions : $\beta(0) = L$, $\beta(1) = N$; $k > 0$; $h > 0$; $\kappa = 2, 4$, or 6 .

Output: A closed spline $\dot{\gamma}_{\mathcal{D}}$ of the κ th order.

Postconditions: Adjacent knots of $\dot{\gamma}_{\mathcal{D}}$ have distances no greater than h ;
 $\forall s \in [0, 1], \|\dot{\gamma}_{\mathcal{D}}(s) - \gamma_{\mathcal{D}}(s)\|_2 = O(h^\kappa)$.

- 1 $m \leftarrow \frac{1}{h} \int_0^1 \|\beta'\|_2$ where the integral is the length of \widetilde{LN}
 - 2 $n_L \leftarrow \frac{k}{h} \|\mathbf{u}(L, t_0)\|_2$; $n_N \leftarrow \frac{k}{h} \|\mathbf{u}(N, t_0)\|_2$
 - 3 $\{p_i^1\}_{i=1}^{m+1} \leftarrow \{\beta\left(\frac{i-1}{m}\right)\}_{i=1}^{m+1}$
 - 4 $\{p_i^2\}_{i=1}^{n_N+1} \leftarrow \text{backwardStreak}\left(\mathbf{u}, N, t_0, k, \left\{t_0 + \frac{i-1}{n_N}k\right\}_{i=1}^{n_N+1}, \kappa, \frac{k}{m}\right)$
 - 5 $\{p_i^3\}_{i=1}^{m+1} \leftarrow \text{flowMap}\left(\mathbf{u}, \{p_j^1\}_{j=m+1}^1, t_0 + k, t_0, -\frac{k}{m}\right)$
 - 6 $\{p_i^4\}_{i=1}^{n_L+1} \leftarrow \text{backwardStreak}\left(\mathbf{u}, L, t_0, k, \left\{t_0 + \frac{i-1}{n_L}k\right\}_{i=1}^{n_L+1}, \kappa, \frac{k}{m}\right)$
 - 7 If the distance between any pair of adjacent points in $\{p_i^j\}$ ($j = 1, 2, 3, 4$) is greater than h , insert additional knots so that no such distance exists
 - 8 If the distance between any pair of adjacent points in $\{p_i^j\}$ ($j = 1, 2, 3, 4$) is less than $h/100$, delete a knot so that no such distance exists
 - 9 reverse the order of points in $\{p_i^4\}$
 - 10 **for** $j = 1 : 4$ **do**
 - 11 | $\gamma^j \leftarrow$ a κ th-order spline fit through $\{p_i^j\}$
 - 12 **end**
 - 13 $\dot{\gamma}_{\mathcal{D}} \leftarrow$ the concatenation of $\gamma^1, \gamma^2, \gamma^3$, and γ^4
 - 14 **return** $\dot{\gamma}_{\mathcal{D}}$
-

FIG. 2.4. A procedure that produces a spline that approximates the flux-generating curve. The “not-a-knot” condition is used for line 10. Line 1 implies that $\frac{k}{m} = O(h)$ in lines 4–6. The postconditions then follow from line 7 and postconditions of `flowMap` and `backwardStreak`. Lines 7 and 8 ensure the well-conditioning of the linear system for the spline fitting at line 11.

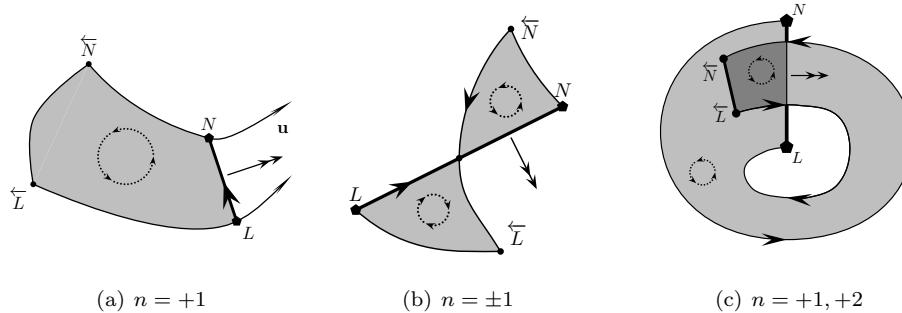


FIG. 2.5. Examples of flux-generating curves, DRs of a simple open curve \widetilde{LN} , and the unit outward normal vector of \widetilde{LN} . A flux-generating curve is represented by a closed solid curvilinear line with arrows. DRs are represented by shaded regions. The orientation of dotted circles with small arrows indicates the sign of the index of DR. In subplot (c), the dark gray region and the light gray region represent $\mathcal{D}_{\widetilde{LN}}^{+2}$ and $\mathcal{D}_{\widetilde{LN}}^{+1}$, respectively. The double arrowed vectors indicate $\mathbf{n}_{\widetilde{LN}}$, the unit normal vector of \widetilde{LN} in Definition 2.5.

$\gamma(s_c) = p(t_c)$ is called a *positive crossing* if, in a sufficiently small open neighborhood of s_c , γ is oriented from the left of the directed path of p to its right; otherwise it is called a *negative crossing* [1]. As shown in [35, section 3], the following definition is equivalent to the common notion of winding numbers.

DEFINITION 2.2 (winding numbers). *Let γ be an oriented closed curve and $\mathbf{x} \notin \gamma$ a fixed position. Let p denote a particle that moves continuously from \mathbf{x} to ∞ . The winding number of γ around \mathbf{x} , written $w(\gamma, \mathbf{x})$, is the number of negative crossings of p through γ minus its number of positive crossings.*

Definition 2.2 has an immediate consequence as follows.

COROLLARY 2.3. *Let γ be an oriented closed curve and \mathbf{x} a fixed point in the unbounded complement of γ . Then $w_\gamma(\mathbf{x}) = 0$.*

The winding number of an oriented closed curve furnishes a family of equivalence classes of points in the plane.

DEFINITION 2.4 (donating regions). *For a continuously differentiable velocity field $\mathbf{u}(\mathbf{x}, t)$, the donating region of index n associated with a simple curve \widetilde{LN} fixed over the time interval $(t_0, t_0 + k)$ is*

$$(2.7) \quad \mathcal{D}_{\widetilde{LN}}^n(t_0, k) := \{\mathbf{x} \in \mathbb{R}^2 : w(\gamma_{\mathcal{D}}, \mathbf{x}) = n\},$$

where $\gamma_{\mathcal{D}}$ is the flux-generating curve in Definition 2.1.

We denote by $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ the union of DRs with nonzero indices,

$$(2.8) \quad \mathcal{D}_{\widetilde{LN}}(t_0, k) = \bigcup_{n \in \mathbb{Z} \setminus \{0\}} \mathcal{D}_{\widetilde{LN}}^n(t_0, k).$$

DEFINITION 2.5. *The unit normal vector of a simple curve \widetilde{LN} , written $\mathbf{n}_{\widetilde{LN}}$, is obtained by a clockwise rotation for $\frac{\pi}{2}$ of the unit tangent vector of \widetilde{LN} .*

Definitions 2.4 and 2.5 are illustrated in Figure 2.5. As a physical interpretation of Definition 2.5, particles originally in a DR of positive indices will cross \widetilde{LN} in a

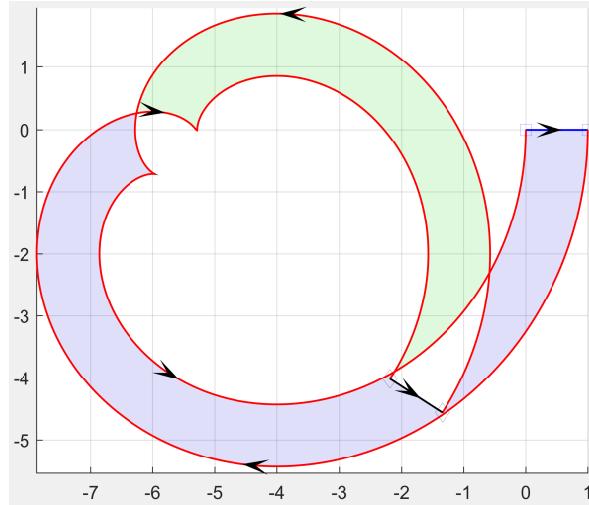


FIG. 2.6. A generating curve with intersecting streaklines and the corresponding nonnormal DR. The velocity field is $\mathbf{u} = (-y, x - t + 5)^T$ with $t_0 = 0$ and $k = 10$. The squares represent the seeding locations of streaklines $L = (0, 0)$ and $N = (1, 0)$, with the two streaklines the same as those in Figure 2.3. The blue curve represents \widetilde{LN} , the black curve represents its preimage, and the red curves represent the streaklines. The arrows indicate the orientation of the generating curve. The light green, light blue, and white regions represent $D_{\widetilde{LN}}^{+1}(t_0, k)$, $D_{\widetilde{LN}}^{-1}(t_0, k)$, and $D_{\widetilde{LN}}^0(t_0, k)$, respectively. See the supplemental movie file *nonNormalDR.gif*.

direction that agrees with the normal vector of \widetilde{LN} , i.e., the inner product of these two vectors is positive. It can be proven that this definition of $\mathbf{n}_{\widetilde{LN}}$ is consistent both with our previous definition [35, Definition 4.7] and with that by Hofherr and Karrasch [7]; see the discussion after Theorem 2.6.

Due to (2.7), the flux-generating curve of \widetilde{LN} is also referred to as the *generating curve* of $D_{\widetilde{LN}}^n$. A DR is *normal* if the two backward streaklines seeded at L and N neither intersect nor self-intersect. All DRs in Figure 2.5 are normal while the DR in Figure 2.6 is not. However, the DR in Figure 2.6 would be normal if k were chosen sufficiently small; see Figure 2.7(a). In other words, the normality of a DR can be guaranteed by choosing the length of the time interval to be sufficiently small [35, Proposition 4.6]. This statement can be strengthened so that, for sufficiently short time intervals, either γ_i is Jordan or each self-intersection of γ_i is an intersection of \widetilde{LN} and its preimage.

In the special case of steady flows, a streakline reduces to a streamline, of which its tangent vector is the velocity. Therefore, streamlines never intersect. Consequently, DRs for steady flows are always normal.

2.6. The LFC algorithm. The theoretical foundation of our new LFC algorithm is the flux identity (1.5), the proof of which is deferred to Theorem 3.8. Algorithmically, our LFC is based on the procedure `generatingCurve` in Figure 2.4, and it simply orchestrates the RHS of (1.5) by recursively invoking Gaussian formulas for 1D quadrature.

