Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

One- and two-step semi-Lagrangian integrators for ALE-FE two-phase flow simulations

G. R. Anjos*[1] | G. P. Oliveira[2] | N. Mangiavacchi[3] | J. R. Thome[4]

[1]COPPE/Department of Mechanical Engineering, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
[2]Graduate Program in Mechanical Engineering, Federal University of Paraíba, Paraíba, Brazil
[3]Group of Environmental Studies in Reservoirs, State University of Rio de Janeiro, Rio de Janeiro, Brazil
[4]École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Correspondence
*G. R. Anjos, Email: gustavo.rabello@coppe.ufrj.br

Abstract

Semi-lagrangian (SL) schemes are of utmost relevance to simulate two-phase flows where advection dominates. The combination of SL schemes with the finite element (FE) method and arbitrary Lagrangian-Eulerian (ALE) dynamic meshes yields a strong ingredient to deal with applications in Earth sciences, environmental engineering, and oil and gas industry whose numerical background is incipient. This paper is intended to propose a second-order time multistep SL/ALE/FE scheme to track particle trajectories traveling amidst single- or two-phase incompressible flows both in fixed and moving mesh situations. The novel scheme is a BDF2-like implementation that considers the mesh velocity as part of its backward-in-time integration. Trajectory departure points are searched through extrapolation and followed by a multistep interpolation corrector that works both for constant and varying time steps. A series of two-phase benchmark flow simulations are carried out by using the cubic element to compare the performance of the new integrator against single-step approximations as well as to analyze enhancements in representing the two-phase flow dynamics. We use the 2D axisymmetric Navier-Stokes equations as underlying model. Error analyses, convergence tests, quantitative and qualitative comparisons are presented and discussed to highlight the superior conservative feature of the novel scheme.

KEYWORDS:
Semi-lagrangian method; finite element method; fluid particle tracking; surface tension; high-order approximation; moving mesh

1 | INTRODUCTION

Modeling the advective transport in two-phase flows is relevant for applications that require accurate tracking of fluid particles. A few examples where advection plays a dominant role include the diffusion of pollutants over the air, the deposition of contaminants on aquifers, and the propagation of saturation fronts in petroleum reservoirs.

Over the years, the semi-Lagrangian method (from now on, SL) was successfully applied to approximate advection dominated flows since its inception in the Earth sciences community[?] a few decades ago. The technique attracted CFD practitioners mainly because of its ability to deal with large time step simulations (high CFL numbers) and accurate trajectory integration, which relies on a persistent tracking of the material motion of fluid particles. The combination between SL and the finite element method (FEM) dates back to the 1980s when numerical algorithms to solve the advection-diffusion equation were introduced[?]. Afterward, several authors devised SL/FEM schemes dedicated to deal with incompressible flows[? ? ? ? ?].

A few more recent methods embodied SL schemes into tailored algorithms to deal with spray combustion[?], weather prediction[?], rheology[?], guiding-center problems[?], and general 3D transport problems[? ? ?]. In two-phase flow models, the SL's presence is somewhat more recent and brings two main features. Firstly, it conserves the same properties of stability and accuracy exhibited in single-phase flow dynamics concerning the integration of the advection term. Secondly, it allows for equivalent accuracy when Level-Set/Volume-Of-Fluid functions are used as phase definition. Although many papers have taken advantage of both situations, we will give a quick overview of recent usage of the SL method for two-phase flows without making point-by-point comparisons.

Discrete models based on the advection of the level-set as phase definition function perhaps are the most popular among those who have chosen to insert the SL method into their applications. Different order SL schemes were applied to improve the accuracy of near-interface points and the advection of the interface itself for bubble and droplet simulations[? ?], implemented to capture free surface in Newtonian and non-Newtonian interfacial rheology[?], and used in combination with second-order interpolation to advect the level-set function[?]. As opposed to level-set functions, other formulations also appear. In[?], for example, the SL method was used together with a 4th-order Runge-Kutta method to advect the volume fraction function under a VOF-PLIC — Volume-Of-Fluid/Piecewise-Linear Interface Calculation — context. Another proposal was based on the CIP/MM/FVM — Constrained Interpolated Profile/Multi-Moment/Finite Volume Method — was used together with SL interpolation to advect particles shaping interfaces and other quantities over general two-phase flow configurations[?]. Bubble deformation subject to magnetic forces was also investigated through a CIP — Constrained Interpolated Profile — conservative SL to solve the flow field equations[?]. A similar CIP/SL method was implemented by[?] to study droplet and bubble merging with free surfaces.

