Highly Accurate Lagrangian Flux Calculation via Algebraic Quadratures on Spline-Approximated Donating Regions

Qinghai Zhang

Department of Mathematics, University of Utah,
155 S. 1400 E., JWB 233, Salt Lake City UT 84103

Abstract

Lagrangian flux through a fixed curve segment within a time interval \((t_n, t_n + k)\) can be formulated as an integral at the initial time \(t_n\) over a compact point set called donating region, in which each particle will pass through the curve during the time interval and contribute to the flux. Based an explicit, constructive, and analytical solution of the donating region, the author proposes algorithms of Lagrangian flux calculation (LFC) as algebraic quadratures over the spline-approximated donating regions. This generic formulation yields high accuracies up to the 8th-order both in time and space. As another benefit, LFC leads to a conservative semi-Lagrangian method whose time step size is free of the Eulerian stability constraint \(C_r \leq 1\). The high accuracy and robustness of the proposed LFC algorithms are demonstrated by various numerical tests with Courant numbers ranging from 1 to 10000 and with accuracies from the 2nd order to the 8th order. LFC might also be useful in designing hybrid Eulerian methods for complex geometries.

Key words: Lagrangian flux; Donating region; Streakline; Scalar advection equation; B-spline; Semi-Lagrangian methods.

1 Introduction

Consider a scalar function \(f(x, t) : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}\) governed by the evolution equation

\[
\frac{df}{dt} = \frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = S(f, x, t),
\]

(1)

Email address: qinghai@math.utah.edu (Qinghai Zhang).
where \( t \) is time, \( \mathbf{x} \) space, \( D \) dimensionality, \( \mathbf{u}(\mathbf{x},t) \) a continuous velocity field, and \( S \) the source term. Since \( df \) is an exact form, its integral is path independent. More precisely, let \( \gamma : (s_0, s_1) \rightarrow \mathbb{R}^D \times \mathbb{R} \) be an oriented curve in the extended phase space with its endpoints as \( \gamma(s_0) = (\mathbf{x}_0, t_0) \) and \( \gamma(s_1) = (\mathbf{x}_1, t_1) \), the generalized Stokes's theorem [70, ch.V] implies

\[
\int_\gamma df = \int_\gamma \left( \nabla f, \frac{\partial f}{\partial t} \right) \cdot (d\mathbf{x}, dt).
\] (2)

Taking \((\mathbf{x}_0, t_0)\) as the initial condition, the solution of (1) can be written as

\[
f(\mathbf{x}_1, t_1) = f(\mathbf{x}_0, t_0) + \int_\gamma \nabla f \cdot (d\mathbf{x} - \mathbf{u} dt) + \int_\gamma S dt.
\] (3)

(3) unifies Lagrangian and Eulerian viewpoints. When \( \gamma \) is chosen to satisfy \( \frac{dx}{dt} = \mathbf{u} \), the first integral in (3) vanishes and this leads to a pure Lagrangian method. When \( \gamma \) is chosen so that \( \mathbf{x} \) is fixed, (3) yields an Eulerian method. \( \gamma \) might also be chosen as a union of fixed curves and particle trajectories; Smolarkiewicz and Pudykiewicz [61] used such a choice in designing a class of semi-Lagrangian methods.

A disadvantage of pure Lagrangian methods is that the distribution of the initial particles might become highly nonuniform and the accuracy of approximating \( f(\mathbf{x}, t) \) deteriorates in regions where the particles are widely spread. For a better control of the distribution, one can choose a new set of particles at the beginning of each time step so that these particles are advected to coincide with a regular grid at the end of the time step. This is an essential idea of semi-Lagrangian methods. Originated half a century ago [26,71,38], semi-Lagrangian methods have been an active research field and have enjoyed tremendous popularity in weather forecasting (e.g. [51] [64] [44]) and plasma simulation (e.g. [4] [8] [52]). In other applications, they are known as Eulerian-Lagrangian methods [46] or the modified method of characteristics [37].

Semi-Lagrangian methods have several distinguishing features from Eulerian methods. For one thing, they are more amenable to non-uniform grids and spherical geometry [23, p.389]. More importantly, in advection-dominated problems, semi-Lagrangian methods allow the time step size to be determined by accuracy considerations rather than stability constraints (see Section 6.2 for a discussion), thus they can be considerably more efficient than Eulerian methods. Ritchie et al. [30] reported that a semi-Lagrangian method with 15-minute time steps gave an accuracy equivalent to that of an Eulerian counterpart with 3-minute time steps, yielding an efficiency improvement of about a factor of four. Nonetheless, if the forcing term \( S \) has important high frequencies that must be resolved, accuracy considerations may lead to similar time steps for both methods, then semi-Lagrangian methods tend to be at a disadvantage due to its additional cost of characteristic tracing [3].
A criticism to early semi-Lagrangian methods is that they typically do not conserve the global and local integrals of the advected quantity. This is not unexpected. After all, the question of which particle will coincide a grid point implies that the unknown to be solved are point values, whereas the conservation of a certain quantity usually involves its integrals over volumes. Hence an alternative strategy for mass conservation seeks trace-back regions at the beginning of the time step such that they overlap the fixed control volumes at the end of the time step. The boundaries of these trace-back regions amount to a different tessellation of the domain, thus achieving global conservation for cell-averaged quantities. Local mass conservation is more difficult, but can still be attained by adjusting the trace-back region so that its total volume equals that of the corresponding control volume; see [1], [77] for two examples.

This paper begins with a more general question: given a time interval \((t_n, t_n + k)\) and a fixed simple curve \(\tilde{LN}\) in the time-dependent velocity field \(u(x, t)\), which set of particles will pass through \(\tilde{LN}\) and contribute to the flux of \(f\) through \(\tilde{LN}\) within the time interval? More precisely, given \(\tilde{LN}\), \((t_n, t_n + k)\), and \(u(x, t)\), find a compact region at \(t_n\) such that the following holds:

\[
\int_{D_{\tilde{LN}}} f(x, t_n) \, d\mathbf{x} = \int_{t_n}^{t_n+k} \int_{\tilde{LN}} f(x, t)u(x, t) \cdot \mathbf{n}_{\tilde{LN}} \, d\mathbf{x} \, dt,
\]

where \(\mathbf{n}_{\tilde{LN}}\) is the normal vector of \(\tilde{LN}\). The LHS and RHS of (4) are the Lagrangian flux and the Eulerian flux, respectively. (4) furnishes an implicit definition of donating region (DR) as the loci of the fluxing particles at time \(t_n\). It is also the dependence domain of \(\tilde{LN}\) for the advection operator. On one hand, DR appears to stem from the Eulerian viewpoint. Indeed, by integrating over control volumes and applying the divergence theorem, one can express the cell-averaged unknowns in terms of fluxes, i.e., the integrals over DRs; see (50). On the other hand, DR connects to the semi-Lagrangian approach in that the trace-track region mentioned in the previous paragraph is a special case of DR when the endpoints of the simple curve are arbitrarily close; see Corollary 10 in Section 3.3 for the precise statement and its proof.

In [73], the author solved the DR problem (4) with an explicit and constructive definition of DR, which is shown to contain, and only contain, all the particles that has a net effect of passing through \(\tilde{LN}\) once. This statement holds not only in the asymptotic range of zero time step size, but also for any finite time interval so long as (i) the backward streaklines of \(L\) and \(N\) neither self-intersect nor properly intersect each other, and (ii) \(\tilde{LN}\) remains as a subset of the DR boundary. Conversely, the time step can be arbitrarily long provided that both (i) and (ii) hold.

Utilizing this analytical solution, this work focuses on a computational aspect of DRs, namely Lagrangian flux calculation (LFC) with high-order accuracies
and long time steps. The motivations come from both Eulerian methods and semi-Lagrangian methods.

Traditional finite-volume methods compute Eulerian fluxes through cell edges from nearby cell-averaged quantities \[76,75\], but have difficulties near irregular boundaries. One possible strategy to cope with this is to restrict traditional Eulerian flux calculation to cells in uniform regions and to employ LFC for cell edges near irregular boundaries. Due to its flexibility, LFC may serve as a useful complement to the traditional Eulerian flux calculation and the combination of them might lead to promising hybrid methods.

In contrast to their high-order spatial accuracies, most (if not all) semi-Lagrangian methods only have second-order temporal accuracies, due to the Strang splitting in converting a multi-dimensional problem into a sequence of one-dimensional equations. As Eulerian methods move to high-order accuracies, the improvement on their efficiency is dramatic: there exist problems with certain accuracy requirements such that a fourth-order finite-volume method \[75\] running on a personal laptop is faster than a second-order version running on the fastest supercomputer in this world! To maintain the efficiency advantage of semi-Lagrangian methods, there is a pressing need to have high-order temporal accuracies. This need has long been recognized by Staniforth and Côté \[64\] as they noted that “it will become important to increase the order of the time discretization, otherwise there will ultimately be no time-step advantage with respect to an analogous Eulerian model.” The LFC algorithms proposed here answer this need. In Section 6.3, a conservative semi-Lagrangian method resulting from high-order LFC is demonstrated to have up to the seventh order accuracies both in time and space for Courant numbers ranging from 10 to 1000.

The rest of this paper is organized as follows. After introducing notation in Section 2, the analytical solution of the DR problem is summarized in Section 3 with some new findings. The fundamental algorithms of constructing streaklines and timelines are detailed in Section 4. The DR boundary is discretized by a number of breakpoints and is then approximated by B-splines that connect these breakpoints. The accuracy of the resulting curvilinear polygon in approximating the exact DR is also analyzed for the linear case and the cubic case. In Section 5, Green’s formula is utilized to reduce the cubature over the approximated DR to nested one-dimensional quadratures, which is then evaluated by standard Gauss-Legendre rules. These steps combine to the proposed LFC algorithms. Their convergence rates from 2 to 8 are demonstrated in Section 6 by various numerical tests for open and closed curves, and for moderate and very large Courant numbers from 1 to 10000. In particular, Section 6.3 shows that LFC naturally leads to a conservative semi-Lagrangian method for the scalar advection equation. Section 7 concludes this paper with future research prospects.
2 Preliminaries and Notation

A vector in a Euclidean space may be interpreted either as a stationary location or as the locus of a moving particle. For clarity the following conventions same as those in [73] are adopted:

- The function form \( p(t) \) denotes the location of a moving particle at time \( t \);
- Lower letters \( p, q \) without function forms are used when the independence on time is irrelevant or unimportant;
- Uppercase letters \( L, M, N \) are reserved for stationary locations;
- Uppercase calligraphic letters are reserved for point sets, which default to open sets unless otherwise specified.

2.1 Flow maps

The ordinary differential equation (ODE)

\[
\frac{dx}{dt} = u(x, t)
\]

admits a unique solution for any given initial time and position so long as \( u \) is continuous in time and Lipschitz continuous in space. The locations of a particle \( p(t) \), after forward tracing (\( \rightarrow \)) or backward tracing (\( \leftarrow \)), can be obtained as:

\[
\overrightarrow{p(t_0, \tau)} = p(t_0 + \tau) = p(t_0) + \int_{t_0}^{t_0+\tau} u(p(t), t) \, dt; \quad (6a)
\]

\[
\overleftarrow{p(t_0, \tau)} = p(t_0 - \tau) = p(t_0) - \int_{t_0}^{t_0-\tau} u(p(t), t) \, dt, \quad (6b)
\]

where \( t_0 \) is the reference time and the time increment \( \tau \) is always positive. (6a) and (6b) are inverse functions to each other in the sense that tracing \( p \) forward and backward with the same time increment returns \( p \) to the initial location,

\[
\overrightarrow{p(t_0, \tau)} = q(t_0 + \tau) \iff \overleftarrow{q(t_0 + \tau, \tau)} = p(t_0). \quad (7)
\]

In the most general setting, both \( t_0 \) and \( \tau \) are independent variables, the tracing functions (6) are also called flow maps, \( \phi : \mathbb{R}^D \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^D \),

\[
\phi_{t_0}^{+\tau}(p(t_0)) = p(t_0 + \tau); \quad \phi_{t_0}^{-\tau}(p(t_0)) = p(t_0 - \tau), \quad (8)
\]

where the ‘+’ and ‘-’ superscripts denote forward and backward tracing respectively. The tracing could be either forward or backward for a superscript without a leading sign. A numerical integrator approximating the action of
the flow map will be denoted by $\hat{\phi}$. The synonym notations (6) and (8) will be chosen based on clarity and simplicity. In particular, when the initial time $t_0$ is fixed, each flow map with fixed $\tau$ is a homeomorphism, i.e., a continuous bijective function, and the set of flow maps

$$\Xi_{t_0} = \{ \phi_{t_0}^\sigma : \sigma \in \mathbb{R} \}$$

(9)

forms a one-parameter group of diffeomorphisms.