THEOREM 2.6. Denote by $\dot{\gamma}_D := \{(\hat{x}_s(\ell), \hat{y}_s(\ell))\}_{s=1}^\ell$ a κ th-order closed spline in piecewise polynomial form where $\ell \in [\ell_s, \ell_{s+1}]$ for the s th segment. Let $\mathcal{R} \supseteq \dot{\gamma}_D$ be a rectangular $\mathcal{R} = [x_1, x_2] \times [y_1, y_2]$ with a fixed $\varsigma \in [x_1, x_2]$. Denote by $\{\lambda_i^z\}_{i=1}^z$ and

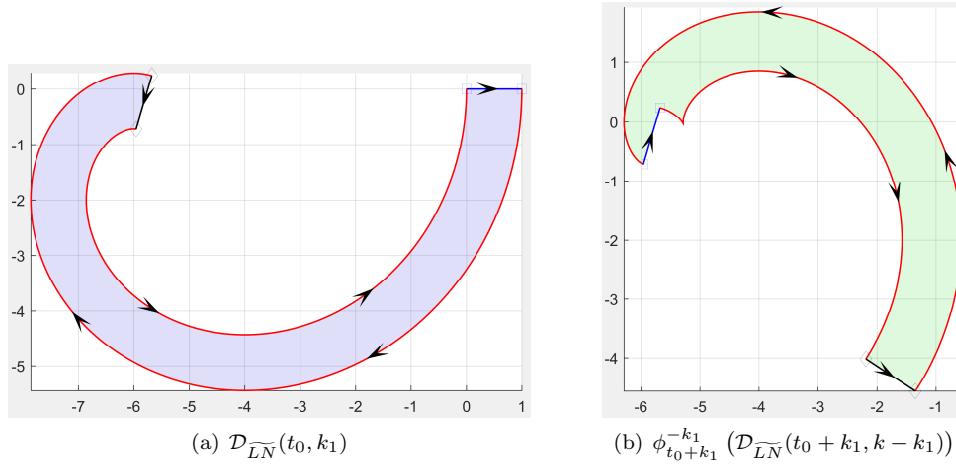


FIG. 2.7. An algebraic decomposition of the non-normal DR $\mathcal{D}_{\overline{LN}}(t_0, k)$ in Figure 2.6 into a normal DR and the image of another normal DR; see Figure 2.6 for the caption. The decomposition is as follows. First, we split the time interval $[t_0, t_0 + k]$ into two subintervals $[t_0, t_0 + k_1]$ and $[t_0 + k_1, t_0 + k]$ so that both DRs $\mathcal{D}_{\overline{LN}}(t_0, k_1)$ and $\mathcal{D}_{\overline{LN}}(t_0 + k_1, k - k_1)$ are normal. Since the flow map is a homeomorphism, a preimage of $\mathcal{D}_{\overline{LN}}(t_0 + k_1, k - k_1)$ is also normal, as shown in subplot (b). Then we define a homological concatenation of two closed curves by canceling their overlapping parts with different orientations; for example, the black line segment in subplot (a) and the blue line segment in subplot (b) cancel each other due to different orientations. This concatenation of the two closed curves in subplots (a) and (b) yields the generating curve in Figure 2.6.

$\{\omega_i^z\}_{i=1}^z$ nodes and weights of a Gauss formula of z nodes within $[-1, 1]$. Then for every bivariate polynomial $P(x, y)$ of degree $\leq q$, the following algebraic quadrature formula is exact:

$$(2.9) \quad \oint_{\dot{\gamma}_{\mathcal{D}}} F_P(x, y) dy = I(\dot{\gamma}_{\mathcal{D}}, P) := \sum_{s=1}^{\ell} \sum_{i=1}^n \sum_{j=1}^m w_{sij} P(x_{sij}, y_{sj}),$$

where $F_P(x, y) := \int_{\zeta}^x P(v, y)dv$, “ $\lceil \cdot \rceil$ ” denotes the ceiling function, and

$$(2.10) \quad \begin{cases} n \geq \lceil \frac{q+1}{2} \rceil, \quad m \geq \lceil \frac{(q+2)(\kappa-1)}{2} \rceil; \\ r_s(\eta) = \frac{\ell_{s+1}-\ell_s}{2}\eta + \frac{\ell_{s+1}+\ell_s}{2}, \quad r_{sj}^m = r_s(\lambda_j^m); \\ x_{sij} = \frac{\hat{x}_s(r_{sj}^m)-\varsigma}{2}\lambda_i^n + \frac{\hat{x}_s(r_{sj}^m)+\varsigma}{2}, \quad y_{sj} = \hat{y}_s(r_{sj}^m); \\ w_{sij} = \frac{\ell_{s+1}-\ell_s}{2} \cdot \frac{\hat{x}_s(r_{sj}^m)-\varsigma}{2} \cdot \omega_i^n \omega_j^m \frac{d\hat{y}_s}{dl}(r_{sj}^m). \end{cases}$$

Proof. We substitute the piecewise polynomial form into (2.9), normalize each integral domain to $[-1, +1]$, and repeatedly apply Gauss quadrature formulas to get

$$\begin{aligned}
\oint_{\hat{\gamma}_{\mathcal{D}}} F_P(x, y) \, dy &= \sum_{s=1}^{\ell} \int_{\ell_s}^{\ell_{s+1}} F_P(\hat{x}_s, \hat{y}_s) \frac{d\hat{y}_s}{d\ell} d\ell \\
&= \sum_{s=1}^{\ell} \int_{-1}^{+1} \frac{\ell_{s+1} - \ell_s}{2} F_P(\hat{x}_s(\eta), \hat{y}_s(\eta)) \frac{d\hat{y}_s}{d\ell}(\eta) d\eta \\
&= \sum_{s=1}^{\ell} \int_{-1}^{+1} \frac{\ell_{s+1} - \ell_s}{2} \frac{d\hat{y}_s}{d\ell}(\eta) \int_{\zeta}^{\hat{x}_s(\eta)} P(x, \hat{y}_s(\eta)) dx d\eta \\
&= \sum_{s=1}^{\ell} \int_{-1}^{+1} \int_{-1}^{+1} \left\{ \frac{\ell_{s+1} - \ell_s}{2} \cdot \frac{\hat{x}_s(\eta) - \varsigma}{2} \cdot \frac{d\hat{y}_s}{d\ell}(\eta) \cdot P(v(\lambda), \hat{y}_s(\eta)) \right\} d\lambda d\eta,
\end{aligned}$$

where the first step follows from the piecewise polynomial form $\dot{\gamma}_{\mathcal{D}} := \{(\hat{x}_s(\ell), \hat{y}_s(\ell))\}_{s=1}^{\nu}$, the second step follows from the variable change

$$\ell = r_s(\eta) = \frac{\ell_{s+1} - \ell_s}{2}\eta + \frac{\ell_{s+1} + \ell_s}{2},$$

the third step follows from $F_P(x, y) := \int_{\varsigma}^x P(v, y)dv$, and the last step follows from

$$x = v(\lambda) = \frac{\hat{x}_s(\ell) - \varsigma}{2}\lambda + \frac{\hat{x}_s(\ell) + \varsigma}{2}.$$

It is well known that the Gauss quadrature formula $I_z(f) = \sum_{i=1}^z w_i^z f(\lambda_i^z)$ has degree of exactness $2z - 1$. Hence to justify the value of n and m in (2.10) we only need to figure out the degrees of the bivariate polynomial in the curly brackets of the fourth step in terms of λ and η . The degree of λ is q because $v(\lambda)$ is linear. The degree of η is $(\kappa - 1)(q + 2) - 1$ because degrees of the factors $\frac{\hat{x}_s(\eta) - \varsigma}{2}$, $\frac{d\hat{y}_s}{d\ell}(\eta)$, and $P(v(\lambda), \hat{y}_s(\eta))$ are $\kappa - 1$, $\kappa - 2$, and $q(\kappa - 1)$, respectively. Finally, the formulas for x_{sij} , y_{sj} , and w_{sij} follows from applying Gauss quadrature rules to the bivariate polynomial. \square

The main idea of the above proof is borrowed from Sommariva and Vianello [25], but there are several subtle differences between our Theorem 2.6 and theirs. First, our integral is a *line integral* over an algebraically closed curve while theirs is an integral over a planar area or a regular set in the plane. Second, our closed curve is slightly more general than the boundary of their planar area [25, Remark 4]; for example, there are no restrictions on intersections of the closed curve. Most importantly, any integral over a regular set in the plane can be expressed as a line integral in the LHS of (2.9) via an appropriate choice of $\dot{\gamma}_{\mathcal{D}}$ and a suitable definition of the interior of $\dot{\gamma}_{\mathcal{D}}$; but the converse is not true. This slightly better generality is furnished by the formulation of the integral as a *line integral*.

Note that, for linear splines ($\kappa = 2$), the number of quadrature nodes in (2.9) can be reduced by compression of multivariate discrete measures [26, 28].

Our new LFC algorithm is summarized in Algorithm 1, which mainly consists of two steps: constructing the generating curve by the procedure `generatingCurve` in Figure 2.4 and calculating the weighted sum of initial values of the scalar according to Theorem 2.6. The sign of the calculated flux entails some explanation.

Let \mathcal{B} and \mathcal{C} be two bases of a vector space V ; denote by $P_{\mathcal{C} \leftarrow \mathcal{B}}$ the change-of-basis matrix from \mathcal{B} to \mathcal{C} . The two bases \mathcal{B} and \mathcal{C} define the *same orientation* of V if and only if $\det P_{\mathcal{C} \leftarrow \mathcal{B}} > 0$; they define the *opposite orientation* of V if and only if $\det P_{\mathcal{C} \leftarrow \mathcal{B}} < 0$. An *orientation* of a vector space V is an equivalence class of bases of V under the equivalence relation of “defining the same orientation” of V . Hence any basis in an equivalence class indicates that orientation of V . By Definition 2.5, a counterclockwise rotation of the unit normal vector of \widetilde{LN} coincides with the unit tangent vector of \widetilde{LN} . Therefore, one can build a basis of the phase space by choosing $\mathbf{n}_{\widetilde{LN}}$ as the first in the list of basis vectors. This counterclockwise orientation agrees with the counterclockwise orientation of any Jordan curve. Physically, this means that any particle originally in a DR of positive indices will cross \widetilde{LN} in a direction such that the inner product of the two vectors in (2.11) is positive; see Figure 2.5.

The postcondition that our LFC algorithm is κ th-order accurate follows directly from the postconditions of the procedure `generatingCurve` and Theorem 2.6.

The distinguishing features of Algorithm 1 are summarized as follows.

Algorithm 1: Lagrangian flux calculation.

Input: a velocity field $\mathbf{u}(\mathbf{x}, t)$ and a scalar field $f(\mathbf{x}, t_0)$ that satisfy (1.2),
 a time interval $[t_0, t_0 + k]$,
 a fixed simple curve $\widetilde{LN} = \beta([0, 1])$,
 a length scale h ,
 a κ th-order ODE solver.

Preconditions : \mathbf{u} and f are κ times continuously differentiable;
 $\mathbf{n}_{\widetilde{LN}}$ is the same in that in Definition 2.5.