We may observe that almost all the SL methodologies presented for two-phase flows as those mentioned above have focused primarily on interface-capturing approaches. On the other hand, numerical experiments involving interface-tracking procedures are so far embryonic. The authors have been reaching considerable outcomes with moving mesh SL/FEM techniques based on a front-tracking Arbitrary Lagrangian-Eulerian (ALE) approach to simulate the dynamics of multiple continuous and dispersed two-phase flow patterns with first-order SL integration[? ? ? ?]. This ensemble SL/ALE/FEM has advantages that come across many challenges faced in two-phase flow modeling[?]. A series of arguments support this assertion. Firstly, interface-fitted elements can track the topological changes with high resolution. Secondly, balanced dynamic meshing operations hamper severe node entanglement due to a purely Lagrangian description or avoid expressive node stagnation caused by a strictly Eulerian description. Thirdly, full control of volume and surface tesselations assure adaption and refinement. Lastly, swirling trajectories and dispersed bodies achieve better integration in a broad scope of two-phase flows. Differently from the traditional interpretation of the SL method, the SL/ALE combination searches for past locations that advance forward in time to match points that were displaced not over a fixed mesh, but dynamically due to remeshing operations.

Regardless of the fluid flow nature, the main numerical problems encompassing the SL method orbit around the pair interpolation/integration in a variable space-time domain. Achieving high-order schemes with computational cost-effectiveness is of utmost importance even though substantial computational resources are at hand.

This paper is intended to propose a multistep SL/ALE/FEM scheme to track particle trajectories traveling amidst single- or two-phase incompressible flows both in fixed and moving mesh situations. The novel scheme is a BDF2-like implementation that applies two backward-in-time integrations that embody the mesh velocity. Departure points are searched through extrapolation and followed by a multistep interpolation corrector that works both for constant and varying time steps. A series of two-phase benchmark flow simulations are carried out by using mixed finite element spaces to compare the performance of the new integrator against single-step approximations as well as to analyze enhancements in representing the two-phase flow dynamics. Error analyses, mesh convergence tests, and qualitative comparisons are presented and discussed to highlight the superior conservative feature of the novel scheme.

2 | METHODOLOGY

2.1 | Mathematical formulation

We start by writing the dimensionless form of the Navier-Stokes – considering that the reference bubble/drop velocity is unknown – and continuity equations for incompressible axisymmetric flows endowed with the surface tension term stemming from the "one-fluid" formulation[?]. According to the ALE description, they read as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{c}\cdot\nabla\mathbf{u}\right) = -\nabla p + \frac{1}{Ar^{1/2}}\mathbf{L}(z,r) + \rho\mathbf{g} + \frac{1}{Eo}\mathbf{f} \tag{1a}$$

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

where ρ is the variable density as a function of z and r, τ is the time, $\mathbf{u}$ is the flow velocity field, $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}$ is the convective velocity ($\hat{\mathbf{u}}$ is the mesh velocity), p is the pressure, $\mathbf{g}$ is the gravitational acceleration, and $\mathbf{f} = \kappa\nabla H$ is the interface force term, with curvature κ and the gradient of the Heaviside function ∇H. The dimensionless groups Ar and Eo stand for Archimedes and Ëortvös numbers, respectively defined as

$$Ar = \frac{\rho_0^2 g_0 r_0}{\mu_0^2}, \quad \text{and} \quad Eo = \frac{\Delta\rho g_0 r_0^2}{\sigma_0},$$