2.2 Curvilinear polygons

A parameterized curve in $\mathbb{R}^D$ is a continuous function $\gamma : (s_1, s_2) \rightarrow \mathbb{R}^D$ where $-\infty < s_1 < s_2 < +\infty$. If $\gamma$ is simple if it is injective. If a simple curve $\gamma$ further has $\gamma(s_1) = \gamma(s_2)$, we say $\gamma$ is a simple closed curve on the interval $(s_1, s_2]$. Although strictly speaking a curve is a map, it will also refer to the image of the map. A simple, non-closed curve is also called a curve segment, denoted by $LN$ with endpoints $L = \gamma(s_1)$, $N = \gamma(s_2)$.

Let $p_1, p_2, \cdots, p_m$ denote $m$ pairwise disjoint points in the plane, connect these points by $m$ curve segments $\overline{p_ip_{i+1}}, \cdots, \overline{p_mp_{m+1}}(p_{m+1} = p_1)$ to form a closed curve $\partial P$. $\partial P$ bounds a curvilinear polygon $P$, whose vertices are the $m$ points, and whose edges are the $m$ curve segments. $\partial P$ is a simple closed curve if and only if for all $i, j = 1, 2, \cdots, m$,

$$\begin{cases} 
  p_i \notin \overline{p_jp_{j+1}}; \\
  i \neq j \iff \overline{p_ip_{i+1}} \cap \overline{p_jp_{j+1}} = \emptyset;
\end{cases}$$

(10)

otherwise $\partial P$ is self-intersecting. Since a curve is a mapping from an open interval to $\mathbb{R}^D$, it does not include its endpoints.

Jordan curve theorem [36] and its generalization, Jordan-Brouwer theorem [13], state that a simple closed curve/surface divides the plane/space into three parts: itself, its interior, and exterior. A point in the plane/space belongs to one of the three parts. The orientation of a simple closed curve/surface can be interpreted as the choice of assigning the signs ‘+’ and ‘−’ to the interior and the exterior. A simple closed curve is defined to be positively oriented if one always has the interior to the left when traversing the curve in the increasing direction of the parameterization.

A curvilinear polygon $P$ always refers to an open point set, i.e. it is the interior of its boundary $\partial P$: $P = \text{int}(\partial P)$. $\|P\|$ denotes its volume. The notation $P(p_1, p_2, \cdots, p_m)$ with the vertex sequence as $(p_1, p_2, \cdots, p_m)$ implies the orientation of $\partial P$ and $P$. 
3 Donating Region Construction: Analysis

Before starting this section, it is emphasized that the velocity field is only assumed to be continuous in time and Lipschitz continuous in space. In particular, the divergence-free condition $\nabla \cdot \mathbf{u} = 0$ is not assumed, hence the analysis here applies equally well to incompressible and compressible flows.

3.1 Streaklines and other characteristic curves

The timeline of a point set $\mathcal{S}$ is a subset of the phase space $\mathbb{R}^D$ defined as

$$\phi_{t_0}^{\pm \tau}(\mathcal{S}) = \{ \phi_{t_0}^{\pm \tau}(q) : q(t_0) \in \mathcal{S} \},$$

where the forward timeline with $+\tau$ and the backward timeline with $-\tau$ are also called the image and preimage of $\mathcal{S}$, respectively; in addition, they are homeomorphic to $\mathcal{S}$

$$\mathcal{S} \cong \phi_{t_0}^{+ \tau}(\mathcal{S}) \cong \phi_{t_0}^{- \tau}(\mathcal{S}),$$

because each flow map in $\Xi_{t_0}$ is a homeomorphism.

A pathline is the curve generated by following a single particle in a time period, hence the points on this curve are at different time instances:

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

In general, the independent parameters of a flow map (8) are the initial time $t_0$, the initial position $p(t_0)$, and the time increment $\tau$; it is bijective in the case of timelines when $t_0$ and $\tau$ are given, but not bijective in the case of pathlines for given $t_0$ and $p(t_0)$ since different $\tau$’s might lead to the same location. However, the uniqueness of the ODE solution dictates that extended pathlines do not intersect in the extended phase space $\mathbb{R}^D \times \mathbb{R}$.

A streamline of a time instant $t_0$ is a curve $\gamma_{t_0}^{sm}(s)$ with its tangent vector parallel to the instantaneous velocity field $\mathbf{u}(\mathbf{x}, t_0)$ [29, p.422],

$$\frac{d\gamma_{t_0}^{sm}(\mathbf{x})}{ds} \times \mathbf{u}(\mathbf{x}, t_0) = 0.$$  

Definition 1 (streakline) A streakline is the locus of all the particles that have passed or will pass continuously through a fixed location:

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

$$\Psi_{t_0}^{-k}(M) = \{ \phi_{t_0-k+\tau}(M) : \tau \in (0, k) \}.$$
Different from a pathline, a streakline is a snapshot: all particles in (15a) are at time \( t_0 + k \) and all particles in (15b) at \( t_0 - k \). Particles in (15a) come from the same initial location \( M \), although they have different initial time \( t_0 + k - \tau \) and different time increment \( \tau \). In contrast, particles in (15b) start at the same initial time \( t_0 - k \) and arrive at the same location \( M \), but their time increments \( \tau \) and their initial positions \( \phi_{t_0 - k + \tau} \) vary. Intuitively, (15a) is the trace of continuously injected dye at a fixed location in a fluid flow. With time and velocity reversed, (15b) can also be thought of as the trace of continuously injected dye at \( M \).

In both (13) and (15), the independent parameter is the time increment \( \tau \), therefore pathlines \( \Phi : (0, k) \to \mathbb{R}^D \) and streaklines \( \Psi : (0, k) \to \mathbb{R}^D \) are indeed curves. Streamlines are also curves obtained by integrating an instantaneous velocity field. In contrast, timelines defined in (11) are isomorphic images of a general point set and are not necessarily curves. For steady flows, the streakline, pathline, and streamline seeded from the same point coincide. See Fig. 1 and Fig. 2 in [66] for illustrations of these characteristic curves in unsteady flows.

3.2 The flux set

**Definition 2 (fluxing particle)** In two dimensions, a fluxing particle to a simple curve \( \widehat{LN} \) over the time interval \( (t_0 - k, t_0) \) is a particle \( p \) whose pathline \( \Phi_{t_0 - k}(p) \) properly intersects \( \widehat{LN} \) at least once. The sign of intersection \( (x_I, t_I) \) is defined as \( S_I = \text{sgn}(n(x_I) \cdot u(x_I, t_I)) \), which determines the type of the flux as an out-flux if \( S_I = +1 \), or an in-flux if \( S_I = -1 \). The index of the fluxing particle is the sum of the signs of all the intersections.

Note that the properness of intersection is necessary for the particle to pass through \( \widehat{LN} \) so that it is indeed a fluxing particle. The index of a fluxing particle is analogous to the concept of winding number in complex analysis.

**Definition 3 (flux set)** The flux set of index \( n \) associated with \( \widehat{LN} \) over the time interval \( (t_0 - k, t_0) \), denoted by \( \mathcal{F}_{\widehat{LN}}^{n}(t_0, k) \), is the area occupied by all the fluxing particles of index \( n \) at time \( t_0 - k \).

3.3 Donating region construction

The analytical, constructive definition of the DR in two-dimensional space is based on timelines, streaklines, and the orientation of simple closed curves.

**Definition 4 (donating region)** For a given velocity field \( u(x, t) \), the DR
(a) The canonical form

(b) The rotational form. \( D = D_1 \cup D_2 \)

Fig. 1. The fundamental forms of DR. The dotted circles with arrows indicate the orientation. For a rotational DR, \( D_1 \) and \( D_2 \) may or may not have opposite orientations.

\( D_{\tilde{N}L}(t_0, k) \) associated with a simple curve \( \tilde{L}\tilde{N} \) over time interval \((t_0 - k, t_0)\) is the open curvilinear polygon \( P(L, N, \tilde{N}, \tilde{L}) \), whose edges are \( \tilde{L}\tilde{N}, \Psi_{t_0}^{-k}(N), \phi_{t_0}^{-k}(\tilde{L}\tilde{N}), \) and \( \Psi_{t_0}^{-k}(L) \); its orientation is determined by the vertex sequence \( L \rightarrow N \rightarrow \tilde{N} \rightarrow \tilde{L} \).

If \( L \) and/or \( N \) have zero velocity over the time interval, then \( \Psi_{t_0}^{-k}(L) = \{L\} \) and/or \( \Psi_{t_0}^{-k}(N) = \{N\} \), thus the number of boundary edges of a DR is not necessarily four. A DR might also degenerate to an empty set if \( \phi_{t_0}^{-k}(\tilde{L}\tilde{N}) \) and \( \tilde{L}\tilde{N} \) are subsets of a single streakline.

**Definition 5** A DR is said to be canonical, or have a canonical form, if and only if its boundary is a simple closed curve.

**Lemma 6** A DR \( D_{\tilde{L}\tilde{N}}(t_0, k) \) is canonical if and only if all of the following holds:

1. Both \( \Psi_{t_0}^{-k}(L) \) and \( \Psi_{t_0}^{-k}(N) \) are simple curves; \( (16a) \)
2. \( \Psi_{t_0}^{-k}(L) \) and \( \Psi_{t_0}^{-k}(N) \) have no proper intersections; \( (16b) \)
3. \( \tilde{L}\tilde{N} \cap \phi_{t_0}^{-k}(\tilde{L}\tilde{N}) = \emptyset; \) \( (17a) \)
4. \( \tilde{L}\tilde{N} \cap \Psi_{t_0}^{-k}(N) = \emptyset, \tilde{L}\tilde{N} \cap \Psi_{t_0}^{-k}(L) = \emptyset; \) \( (17b) \)
5. \( \phi_{t_0}^{-k}(\tilde{L}\tilde{N}) \cap \Psi_{t_0}^{-k}(N) = \emptyset, \phi_{t_0}^{-k}(\tilde{L}\tilde{N}) \cap \Psi_{t_0}^{-k}(L) = \emptyset; \) \( (17c) \)
6. \( L \neq \tilde{N}(t_0, k), L \notin \Psi_{t_0}^{-k}(N), \) \( N \neq \tilde{L}(t_0, k), N \notin \Psi_{t_0}^{-k}(L). \) \( (18) \)

**Proof.** Since \( \tilde{L}\tilde{N} \) is simple, its homeomorphic preimage \( \phi_{t_0}^{-k}(\tilde{L}\tilde{N}) \) is also simple; together with \( (16a) \), all the four edges are simple. \( (16b) \) and \( (18) \) imply \( \Psi_{t_0}^{-k}(L) \cap \Psi_{t_0}^{-k}(N) = \emptyset \), which, together with \( (17) \), implies that the edges are pairwise disjoint for all enumerative possibilities. \( (18) \) guarantees
disjointness of vertices and also prevents self-intersection at the vertices. The rest of the proof follows from (10).