Output: a real number I_κ as an estimate of the Eulerian flux

$$(2.11) \quad I_E := \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f(s, t) \mathbf{u}(s, t) \cdot \mathbf{n}_{\widetilde{LN}} \, ds \, dt.$$

Postconditions: $|I_\kappa - I_E| = O(h^\kappa)$.

- 1 $\dot{\gamma}_D \leftarrow \text{generatingCurve } (\mathbf{u}, \beta, t_0, k, h, \kappa)$
 - 2 $I_\kappa \leftarrow I(\dot{\gamma}_D, f(\mathbf{x}, t_0))$ as in Theorem 2.6
 - 3 **return** I_κ
-

- Simplicity: neither self-intersections of the DR boundary nor the winding numbers are computed.
- High accuracy: results of $\kappa = 4, 6$ are much more accurate than those of $\kappa = 2$; see section 4.
- Excellent conditioning: (a) potential ill-conditioning caused by intersecting curve segments are avoided; (b) the linear system for spline fitting in the procedure `generatingCurve` is well-conditioned because the distance variation between adjacent knots are within two orders of magnitudes; (c) with explicit Runge–Kutta methods for time integration, the final formula (2.9) only involves function evaluation and weighted sums. It is also recommended that any implementation of Algorithm 1 avoid potential catastrophic cancellation of floating-point numbers by adopting special summation algorithms such as that in [19].
- Amenability to parallel computing: each step of Algorithm 1 can be divided into almost independent tasks so that the synchronization from local processors into the global result requires a minimum amount of time.

We note in passing that, when the scalar f is not smooth, it might be better to use a composite formula such as a Newton–Cotes formula rather than a Gauss formula with a high degree of exactness.

3. Analysis. The first three subsections contains prerequisites for analysis in subsequent subsections. In section 3.4, we derive two useful streakline identities, which further show that the generating curves form a homotopy class of moving loops. Consequently, the Reynolds transport theorem can be applied to the homotopy class of deforming DRs. Finally, the flux identities (1.5) and (1.4) are proved in section 3.5 by a method reminiscent of homology.

3.1. The Hopf theorem. A *path* in a topological space \mathcal{X} is a continuous map $\zeta : [0, 1] \rightarrow \mathcal{X}$. A *homotopy* of paths in \mathcal{X} is a family of paths $\gamma_t : [0, 1] \rightarrow \mathcal{X}$, $t \in [0, 1]$, such that the endpoints $\gamma_t(0)$, $\gamma_t(1)$ are independent of time and the associated map $H : [0, 1]^2 \rightarrow \mathcal{X}$ defined by $H(s, t) = \gamma_t(s)$ is continuous. The relation of homotopy on

paths with fixed endpoints in any space is an equivalence relation. The equivalence class of a path ζ under this equivalence relation is called the *homotopy class* of ζ . For two paths ζ_1 and ζ_2 satisfying $\zeta_1(1) = \zeta_2(0)$, the *composition* of ζ_1 and ζ_2 , denoted $\zeta_1 \cdot \zeta_2$, is defined as

$$(3.1) \quad (\zeta_1 \cdot \zeta_2)(s) := \begin{cases} \zeta_1(2s), & s \in [0, \frac{1}{2}], \\ \zeta_2(2s - 1), & s \in [\frac{1}{2}, 1]. \end{cases}$$

This composition preserves homotopy, i.e. $[\zeta_1 \cdot \zeta_2]$ remains a homotopy class [6, page 26].

A *loop* or *cycle* is a path whose endpoints coincide; this common endpoint is the *basepoint* of the loop. Two loops are homotopic in \mathcal{X} if they are homotopic in \mathcal{X} as paths; in this case, the basepoint of the loop homotopy must be fixed. In comparison, a *free homotopy* between two loops γ_1 and γ_2 in \mathcal{X} is a function $H_f : [0, 1]^2 \rightarrow \mathcal{X}$ such that $H_f(0, t) = \gamma_1(t)$ and $H_f(1, t) = \gamma_2(t)$ for all t , and $H_f(s, 0) = H_f(s, 1)$ for all s . Then γ_1 and γ_2 are said to be *freely homotopic* in \mathcal{X} . Clearly, any path homotopy between two loops can be interpreted as a free homotopy between them.

THEOREM 3.1 (Hopf theorem [8]). *Let a point $\mathbf{x} \in \mathbb{R}^2$ be given. Two closed curves γ_1 and γ_2 are freely homotopic in $\mathbb{R}^2 \setminus \{\mathbf{x}\}$ if and only if $w(\gamma_1, \mathbf{x}) = w(\gamma_2, \mathbf{x})$.*

Theorem 3.1 states that a closed curve can be continuously deformed to another *without crossing* \mathbf{x} in \mathbb{R}^2 if and only if $w(\gamma_1, \mathbf{x}) = w(\gamma_2, \mathbf{x})$. We emphasize that the underlying space of the free homotopy must be $\mathbb{R}^2 \setminus \{\mathbf{x}\}$ instead of \mathbb{R}^2 ; see [35, Figure 2] for an illustration.

3.2. Jacobian of flow maps. The following lemma is well known and its proof can be found in many texts such as [14, page 6].

LEMMA 3.2. *Let $\mathbf{u}(\mathbf{x}, t)$ be a velocity field that is continuous in time and continuously differentiable in space. Let $\phi_{t_0}^{+(t-t_0)}$ be the associated flow map with fixed initial time t_0 , and label a Lagrangian particle $X(\mathbf{x}, t) = \phi_{t_0}^{+(t-t_0)}(\mathbf{x})$ by its initial position \mathbf{x} . Denote by $J(\mathbf{x}, t)$ the Jacobian determinant of the flow map $\phi_{t_0}^{+(t-t_0)}(\mathbf{x})$. Then*

$$(3.2) \quad \forall t > t_0, \quad \frac{dJ}{dt}(\mathbf{x}, t) = J(\mathbf{x}, t)(\nabla \cdot \mathbf{u})(X(\mathbf{x}, t), t).$$

In particular, $J = 1$ for incompressible flows since $\frac{dJ}{dt} = 0$ and then $\phi_{t_0}^0$ is the identity operator.

LEMMA 3.3. *Let \mathcal{M} denote a bounded open set whose boundary consists of a countable number of pairwise disjoint Jordan curves. Let $f(\mathbf{x}, t)$ be an integrable scalar function satisfying*

$$(3.3) \quad \frac{d}{dt} \int_{\phi_{t_0}^{+(t-t_0)}(\mathcal{M})} f(X, t) dX = 0.$$

If $f(\mathbf{x}, t)$ is further differentiable, then (3.3) is equivalent to the scalar hyperbolic conservation law (1.2).

Proof. Change the integral variable in (3.3) so that the integral is over the fixed domain \mathcal{M} . The chain rule and Lemma 3.2 yield

$$\begin{aligned} \frac{d}{dt} \int_{\phi_{t_0}^{+(t-t_0)}(\mathcal{M})} f(X, t) dX &= \frac{d}{dt} \int_{\mathcal{M}} f(X(\mathbf{x}, t), t) J d\mathbf{x} \\ &= \int_{\mathcal{M}} \left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial X} \frac{dX}{dt} + f(\nabla \cdot \mathbf{u}) \right] J d\mathbf{x} \\ &= \int_{\phi_{t_0}^{+(t-t_0)}(\mathcal{M})} \left[\frac{\partial f}{\partial t} + \nabla \cdot (f \mathbf{u}) \right] dX = 0. \end{aligned}$$

Equation (1.2) must hold because t and \mathcal{M} are both arbitrary. \square

(3.3) is more general than (1.2) in that the scalar function f needs not to be differentiable in order for its integral on a moving region to be conserved.

3.3. Reynolds transport theorem on the homotopy class of moving regions. A *moving Jordan loop* is a homotopy class of loops

$$(3.4) \quad \dot{\Gamma} = \{\dot{\gamma}(\tau) : \tau \in [0, k]\},$$

where (i) each loop $\dot{\gamma}(\tau) : [0, 1] \rightarrow \mathbb{R}^2$ is an oriented piecewise smooth Jordan curve, (ii) the map associated with the homotopy is smooth in time, and (iii) all the Jordan curves have a consistent orientation.

A *moving region in the plane* is a family of regular subsets of \mathbb{R}^2 ,

$$(3.5) \quad \{\mathcal{M}(\tau) : \tau \in [0, k]\},$$

such that $\{\partial\mathcal{M}(\tau) : \tau \in [0, k]\}$ is a set of moving Jordan loops. The following theorem is also known as the multidimensional Leibniz integral rule [22].

THEOREM 3.4 (Reynolds transport theorem [16]). *Let $\{\mathcal{M}(\tau) : \tau \in [0, k]\}$ denote a moving region in the plane and $g(\cdot, \tau) : \mathcal{M}(\tau) \rightarrow \mathbb{R}$ a scalar function. We have*

$$(3.6) \quad \frac{d}{d\tau} \int_{\mathcal{M}(\tau)} g(\mathbf{x}, \tau) d\mathbf{x} = \int_{\mathcal{M}(\tau)} \frac{\partial g(\mathbf{x}, \tau)}{\partial \tau} d\mathbf{x} + \int_{\partial\mathcal{M}(\tau)} g(s, \tau) V_n(s, \tau) ds,$$

where $V_n(s, \tau)$ is the speed of $\partial\mathcal{M}(\tau)$ in the outward normal direction.

Proof. See [13] for a proof via differential forms and Stokes' theorem. \square

In proving Lemmas 3.2 and 3.3, each pair of elements in the moving region must have a bijective correspondence through the flow map. In the Reynolds transport theorem, however, the bijective correspondence need not hold for the moving region under consideration; this is a key of our proof of the flux identity in section 3.5.

3.4. The homotopy classes of generating curves. By (2.4), the backward streakline $\Psi_{t_0+k}^{-k}(M)$ contains the initial positions of all particles that will pass M within (t_0, t_0+k) . At $t = t_0 + \tau$ with $\tau \in (0, k)$, these particles can be classified into three types: those that will pass M within $(t_0 + \tau, t_0 + k)$, the one that coincides with M , and those that have passed M within $(t_0, t_0 + \tau)$. Under the mapping of the diffeomorphism $\phi_{t_0}^\tau$, these three sets of particles are $\Psi_{t_0+k}^{-(k-\tau)}(M)$, M , $\Psi_{t_0}^{+\tau}(M)$, respectively. Hence the image of the backward streakline $\Psi_{t_0+k}^{-k}(M)$ can be decomposed as

$$(3.7) \quad \forall \tau \in (0, k), \quad \phi_{t_0}^\tau (\Psi_{t_0+k}^{-k}(M)) = \Psi_{t_0+k}^{-(k-\tau)}(M) \cup M \cup \Psi_{t_0}^{+\tau}(M).$$

For two time increments $0 < r < s$, similar arguments yield

$$(3.8) \quad \phi_{t_0+r}^{-r} (\Psi_{t_0+s}^{-(s-r)}(M)) = \Psi_{t_0+s}^{-s}(M) \setminus \Psi_{t_0+r}^{-r}(M) \setminus \phi_{t_0+r}^{-r}(M).$$

The streakline inside the LHS parenthesis is at time $t_0 + s - (s - r) = t_0 + r$ while those streaklines on the RHS are at time $t_0 + s - s = t_0 + r - r = t_0$. Then the flow map $\phi_{t_0+r}^{-r}$ carries the former streakline to the RHS because they are exactly the same set of Lagrangian particles!