where the subscript stands for reference values, $\Delta\rho_0$ is the difference of densities among phases and σ is the surface tension. Hence, for single-phase flows employing a static mesh, $\hat{\mathbf{u}} = \mathbf{0} \Rightarrow \mathbf{c} = \mathbf{u}$, since there is no mesh motion. Moreover, the operators read as:

$$\begin{aligned}
\nabla &= [\partial_z, \partial_r], \\
\nabla\cdot &= [\partial_z, r^{-1} + \partial_r]^T, \\
\mathbf{L}(z,r) &= [L_z(z,r), L_r(z,r)]^T, \\
L_z(z,r) &= r^{-1}[\partial_r(\mu r(\partial_r(\mathbf{u}\cdot\mathbf{e}_z) + \mu\partial_z(\mathbf{u}\cdot\mathbf{e}_r)) + 2r\partial_z(\mu\partial_z(\mathbf{u}\cdot\mathbf{e}_z))] \;\; \text{and} \\
L_r(z,r) &= r^{-1}[2\partial_r(\mu r(\partial_r(\mathbf{u}\cdot\mathbf{e}_r))) + r\partial_z(\mu\partial_r(\mathbf{u}\cdot\mathbf{e}_z)) + r\partial_z(\mu\partial_z(\mathbf{u}\cdot\mathbf{e}_r)) - 2r^{-1}(\mathbf{u}\cdot\mathbf{e}_r)],
\end{aligned}$$

for axial and radial (z, r) coordinates, unitary vectors $\mathbf{e}_z$ and $\mathbf{e}_r$, and μ the variable dynamic viscosity as a function of z and r, respectively.

To lay out the mathematical background for the fluid flows studied here, let us consider that Eqs. (1a-1b) hold over the space-time domain $\Omega \times \tau$, with $\Omega \subset \mathbb{R}^2$ compact and having (outer) boundary Γ, and $\tau \subset \mathbb{R}$ finite. More generally, we can define $\Omega_b \subset \Omega$ as the dispersed phase, so that $\Omega\backslash\Omega_b$ corresponds to the continuous phase and Γ_b corresponds to the (inner) boundary of the dispersed phase, also defined as fluid-fluid interface. Due to the axisymmetry, we have assumed that $\mathbf{u}\cdot\mathbf{e}_r = 0 = \partial_r(\mathbf{u}\cdot\mathbf{e}_z)$ on the symmetry axis $(z, r = 0)$. Additionally, no-slip boundary condition with zero velocities is assumed at top and side boundaries when the bubble/drop is confined and a fixed reference frame is used, while a slip condition with opposite axial velocity set by the bubble/drop center of mass velocity is used at top boundary when bubble center of mass location is fixed in space (moving reference frame). A zero pressure Dirichlet boundary condition is used at the outflow (left boundary), and an inflow condition is set with the axial velocity of the bubble/drop center of mass velocity at the right boundary, therefore keeping the bubble/drop fixed in space when the moving reference frame is used. A generic model for this kind of flow is depicted in Figure 1.

2.2 | Finite element discretization

Let $(E, \mathcal{P}, \mathcal{D})$ be a finite element, where E is a triangle, $\mathcal{P}$ a shape function space defined over E and $\mathcal{D}$ a set of degrees of freedom. By calling $\mathcal{T}_h = \{E_h\}$ a triangulation, h a mesh refinement factor and Δt_n a time step varying with n, we obtain the following equations, which, for discretization purposes, redefine the continuous space-time domain as union-sets of triangulations