**Definition 7** A DR is said to be rotational or have a rotational form if and only if it satisfies (16) and (17) but not (18).

See Fig. 1 for two examples of the canonical and rotational forms.

**Theorem 8 (The Donating Region Theorem)** If a DR $\mathcal{D}_{\tilde{L}N}(t_0, k)$ satisfies (16), $\tilde{L}N \subset \partial \mathcal{D}_{\tilde{L}N}$, and the number of self-intersections of $\partial \mathcal{D}_{\tilde{L}N}$ is countable, then $\mathcal{D}_{\tilde{L}N}(t_0, k)$ consists of a countable number of simply-connected regions

$$
\mathcal{D}_{\tilde{L}N}(t_0, k) = \mathcal{D}^+_{\tilde{L}N} \cup \mathcal{D}^-_{\tilde{L}N}, \quad \mathcal{D}^+_{\tilde{L}N} = \bigcup_{i=1}^{m^+} \mathcal{D}^+_i; \quad \mathcal{D}^-_{\tilde{L}N} = \bigcup_{i=1}^{m^-} \mathcal{D}^-_i, \tag{19}
$$

where $\mathcal{D}^+_i$'s and $\mathcal{D}^-_i$'s have positive and negative orientations, respectively. $m^\pm \in \mathbb{N}$ are the number of canonical sub-donating-regions, $m^\pm = 0$ implies $\mathcal{D}^\pm_{\tilde{L}N} = \emptyset$. Furthermore,

$$
\mathcal{D}^+_{\tilde{L}N} \equiv \mathcal{F}^+_{\tilde{L}N}(t_0, k); \quad \mathcal{D}^-_{\tilde{L}N} \equiv \mathcal{F}^-_{\tilde{L}N}(t_0, k). \tag{20}
$$

**PROOF.** The proof is an induction on the number of intersections. The induction basis is that (20) holds for both the canonical form and the rotational form. In the induction step, each possible intersection, i.e. that of a streakline to the timeline, that of a streakline to the fixed curve, or that of the time line to the fixed curve, are shown to lead to a decomposition of the DR so that the induction hypothesis still holds. The reader is referred to [73] for more details.

**Remark 9** As shown in Fig. 2, the condition $\tilde{L}N \subset \partial \mathcal{D}_{\tilde{L}N}$ in Theorem 8 excludes from $\mathcal{D}_{\tilde{L}N}(t_0, k)$ any particle with its fluxing index $n \geq 2$ or $n \leq -2$. By the rectification theorem [2, ch.2], this condition can always be satisfied by reducing $k$.

Theorem 8 states that, so long as $\tilde{L}N \subset \partial \mathcal{D}_{\tilde{L}N}$, the backward streaklines of $L$ and $N$ neither self-intersect nor properly intersect each other, the DR $\mathcal{D}_{\tilde{L}N}(t_0, k)$ contains, and only contains, all the fluxing particles of $\tilde{L}N$ with fluxing indices $\pm 1$. The tracing of the particles in the flux set is thus reduced to that of the DR boundary. In the asymptotic case of $\tilde{L}N$ becoming a simple closed curve, the DR is the exclusive disjunction of the interior of $\tilde{L}N$ and its preimage. Thus the flux sets of index $\pm 1$ for the simple closed curve are the mutual difference of its interior and the preimage of its interior.
Fig. 2. An example of $\widetilde{LN} \not\subset \partial D_{\widetilde{LN}}$. The particles in the dark gray shade go across $\widetilde{LN}$ twice while those in the light gray shade only once. The arrows indicate the orientation of the curves. The boundary of the white region is a simple closed curve containing $L$ and the boundary of the gray region is another simple closed curve containing $N$. The former is entirely contained in the latter.

Fig. 3. The connection between the DR and the preimage of a simple closed curve. As $N \to L$, the volume of the streaktube (the shaded region) approaches zero. In the case of $\mathcal{C} \cap \phi_{t_0}^{-k} (\mathcal{C}) = \emptyset$, the DR then consists of two disjoint opposite-oriented compact regions: $\mathcal{C}$ is the out-flux set and $\phi_{t_0}^{-k} (\mathcal{C})$ the in-flux set.

**Corollary 10** Let $\partial \mathcal{C}$ be the simple closed curve as the boundary of $\mathcal{C}$, then

$$
\mathcal{F}_{\partial \mathcal{C}}^{-1} (t_0, k) = \mathcal{C} \setminus \phi_{t_0}^{-k} (\mathcal{C}), \quad \mathcal{F}_{\partial \mathcal{C}}^{-1} (t_0, k) = \phi_{t_0}^{-k} (\mathcal{C}) \setminus \mathcal{C}.
$$

In particular, if $\mathcal{C} \cap \phi_{t_0}^{-k} (\mathcal{C}) = \emptyset$, $\mathcal{F}_{\gamma_C}^{-1} (t_0, k) = \mathcal{C}$ and $\mathcal{F}_{\gamma_C}^{-1} (t_0, k) = \phi_{t_0}^{-k} (\mathcal{C})$.

**PROOF.** Let a simple curve $\widetilde{LN} \subset \partial \mathcal{C}$ have the same orientation as that of $\partial \mathcal{C}$. It follows from the opposite orientation of the streaklines $\Psi_{t_0}^{r-k} (L)$ and $\Psi_{t_0}^{r-k} (N)$ in Definition 4 that, as $N \to L$,

$$
\lim_{\widetilde{LN} \to \partial \mathcal{C}} D_{\widetilde{LN}} (t_0, k) = \mathcal{C} \oplus \phi_{t_0}^{-k} (\mathcal{C}) = D^+ \cup D^-,
$$

where $D^+ = \mathcal{C} \setminus \phi_{t_0}^{-k} (\mathcal{C})$ and $D^- = \phi_{t_0}^{-k} (\mathcal{C}) \setminus \mathcal{C}$ have the same and opposite orientations as that of $\partial \mathcal{C}$, respectively. The case of $\mathcal{C} \cap \phi_{t_0}^{-k} (\mathcal{C}) = \emptyset$ is illustrated in Fig. 3. Applying Theorem 8 completes the proof.
4 Donating Region Construction: Spline Approximation

In practice the analytical forms of the streaklines and timelines are seldom available and an approximation of DRs is thus desirable. The discrete representation of DRs in this work are based on B-splines [56,57], the notation of which is introduced just enough to ease exposition and to make the proposed LFC algorithms precise; for complete theories see [10], [43, ch.2], [58].

4.1 B-splines

Let $\kappa \in \mathbb{N}^+$ be a positive integer, $\mathbb{P}_{\kappa-1}$ or $\mathbb{P}_{<\kappa}$ the linear space of all polynomials of degree $\kappa - 1$ (or order $\kappa$), $\xi = (\xi_j)_{j=1}^{l+1}$ a strictly increasing sequence in $\mathbb{R}$. A piecewise polynomial (pp) function has the form

$$ P(\xi) = P_j(\xi), \quad \xi \in (\xi_j, \xi_{j+1}], \quad (23) $$

where $P_j \in \mathbb{P}_{<\kappa}$ for each $j$. The linear space of all such $P$’s is denoted by $\Pi_{\kappa, \xi, \nu}$; then $\xi_j$ is called a break, $\xi$ the break sequence, $\nu = (\nu_j)_{j=1}^{l}$ the continuity vector whose elements count the number of continuity conditions at $\xi$:

$$ \frac{d^{(i)}}{d\xi^{(i)}} P_{j-1}(\xi_j) = \frac{d^{(i)}}{d\xi^{(i)}} P_j(\xi_j), \quad \forall j = 2, \cdots, l; \quad \forall i = 0, 1, \cdots, \nu_j - 1. \quad (24) $$

In particular, $\nu_j = 1$ means $P_{j-1}(\xi_j) = P_j(\xi_j)$. In this work a pp function is always continuous, i.e., $\nu_j \geq 1$. By restricting the indices of $\nu$ to $2, \cdots, l$, I have assumed that $P(\xi)$ is smooth at $\xi_1$ and $\xi_{l+1}$. Then $\Pi_{\kappa, \xi, \nu}$ consists of $l$ pp functions, each has order $\kappa$. Hence the dimension of $\Pi_{\kappa, \xi, \nu}$ is

$$ \dim \Pi_{\kappa, \xi, \nu} = \kappa l - \sum_{j=2}^{l} \nu_j. \quad (25) $$

One basis of $\Pi_{\kappa, \xi, \nu}$ is formed by the truncated power functions:

$$ (\sigma - \xi)^\kappa_+ = ((\sigma - \xi)_0^+)\kappa, \quad (\sigma - \xi)_+ = \max(0, \sigma - \xi), \quad 0^0 = 0. \quad (26) $$

However, the value of a pp function may involve too many truncated power functions when $l$ is large; in other words, the truncated power basis does not have compact support. Furthermore, when two breaks are relatively close, the linear system of the truncated power expansion is ill-conditioned [10, p.85].

Definition 11 (divided difference) The $k$th divided difference of a function $g(\xi)$ at sites $\sigma_j, \cdots, \sigma_{j+k}$, denoted by $[\sigma_j, \cdots, \sigma_{j+k}]g$, is the leading coefficient (i.e. the coefficient of $\xi^k$) of the polynomial of order $\kappa + 1$ that agrees with $g(\xi)$ at the sequence $(\sigma_j, \cdots, \sigma_{j+k})$. 

12
In particular, \([\sigma_1] = g(\sigma_1)\). The sites are usually organized as a monotonically increasing sequence with repeated sites occurring together. Then
\[
[\sigma_j, \cdots, \sigma_j+\kappa] g = \begin{cases} 
\frac{g^{(\kappa)}(\sigma_j)}{\kappa!}, & \sigma_j = \sigma_{j+1} = \cdots = \sigma_{j+\kappa}; \\
[\sigma_{j+1}, \cdots, \sigma_{j+\kappa}] g - [\sigma_j, \cdots, \sigma_{j+\kappa-1}] g, & \sigma_{j+\kappa} - \sigma_j, 
\end{cases}
\]
otherwise. \(27\)

The above formula naturally leads to a polynomial of the Newton form to approximate the function \(g\):
\[
g(x) \approx \sum_{j=1}^{\kappa} (x - \sigma_1)(x - \sigma_2)\cdots(x - \sigma_{\kappa-1})[\sigma_1, \cdots, \sigma_j] g. \quad 28\)

See [10, ch.I] for more details on divided difference.

A remarkable improvement of the truncated power functions can be achieved by applying divided differences to them: this is the gist of B-splines. Its definition by de Boor [10, p.87] differs from the original one [19] in its normalization and is more amenable to numerical calculation.

**Definition 12 (B-spline)** Let \(\sigma = (\sigma_j)\) be a non-decreasing sequence. The \(j\)th normalized B-spline of order \(\kappa\) for the knot sequence \(\sigma\) is defined as
\[
B_{j,\kappa, \sigma}(\xi) = (\sigma_{j+\kappa} - \sigma_j)[\sigma_j, \cdots, \sigma_{j+\kappa}](\cdot - \xi)^{\kappa-1}. \quad 29\)

In the above definition, the divided difference should be interpreted as a functional acting on functions of \(\sigma\). The placeholder notation "\(\cdot\)" indicates that the \(\kappa\)th divided difference of \((\sigma - \xi)^{\kappa-1}\) is to be taken by fixing \(\xi\) and considering \((\sigma - \xi)^{\kappa-1}\) as a function of \(\sigma\) alone. The resulting B-spline function obviously depends only on \(\xi\).

B-splines have many attractive properties such as compact support, positivity, unity partition, variation diminishing and so on. Most importantly, B-splines form a natural basis for \(\Pi_{<\kappa, \xi, \nu}\) (hence B stands for basis).