LEMMA 3.5. *The set of generating curves of DRs for a simple open curve*

$$(3.9) \quad \dot{\Gamma}_{\widetilde{LN}}(t_0, k) := \left\{ \gamma_{\mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} : \tau \in [0, k] \right\}$$

is a homotopy class in \mathbb{R}^2 .

Proof. Construct a composite path via concatenation as

$$\zeta_r := \left(L \cup \Psi_{t_0+k}^{-(k-rk)}(L) \cup \overleftarrow{L} \right) \cdot \left(\overleftarrow{L} \cup \phi_{t_0+k}^{-(k-rk)}(\widetilde{LN}) \cup \overleftarrow{N} \right) \cdot \left(\overleftarrow{N} \cup \Psi_{t_0+k}^{-(k-rk)}(N) \cup N \right).$$

By (3.7), the set of backward streaklines $\{\Psi_{t_0+k}^{-(k-rk)}(L) : r \in [0, 1]\}$ are the loci of a continuously diminishing set of particles that will pass L during the continuously shortened time intervals $\{[t_0 + rk, t_0 + k] : r \in [0, 1]\}$. The set of the preimages of \widetilde{LN} , $\{\phi_{t_0+k}^{-(k-rk)}(\widetilde{LN}) : r \in [0, 1]\}$, is also continuous with respect to r . Hence the map $H(s, r) = \zeta_r(s)$ is continuous, and $[\zeta_r]$ is a homotopy class. For the static path $\zeta_{\widetilde{LN}} := L \cup \widetilde{LN} \cup N$, the identity map $I(s, r) = \zeta_{\widetilde{LN}}(s)$ is clearly continuous, and $[\zeta_{\widetilde{LN}}]$ is also a homotopy class. Hence their composition paths $\dot{\Gamma}_{\widetilde{LN}} = \zeta_r \cdot \zeta_{\widetilde{LN}}$ form a homotopy class of loops. \square

Lemma 3.5 states that generating curves of DRs of the same family are a moving loop and consequently the corresponding DRs form a moving region.

3.5. The flux identities. A DR is *canonical* if its generating curve is Jordan. We prove the flux identities (1.5) and (1.4) in four steps as follows. This sequence reflects our opinion that (1.5) is more important and more useful than (1.4).

- The flux identity (1.4) for canonical DRs is proved in Lemma 3.6.
- The flux identity (1.5) for canonical DRs is derived in Corollary 3.7.
- The flux identity (1.5) for arbitrary DRs is proved in Theorem 3.8.
- The flux identity (1.4) for arbitrary DRs is derived in Theorem 3.9.

LEMMA 3.6 (the flux identity for canonical DRs). *Let f be a scalar function conserved by a nonautonomous flow $\mathbf{u}(\mathbf{x}, t)$ such that (1.2) holds. If the DR $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ for a fixed simple open curve \widetilde{LN} is canonical, then*

$$(3.10) \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt = m \int_{\mathcal{D}_{\widetilde{LN}}^m(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x},$$

where m is either -1 or 1 , and $\mathbf{n}_{\widetilde{LN}}$ is the unit normal vector of \widetilde{LN} in Definition 2.5.

Proof. By Lemma 3.5, the DRs $\{\mathcal{D}_{\widetilde{LN}}(t_0 + \tau, k - \tau) : \tau \in (0, k)\}$ form a moving region in the plane. For the case of $m = +1$, we have

$$\begin{aligned} & \frac{d}{d\tau} \int_{\mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} f(\mathbf{x}, t_0 + \tau) d\mathbf{x} \\ &= \int_{\mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} \frac{\partial f(\mathbf{x}, t_0 + \tau)}{\partial \tau} d\mathbf{x} + \int_{\partial \mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} f(s, t_0 + \tau) V_n(s, t_0 + \tau) ds \\ &= - \int_{\mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} \nabla \cdot (f \mathbf{u}) d\mathbf{x} + \int_{\partial \mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} f(s, t_0 + \tau) V_n(s, t_0 + \tau) ds \\ &= \int_{\partial \mathcal{D}_{\widetilde{LN}}(t_0+\tau, k-\tau)} f(s, t_0 + \tau) (V_n(s, t_0 + \tau) - \mathbf{u} \cdot \mathbf{n}) ds \\ &= - \int_{\widetilde{LN}} f(s, t_0 + \tau) \mathbf{u}(s, t_0 + \tau) \cdot \mathbf{n} ds, \end{aligned}$$

where the first step follows from the Reynolds transport theorem (Theorem 3.4), the second step from the hyperbolic conservation law (1.2), and the third step from the divergence theorem, $m = +1$, and Definition 2.5. It is important to distinguish the normal speed of the moving Jordan loop $\hat{\Gamma}_{\widetilde{LN}}(t_0, k)$ in (3.9) from those of the underlying flow. Because all Lagrangian particles in the initial DR $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ are passively advected by the velocity field \mathbf{u} , we have $V_n = \mathbf{u} \cdot \mathbf{n}$ for every point in $\partial\mathcal{D}_{\widetilde{LN}}(t_0 + \tau, k - \tau) \setminus \widetilde{LN}$. In comparison, $V_n = 0$ holds everywhere on the stationary curve \widetilde{LN} . Therefore, $V_n - \mathbf{u} \cdot \mathbf{n} = -\mathbf{u} \cdot \mathbf{n}$ holds on \widetilde{LN} , and $V_n - \mathbf{u} \cdot \mathbf{n} = \mathbf{0}$ holds everywhere else on $\partial\mathcal{D}_{\widetilde{LN}}(t_0 + \tau, k - \tau)$. In the last step, Definition 2.5 implies that the unit normal of \widetilde{LN} is exactly the same as the outward unit normal of $\mathcal{D}_{\widetilde{LN}}(t_0 + \tau, k - \tau)$. Finally, integrating the above equation in $[t_0, t_0 + k]$ yields

$$\begin{aligned} & \int_0^k \left[\frac{d}{d\tau} \int_{\mathcal{D}_{\widetilde{LN}}(t_0 + \tau, k - \tau)} f(\mathbf{x}, t_0 + \tau) d\mathbf{x} \right] d\tau = - \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f(s, t) \mathbf{u}(s, t) \cdot \mathbf{n}_{\widetilde{LN}} ds dt \\ & \Leftrightarrow \int_{\mathcal{D}_{\widetilde{LN}}(t_0 + k, 0)} f(\mathbf{x}, t_0 + k) d\mathbf{x} - \int_{\mathcal{D}_{\widetilde{LN}}(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x} = - \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt \\ & \Leftrightarrow \int_{\mathcal{D}_{\widetilde{LN}}(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x} = \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt, \end{aligned}$$

where the first step follows from the fundamental theorem of calculus and the second step from the fact that $\mathcal{D}_{\widetilde{LN}}(t_0 + k, 0) = \emptyset$.

The case of $m = -1$ can be proved by the same argument, the only difference being that the outward unit normal of the DR is now $-\mathbf{n}$ and $V_n = -\mathbf{u} \cdot \mathbf{n}$. \square

The sign of the Lagrangian integral on the RHS of (3.10) can also be taken care of by the orientation of the generating curve.

COROLLARY 3.7. *With the same conditions in Lemma 3.6, we have*

$$(3.11) \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt = \oint_{\gamma_{\mathcal{D}}} F(x, y, t_0) dy,$$

where $\gamma_{\mathcal{D}}$ is an oriented Jordan curve that generates the canonical DR $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ and

$$(3.12) \quad F(x, y, t_0) := \int_{\varsigma}^x f(v, y, t_0) dv$$

with ς being an arbitrary fixed real number.

Proof. By the fundamental theorem of calculus, we have $f(x, y) = \frac{\partial F}{\partial x}$. Since $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ is canonical, its generating curve is Jordan and equals its boundary. Hence we can apply Green's theorem to get

$$\int_{\mathcal{D}_{\widetilde{LN}}(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x} = \pm \oint_{\gamma_{\mathcal{D}}} F(x, y, t_0) dy,$$

where “ \pm ” is “ $+$ ” for $m = +1$ (i.e., $\gamma_{\mathcal{D}}$ is positively oriented) and is “ $-$ ” for $m = -1$ (i.e., $\gamma_{\mathcal{D}}$ is negatively oriented). The rest follows from Lemma 3.6. \square

The following theorem is the central analytic result of this paper.

THEOREM 3.8 (the flux identity for LFC). *Let f be a scalar function conserved by a nonautonomous flow $\mathbf{u}(\mathbf{x}, t)$ such that (1.2) holds. Then we have*

$$(3.13) \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt = \oint_{\gamma_{\mathcal{D}}} F(x, y, t_0) dy,$$

where γ_D is the oriented generating curve of the DR in Definition 2.4, $\mathbf{n}_{\widetilde{LN}}$ is the unit normal vector of \widetilde{LN} in Definition 2.5, and F is defined in (3.12).

Proof. As the main idea of the proof, we divide $[t_0, t_0 + k]$ and \widetilde{LN} into subintervals and subcurves so that the LHS of (3.13) is the sum of a number of subintegrals, each of which equals the integral of f over a sub-DR for some subcurve and some subinterval. As oriented Jordan curves, the generating curves of these sub-DRs concatenate to γ_D . The details are as follows.

First, for any given $k > 0$ and real numbers k_1, k_2, \dots, k_m ,

$$0 = k_0 < k_1 < k_2 < \dots < k_m < k_{m+1} = k,$$

denote by γ_i the generating curve of $\mathcal{D}_{\widetilde{LN}}(t_0 + k_i, k_{i+1} - k_i)$. For a sufficiently small time increment, we know from [35, Proposition 4.6] that the two streaklines of any DR do not intersect each other, neither do they self-intersect, nor do they intersect \widetilde{LN} and its preimage $\phi_{t_0+k}^{-k}(\widetilde{LN})$. Hence we can choose k_i 's such that either γ_i is Jordan or each self-intersection of γ_i belongs to $\phi_{t_0+k}^{-(k-k_i)}(\widetilde{LN}) \cap \phi_{t_0+k}^{-(k-k_{i+1})}(\widetilde{LN})$.