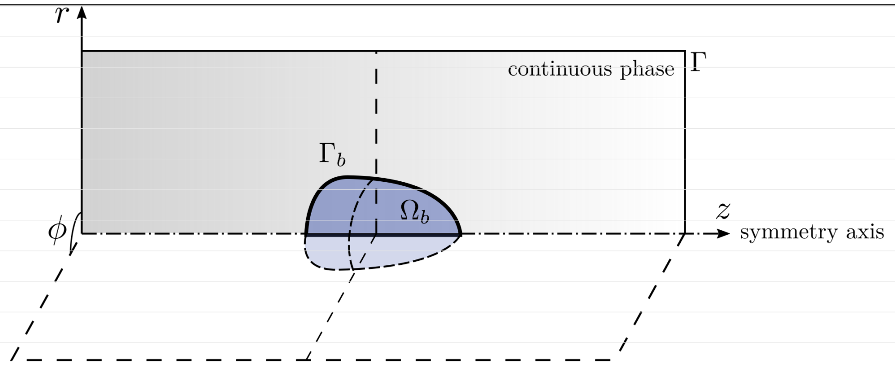


FIGURE 1 Generic model for an axisymmetric continuous-dispersed two-phase flow. Dispersed bodies are represented by the inner region Ω_b with boundary Γ_b that moves along the symmetry axis z and undergoes interface changes along the radial coordinate r.

and variable length time steps:

$$\Omega = \mathcal{T}_{h_b} \cup \mathcal{T}_h, \text{ with } E_{h_b} \in \Omega_b, \;\; E_h \in \Omega \backslash \Omega_b \tag{2}$$

$$T = \cup_{n=0}^{N-1} \Delta t_n, \;\; \Delta t_n = t^{n+1} - t^n, \tag{3}$$

where element-wise continuity is ensured. Furthermore, let $\mathbf{x}$ be an arbitrary mesh node and $H(\mathbf{x}) = \{0, 1/2, 1\}$ if the node belongs to the continuous phase, interface or dispersed phase, in this order.

The material derivative is discretized with the Semi-Lagrangian (SL) method, therefore ensuring unconditionally numerical stability for first order approximation. Despite allowing relative large time step and mesh size, high order SL schemes do not assure stability[?]. Common steps concerning the classical FE workflow allow us to define the function spaces

$$\mathcal{U} := \{\mathbf{u} \in [\mathcal{H}^1]^2(\Omega) \; ; \; \mathbf{u} = \mathbf{u}_D \text{ on } \partial\Omega_D\}$$

$$\mathcal{V} := \{\mathbf{v} \in [\mathcal{L}_0]^2(\Omega) \; ; \; \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}$$

$$\mathcal{Q} := \left\{ q \in \mathcal{L}^2(\Omega) \; ; \; \int_\Omega qr \, d\mathbf{z} = 0 \right\},$$

where $\mathcal{H}^1$ is the usual Sobolev function space and $\mathcal{L}_0 := \{v \in \mathcal{L}^2 \; ; \; \int_\Omega \mathbf{v}^2 r^\alpha \, d\mathbf{z} < \infty\}$, for $\alpha \in \mathbb{R}$, is a square-integrable function space in the context of axisymmetric FEM[? ?], with $d\mathbf{z} = dzdr$. Moreover, we derive a weak form based on the Galerkin method[?], thereby leading to

$$M_\rho \left(\frac{\hat{D}\mathbf{u}}{Dt}, \mathbf{v} \right) + G(p, \mathbf{v}) + \frac{1}{Ar^{1/2}} K(\mathbf{u}, \mathbf{v}) - M_\rho(\mathbf{g}, \mathbf{v}) - \frac{1}{Eo} G(H, \kappa \mathbf{v}) = \mathbf{0} \tag{4a}$$

$$D(q, \mathbf{u}) = 0, \tag{4b}$$

whose bilinear forms are

$$M_\rho(\mathbf{u},\mathbf{v}) = \int_\Omega \rho\mathbf{u}\cdot\mathbf{v}r\,d\mathbf{z}, \tag{5a}$$

$$K(\mathbf{u},\mathbf{v}) = \int_\Omega \mu r(\nabla\mathbf{u} : \nabla\mathbf{v}^T)\,d\mathbf{z} + 2\int_\Omega \mu r^{-1}(\mathbf{u}\cdot\mathbf{e}_r)(\mathbf{v}\cdot\mathbf{e}_r)\,d\mathbf{z}, \tag{5b}$$

$$G(p,\mathbf{v}) = \int_\Omega \nabla p\cdot\mathbf{v}r\,d\mathbf{z}, \quad \text{and} \tag{5c}$$

$$D(q,\mathbf{u}) = \int_\Omega q\nabla\cdot\mathbf{u}r\,d\mathbf{z} + \int_\Omega q\mathbf{u}\cdot\mathbf{e}_r\,d\Omega \tag{5d}$$

The operator $\frac{\hat{D}}{Dt}$ is a compact notation for the discrete ALE material derivative that will be treated as a SL term and κ is the curvature of the bubble/drop. After assembling the resulting matrices, the previous equations are converted into a system of equations that must be solved for convenient boundary conditions.