**Theorem 13 (Curry and Schoenberg [20])** Given a strictly increasing break sequence \(\xi = (\xi_j)_{j=1}^{l+1}\) and the continuity vector \(\nu = (\nu_j)_{j=1}^{l}\) with \(\nu_j \leq \kappa\). Denote \(n = \dim \Pi_{<\kappa, \xi, \nu}\) as in (25). If the knot sequence \(\sigma = (\sigma_j)_{j=1}^{l+\kappa}\) obtained from the break sequence \(\xi\) satisfies

- \(\forall j = 2, \ldots, l, \xi_j\) occurs exactly \(\kappa - \nu_j\) times in \(\sigma\);
- \(\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_\kappa \leq \xi_1\) and \(\xi_{l+1} \leq \sigma_{n+1} \leq \sigma_{n+2} \leq \cdots \leq \sigma_{n+\kappa}\),

then, the B-splines \((B_{j,\kappa, \sigma})_{j=1}^{n}\) span \(\Pi_{<\kappa, \xi, \nu}\). More precisely, for each \(\mathcal{P}(\xi) \in\)
\[ \Pi_{\kappa, \xi, \nu}, \text{there exist unique coefficients} \alpha_j \in \mathbb{R}, j = 1, 2, \cdots, n, \text{such that} \]
\[ \mathcal{P}(\xi) = \sum_{j=1}^{n} \alpha_j B_{j, \kappa, \sigma}, \quad \text{on } [\xi_1, \xi_{l+1}]. \quad (30) \]

The coefficients \( \alpha_j \)'s can be obtained via the well-known Marsden’s identity and the “dual functionals” property of B-splines [11]; nonetheless, a stable and well-conditioned algorithm requires exploiting a recurrence relation of B-splines instead of directly evaluating the divided differences [9,17]. Such an algorithm is implemented in the MATLAB® Spline Toolbox.

4.2 Approximating timelines and streaklines

Hereafter I will denote by \( \hat{\cdot} \) the numerical approximation of an entity, e.g., \( \hat{L} \) and \( \hat{\phi}_{t_0}^{t_\tau}(p) \) approximate \( \bar{L} \) and \( \phi_{t_0}^{t_\tau}(p) \), respectively.

The first step towards discrete DR representation is to obtain a sequence of breakpoints for the closed curve of the DR boundary. The subroutine

\[ p(t_e) \approx \hat{\phi}_{t_s}^{t_e-t_s}(p) = \text{RungeKutta}(u, p, t_s, t_e) \quad (31) \]

integrates along the pathline of a particle \( p \) from \( t_s \) to \( t_e \) using a Runge-Kutta method. In the notation \( \hat{\phi}_{t_s}^{t_e-t_s}(p) \), the initial time is \( t_s \), the initial position \( p(t_s) \), the length of the integration \( |t_e - t_s| \), and the ending time \( t_e \).

As shown in Algorithm 1 and Fig. 4, it is straightforward to integrate (5) to obtain the preimage of a fixed curve. In contrast, backward streakline approximation, as shown in Algorithm 2 and Fig. 5, are much less intuitive. It is illuminating to compare Algorithm 1 to Algorithm 2: the numerical integrations in the former have different initial positions, but the same starting and ending time; those in the latter have the same initial position and ending time, but different initial times and time increments. For simplicity, this work divides the intervals \( (\tau_1 = 0, \tau_{\iota+1} = k) \) and \( (s_1, s_{\iota+1}) \) into equidistant sub-intervals. Also note that forward streaklines are never used in this work.

Algorithm 2 is sufficient to build backward streakline breakpoints for the purpose of LFC. Meanwhile it is informative to point out that streakline construction is well-studied in the community of flow visualization [39] [18] [69]. A common approach produces streaklines by intersecting hyperplanes with constant time instants to a path surface. Some authors also use the term ‘stream surface’ as a synonym of path surface, this is probably due to the fact that the seminal paper by Hultquist [34] was on stream surface construction for steady flows.
Fig. 4. Approximating the preimage of a fixed curve $\tilde{LN}$, $p_i = \gamma(s_i)$ in Algorithm 1. The dashed line represents the exact preimage of $\tilde{LN}$, with the preimages of the points $p_i$ represented by the solid dots. The hollow dots represent the B-spline breakpoints $\tilde{p}_i = \phi^{-k}_{t_0}(p_i)$ obtained by numerical time integration.

**Algorithm 1**: Breakpoints of the preimage of a fixed curve.

**Input**: The velocity field $u$,
the time interval $(t_0 - k, t_0)$,
a fixed parametrized curve $\tilde{LN} = \gamma(s)$,
a strictly monotonically increasing sequence
\[
\{s_i : i = 1, \ldots, t + 1; \, \gamma(s_1) = L, \, \gamma(s_{t+1}) = N, \, \gamma(s_j) \in \tilde{LN}, j = 2, \ldots, t \}.
\]

**Output**: A sequence of points $\{\phi^{-k}_{t_0}(s_i) : i = 1, \ldots, t + 1\}$ as the breakpoints of $\phi^{-k}_{t_0}(\tilde{LN})$, i.e. the preimage of $\tilde{LN}$.

1. foreach $i = 1, \ldots, t + 1$ do
2.  \[ \phi^{-k}_{t_0}(s_i) \leftarrow \text{RungeKutta}(u, \gamma(s_i), t_0, t_0 - k) \]
3. end

**Algorithm 2**: Breakpoints of a backward streakline.

**Input**: The velocity field $u$,
the time interval $(t_0 - k, t_0)$,
the seeding position $M$,
a strictly monotonically increasing sequence
\[
\{\tau_i : \tau_1 = 0, \tau_{i+1} = k, \, i = 1, \ldots, t + 1 \}.
\]

**Output**: A sequence of points $\{\phi^{-\tau_i}_{t_0-k}(M) : i = 1, \ldots, t + 1\}$ as the breakpoints of the backward streakline $\Psi^{-k}_{t_0}(M)$.

1. $\phi^{-\tau_1}_{t_0-k}(M) \leftarrow M$
2. foreach $i = 2, \ldots, t + 1$ do
3.  \[ \phi^{-\tau_i}_{t_0-k}(M) \leftarrow \text{RungeKutta}(u, M, t_0 - k + \tau_i, t_0 - k) \]
4. end
Fig. 5. Backward streakline approximation in the extended phase space \((x,t)\) of one-dimensional space \(x\). The dotted lines represent extended pathlines, the hollow points the breakpoints of the streakline, and the solid points the particles that arrives at \(M\) at different time instants. Along the \(x\)-axis, \(M = \frac{\dot{\phi}_{t_0-k+\tau_i}}{\tau_i} (M) = \frac{\dot{\phi}_{t_0-k+\tau_{i-1}}}{\tau_{i-1}} (M)\), \(q(t_0 - k) = \frac{\dot{\phi}_{t_0-k+\tau_i}}{\tau_i} (M)\). The double arrow denotes the direction of the velocity of a particle \(q\) with \(q(t_0 - k + \tau_i) = M\); the single arrow denotes the direction of Runge-Kutta integration for another particle \(q'\) with \(q'(t_0 - k + \tau_{i-1}) = M\). To sum up, Algorithm 2 picks a different particle coinciding with \(M\) for each initial time \(t_s = t_0 - k + \tau_i\) and integrate along its pathline until \(t_e = t_0 - k\). Hence a streakline is the intersection of a constant time plane to a path-surface.

More recently, WeinKauf and Theisel [69] showed that a streakline can be expressed as the tangent curve of a \((D + 2)\)-dimensional vector field in the twice-extended phase space \(\mathbb{R}^D \times \mathbb{R} \times \mathbb{R}\):

\[
\mathbf{q}(x, t, \tau) = \begin{pmatrix} (\nabla \phi^\tau_t(x))^{-1} \cdot \frac{\partial \phi^\tau_t(x)}{\partial \tau} + \mathbf{u}(x, t) \\ 0 \\ -1 \end{pmatrix},
\]

(32)

where \(\nabla \phi^\tau_t\) is the spatial gradient of the flow map and \(\text{"\text{\text{\text{-}}}1\text{"}}\) refers to time increment. The time component being zero implies that all points of a streakline is at a constant time. Equation (32) exemplifies that a non-autonomous system of ODEs can always be converted to an autonomous system by adding additional dimensions. The significance is that, as an alternative to intersecting the path surface to constant time hyperplanes, one can integrate the streak vector.
to obtain streaklines in the same manner of integrating the velocity vector to obtain pathlines. More precisely, given the seeding position \( M \), the initial time \( t_0 \), and a time increment \( \tau \), one can integrate \( \bar{q}(x, t, \tau) \) in \( D+2 \) dimensions with the initial condition \((M, t_0, 0)\) until the last component reaches \( \tau \), then the streakline is obtained by projecting the resulting \((D+2)\)-dimensional curve into \( \mathbb{R}^D \), i.e. removing its \( t \) and \( \tau \) coordinates. Although preparing the streak vector field requires additional memory, this approach is more computationally efficient than the classical approach; see [69] for more details.

4.3 Approximating donating regions

In this subsection the breakpoints of the timeline and the streaklines are first assembled into a coherent breakpoint sequence for a simple closed DR boundary. Each coordinate of the DR boundary is then considered as a function of the cumulative chordal distance so that it is easily approximated by a B-spline. Finally the accuracy of this approximation is analyzed for two common cases.

Definition 14 (breakpoint sequence of DR boundary) Given a curve segment \( LN \), its parametrization \( \gamma : (s_1, s_{i+1}) \to LN \) with a strictly monotonically increasing sequence \((s_i)_{i=1}^{i+1}\), another strictly monotonically increasing sequence \((\tau_i)_{i=1}^{i+1}\) with \( \tau_1 = 0, \tau_{i+1} = k \), and a \( \kappa \)th order numerical integrator \( \phi \).

The \( \kappa \)th order breakpoint sequence \((\hat{q}_j)_{j=1}^{i+1} = (\hat{x}_j, \hat{y}_j)_{j=1}^{i+1}\) for the boundary of \( D_{LN}(t_0, k) \) is ordered as

\[
\hat{q}_j = \begin{cases} 
\hat{\phi}_{t_0-k+\tau_j}^{-\tau_j} (N) , & j = 1, 2, \ldots, t + 1; \\
\hat{\phi}_{t_0}^{-k} (\gamma(s_{2t+2-j})) , & j = t + 1, t + 2, \ldots, 2t + 1; \\
\hat{\phi}_{t_0}^{-\tau_{3t+2-j}} (L) , & j = 2t + 1, 2t + 2, \ldots, 3t + 1; \\
\gamma(s_{4t}) , & j = 3t + 1, 3t + 2, \ldots, 4t + 1,
\end{cases}
\]

where the breakpoints in the first and third rows are obtained from Algorithm 2, those in the second row from Algorithm 1, and those in the fourth row from the parameterization \( \gamma \).

In particular, \( \hat{q}_1 = N, \hat{q}_{i+1} = \hat{N}, \hat{q}_{2i+1} = \hat{L}, \) and \( \hat{q}_{3i+1} = L \). The indices in the third row of (33) are reversed to comply with the order of the breakpoints in Definition 4. If \( LN \) is a line segment, the last \( i \) breakpoints may be omitted.

Define the cumulative chordal distance [25] as

\[
\ell_j = \begin{cases} 
0 , & j = 1; \\
\ell_{j-1} + \sqrt{(\hat{x}_j - \hat{x}_{j-1})^2 + (\hat{y}_j - \hat{y}_{j-1})^2}, & j > 1,
\end{cases}
\]

17
where \((\hat{x}_j, \hat{y}_j) = \hat{q}_j\) is the \(j\)th breakpoint, and approximating the DR boundary reduces to that of each coordinate, \(\hat{x}(\ell)\) or \(\hat{y}(\ell)\), by a single spline function.