Second, images of the generating curves $\phi_{t_0+k_i}^{-k_i}(\gamma_i)$ for $i = 0, 1, \dots, m$ concatenate to γ_D , the generating curve of $\mathcal{D}_{\widetilde{LN}}(t_0, k)$. To see this, we calculate algebraically for a single image as

$$\begin{aligned} \phi_{t_0+k_i}^{-k_i}(\gamma_i) &= \phi_{t_0+k_i}^{-k_i} \left(\widetilde{LN} \cup \Psi_{t_0+k_{i+1}}^{-(k_{i+1}-k_i)}(N) \cup \phi_{t_0+k_{i+1}}^{-(k_{i+1}-k_i)}(\widetilde{NL}) \cup \Psi_{t_0+k_{i+1}}^{-(k_{i+1}-k_i)}(L) \right) \\ &= \phi_{t_0+k_i}^{-k_i} \left(\widetilde{LN} \right) \cup \left(\Psi_{t_0+k_{i+1}}^{-(k_{i+1}-k_{i-1})}(N) \setminus \Psi_{t_0+k_i}^{-(k_i-k_{i-1})}(N) \right) \\ &\quad \cup \phi_{t_0+k_{i+1}}^{-k_{i+1}} \left(\widetilde{NL} \right) \cup \left(\Psi_{t_0+k_{i+1}}^{-(k_{i+1}-k_{i-1})}(L) \setminus \Psi_{t_0+k_i}^{-(k_i-k_{i-1})}(L) \right), \end{aligned}$$

where the first step follows from (2.7) and the second step follows from (3.8) and a group property of the flow map, i.e., $\phi_{t_0+k_i}^{-k_i} \circ \phi_{t_0+k_{i+1}}^{-(k_{i+1}-k_i)} = \phi_{t_0+k_{i+1}}^{-k_{i+1}}$. To avoid cluttering, we have ignored single points in the above derivation. It follows that the concatenation $\gamma_0 \sqcup \phi_{t_0+k_1}^{-k_1}(\gamma_1) \sqcup \dots \sqcup \phi_{t_0+k_m}^{-k_m}(\gamma_m)$ is

$$\begin{aligned} &\widetilde{LN} \sqcup \Psi_{t_0+k_1}^{-k_1}(N) \sqcup \phi_{t_0+k_1}^{-k_1} \left(\widetilde{NL} \right) \sqcup \Psi_{t_0+k_1}^{-k_1}(L) \\ \sqcup &\phi_{t_0+k_1}^{-k_1} \left(\widetilde{LN} \right) \sqcup \Psi_{t_0+k_2}^{-k_2}(N) \setminus \Psi_{t_0+k_1}^{-k_1}(N) \sqcup \phi_{t_0+k_2}^{-k_2} \left(\widetilde{NL} \right) \sqcup \Psi_{t_0+k_2}^{-k_2}(L) \setminus \Psi_{t_0+k_1}^{-k_1}(L) \\ \sqcup &\dots \\ \sqcup &\phi_{t_0+k_m}^{-k_m} \left(\widetilde{LN} \right) \sqcup \Psi_{t_0+k}^{-k}(N) \setminus \Psi_{t_0+k_m}^{-k_m}(N) \sqcup \phi_{t_0+k}^{-k} \left(\widetilde{NL} \right) \sqcup \Psi_{t_0+k}^{-k}(L) \setminus \Psi_{t_0+k_m}^{-k_m}(L) \\ = &\widetilde{LN} \sqcup \Psi_{t_0+k}^{-k}(N) \sqcup \phi_{t_0+k}^{-k} \left(\widetilde{NL} \right) \sqcup \Psi_{t_0+k}^{-k}(L); \end{aligned}$$

see Figures 3.1 and 2.7 for two illustrations. Here we have used the notation “ \sqcup ” to emphasize the homological convention that two curves concatenate to the empty set if they have the same image points but different orientations.

Third, self-intersections of γ_i and their preimages can be exploited to divide γ_i into the concatenation of a number of closed curves $\gamma_{i,0}, \gamma_{i,1}, \dots, \gamma_{i,r_i}$. Each of these closed curves might be degenerate, i.e., its complement does not contain any bounded connected component. In addition, it might contain a whisker, a special type of 1D self-intersection where each self-intersection point has an even degree. These whisker arcs do not have any contribution to the line integral as the even degree causes the line integral to cancel. Remove the degenerate parts from the list of $\gamma_{i,j}$'s, and we have a number of pairwise almost disjoint Jordan curves, each of which generates a sub-DR. See subplot (a) of Figure 3.1 and [34, Figure 4.3 and Figure 4.4] for illustrations.

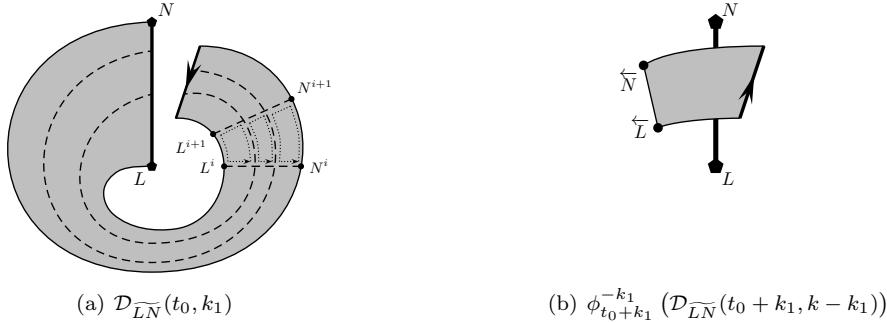


FIG. 3.1. Decomposing the DR $\mathcal{D}_{\overline{LN}}(t_0, k)$ in Figure 2.5(c) into a canonical DR and the image of another canonical DR, respectively shown in the two subplots. The two closed curves concatenate to the generating curve of $\mathcal{D}_{\overline{LN}}(t_0, k)$ due to the different orientations of the two thick arrowed line segments. In subplot (a), dashed line segments represent preimages of \overline{LN} , dashed curves represent streaklines seeded at breakpoints on \overline{LN} , and dotted closed curves with arrows represent oriented generating curves of sub-DRs. $L^i := \phi_{t_0+k}^{-(k-k_i)}(L)$.

Lastly, we have

$$\begin{aligned} \oint_{\gamma_D} F &= \sum_{i=0}^m \oint_{\gamma_i} F = \sum_{i=0}^m \sum_{j=0}^{r_i} \oint_{\gamma_{i,j}} F \\ &= \sum_{i=0}^m \sum_{j=0}^{r_i} \oint_{\phi_{t_0+k_i}^{-(k-k_i)}(\gamma_{i,j})} F \\ &= \sum_{i=0}^m \sum_{j=0}^{r_i} \int_{t_0+k_i}^{t_0+k_{i+1}} \int_{\widetilde{L_jN_j}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{L_jN_j}} ds dt \\ &= \sum_{i=0}^m \int_{t_0+k_i}^{t_0+k_{i+1}} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt \\ &= \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt, \end{aligned}$$

where the first line follows from the second and third steps, the second line from Lemma 3.3, and the third line from Corollary 3.7 and the construction $L = L_0$, $N_j = L_{j+1}$, and $N = N_{r_i}$. \square

Theorem 3.8 yields Theorem 3.9, which has been proved in [7] with a different approach.

THEOREM 3.9 (the flux identity for arbitrary DRs). *With the same conditions in Theorem 3.8, we have*

$$(3.14) \quad \forall k > 0, \quad \int_{t_0}^{t_0+k} \int_{\widetilde{LN}} f \mathbf{u} \cdot \mathbf{n}_{\widetilde{LN}} ds dt = \sum_{n \in \mathbb{Z}} n \int_{\mathcal{D}_{\overline{LN}}^n(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x},$$

where $\mathcal{D}_{\overline{LN}}^n(t_0, k)$ is the DR of index n in Definition 2.4.

Proof. By the third step in the proof of Theorem 3.8, we can assume that γ_D is an oriented closed curve with a finite number of self-intersection points. Then it follows from [35, Proposition 2.5] that γ_D can be decomposed into a set Γ_D of oriented, pairwise noncrossing Jordan curves via cycle decomposition of an oriented closed curve, which is also known as the Seifert decomposition [23]. A Jordan curve γ_k is said to *include* γ_ℓ if the bounded complement of γ_ℓ is a subset of that of γ_k . This inclusion relation is a partial order on Γ_D and can be utilized to sort the set into a forest of trees.

A positively and negatively oriented Jordan curve is said to have *sign* +1 and -1, respectively. For each Jordan curve $\gamma_k \in \Gamma_D$, let n_k denote the sum of the signs of

oriented Jordan curves in $\Gamma_{\mathcal{D}}$ that includes γ_k . Then we know from Corollary 3.7 and Lemma 3.3 that each Jordan curve $\gamma_k \in \Gamma_{\mathcal{D}}$ generates a sub-DR \mathcal{D}_k associated with some subcurve of \widetilde{LN} and some subinterval of $[t_0, t_0 + k]$. Lemma 3.6 further implies $\oint_{\gamma_k} F = \text{sign}(\gamma_k) \int_{\mathcal{D}_k} f$. Consequently, we have

$$\oint_{\gamma_{\mathcal{D}}} F = \sum_{\gamma_k \in \Gamma_{\mathcal{D}}} \oint_{\gamma_k} F = \sum_{\gamma_k \in \Gamma_{\mathcal{D}}} \text{sign}(\gamma_k) \int_{\mathcal{D}_k} f = \sum_{n \in \mathbb{Z}} n \int_{\mathcal{D}_{\widetilde{LN}}^n(t_0, k)} f(\mathbf{x}, t_0) d\mathbf{x},$$

where the last step follows from the inclusion relation and Definition 2.2. In other words, the integral on the sub-DR generated by any Jordan curve that is included in other Jordan curves has to be counted multiple times according to the index of the sub-DR. The rest of the proof follows from Theorem 3.8. \square

4. Numerical tests. In this section we perform a variety of numerical tests for incompressible and compressible flows, for regular and chaotic flows, for normal and nonnormal DRs, and for DRs with degree 1 or higher. Results of these tests confirm the validity of Theorem 3.8 and demonstrate the high accuracy and efficiency of Algorithm 1.

We define the *relative LFC error* of Algorithm 1 as

$$(4.1) \quad E_{\kappa}(h) := \left| \frac{I_{\kappa}(h) - I_E}{I_E} \right|$$

and the corresponding *LFC convergence rate* as

$$(4.2) \quad \mathcal{O}_{\kappa}(h) = \log_r \frac{E_{\kappa}(rh)}{E_{\kappa}(h)},$$

where h is the length scale as an input parameter for Algorithm 1 and we set the refinement factor $r = 2$ in this work.

In section 2.6, we already showed that the convergence rate of Algorithm 1 is an even integer κ ; here we confirm this by numerical results for $\kappa = 2, 4, 6$. Of course the splines need to be linear, cubic, and quintic, respectively, and we employ as ODE solvers of matching orders the modified Euler method [11, page 155], the classical fourth-order Runge–Kutta method [5, page 138], and Verner’s method of order 6(5) [29], respectively. These ODE solvers are chosen for the sole reason of easy implementation; other time integrators with the same order of accuracy and larger stability regions would yield qualitatively the same results.