2.3 | Mixed finite element spaces

In this section, we write down the stable element used in this paper to deal with Eqs. (4a-4b). By following a generic notation for nodes of K_h, let us consider: $\mathbf{x}_i$ a *vertex node*; $\mathbf{x}_{iij} = \frac{1}{3}(2\mathbf{x}_i + \mathbf{x}_j)$, with $i \neq j$, an *edge node*; and $\mathbf{x}_{ijk} = \frac{1}{3}(\mathbf{x}_i + \mathbf{x}_j + \mathbf{x}_k)$ a *center node*. For all cases, $i, j = 1, 2, 3$. These generalities allow us to establish the “10-node triangle” (`cubic` element), whose shape functions for velocity and the scalar transport (used later in the result section) are written in terms of the barycentric coordinates $\lambda_{i,j,k}$ are respectively of degree 3 (cubic), whereas the shape functions for pressure p are of degree 1 (linear). The chosen element is Ladyzhenskaya–Babuška–Brezzi (LBB) stable – pair (P_3, P_1) –. Therefore, it does not require any other artificial stabilization technique, such as the Pressure Stabilized Petrov-Galerkin (PSPG). Moreover, the third order piece-wise polynomials used for velocity/scalar interpolation assures high numerical accuracy. Figure 2 illustrates the 10-node triangle in terms of the ijk-notation.

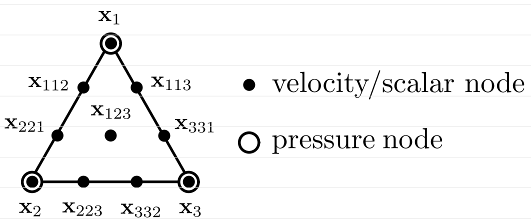


FIGURE 2 Stable element used in this paper: the “10-node triangle”, (P_3, P_1), or `cubic` element.

According to the shape functions and degrees of freedom (DOFs), we can form element-wise interpolants to compute flow variables at non-mesh points and resolve the partial requirement of interpolation of the SL method as we will discuss later.

Therefore, the triplet $(K, \mathcal{P}, \mathcal{D})$ is defined by

$$K = \text{10-node triangle} \tag{6a}$$

$$\mathcal{P} = \mathcal{P}_3 \oplus \mathcal{P}_1 \tag{6b}$$

$$\mathcal{D} = \{\mathbf{u}(\mathbf{x}_i), \mathbf{u}(\mathbf{x}_{ij}), \mathbf{u}(\mathbf{x}_{iij}), \mathbf{u}(\mathbf{x}_{ijk}), \varphi(\mathbf{x}_i)\}, \tag{6c}$$

whereas the interpolants $\mathcal{I}[\cdot]$ for velocity and scalar fields are, respectively, given by

$$\begin{aligned}\mathcal{I}[\mathbf{u}] &= \frac{1}{2}\sum_i \lambda_i(3\lambda_i - 1)(3\lambda_i - 2)\mathbf{u}(\mathbf{x}_i) + \\ &+\frac{9}{2}\sum_{i\neq j} \lambda_i\lambda_j(3\lambda_i - 1)\mathbf{u}(\mathbf{x}_{iij}) + \\ &+27\sum_{i<j<k} \lambda_i\lambda_j\lambda_k\mathbf{u}(\mathbf{x}_{ijk}) \text{ and} \end{aligned}\tag{7a}$$

$$\mathcal{I}[\varphi] = \sum_i \lambda_i\varphi(\mathbf{x}_i), \tag{7b}$$

where φ is an arbitrary scalar field to be interpolated. In this paper, it is essentially a placeholder for the pressure field, or the generic scalar field ψ as presented in Eq. (30).