**Definition 15 (spline approximation of donating regions (SADR))** Let \(D_{LN}^- (t_0, k)\) be a canonical DR as in Definition 5 and \(\kappa \geq 2\) a positive integer. The \(\kappa\)th order spline approximation of \(D_{LN}^- (t_0, k)\), denoted by \(\mathcal{D}_{LN}^\kappa (t_0, k)\), is the interior of a closed curve \(\hat{\gamma} : (0, \ell_{4\iota+1}) \rightarrow \mathbb{R}^2\), whose coordinates \(\hat{x}(\ell), \hat{y}(\ell)\) are pp functions obtained from the break sequence and the continuity vector,

\[
\xi = (\ell_j)_{j=1}^{4\iota+1},
\]

\[
\begin{aligned}
\nu_j &= 1, & j &= 1, \iota + 1, 2\iota + 1, 3\iota + 1; \\
\nu_j &\geq \kappa - 1 & \text{otherwise,}
\end{aligned}
\]

where \(\hat{q}_j = (\hat{x}_j, \hat{y}_j)\) are the \(\kappa\)th order break sequence as in Definition 14 and \(\ell_j\) the cumulative chordal distance as defined in (34).

In particular, two popular cases of \(\kappa = 2\) and \(\kappa = 4\) are considered.

**Definition 16** The linear approximation of a DR is the SADR with \(\kappa = 2\), and \(\nu_j = 1, \forall j = 1, 2, \cdots, 4\iota + 1\).

For \(\kappa = 4\), \(\nu\) can be determined by the “not-a-knot” end condition [6,7] as

\[
\begin{aligned}
\nu_j &= 1, & j &= 1, \iota + 1, 2\iota + 1, 3\iota + 1; \\
\nu_j &= \kappa, & j &= 2, \iota + 1 \pm 1, 2\iota + 1 \pm 1, 3\iota + 1 \pm 1, 4\iota; \\
\nu_j &= \kappa - 1, & \text{otherwise.}
\end{aligned}
\]

**Theorem 17 (accuracy of SADR)** Referring to Definition 14, denote \(\bar{k} = \max_i |\tau_i - \tau_{i+1}|\) and \(\bar{h} = \max_j ||\hat{q}_j - \hat{q}_{j+1}||_2\) where \(j \in [\iota + 1, 2\iota] \cup [3\iota + 1, 4\iota]\).

Assume that each edge of the exact DR \(D_{LN}^- (t_0, k)\) is \(C^{\kappa - 1}\) continuous. For the linear approximation and cubic approximation with (36), the volume difference between the exact and approximated DRs is

\[
E_D = \left\| \mathcal{D}_{LN}^{\kappa} (t_0, k) \oplus D_{LN}^- (t_0, k) \right\| = O(hk^\kappa + k^{\kappa+1}) + O(i\bar{h}^{\kappa+1} + \bar{k}^{\kappa+1}).
\]

In particular, for fixed \(\iota\),

\[
h \sim k \Rightarrow E_D = O(k^{\kappa+1}).
\]

---

\(^2\) some authors define a \(\kappa\)th order spline function as a pp function with \(\nu = (\kappa - 1, \cdots, \kappa - 1)\), however, there exist splines without maximum smoothness, e.g., Hermite splines. In this work, a spline function means a linear combination of B-splines and hence no intrinsic restriction on \(\nu\) is implied by its name.

\(^3\) Note that a break \(\xi_j\) with \(\nu = \kappa\) is absent in the knot sequence due to Theorem 13, hence the name “not-a-knot” follows.
PROOF. From (35), the closed boundary of $D_{LN}^\kappa \oplus D_{LN}^\kappa$ can be divided into six closed curves, each of which bounds a curvilinear polygon and leads to a certain type of representation error as follows

$$E_{\text{streak}}^\text{ODE} = \sum_{M=L,N} \left\| \mathcal{P}\left( \phi_{t_0-k+\tau_1}(M), \cdots, \phi_{t_0-k+\tau_{\kappa+1}}(M), \phi_{t_0-k+\tau_{\kappa+1}}(M), \cdots, M \right) \right\|;$$  

(39a)

$$E_{\text{streak}}^\text{REP} = \sum_{M=L,N} \sum_{i=1}^{\kappa} \left\| \mathcal{P}\left( \phi_{t_0-k+\tau_{i+1}}(M), \phi_{t_0-k+\tau_i}(M), \phi_{t_0-k+\tau_{i+1}}(M) \right) \right\|;$$  

(39b)

$$E_{\text{image}}^\text{ODE} = \left\| \mathcal{P}\left( \bar{p}_1 = \hat{L}, \bar{p}_2, \cdots, \bar{p}_i, \bar{p}_{i+1} = \hat{N}, \bar{p}_{i+1} = \hat{N}, \bar{p}_t, \cdots, \bar{p}_2, \bar{p}_1 = \hat{L} \right) \right\|;$$  

(39c)

$$E_{\text{image}}^\text{REP} = \sum_{i=1}^{\kappa} \left\| \mathcal{P}\left( \bar{p}_i, \bar{p}_{i+1}, \bar{p}_t \right) \right\|;$$  

(39d)

where $E_{\text{streak}}^\text{ODE}$ and $E_{\text{image}}^\text{ODE}$ are caused by $\phi$, $E_{\text{streak}}^\text{REP}$ and $E_{\text{image}}^\text{REP}$ by representing the exact streaklines and timelines using B-splines. Each curvilinear polygon in (39d) consists of two edges, one is the exact timeline segment while the other its approximation by a B-spline; a similar statement holds for (39b) with respect to streaklines. Fig. 6 shows an example with $\kappa = 2$ and $\iota = 1$.

The volume error is then bounded as

$$E_D \leq E_{\text{streak}}^\text{ODE} + E_{\text{streak}}^\text{REP} + E_{\text{image}}^\text{ODE} + E_{\text{image}}^\text{REP},$$  

(40)

where the inequality is justified by the possibility that the regions of different error types overlap each other, and the equality that they might also be pairwise disjoint. Since the numerical time integrator is $\kappa$th-order accurate and
In a finite time interval, \( |\hat{p}_t - \tilde{p}_t| = O(k^\kappa) \), hence \( E_{\text{image}}^{\text{ODE}} = O(hk^\kappa) \). Similarly \( E_{\text{streak}}^{\text{ODE}} = O(k^{\kappa+1}) \) since the length of a streakline is of \( O(k) \). As for the representation error, \( E_{\text{image}}^{\text{REP}} = O(2\bar{h}k^\kappa) \) and \( E_{\text{streak}}^{\text{REP}} = O(2\bar{h}k^{\kappa+1}) \), which follow from the midpoint rule for \( \kappa = 1 \) and Theorem 1 in [53] for \( \kappa = 3 \).

Finally, (38) clearly follows from (37) for a fixed number of polynomial pieces.

The numerical tests in section 6 confirms (38), even for the cases of \( \kappa \neq 2, 4 \). Note that the estimate in (37) might not be sharp, i.e., the order-of-accuracy could be higher. This is based on the observation that the parameterization of a curve is not unique and can be suitably modified to improve the accuracy, analogous to the improvement of Gaussian quadrature over Newton-Cotes quadrature by utilizing the extra freedom of node placements. Indeed, the error of a cubic parametric spline in approximating an arbitrary planar curve was first shown to be \( O(h^3) \) [32] and then improved to \( O(h^4) \) [53] by a re-parameterization. It has been conjectured [33] that, using \( \kappa \)th-order pp functions, a smooth regular curve in \( \mathbb{R}^D \) can be approximated to an order as high as \( \kappa + \lceil (\kappa - 2)/(D - 1) \rceil \). This conjecture has been confirmed for a number of special cases [12,48,35]. In two dimensions, the optimal order of \( 2\kappa - 2 \) is very promising. However, due to the nonlinear nature of the re-parameterization, it is nontrivial to apply this approach to the problem of DR approximation.

5 Algebraic Cubature over Approximated Donating Regions

Most researches of multivariate quadrature have been focused on regular regions [65] [15]. For irregular regions, simple schemes based on Monte Carlo methods are asymptotically slow to converge [47, Ch4]; other accurate methods often require the area of integration be simplexes such as triangles and tetrahedron [42] [49], [50], due to their finite-element applications. Conceptually one could decompose the irregular area into simplexes and sum up the sub-integrals on them, but this incurs a high number of unnecessary integration boundaries and function evaluations of the integrand. Computing a multivariate quadrature on an irregular area appears to be a daunting task.

Fortunately, in recent years the efficiency and accuracy of multivariate quadrature has been drastically improved, cf. [21], [16], [59], [41], [62,63,54]. Together with these papers, various quadrature software are readily available for irregular regions with piecewise linear boundaries [62] as well as spline boundaries [63], and for irregular regions in two dimensions [59] as well as n dimensions [16]. One popular approach [21] [63] repeatedly applies the divergence theorem to convert the multiple integrals to a sequence of nested one-dimensional...
integrals on boundary wireframes; these one-dimensional integral are then calculated by interpolatory quadrature formulas [45] based on orthogonal polynomials [28].

In two dimensions, the integral over a SADR can be transformed by Green’s theorem to another integral on its boundary:

\[
\int_{\mathcal{D}_{\text{LN}}} f(x, y) \, dx \, dy = \int_{\partial\mathcal{D}_{\text{LN}}} F(x, y) \, dy,
\]

(41a)

\[
F(x, y) = \int_{\varsigma} f(v, y) \, dv,
\]

(41b)

where \(\varsigma\) is fixed. The \(x\)-primitive \(F(x, y)\) and its integral can be calculated by Gauss-Legendre quadratures or Clenshaw-Curtis quadratures [67] [24].

**Theorem 18 (cubature over SADR)** Let \(\mathcal{D}_{\text{LN}}\) be a \(\kappa\)-th-order SADR as in Definition 15, and \((\hat{x}_s, \hat{y}_s)_{s=1}^{4}\) its pp-form with \(\ell_s, \ell_{s+1}\) for the \(s\)th segment. Let \(\mathcal{R} \supseteq \mathcal{D}_{\text{LN}}\) be a rectangular area \(\mathcal{R} = [x_1, x_2] \times [y_1, y_2]\) with a fixed \(\varsigma \in [x_1, x_2]\). Let \(\{\lambda_i\}_{i=1}^{z} \) and \(\{\omega_i\}_{i=1}^{z} \) be the nodes and weights of the Gauss-Legendre rule within \([-1, 1]\). Then for every bivariate polynomial \(P(x, y)\) of order \(\leq \kappa\), the following algebraic cubature formula is exact:

\[
\int_{\mathcal{D}_{\text{LN}}} P(x, y) \, dx \, dy = I_{\kappa}(P) = \sum_{s=1}^{4} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{sij} P(x_{sij}, y_{sij})
\]

(42a)

\[
n \geq \left\lceil \frac{\kappa}{2} \right\rceil, \quad m \geq \left\lceil \frac{\kappa^2 - 1}{2} \right\rceil;
\]

(42b)

\[
r_s(\eta) = \frac{\ell_{s+1} - \ell_s}{2} \eta + \frac{\ell_{s+1} + \ell_s}{2}, \quad r_{sij} = r_s(\lambda_j^m);
\]

(42c)

\[
x_{sij} = \frac{\hat{x}_s \left(r_{sij}^m\right) - \varsigma}{2} \lambda_i^n + \frac{\hat{x}_s \left(r_{sij}^m\right) + \varsigma}{2}; \quad y_{sij} = \hat{y}_s \left(r_{sij}^m\right);
\]

(42d)

\[
w_{sij} = \left(\frac{\ell_{s+1} - \ell_s}{2}\right) \left(\frac{\hat{x}_s \left(r_{sij}^m\right) - \varsigma}{2}\right) \omega_i^n \omega_j^m \frac{d\hat{y}_s}{d\ell} \left(r_{sij}^m\right).
\]

(42e)