As another important detail, the degree q of the bivariate polynomial $P(x, y)$ in Theorem 2.6 is set to be higher than the input parameter κ of Algorithm 1, so that the approximation error of the scalar function by $P(x, y)$ is not the leading term. This helps us to focus on the main purpose of this section.

4.1. A canonical DR in exponentially divergent flow. The velocity field of this compressible flow and the passively advected scalar function are given as

$$(4.3) \quad \mathbf{u}(x, y, t) = \begin{bmatrix} \exp(x) - 1 - t \\ \exp(y) + t \end{bmatrix},$$

$$(4.4) \quad f(x, y, t) = \exp(-(x + y + t)).$$

It is easily checked that \mathbf{u} and f satisfy the scalar conservation law (1.2).

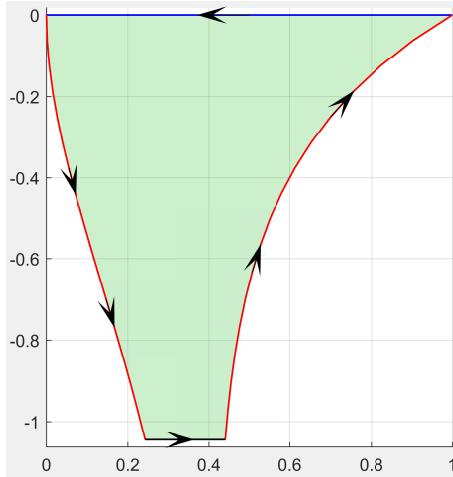


FIG. 4.1. The DR and its generating curve in the exponentially divergent flow (4.3) constructed by our algorithm in Figure 2.4 with $[t_0, t_0 + k] = [0, 1]$, $\kappa = 2$, $h = \frac{1}{8}$, and \overline{LN} being the blue line connecting $(1, 0)$ and $(0, 0)$. The streaklines and the preimage of \overline{LN} are represented by the red curves and the black line segment, respectively. The shaded region represents $D_{\overline{LN}}^{+1}(t_0, k)$.

TABLE 4.1
Errors and convergence rates of Algorithm 1 for the exponentially divergent test in Figure 4.1.

κ	$E_\kappa(\frac{1}{16})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{32})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{64})$
2	7.73e-04	2.04	1.87e-04	2.02	4.62e-05
4	9.84e-07	4.03	6.03e-08	4.02	3.72e-09
6	1.44e-09	6.48	1.61e-11	6.40	1.91e-13

For the line segment that connects $L = (1, 0)$ and $N = (0, 0)$, we use Algorithm 1 to calculate I_κ as an estimate of the flux integral I_E ; see Figure 4.1 for the corresponding generating curve and DR. As shown in Table 4.1, all of the desired convergence rates $\kappa = 2, 4, 6$ are achieved. In particular, the LFC errors of cubic ($\kappa = 4$) and quintic ($\kappa = 6$) splines are smaller than those of linear polygons ($\kappa = 2$) by orders of magnitude.

4.2. A nonnormal DR of degree 1 in an incompressible flow. The velocity field of this test

$$(4.5) \quad \mathbf{u}(x, y, t) = \begin{bmatrix} -y \\ x - t + 5 \end{bmatrix}$$

has already been used in Figure 2.6 to generate a DR within the time interval $[t_0, t_0 + k] = [0, 10]$ for the line segment connecting $L = (0, 0)$ and $N = (1, 0)$. As shown in Figure 2.6, the streakline $\Psi_{t_0+k}^{-k}(L)$ not only is self-intersecting but also intersects the other streakline $\Psi_{t_0+k}^{-k}(N)$. Hence the resulting DR is indeed non-normal.

The scalar function is chosen as

$$(4.6) \quad f(x, y, t) = [(x - t + 5) \cos t + (y + 1) \sin t - 5]^2 + [(y + 1) \cos t - (x - t + 5) \sin t - 1]^2,$$

and it is easily verified that (4.6) and (4.5) satisfy the scalar conservation law (1.2).

We set values of the length scale h of Algorithm 1 to be much smaller than those in the previous test. The corresponding LFC errors and convergence rates are listed in Table 4.2, where the second-order and fourth-order convergence rates are verified.

In the case of $\kappa = 6$, results in Table 4.2 shows a temporary deterioration of convergence rates on the first three grids; this is probably caused by the C^1 discontinuities (i.e., the cusps) of the streaklines in Figure 2.6 and our lack of special treatments for these cusps. We defer a thorough investigation and treatment of this accuracy deterioration for $\kappa = 6$ to future research.

4.3. Normal DRs of degree greater than 1 in compressible flows. In this test, the velocity field and the passively advected scalar function are

$$(4.7) \quad \mathbf{u}(x, y, t) = \begin{bmatrix} x + 2\pi y \\ -2\pi x + y \end{bmatrix},$$

$$(4.8) \quad f(x, y, t) = (x^2 + y^2)e^{-4t}.$$

It is easily checked that $\nabla \cdot \mathbf{u} \neq 0$ and that \mathbf{u} and f satisfy the scalar conservation law (1.2).

For time intervals $[t_0, t_0+k] = [0, 1.25]$, $[0, 2.25]$ and the line segment that connects $L = (1, 0)$ and $N = (0, 0)$, we plot two corresponding DRs in Figure 4.2. Due to the

TABLE 4.2
Errors and convergence rates of LFC Algorithm 1 for the nonnormal DR test of section 4.2;
see Figure 2.6 for the corresponding DR.

κ	$E_\kappa(\frac{1}{64})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{128})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{256})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{512})$
2	7.54e-03	2.01	1.88e-03	2.03	4.60e-04	2.02	1.13e-04
4	9.44e-06	4.01	5.88e-07	4.17	3.28e-08	3.92	2.17e-09
6	8.18e-07	4.32	4.10e-08	-0.75	6.90e-08	5.74	1.29e-09

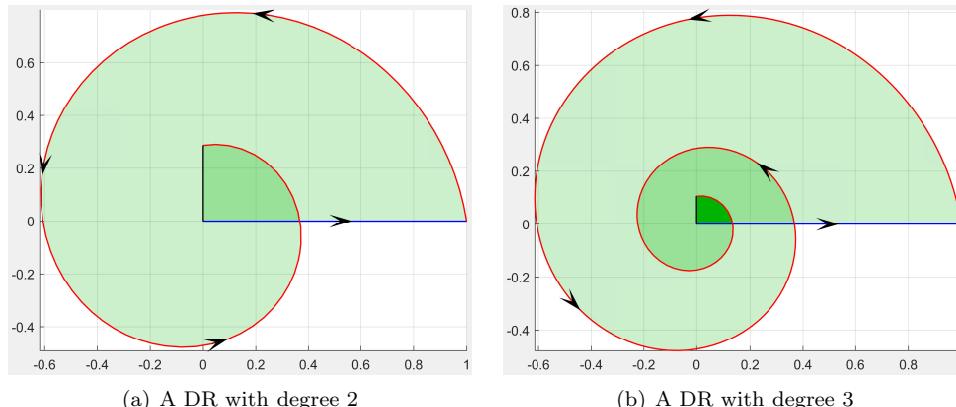


FIG. 4.2. DRs in the compressible flow (4.7) constructed by our algorithm in Figure 2.4 with $\kappa = 2$, $h = \frac{1}{8}$, and \widehat{LN} being the blue line segment connecting $L = (0, 0)$ and $N = (1, 0)$. The time interval is $[0, 1.25]$ for subplot (a) and $[0, 2.25]$ for subplot (b). The streaklines and the preimage of \widehat{LN} are represented by the red curves and the black line segment, respectively. Note that the streakline seeded at L degenerates to the point L since the velocity is a constant zero at L . The light green, green, and dark green regions represent D_{LN}^{+1} , D_{LN}^{+2} , and D_{LN}^{+3} , respectively.

TABLE 4.3

Errors and convergence rates of LFC Algorithm 1 for tests in section 4.2 where the DRs have degrees greater than one.

κ	$E_\kappa(\frac{1}{64})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{128})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{256})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{512})$
DRs with degree 2, $[t_0, t_0 + k] = [0, 1.25]$							
2	3.56e-03	1.73	1.07e-03	1.88	2.91e-04	1.95	7.56e-05
4	6.12e-06	3.87	4.18e-07	3.94	2.72e-08	3.97	1.74e-09
6	2.62e-10	6.03	4.00e-12	6.03	6.13e-14	4.11	3.56e-15
DRs with degree 3, $[t_0, t_0 + k] = [0, 2.25]$							
2	7.94e-03	1.36	3.10e-03	1.76	9.13e-04	1.90	2.45e-04
4	5.70e-05	3.74	4.27e-06	3.89	2.88e-07	3.95	1.87e-08
6	9.66e-09	6.09	1.42e-10	6.03	2.16e-12	6.10	3.15e-14

singular velocity at N , the streakline $\Psi_{t_0+k}^{-k}(N)$ reduces to the singleton point set $\{N\}$. In contrast, the large magnitude of the velocity field away from N leads to the long spiraling streakline $\Psi_{t_0+k}^{-k}(L)$, and consequently the DRs have degrees greater than 1; see Figure 4.2.

4.3.1. Results of LFC. We input four small length scales into Algorithm 1 to resolve fine details of the long streakline seeded at L . In Table 4.3, all the desired convergence rates $\kappa = 2, 4, 6$ are achieved. In particular, the LFC error for the degree-2 DR on the finest grid is very close to machine precision. Apart from explaining the slightly reduced convergence rate “4.11,” this extremely small error strongly supports the discussion in section 2 that our LFC Algorithm 1 has excellent conditioning.

Test results in Tables 4.1, 4.2, and 4.3 confirm the validity of Theorem 3.8 and Algorithm 1. In addition, we observe that our LFC algorithm based on cubic or quintic splines is much more accurate than the one based on linear polygons.

4.3.2. An accuracy and efficiency comparison of LFC with LFC-HK. Hofherr and Karrasch [7, section 4] have tested their algorithm LFC-HK with the same velocity field in (4.7) and a scalar field slightly different from (4.8). Their results on relative errors and CPU time [7, Table 1] provide a baseline for comparing LFC to LFC-HK with respect to accuracy and efficiency.

As shown in Table 4.4, LFC has better efficiency and accuracy than LFC-HK. In the third line, the CPU time of LFC is less than half of that of LFC-HK, yet the error of LFC is 200 times smaller than that of LFC-HK. From the third line to the fourth line, the CPU time increases by a factor over 8 for LFC-HK, yet its error is reduced less than a factor of 2. As for the LFC algorithm, an increase of CPU time by a factor of 3.2 leads to a reduction of the error by a factor of 67.