2.4 | Semi-lagrangian method

SL/FEM approaches are underpinned by two subsequent steps: i) a backward-in-time integration, whereby fluid particle trajectories crossing a patch of elements are traced backwards until their departure points; ii) an interpolation of the fluid flow variables at the departure points. However, under the ALE viewpoint, the arbitrary mesh motion becomes a complicating factor both to integration and interpolation. To discuss the role played by the mesh velocity on a hybridized SL/ALE/FEM approach, we firstly recall a few mathematical aspects concerning the standard SL method with a simpler notation.

Throughout the continuum, it is usual to identify by $\mathbf{x}$ a spatial position in Ω so that $\mathbf{X} = \{(\mathbf{x}, \tau)\}_{\tau=f}^{\tau=h}$ is the trajectory formed by all points visited by a given particle in the interval $f \leq \tau \leq h$. The trajectory $\mathbf{X}$ is also the solution of the differential equation

$$\frac{d\mathbf{X}(\tau)}{d\tau} = \mathbf{u}(\mathbf{X}(\tau), \tau), \tag{8}$$

or a characteristic line[??]. We call $\mathbf{x}_*$ the "foot" of $\mathbf{X}$, i.e. the position occupied by the particle at the instant $\tau = f$, while $\mathbf{x}_a$ is called the "head" of $\mathbf{X}$, i.e. the position where the particle will be at the future time instant $\tau = h$. Then, within the interval $f \leq \tau \leq h$, the arrival point $\mathbf{x}_a$ is obtained through integration straightforwardly provided that $\mathbf{x}_*$ is known.

After integrating Eq. (8), we should verify that the equality

$$\mathbf{x}_* = \mathbf{x}_a - \int_f^h \mathbf{u}(\mathbf{X}(\tau), \tau)\, d\tau \tag{9}$$

is valid. Numerically, we should think of τ as a parameter that will define arbitrary time steps. For brevity, let us consider $f = t^n$ and $h = t^{n+1}$ as two successive time instants. The SL method enforces that $\mathbf{x}_a$ matches a mesh node at t^{n+1}, but demands that an approximation for $\mathbf{x}_*$ is sought at t^n since this exact departure point generally does not coincide with a mesh node. Such approximated departure point, denoted by $\mathbf{x}_d$, must be as close as possible to the true departure point $\mathbf{x}_*$. When replacing Eq. (9) by a simple discrete shift computed as

$$\mathbf{x}_* \approx \mathbf{x}_d = \mathbf{x}_a - \boldsymbol{\alpha}, \tag{10}$$

so that $\boldsymbol{\alpha} \approx \int_{t^n}^{t^{n+1}} \mathbf{u}(\mathbf{X}(\tau), \tau)\, d\tau$ turns into a backward displacement vector and produces a first-order scheme. Several schemes to compute $\boldsymbol{\alpha}$ are known in the literature, but high-order schemes are more interesting because use multiple time steps and reduce the numerical dissipation[?][?][?].

A singular feature of the SL method is the possibility of handling the material derivative appearing in Eq. (4a) by

$$\frac{D\varphi}{D\tau} = \frac{\partial \varphi}{\partial \tau} + \mathbf{c} \cdot \nabla \varphi \approx \frac{\varphi(\mathbf{x}_a, t^{n+1}) - \varphi(\mathbf{x}_d, t^n)}{\Delta t_n}. \tag{11}$$

This way, the spatial gradient term of any fluid variable $\varphi : \Omega \times \tau \to \mathbb{R}$ is suppressed, meaning that both the temporal and convective rates of change occur instantaneously in a combined effect.

From this point on, we assume that $\varphi(\mathbf{x}_i, t^n)$ will be denoted by the reduced discrete notation φ_i^n. Because perfect matches of departure points and mesh nodes are implausible, the value φ_d^n should be interpolated. That is why interpolation is the second feature inherent to any SL scheme we should care about in terms of effectiveness[?].