**PROOF.** The proof is due to Sommariva and Vianello [63], it is rephrased here for clarity. Consider (41a) along the \(s\)th segment with the pp-form \((\hat{x}_s, \hat{y}_s)\),

\[
I_s(P) = \int_{\ell_s}^{\ell_{s+1}} F(\hat{x}_s, \hat{y}_s) \, \frac{d\hat{y}_s}{d\ell} \, d\ell = \int_{-1}^{1} \frac{\ell_{s+1} - \ell_s}{2} F(\hat{x}_s, \hat{y}_s) \, \frac{d\hat{y}_s}{d\ell} \, d\eta,
\]

(43)

where the second step follows from the variable change \(\ell = r_s(\eta)\). By the quadrature rule, we can then express the integral as a linear combination of
point values of the integrand:
\[ I_s(P) = \frac{\ell_{s+1} - \ell_s}{2} \sum_{j=1}^{m} w_j^m F(\hat{x}_s, y_{sj}) \frac{\partial y_s}{\partial \nu}(r_{sj}^m) + O(h^\kappa), \]  
(44)

where the \( \kappa \)-th-order accuracy requires \( m \geq \left\lceil \frac{k^2-1}{2} \right\rceil \), \( r_{sj}^m = \ell(\lambda_{sj}^m) \), and \( y_{sj} = \hat{y}_s(\ell(\lambda_{sj}^m)) \). Identifying \( f \) in (41) with \( P \), the \( x \)-primitive function \( F(x) \) is then
\[ F(\hat{x}_s, y_{sj}) = \int_{\hat{x}_s - \varsigma}^{\hat{x}_s + \varsigma} P(v, y_{sj}) dv = \int_{-1}^{+1} \frac{\hat{x}_s - \varsigma}{2} P(v(\lambda), y_{sj}) d\lambda, \]  
(45)

where the second step follows from the change of variable,
\[ v(\lambda) = \frac{\hat{x}_s (r_{sj}^m) - \varsigma}{2} \lambda + \frac{\hat{x}_s (r_{sj}^m) + \varsigma}{2}. \]

By the one-dimensional quadrature rule, a choice of \( n \geq \left\lceil \frac{k}{2} \right\rceil \) yields \( k \)-th-order accuracy,
\[ F(\hat{x}_s, y_{sj}) = \frac{\hat{x}_s (r_{sj}^m) - \varsigma}{2} \sum_{i=1}^{n} w_i^n P(x_{sij}, y_{sj}) + O(h^\kappa). \]  
(46)

Since \( P(x, y) \) is a polynomial with its order at most \( \kappa \), substituting (46) into (44) and summing over all spline segments yield (42a).

**Theorem 19 (accuracy of cubature over SADR)** For the \( \kappa \)-th order SADR, assume \( k \sim h \) and the conclusion (38) of Theorem 17 holds. For a bounded scalar function \( f \in C^\kappa(\mathbb{R}), \mathbb{R} \supseteq D_{\tilde{L}N} \) as in Theorem 18,
\[ \left| \int_{D_{\tilde{L}N}} f(x, y) dx dy - I_\kappa(f) \right| = O(k^{\kappa+1}). \]  
(47)

**PROOF.** Let \( P^* \) be the bivariate polynomial of order \( \kappa \) that best approximates \( f \) over \( \mathcal{R} \). The triangle inequality yields
\[ \left| \int_{D_{\tilde{L}N}} f(x, y) dx dy - I_\kappa \right| \leq \left| \int_{D_{\tilde{L}N}} \left( f(x, y) - P^*(x, y) \right) dx dy \right| \]
\[ + \left| \int_{D_{\tilde{L}N}} P^*(x, y) dx dy - \int_{\tilde{D}_{\tilde{L}N}} P^*(x, y) dx dy \right| + \left| I_\kappa(P^*) - I_\kappa(f) \right| , \]
where I have used Theorem 18 that \( I_\kappa(P^*) = \int_{\tilde{D}_{\tilde{L}N}} P^*(x, y) dx dy \). Clearly \( \| D_{\tilde{L}N} \| = O(k^2) \). Since \( P^* \) is the best approximation, \( f(x, y) - P^*(x, y) = O(k^\kappa) \), hence the first RHS term is \( O(k^2 k^\kappa) = O(k^{\kappa+2}) \). Since \( P^* \) is bounded on
R and I have assumed that Theorem 17 holds for $\kappa$, the second term is $O(k^{\kappa+1})$. The third RHS term $I_2(|f - P^*|)$ is also $O(k^{\kappa+2})$ because $|f - P^*| = O(k^\kappa)$ and $w_{ij} = O(k^2)$. Clearly, the error of the LFC algorithms dominates that of approximating $f$ with the polynomial $P$.

For a self-intersecting DR, I divide it into simply connected regions, apply (42) to each of them, and then sum up individual cubature as the result. Finally, the proposed LFC algorithms is summarized as follows.

**Definition 20 (Lagrangian flux calculation)** The proposed LFC algorithms consist of Algorithms 1 and 2 for streaklines and timelines, the SADR algorithms in Definitions 14 and 15, and the algebraic cubature formula (42).

### 6 Numerical Tests

Setting the source term to zero in (1) yields the scalar advection equation

$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = 0,$$

(48)

which is solved for a variety of test problems to verify the correctness, accuracy, and stability of the proposed LFC algorithms. In particular, the difference between the Eulerian fluxes, i.e. the RHS of (4), and the calculated Lagrangian fluxes for simple open curves are shown in Section 6.1 to converge to zero at a rate designated by the order of SADR construction. Section 6.2 exhibits the stability of LFC algorithms for large time steps with Courant numbers ranging from 10 to 10000. In Section 6.3, a semi-Lagrangian method is proposed by applying the LFC algorithms to closed curves; its high convergence rates and independence of the CFL condition are also demonstrated.

#### 6.1 LFC for simple open curves with small time steps

The four test cases of this subsection are detailed in Table 1. For the first two tests, characteristic tracing yields an analytical expression of the Eulerian flux, which, in the strain test, is obtained by a change of variables [81]: $x = \exp(\xi)$, $y = \exp(-\eta)$, and observing that $\frac{dx}{dt} = \frac{dy}{dt} = t$ be the characteristics of the transformed advection equation. In the third deformation test, the velocity field is borrowed from Smolarkiewicz [60] and an animation of it can be found in Fig. 18 of [77]; since the initial scalar function is the constant one, no characteristic tracing is needed and the flux reduces to the total volume of the DR. In the last vortex test, the velocity field is a modified version of that in
Table 1
Test cases of LFC for simple open curves with small time steps.

![Diagram of the test polygon.](image)

<table>
<thead>
<tr>
<th>tests</th>
<th>velocity, initial scalar field, and characteristic tracing formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>rotation</td>
<td>( \mathbf{u}(x, y, t) = \frac{1}{1+t^2} \begin{bmatrix} -y \ x \end{bmatrix} ); ( f_{t_0}(x, y) = 100xy ); ( \begin{bmatrix} x(t_0) \ y(t_0) \end{bmatrix} = \begin{bmatrix} \cos \theta &amp; \sin \theta \ -\sin \theta &amp; \cos \theta \end{bmatrix} \begin{bmatrix} x(t) \ y(t) \end{bmatrix} ), ( \theta = \tan^{-1}(t) );</td>
</tr>
<tr>
<td>strain</td>
<td>( \mathbf{u}(x, y, t) = t \begin{bmatrix} x \ -y \end{bmatrix} ); ( f_{t_0}(x, y) = (4x^2 + 4y^2)^5 ); ( \begin{bmatrix} x(t_0) \ y(t_0) \end{bmatrix} = \begin{bmatrix} x(t) \exp \left{ -\frac{(t-t_0)^2}{2} \right} \ y(t) \exp \left{ \frac{(t-t_0)^2}{2} \right} \end{bmatrix} );</td>
</tr>
<tr>
<td>deformation</td>
<td>( \mathbf{u}(x, y, t) = - \cos \left( \frac{\pi t}{2} \right) \begin{bmatrix} \sin(4\pi(x + 0.5)) \sin(4\pi(y + 0.5)) \ \cos(4\pi(x + 0.5)) \cos(4\pi(y + 0.5)) \end{bmatrix} ); ( f_{t_0}(x, y) = 1 ); characteristic tracing is unnecessary;</td>
</tr>
<tr>
<td>vortex</td>
<td>( \mathbf{u}(x, y, t) = \begin{bmatrix} \sin^2(\pi(x + 0.5t)) \sin(2\pi y), \ -\sin(2\pi(x + 0.5t)) \sin^2(\pi y) \end{bmatrix} ); ( f_{t_0}(x, y) = \exp \left{ -(x - 0.5)^2 - (y - 0.5)^2 \right} ); characteristic tracing uses a higher order Runge-Kutta method;</td>
</tr>
</tbody>
</table>

[5] so that a critical point moves along the top edge of the test polygon. Due to the complexities of the velocity and the initial function, an analytical form of Eulerian flux is unavailable for this test.

For each test, SADR}s are constructed for all edges of the test polygon shown in Table 1. The construction follows Definitions 14 and 15 with explicit Runge-Kutta methods of various orders: the modified Euler and Heun’s formula [40, p. 155], the classical fourth-order Runge-Kutta method [31, p. 138], Verner’s method of order 6(5) [68], and Dormand & Prince’s method of order 8(7) [22]. These methods are chosen randomly to demonstrate the insensitivity of the LFC algorithms to the type of numerical time integrators. Nonetheless, the time step size will have to lie in the stability region of a chosen method. When...
Table 2
Results of LFC for the test cases in Table 1. $\kappa$ is the order of SADR. Each edge of the test polygon is divided into $n_s$ segments with equal lengths, and each segment has $2\kappa$ breakpoints. $E_L(n_s)$ is the maximum norm of the error over all $5n_s$ segments and all $n_s$ time steps. $O_L$ is the resulting convergence rate. The Courant number is $C_r = \max ||u||_2 h/k$ where $h$ is the maximum length of an edge segment and $k$ the time step size. $k \sim h$ is chosen for $C_r$ to be a constant in each test.