In both LFC-HK and our LFC algorithm, the most time-consuming part is the construction of the generating curves (especially the streaklines) by repeatedly calling a time integrator. Two reasons for the better cost-effectiveness of our LFC algorithm are Gauss quadrature formulas and the cubic/quintic splines. Furthermore, the LFC-HK algorithm appears to have difficulty in continuing its (approximate) second-order convergence rate once its errors get close to 10^{-7} . In contrast, our LFC algorithm does not show accuracy deterioration, thanks to its excellent conditioning; see the corresponding discussion in section 2.6.

4.3.3. An accuracy and efficiency comparison of LFC to FV-MOL. In this subsection we demonstrate (UFI.3) mentioned in section 1. To compute the Eulerian flux over a time interval $[t_0, T]$ through a line segment \widetilde{LN} inside a domain,

TABLE 4.4

Results of CPU time and relative errors of LFC-HK and LFC for the test of (4.7) and (4.8). For our MATLAB code of Algorithm 1, we use $\kappa = 6$ and set the spatial grid size to $\frac{1}{N}$ and the time step size to $\frac{2.25}{N}$, where N is the number of segments approximating \widetilde{LN} . For LFC-HK, we download the MATLAB code [7, page 542] from GitHub and run it on the same laptop to produce the results in the first three columns. See [7, page 542] for the meaning of the distance threshold. All tests are performed on a single core of an Intel Xeon E5 – 2698 v3 2.30GHz processor; in particular, parallel computing is disabled for both algorithms. The reader is invited to reproduce the results using our demo code freely available at <https://github.com/wdachub/LFC2D>.

LFC-HK			LFC (Algorithm 1)		
Distance threshold	Relative error	CPU time (s)	N	Relative error	CPU time (s)
0.01	2.25e-05	1.14	20	1.52e-05	0.32
0.005	8.63e-06	1.34	40	1.79e-07	0.62
0.001	4.93e-07	4.52	80	2.46e-09	1.80
0.0001	2.76e-07	36.54	160	3.67e-11	5.82

one can adopt the FV-MOL approach to solve the scalar conservation law (1.2) in $[t_0, T]$ over the *entire* domain, with the domain partitioned in a way such that \widetilde{LN} is always the union of a number of cell faces; then the Eulerian flux through \widetilde{LN} can be estimated as a numerical quadrature in time by a Newton–Cotes formula with its temporal nodal values as spatial sums of face-averages of $f\mathbf{u} \cdot \mathbf{n}$. Clearly, the efficiency of this FV-MOL approach [36] depends heavily on the ratio of the length scale of \widetilde{LN} over that of the domain. As the ratio gets smaller and smaller, FV-MOL becomes less and less efficient in that most CPU time is spent on irrelevant regions; this is demonstrated by the first three rows of Table 4.5.

In comparison, the CPU time and relative errors of our LFC algorithm remain the same while the domain grows larger and larger in Table 4.5. This is not surprising since no information on the DR of index zero is used. We conclude that the proposed LFC algorithm is very flexible and efficient for fluxes calculation in *local* regions.

Another method to calculate the Eulerian flux on the LHS of (1.4) and (1.5) is to scatter a number of particles at the initial time, to advect them, to calculate the instant of their crossing \widetilde{LN} , and to approximate the integral with the trapezoidal rule. Hofherr and Karrasch [7, page 538] have already shown that this Eulerian method of flux calculation is inferior to LFC-HK in terms of efficiency.

4.4. The oscillating vortex pair (OVP) flow for lobe dynamics. We have been focusing on the demonstration of the high accuracy and efficiency of the proposed LFC algorithm with regular flows. In this section we further illustrate utilities of our LFC algorithm, in particular (UFI.5), for more realistic flows with heterogeneous mixing, which are chaotic in local regions and regular in others. After all, it is in these cases that a partition of the phase space is mostly valued.

Following [17], we study the OVP flow that consists of a pair of oscillating vortices and an ambient strain-rate field:

$$(4.9) \quad \begin{cases} \frac{dx}{dt} = - \left[\frac{y-y_v}{(x-x_v)^2 + (y-y_v)^2} - \frac{(y+y_v)}{(x-x_v)^2 + (y+y_v)^2} \right] - \nu_v + \frac{\epsilon x}{\gamma} \sin \frac{t}{\gamma}; \\ \frac{dy}{dt} = (x-x_v) \left[\frac{1}{(x-x_v)^2 + (y-y_v)^2} - \frac{1}{(x-x_v)^2 + (y+y_v)^2} \right] - \frac{\epsilon y}{\gamma} \sin \frac{t}{\gamma}, \end{cases}$$

where ϵ is the strain rate, γ indicates the strength of the vortices, $\nu_v = \frac{\epsilon^\epsilon}{2I_0(\epsilon)}$ is the averaged velocity of the vortex pair, $I_0(\epsilon)$ is the modified Bessel function of order zero, and $(x_v(t), \pm y_v(t))$ are the vortex positions,

TABLE 4.5

A comparison of FV-MOL and LFC with respect to CPU time in computing the flux through a static line segment \overline{LN} inside a growing domain $[0, L_D]^2$. The fixed points $L = (1, 1)$, $N = (1, 2)$, the fixed time interval $[0, 0.1]$, the velocity in (4.7), and the initial value of scalar function f in (4.8), i.e., $f_0 := f(0) = x^2 + y^2$, are known to both methods. We use the fourth-order FV-MOL method [36] on the square computation domain $[0, L_D]^2$ to solve the scalar conservation law (1.2) with Dirichlet conditions from (4.8). Since the grid size and time step size are fixed at $\frac{1}{8}$ and $\frac{1}{30}$, \overline{LN} is always the union of a number of cell faces; thus the flux through \overline{LN} over the time interval can be estimated in an Eulerian manner by a spatial sum of the face-averages of $f\mathbf{u} \cdot \mathbf{n}$ and a numerical quadrature in time by Simpson's $\frac{3}{8}$ rule. For LFC, we invoke Algorithm 1 with $\kappa = 4$ and $h = \frac{1}{4}$. Results in this table can be reproduced using our demo code freely available at <https://github.com/wdachub/LFC2D>.

	CPU time (s)	Relative error
FV-MOL $L_D = 16$	0.24	3.94e-06
FV-MOL $L_D = 32$	1.07	3.94e-06
FV-MOL $L_D = 64$	3.60	3.94e-06
LFC (Algorithm 1)	0.018	4.39e-06

$$(4.10) \quad \begin{cases} x_v(t) = \frac{1}{2}\gamma e^{\epsilon - \epsilon \cos(\frac{t}{\gamma})} \int_0^{\frac{t}{\gamma}} [1 - 2\nu_v e^{\epsilon \cos(s) - \epsilon}] ds; \\ y_v(t) = e^{\epsilon \cos(\frac{t}{\gamma}) - \epsilon}. \end{cases}$$

For the special case of $\epsilon = 0$, we have $x_v = 0$, $y_v = 1$, and $\nu_v = \frac{1}{2}$. When ϵ is small, the last terms in (4.9) can be interpreted as a perturbation to the velocity field. Overall, the flow contains a regular free flow region, two vortex cores, and a chaotic mixing region. Throughout this work, we use $\epsilon = 0.1$ and $\gamma = 0.5$.

In Figure 4.3 we plot the generating curves and DRs for varying cases of \widetilde{LN} . Lengths of the associated time intervals are carefully selected so that the main features are not too complicated yet are still sufficiently interesting. The lobes in Figure 4.3(a) are qualitatively similar to those shown in [17, Figures 3 and 20]. The chaotic flow pattern in the mixing region is illustrated in Figure 4.3(b). If we shift \widetilde{LN} of Figure 4.3(b) slightly left, we obtain Figure 4.3(c), where a streakline has a large portion of self-overlap around the vortex. The chaotic particle motion is very well illustrated in Figure 4.3(d), which is simply a zoom-in of Figure 4.3(c) at a neighborhood of N .

A number of observations follow from Figure 4.3. First, the generating curves contains much of the information on stable and unstable manifold in the lobe dynamics discussed in detail by Rom-Kedar and colleagues [17]. Second, it is probably not a good idea for an LFC algorithm to calculate curve intersections in realistic flows, particularly when two long segments of the generating curve coincide (or are very close to) each other, as in the cases of Figure 4.3(a) and (c). Lastly, the partition of the plane by the generating curve into DRs with different indices [35, 7] is valuable in determining the direction and the number of a particle crossing LN . As a confirmation of (UFI.5), the coloring of different DRs according to their indices vividly illustrates the flow dynamics in a meaningful manner. Although self-intersections of generating curves still have to be calculated in coloring the DRs, the regions associated with ill-conditioned self-intersections are hardly visible, and thus the potential accuracy deterioration from intersecting segments is not an issue in picture plotting.

We also test the accuracy of our LFC algorithm for the case of Figure 4.3(c) and list resulting errors and convergence rates in Table 4.6. Thanks to the absence of curve intersection in Algorithm 1, we obtain high-order convergence rates in most