<table>
<thead>
<tr>
<th>$\kappa$</th>
<th>$|E_L(1)|_\infty$</th>
<th>$O_L$</th>
<th>$|E_L(2)|_\infty$</th>
<th>$O_L$</th>
<th>$|E_L(4)|_\infty$</th>
<th>$O_L$</th>
<th>$|E_L(8)|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>rotation test ($t \in [0, 0.2]$, $C_r \approx 0.7$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.49e-03</td>
<td>2.55</td>
<td>4.26e-04</td>
<td>2.38</td>
<td>8.17e-05</td>
<td>2.24</td>
<td>1.73e-05</td>
</tr>
<tr>
<td>4</td>
<td>1.25e-06</td>
<td>5.33</td>
<td>3.10e-08</td>
<td>5.03</td>
<td>9.52e-10</td>
<td>4.56</td>
<td>4.05e-11</td>
</tr>
<tr>
<td>6</td>
<td>1.16e-08</td>
<td>6.06</td>
<td>1.75e-10</td>
<td>5.62</td>
<td>3.56e-12</td>
<td>5.90</td>
<td>5.96e-14</td>
</tr>
<tr>
<td>strain test ($t \in [0, 0.24]$, $C_r \approx 1.0$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5.50e-03</td>
<td>1.83</td>
<td>1.55e-03</td>
<td>1.94</td>
<td>4.05e-04</td>
<td>1.98</td>
<td>1.03e-04</td>
</tr>
<tr>
<td>3</td>
<td>1.98e-04</td>
<td>3.08</td>
<td>2.35e-05</td>
<td>3.01</td>
<td>2.91e-06</td>
<td>3.01</td>
<td>3.61e-07</td>
</tr>
<tr>
<td>4</td>
<td>1.31e-04</td>
<td>3.82</td>
<td>9.27e-06</td>
<td>3.94</td>
<td>6.06e-07</td>
<td>3.98</td>
<td>3.85e-08</td>
</tr>
<tr>
<td>5</td>
<td>1.07e-06</td>
<td>4.97</td>
<td>3.40e-08</td>
<td>5.00</td>
<td>1.06e-09</td>
<td>5.00</td>
<td>3.32e-11</td>
</tr>
<tr>
<td>6</td>
<td>8.16e-07</td>
<td>5.82</td>
<td>1.45e-08</td>
<td>5.93</td>
<td>2.36e-10</td>
<td>5.98</td>
<td>3.76e-12</td>
</tr>
<tr>
<td>7</td>
<td>1.54e-09</td>
<td>6.99</td>
<td>1.22e-11</td>
<td>7.00</td>
<td>9.54e-14</td>
<td>4.08</td>
<td>5.65e-15</td>
</tr>
<tr>
<td>strain test ($t \in [0, 0.96]$, $C_r \approx 4.0$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.69e-02</td>
<td>3.95</td>
<td>2.38e-03</td>
<td>4.12</td>
<td>1.37e-04</td>
<td>4.10</td>
<td>7.98e-06</td>
</tr>
<tr>
<td>5</td>
<td>1.76e-04</td>
<td>2.79</td>
<td>2.55e-05</td>
<td>5.71</td>
<td>4.86e-07</td>
<td>5.11</td>
<td>1.41e-08</td>
</tr>
<tr>
<td>6</td>
<td>1.18e-03</td>
<td>6.78</td>
<td>1.07e-05</td>
<td>6.36</td>
<td>1.30e-07</td>
<td>6.23</td>
<td>1.74e-09</td>
</tr>
<tr>
<td>7</td>
<td>3.27e-05</td>
<td>8.59</td>
<td>8.48e-08</td>
<td>8.47</td>
<td>2.38e-10</td>
<td>8.38</td>
<td>7.13e-13</td>
</tr>
<tr>
<td>deformation test ($t \in [0, 0.2]$, $C_r \approx 1.4$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8.61e-03</td>
<td>1.96</td>
<td>2.21e-03</td>
<td>2.22</td>
<td>4.76e-04</td>
<td>2.24</td>
<td>1.01e-04</td>
</tr>
<tr>
<td>3</td>
<td>1.54e-02</td>
<td>4.61</td>
<td>6.30e-04</td>
<td>3.87</td>
<td>4.32e-05</td>
<td>3.68</td>
<td>3.37e-06</td>
</tr>
<tr>
<td>4</td>
<td>1.90e-03</td>
<td>3.28</td>
<td>1.95e-04</td>
<td>3.90</td>
<td>1.31e-05</td>
<td>4.12</td>
<td>7.52e-07</td>
</tr>
<tr>
<td>5</td>
<td>2.43e-03</td>
<td>6.73</td>
<td>2.29e-05</td>
<td>6.72</td>
<td>2.17e-07</td>
<td>5.49</td>
<td>4.82e-09</td>
</tr>
<tr>
<td>6</td>
<td>2.23e-03</td>
<td>6.83</td>
<td>1.96e-05</td>
<td>7.07</td>
<td>1.46e-07</td>
<td>6.46</td>
<td>1.66e-09</td>
</tr>
<tr>
<td>7</td>
<td>9.63e-05</td>
<td>8.65</td>
<td>2.39e-07</td>
<td>6.95</td>
<td>1.94e-09</td>
<td>6.87</td>
<td>1.66e-11</td>
</tr>
<tr>
<td>vortex test ($t \in [0, 0.15]$, $C_r \approx 1.0$)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.98e-03</td>
<td>2.41</td>
<td>3.74e-04</td>
<td>2.47</td>
<td>6.76e-05</td>
<td>2.16</td>
<td>1.51e-05</td>
</tr>
<tr>
<td>4</td>
<td>5.64e-05</td>
<td>4.30</td>
<td>2.86e-06</td>
<td>3.82</td>
<td>2.03e-07</td>
<td>3.93</td>
<td>1.34e-08</td>
</tr>
<tr>
<td>6</td>
<td>1.16e-05</td>
<td>6.58</td>
<td>1.21e-07</td>
<td>6.65</td>
<td>1.20e-09</td>
<td>6.08</td>
<td>1.78e-11</td>
</tr>
</tbody>
</table>
Fig. 7. Examples of self-intersecting SADR $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ with $\kappa = 4$. The axes are the usual Cartesian coordinates $x$ and $y$. A thick solid line represents $\widetilde{LN}$, a dotted line a streakline and a thin solid line a timeline of $LN$. ‘◦’ represents a streakline breakpoint and ‘×’ a timeline breakpoint. $\mathcal{D}_{\widetilde{LN}}(t_0, k)$ is divided into simply connected regions before applying the cubature formula (42).

the velocity field is non-stiff and the time step size is not very large, an explicit Runge-Kutta method is a good choice. For large time step sizes, implicit Runge-Kutta methods might be more efficient since their stability regions are usually larger than those of the explicit methods. Stiff problems often require an implicit Runge-Kutta method to avoid the time step size being restrictively small. As a general principle for efficiency, the time step size should be determined by accuracy criteria, not by stability constraints. To achieve this, one can adaptively change the time step size from the estimation of the local truncation error either by Richardson extrapolation or by an embedded method; see [14], [31, ch. II.4] for some examples.
A constructed SADR might be self-intersecting. For one possibility, $\tilde{\mathcal{L}} \mathcal{N}$ might intersect its timeline due to critical points, as shown in Fig. 7 (a), (c), and (d); for another, the streaklines might intersect each other due to inexactness of floating-point calculations, as shown in Fig. 7(b). These cases are deliberately chosen to test the robustness of the implementation. As discussed at the end of Section 5, a self-intersecting SADR is decomposed into simply-connected regions, with their individual contribution computed by (42) and summed up as the Lagrangian flux. The result is then compared to the exact solution to evaluate the errors and convergence rates. For the last vortex test, standard Richardson extrapolation is used since the exact solution is unavailable.

The errors and convergence rates based on $\infty$-norms are listed in Table 2. The conclusions in Theorem 17 and Theorem 19 are clearly verified for all test cases. The convergence rates are $\kappa$ instead of $\kappa + 1$ because of the temporal accumulation of the error over the $O(\frac{1}{k})$ steps. In the strain tests, the reduced convergence rate ‘4.08’ with $\kappa = 7, C_r \approx 1.0$ is caused by the limited precision of floating-point numbers. The excellent convergence rates for $C_r \approx 4.0$ suggests that the LFC algorithms can have time steps longer than those of Eulerian methods. In fact, the errors $E_L(4)$ and $E_L(8)$ with $C_r \approx 4.0$ are even smaller than $E_L(1)$ and $E_L(2)$ with $C_r \approx 1.0$, especially in the cases of $\kappa > 4$. This is not surprising since the timeline is better resolved with more breakpoints while the streaklines are approximated with roughly the same accuracy. For the vortex test, the convergence rates are also satisfactory despite the presence of a moving critical point.

6.2 LFC for simple open curves with large time steps

The Courant-Friedrichs-Lewy (CFL) condition dictates that the numerical domain of dependence of any point in space and time must contain the analytical domain of dependence so that the numerical method can access the information needed in forming the solution. For advection-dominated problems, pathline tracing provide the means of accessing the analytical domain of dependence. In (3), if $S = 0$ and $\gamma$ is the pathline from $(x_0, t_0)$ to $(x_1, t_1)$, then $f(x_1, t_1) = f(x_0, t_0)$. Hence $f(x_1, t_1)$ depends on $f(x_0, t_0)$, or that the dependence domain of $(x_1, t_1)$ is $(x_0, t_0)$. Starting from $(x_1, t_1)$, if a numerical method can follow a unique path back to $(x_0, t_0)$, the time step size can be arbitrarily long. In my opinion, this long time step neither violates nor circumvents the CFL condition, but rather confirms it.

However, the time step size $k$ can only be so large as the backward pathline tracing reaches a bifurcation, then it is not clear to which direction the dependence domain lies. For LFC, the dependence domain of $\tilde{\mathcal{L}} \mathcal{N}$ is the flux sets of $\tilde{\mathcal{L}} \mathcal{N}$ and, according to Theorem 8, is available provided that $\tilde{\mathcal{L}} \mathcal{N} \subset \partial \mathcal{D}_{\tilde{\mathcal{L}} \mathcal{N}}$ and...
Table 3
Test results of LFC for simple open curves with high Courant numbers. The velocity field and the initial function are the same as those of the strain test in Table 1. DRs are constructed for a single line segment with endpoints \((c_s, c_s)^T\) and \((1, 1)^T\). This single line segment is equally divided into \(n_s\) segments. Since \(n_s\) is also the number of time steps of the whole simulation \((t \in [0, 1])\), \(C_r = \max \|u\|_2 k/h = \frac{1}{1-C_s}\). \(E_L(n_s)\) is the maximum norm of the LFC errors and \(O_L\) the corresponding convergence rate. It is expected that \(O_L = \kappa\), the order of SADR.

<table>
<thead>
<tr>
<th>(\kappa)</th>
<th>(|E_L(1)|_\infty)</th>
<th>(O_L)</th>
<th>(|E_L(2)|_\infty)</th>
<th>(O_L)</th>
<th>(|E_L(4)|_\infty)</th>
<th>(O_L)</th>
<th>(|E_L(8)|_\infty)</th>
<th>(O_L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5.31e+01</td>
<td>3.37</td>
<td>5.14e+00</td>
<td>3.81</td>
<td>3.66e-01</td>
<td>4.04</td>
<td>2.23e-02</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9.05e-01</td>
<td>4.13</td>
<td>5.16e-02</td>
<td>5.96</td>
<td>8.28e-04</td>
<td>6.26</td>
<td>1.08e-05</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.56e-02</td>
<td>7.58</td>
<td>8.16e-05</td>
<td>8.41</td>
<td>2.40e-07</td>
<td>8.37</td>
<td>7.22e-10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c_s = 0.99, C_r = 100)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8.21e+00</td>
<td>3.39</td>
<td>7.86e-01</td>
<td>3.83</td>
<td>5.53e-02</td>
<td>4.06</td>
<td>3.33e-03</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.45e-01</td>
<td>4.13</td>
<td>8.27e-03</td>
<td>5.96</td>
<td>1.33e-04</td>
<td>6.25</td>
<td>1.75e-06</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.47e-03</td>
<td>7.57</td>
<td>1.30e-05</td>
<td>8.40</td>
<td>3.84e-08</td>
<td>8.58</td>
<td>1.01e-10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c_s = 0.999, C_r = 1000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8.61e-01</td>
<td>3.39</td>
<td>8.23e-02</td>
<td>3.83</td>
<td>5.79e-03</td>
<td>4.06</td>
<td>3.48e-04</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.52e-02</td>
<td>4.13</td>
<td>8.69e-04</td>
<td>5.95</td>
<td>1.40e-05</td>
<td>6.25</td>
<td>1.84e-07</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.60e-04</td>
<td>7.57</td>
<td>1.37e-06</td>
<td>8.41</td>
<td>4.03e-09</td>
<td>7.35</td>
<td>2.47e-11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c_s = 0.9999, C_r = 10000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8.65e-02</td>
<td>3.39</td>
<td>8.27e-03</td>
<td>3.83</td>
<td>5.82e-04</td>
<td>4.06</td>
<td>3.49e-05</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.53e-03</td>
<td>4.13</td>
<td>8.74e-05</td>
<td>5.95</td>
<td>1.41e-06</td>
<td>6.25</td>
<td>1.85e-08</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.61e-05</td>
<td>7.57</td>
<td>1.37e-07</td>
<td>8.42</td>
<td>4.02e-10</td>
<td>3.70</td>
<td>3.10e-11</td>
<td></td>
</tr>
</tbody>
</table>

the two streaklines do not intersect. Hence in numerically constructing the DR, \(k\) may keep increasing so long as none of the two conditions is violated. The former can be checked by whether the DR boundary contains multiple simple closed curves with one inside another, as illustrated in Fig. 2. As for the latter condition, \(k\) is only limited by “bona fide” streakline intersections due to critical points, not by those from inexact arithmetic, as already shown in Section 6.1. In summary, the two conditions of Theorem 8 decide the maximum stable \(k\) for LFC.

The strain tests in the previous section are repeated for a single line segment with \(C_r = 10, 100, 1000, 10000\). The corresponding DRs are long and thin stripes; the length-width ratio increases as \(C_r\) increases. Table 3 shows the corresponding errors and convergence rates for \(\kappa = 4, 6, 8\). In the case of \(\kappa = 8\) and \(C_r = 10000\), the convergence rate 3.70 for the finest two grids is much less than the expected rate of 8.0; this might be caused by the oscillation of the
(a) The control volume $C_{st}$

(b) $\phi^{-k}_{t_{n+k}}(C_{st})$ for $r_C = 10^{-1}, C_r \approx 10$

(c) $\phi^{-k}_{t_{n+k}}(C_{st})$ for $r_C = 10^{-2}, C_r \approx 10^2$

(d) $\phi^{-k}_{t_{n+k}}(C_{st})$ for $r_C = 10^{-3}, C_r \approx 10^3$

Fig. 8. A star-shaped control volume and its preimages. The axes are the usual Cartesian coordinates $x$ and $y$. $O_C = (0.25, 0.25)^T$ is fixed; $r_C$ varies to change $C_r$. The test time interval is $[0, 1.7]$. The velocity field is the same as that of the deformation test in Table 1. In subplots (b), (c), and (d), '*'s represent the preimages of the vertices of the star polygon while 'x's the knots for timelines of its edges.

8th order polynomials, cf. the well-known Runge phenomenon. For all other tests, the convergence rates meet the expectation.

6.3 LFC for simple closed curves: an efficient semi-Lagrangian method for the scalar advection equation

From Corollary 10, the DR of a simple closed curve is the exclusive disjunction of its interior and the preimage of its interior. As shown in Fig. 3, if the time step size is big enough, these two regions are disjoint: $C$ is the set of outflux particles through $\partial C$ and its preimage $\phi^{-k}_{t_0}(C)$ is the set of influx particles. By (12), $C$ is homeomorphic to $\phi^{-k}_{t_0}(C)$ and $\partial \left( \phi^{-k}_{t_0}(C) \right) = \phi^{-k}_{t_0} (\partial C)$. Clearly, $\partial \left( \phi^{-k}_{t_0}(C) \right)$ does not contain any streaklines, but only the timelines of the boundary edges of $C$. 

29
Table 4
Results of the proposed semi-Lagrangian method (49) for the polygon in Fig. 8 (a). Similar to the tests in Section 6.1, when the number of segments \( n_s \) per edge is doubled, the number of time steps is also doubled. \( E_L(n_s - 2n_s) \) is the max-norm of the difference between the results of two consecutively refined grids. \( \kappa \) is both the accuracy of SADR and the expected convergence rate.

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
\kappa & \|E_L(4 - 8)\|_\infty & O_L & \|E_L(8 - 16)\|_\infty & O_L & \|E_L(16 - 32)\|_\infty & O_L & \|E_L(32 - 64)\|_\infty \\
\hline
2 & 1.26e-03 & 2.12 & 2.90e-04 & 2.10 & 6.75e-05 & 2.06 & 1.62e-05 \\
3 & 5.07e-04 & 2.91 & 6.73e-05 & 3.04 & 8.20e-06 & 3.03 & 1.00e-06 \\
4 & 1.89e-04 & 3.83 & 1.33e-05 & 3.92 & 8.77e-07 & 3.96 & 5.63e-08 \\
5 & 5.76e-07 & 3.66 & 4.57e-08 & 4.89 & 1.54e-09 & 5.03 & 4.74e-11 \\
6 & 5.39e-06 & 6.44 & 6.23e-08 & 6.27 & 8.06e-10 & 6.10 & 1.18e-11 \\
7 & 7.05e-08 & 6.96 & 5.67e-10 & 7.16 & 3.97e-12 & 8.47 & 1.12e-14 \\
\hline
\kappa & \|E_L(1 - 2)\|_\infty & O_L & \|E_L(2 - 4)\|_\infty & O_L & \|E_L(4 - 8)\|_\infty & O_L & \|E_L(8 - 16)\|_\infty \\
\hline
2 & 7.59e-06 & 1.31 & 3.06e-06 & 1.74 & 9.12e-07 & 1.89 & 2.47e-07 \\
3 & 1.31e-06 & 2.95 & 1.69e-07 & 2.98 & 2.14e-08 & 2.99 & 2.70e-09 \\
4 & 2.29e-08 & 3.70 & 1.76e-09 & 3.87 & 1.21e-10 & 3.94 & 7.89e-12 \\
5 & 1.55e-10 & 4.77 & 5.66e-12 & 4.95 & 1.83e-13 & 5.00 & 5.72e-15 \\
6 & 2.50e-11 & 7.41 & 1.47e-13 & 8.24 & 4.86e-16 & – & 2.18e-16 \\
7 & 7.53e-14 & 7.09 & 5.52e-16 & – & 2.48e-16 & – & 4.60e-16 \\
\hline
\kappa & \|E_L(1 - 2)\|_\infty & O_L & \|E_L(2 - 4)\|_\infty & O_L & \|E_L(4 - 8)\|_\infty & O_L & \|E_L(8 - 16)\|_\infty \\
\hline
2 & 6.35e-10 & 1.94 & 1.65e-10 & 1.97 & 4.21e-11 & 1.99 & 1.06e-11 \\
3 & 1.77e-11 & 3.00 & 2.22e-12 & 3.00 & 2.78e-13 & 3.00 & 3.48e-14 \\
4 & 1.37e-14 & 4.12 & 7.86e-16 & – & 1.73e-16 & – & 6.79e-17 \\
5 & 1.01e-16 & 4.67 & 3.96e-18 & – & 2.05e-17 & – & 6.95e-17 \\
6 & 1.11e-17 & – & 3.59e-17 & – & 1.06e-17 & – & 4.65e-17 \\
7 & 2.97e-17 & – & 5.10e-17 & – & 1.26e-18 & – & 1.78e-18 \\
\hline
\end{array}
\]

\( \phi^{-k}_{t_0}(C) \) is the answer to the question discussed in Section 1: which set of particles at \( t_0 - k \) will overlap the control volume \( C \) at \( t_0 \)? For incompressible flows, a conservative semi-Lagrangian method can thus be formulated for the advection equation (48) as

\[
\langle f(t_n + k) \rangle_C = \langle f(t_n) \rangle_{\phi^{-k}_{t_0+k}(C)},
\]

(49)
where $\langle f \rangle_C = \frac{1}{\|C\|} \int_C f \, d\mathbf{x}$ is the spatial average of $f$ over $C$. Note that (49) is still an exact relation without any discretization errors.

The procedures of this semi-Lagrangian method is almost identical to the LFC algorithms: the preimage of the control volume is first approximated by B-splines as defined in Definitions 14 and 15, then (42) yields the average of $f$ over the approximated preimage $\langle f(t_n) \rangle_{\hat{\phi}_{t_n+k}(C)}$. An important difference is that $\hat{\phi}_{t_n+k}(C)$ consists of no streaklines and the time step size is thus free of the streakline restrictions. However, this does not mean that the time step size can be arbitrarily large. Since the flow map between timelines is a diffeomorphism, $\hat{\phi}_{t_n+k}(C)$ should be homeomorphic to $C$ to preserve the mathematical correctness of the dependence domain. Therefore the time step size of the proposed semi-Lagrangian method is only subject to the constraint of $\hat{\phi}_{t_n+k}(C)$ being simply connected.

A numerical example is shown in Fig. 8, where a star-shaped control volume is fixed in the deformation velocity field of Table 1. Across different test runs, the center of the star polygon does not change while its radius is reduced to generate larger and larger Courant numbers. Three preimages for Courant numbers ranging from 10 to 1000 are also shown in Fig. 8. Since no exact solution is available, Richardson extrapolation is used to evaluate the error norms and corresponding convergence rates. As shown in Table 4, the convergence rates for all tests of $C_r = 10$ are satisfactory. For other cases with larger Courant numbers, the high-order accuracies and fine resolutions lead to well-resolved results with the errors under machine precision. These results not only confirm the high convergence rates of the proposed semi-Lagrangian method, but also show the robustness of the implementation, especially for the cases of Fig. 8 (b) and (c) where the preimages contain thin and long stripes.

As discussed in Section 1, seeking the preimages of the boundaries of control volumes amounts to a different tessellation of the computational domain. Therefore the proposed semi-Lagrangian method is globally conservative. Furthermore, the unknowns in the finite-volume formulation are cell averages instead of point values, hence the local mass-conservation error is included in the solution error. Table 4 shows the small bounds of the local mass-conservation errors, particularly for high-order LFC algorithms.

SADoRe [74], the MATLAB® package accompanying this work, has been made freely available online. The reader can use it to find missing implementation details or reproduce the numerical results presented in this section.
7 Concluding Remarks

Based on the DR analysis, the author has proposed a set of LFC algorithms for computing fluxes through fixed curves and demonstrated their accuracy and robustness by various test problems. The DR of a fixed curve relates to both Eulerian and semi-Lagrangian formulations. On the one hand, the proposed semi-Lagrangian method of (49) is derived by considering the Lagrangian flux of closed curves. On the other hand, the advection equation (48) can be solved by summing up the fluxes through a set of simple curves that constitute the boundary of the control volume,

$$
\langle f(t_{n+k}) \rangle_C = \langle f(t_n) \rangle_C - \frac{1}{\|C\|} \sum_{\tilde{LN} \subset \partial C} \int_{\tilde{D}_{\tilde{LN}}} f(x, t_n) \, dx.
$$

Since (50) is still an analytic relation with no discretization errors, the accuracy of the solution only depends on the LFC algorithms that evaluate the RHS integrals. As demonstrated in Section 6.2 and Section 6.3, both methods of (49) and (50) can have very large time steps, and thus could be much more efficient than classical Eulerian methods.

Four prospects of future research follow. Firstly, some explicit interface tracking methods such as the polygonal area mapping (PAM) method [77] and volume-of-fluid (VOF) methods advect a discrete color function $f$ that marks the tracked material. Since the value of $f$ is either 0 or 1, the Lagrangian flux of $f$ through a cell edge reduces to the volume of the intersection of the DR to the region occupied by the tracked material. Due to this purely geometric nature, SADR might be useful in generalizing PAM and VOF methods to high order accuracies. In fact, a fourth-order VOF advection algorithm based on SADR is already proposed in [72]. Secondly, the LFC algorithms are extremely flexible: the cell edges can be unstructured or even curve segments, and computing fluxes for partial cell edges near irregular boundaries incurs no algorithmic changes. As mentioned in Section 1, LFC might serve as a useful complement to the traditional Eulerian flux calculation in a domain decomposition approach, where fluxes through regular edges are computed by nearby cell averages and those through irregular edges by LFC. Thirdly, hybrid semi-Lagrangian methods based on LFC are currently being developed for solving the advection-diffusion-reaction equation and the Navier-Stokes equations in the presence of irregular and moving boundaries. Two targeting problems are the fluid-solid interface [80] and the fluid-air free surface flows [78,79]. Finally, considerable challenges arise in three dimensions, due to the algorithmic complexity of polyhedra routines, e.g., a self-intersecting polyhedron needs to be decomposed into simply-connected ones to facilitate numerical quadrature. The extension of the proposed LFC algorithms to three dimensions will be reported in a future paper.
Acknowledgements

The author would like to acknowledge insightful comments from two anonymous referees. The author also thanks Dr. John Tinsley Oden for his communication on this paper.

References


36