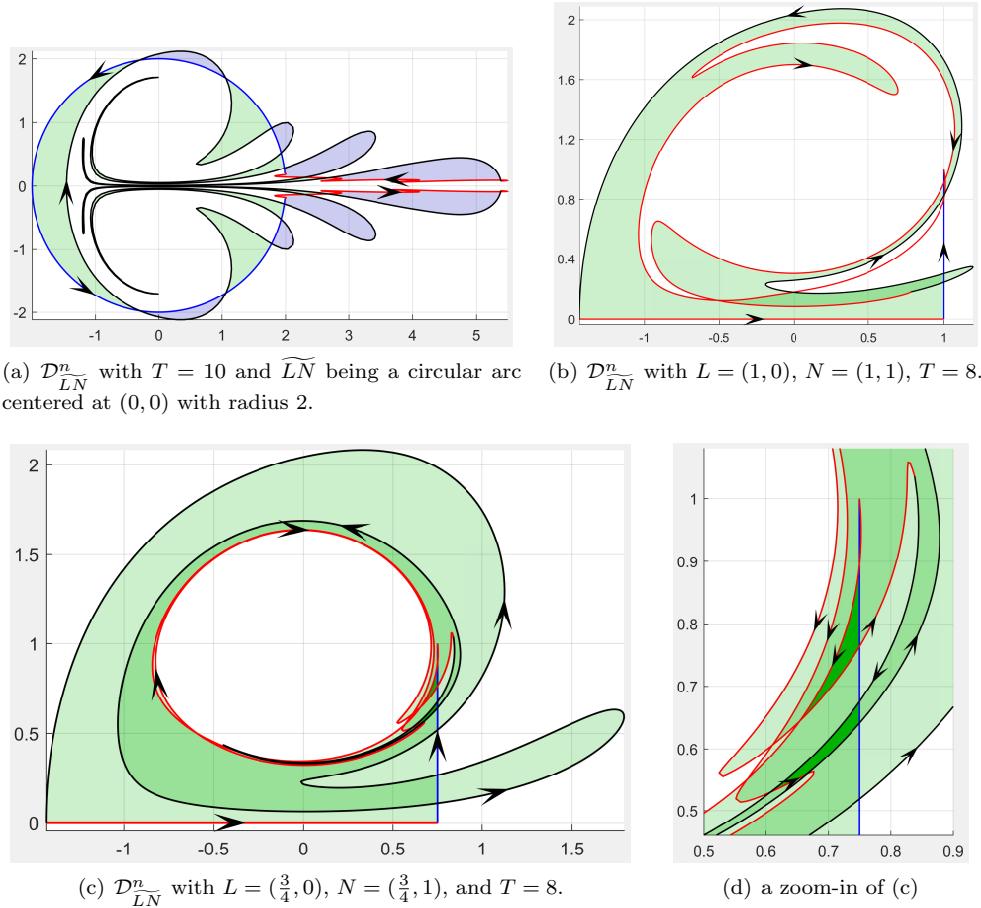


FIG. 4.3. Generating curves and DRs of the OVP flow (4.9) over the time interval $[0, T]$ with $\epsilon = 0.1$ and $\gamma = 0.5$. The blue curve represents \widehat{LN} , the black curve represents its preimage, and the red curves represent the streaklines. The arrows indicate the orientation of the generating curve. The light blue, white, light green, green, and dark green regions represent $\mathcal{D}_{\widehat{LN}}^{-1}(t_0, k)$, $\mathcal{D}_{\widehat{LN}}^0(t_0, k)$, $\mathcal{D}_{\widehat{LN}}^{+1}(t_0, k)$, $\mathcal{D}_{\widehat{LN}}^{+2}(t_0, k)$, and $\mathcal{D}_{\widehat{LN}}^{+3}(t_0, k)$, respectively.

TABLE 4.6

Relative errors and convergence rates of Algorithm 1 for flux calculation in the case of Figure 4.3(c) with (4.9) and $f(x, y, t) \equiv 1$. Here $E_\kappa(2h, h)$ denotes the absolute difference between the results of our LFC Algorithm 1 for the two input parameters $2h$ and h . Accordingly, the convergence rates here should be interpreted in the sense of Richardson extrapolation. The superconvergence for $\kappa = 4$ is probably due to the simple scalar $f \equiv 1$ and the generating curve being fully resolved by cubic splines. On the other hand, the order reduction for $\kappa = 6$ may be caused by the cusps in subplot (d).

κ	$E_\kappa(\frac{1}{20}, \frac{1}{40})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{40}, \frac{1}{80})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{80}, \frac{1}{160})$	\mathcal{O}_κ	$E_\kappa(\frac{1}{160}, \frac{1}{320})$
4	9.03e-02	5.06	2.70e-03	5.60	5.58e-05	8.49	1.55e-07
6	5.94e-02	7.44	3.42e-04	5.21	9.20e-06	2.74	1.37e-06

cases. The results in Table 4.6 suggest that it is practical and accurate to use our LFC algorithm with $\kappa = 4$ for realistic flows.

5. Conclusion. By extending our previous work on DRs [34, 32, 35], we prove a flux identity for scalar conservation laws and propose an LFC algorithm for calculating flux integrals through a simple static curve for a time interval that can be arbitrarily long. Our LFC algorithm requires no evolution information of the scalar function during the time interval except at the very initial time. Thanks to the generality of the flux identity, our LFC algorithm computes neither segment intersection nor winding numbers of sub-DRs. Other notable advantages include its simplicity, flexibility, excellent conditioning, efficiency, and high accuracy. All numerical results presented in this paper can be reproduced using the demo MATLAB package freely available at <https://github.com/wdachub/LFC2D>.

We are currently working on an algorithm for calculating fluxes through moving curves, via a direct augmentation of the proposed LFC algorithm. In future research, we will provide a computational framework and a software package for LFCs on static and moving manifolds with boundaries.

Acknowledgments. Qinghai Zhang thanks Prof. James Keener at the University of Utah, who pointed out a mistake in the early stage of this research. The authors also thank two anonymous referees, whose comments improved the quality of the manuscript.

REFERENCES

- [1] J. W. ALEXANDER, *Topological invariants of knots and links*, Trans. Amer. Math. Soc., 30 (1928), pp. 275–306.
- [2] S. BALASURIYA, *Transport between two fluids across their mutual flow interface: The streakline approach*, SIAM J. Appl. Dyn. Syst., 16 (2017), pp. 1015–1044.
- [3] R. COMMIL, J. SPANGENBERG, AND J. H. HATTEL, *Cellwise conservative unsplit advection for the volume of fluid method*, J. Comput. Phys., 283 (2015), pp. 582–608.
- [4] R. FJORTOFT, *On a numerical method of integrating the barotropic vorticity equation*, Tellus, 4 (1952), pp. 179–194, doi:10.1111/j.2153-3490.1952.tb01003.x.
- [5] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I: Non-stiff Problems*, 2nd ed., Springer-Verlag, Berlin, 1993.
- [6] A. HATCHER, *Algebraic Topology*, Cambridge University Press, Cambridge, UK, 2001.
- [7] F. HOFHERR AND D. KARRASCH, *Lagrangian transport through surfaces in compressible flows*, SIAM J. Appl. Dyn. Syst., 17 (2018), pp. 526–546.
- [8] H. HOPF, *Über die drehung der tangenten und sehnen ebener kurven*, Comp. Math., 2 (1935), pp. 50–62.
- [9] J. DOUGLAS, JR., AND 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 (1982), pp. 871–885.
- [10] D. KARRASCH, *Lagrangian transport through surfaces in volume-preserving flows*, SIAM J. Appl. Math., 76 (2016), pp. 1178–1190.
- [11] J. D. LAMBERT, *Numerical Methods for Ordinary Differential Equations: The Initial Value problem*, John Wiley & Sons, Hoboken, NJ, 1991.
- [12] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK, 2002.
- [13] P. LIDSTRÖM, *Moving regions in Euclidean space and Reynolds' transport theorem*, Math. Mech. Solids, 16 (2011), pp. 366–380.
- [14] J. C. NEU, *Training Manual on Transport and Fluids*, Grad. Stud. Math. 109, American Mathematical Society, Providence, RI, 2010.
- [15] A. OLIVEIRA AND A. M. BAPTISTA, *A comparison of integration and interpolation Eulerian-Lagrangian methods*, Internat. J. Numer. Methods Fluids, 21 (1995), pp. 183–204.
- [16] O. REYNOLDS, *Papers on Mechanical and Physical Subjects: The Sub-mechanics of the Universe, Collected Work*, Vol. III., Cambridge University Press, Cambridge, UK, 1903.
- [17] V. ROM-KEDAR, A. LEONARD, AND S. WIGGINS, *An analytical study of transport, mixing and chaos in an unsteady vortical flow*, J. Fluid Mech., 214 (1990), pp. 347–394.

- [18] V. ROM-KEDAR AND S. WIGGINS, *Transport in two-dimensional maps*, Arch. Ration. Mech. Anal., 109 (1990), pp. 239–298.
- [19] S. M. RUMP, T. OGITA, AND S. OISHI, *Accurate floating-point summation part I: Faithful rounding*, SIAM J. Sci. Comput., 31 (2008), pp. 189–224.
- [20] E. N. SARMIN AND L. A. CHUDOV, *On the stability of the numerical integration of systems of ordinary differential equations arising in the use of the straight line method*, Comput. Math. Math. Phys., 3 (1963), pp. 1537–1543.
- [21] W. E. SCHIESSEER AND G. W. GRIFFITHS, *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with MATLAB*, Cambridge University Press, Cambridge, UK, 2009.
- [22] G. SCOVAZZI AND T. HUGHES, *Lecture Notes on Continuum Mechanics on Arbitrary Moving Domains*, Technical Report SAND-2007-6312P, Sandia National Laboratories, Albuquerque, NM, 2007.
- [23] H. SEIFERT, *Konstruktion dreidimensionaler geschlossener raume*, Ber. Sächsischen Akad. Wiss., 83 (1931), pp. 26–66.
- [24] A. SOMMARIVA AND M. VIANELLO, *Splinegauss: A MATLAB code for Gauss-Green Cubature over Spline Curvilinear Polygons*, 2009, <http://www.math.unipd.it/~alvise/software.html>.
- [25] A. SOMMARIVA AND M. VIANELLO, *Gauss-Green cubature and moment computation over arbitrary geometry*, J. Comput. Appl. Math., 231 (2009), pp. 886–896, doi:10.1016/j.cam.2009.05.014.
- [26] A. SOMMARIVA AND M. VIANELLO, *Compression of multivariate discrete measures and applications*, Numer. Funct. Anal. Optim., 36 (2015), pp. 1198–1223.
- [27] A. STANIFORTH AND J. CÔTÉ, *Semi-Lagrangian integration schemes for atmospheric models – a review*, Monthly Weather Rev., 119 (1991), pp. 2206–2223.
- [28] Y. SUDHAKAR, A. SOMMARIVA, M. VIANELLO, AND W. A. WALL, *On the use of compressed polyhedral quadrature formulas in embedded interface methods*, SIAM J. Sci. Comput., 39 (2017), pp. B571–B587.
- [29] J. H. VERNER, *Explicit Runge–Kutta methods with estimates of the local truncation error*, SIAM J. Numer. Anal., 15 (1978), pp. 772–790.
- [30] T. WEINKAUF AND H. THEISEL, *Streak lines as tangent curves of a derived vector field*, IEEE Trans. Vis. Comput. Graph., 16 (2010), pp. 1225–1324, doi:10.1109/TVCG.2010.198.
- [31] WIKIPEDIA, *Streamlines, Streaklines, and Pathlines*, 2017, https://en.wikipedia.org/wiki/Streamlines,_streaklines,_and_pathlines.
- [32] Q. ZHANG, *Highly accurate Lagrangian flux calculation via algebraic quadratures on spline-approximated donating regions*, Comput. Methods Appl. Mech. Engrg., 264 (2013), pp. 191–204.
- [33] Q. ZHANG, *On a family of unsplit advection algorithms for volume-of-fluid methods*, SIAM J. Numer. Anal., 51 (2013), pp. 2822–2850.
- [34] Q. ZHANG, *On donating regions: Lagrangian flux through a fixed curve*, SIAM Rev., 55 (2013), pp. 443–461.
- [35] Q. ZHANG, *On generalized donating regions: Classifying Lagrangian fluxing particles through a fixed curve in the plane*, J. Math. Anal. Appl., 424 (2015), pp. 861–877.
- [36] Q. ZHANG, H. JOHANSEN, AND P. COLELLA, *A fourth-order accurate finite-volume method with structured adaptive mesh refinement for solving the advection-diffusion equation*, SIAM J. Sci. Comput., 34 (2012), pp. B179–B201.