The ALE motion brings a further peculiarity into Eq. (11) that leads to a somewhat complicated interpretation of the material derivative due to the relative velocity $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}$. When $\hat{\mathbf{u}} = \mathbf{0}$, the finite element mesh is maintained fixed as if it were describing a pure Eulerian motion, but a Lagrangian mesh is virtually produced by the union of all the “foot” nodes. That is to say, we can observe the existence of a distorted mesh at each time step[?][?]. On the other hand, when $\hat{\mathbf{u}} \neq \mathbf{0}$, the integration should embed the relative motion and take both the Lagrangian and Eulerian parcels into consideration.

Figure 3 is a scheme that helps us to interpret the SL integration over a space-time perspective by using some vector algebra. To ease the comprehension of the ALE-based integration, we divide the geometrical entities in a few groups, namely slices, points, vectors, geometrical loci, and time intervals. The parallel slices represent spatial events that occur at a given temporal coordinate. Circle-marked points denote positions over the continuous medium, whereas the square-marked points denote approximated positions of discrete interest that were included for completeness. Vectors stand for displacements, while geometrical loci are shaded regions that identify material points in different motion descriptions. Time intervals mark standard divisions along the time domain.

Given that, we can proceed to recognize each entity and take advantage of the illustration to approach the two-time scheme to be introduced later. Slices at $\tau = t^{n+1}$, $\tau = t^n$, and $\tau = t^{n-1}$, represent the flow dynamics after distinct interstices of $\Delta\tau_2$ and

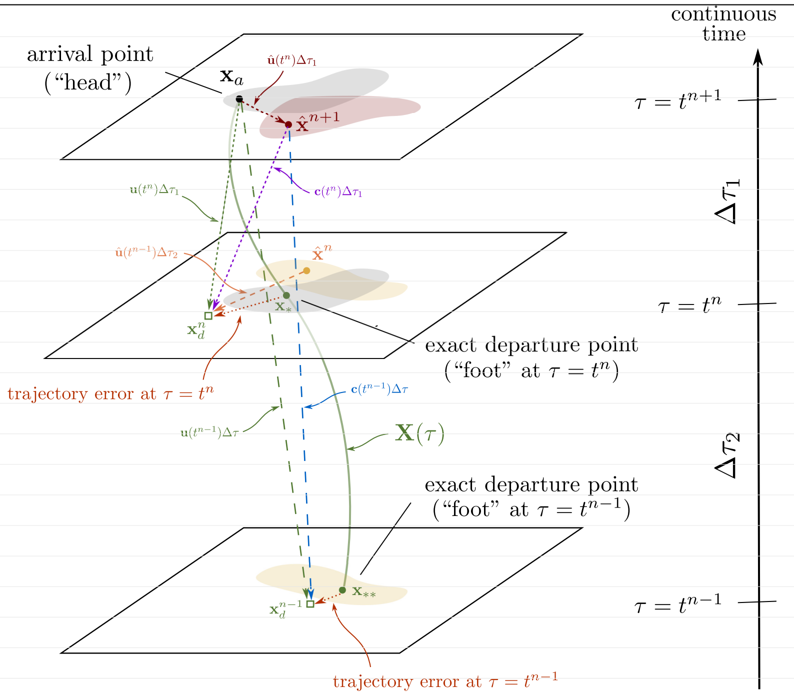

continuous time
arrival point (“head”)
$\mathbf{x}_a$
$\hat{\mathbf{u}}(t^n)\Delta\tau_1$
$\hat{\mathbf{x}}^{n+1}$
$\tau = t^{n+1}$
$\mathbf{u}(t^n)\Delta\tau_1$
$\mathbf{c}(t^n)\Delta\tau_1$
$\Delta\tau_1$
$\hat{\mathbf{u}}(t^{n-1})\Delta\tau_2$
$\hat{\mathbf{x}}^n$
$\mathbf{x}_*$
$\mathbf{x}_d^n$
$\tau = t^n$
exact departure point (“foot” at $\tau = t^n$)
trajectory error at $\tau = t^n$
$\mathbf{c}(t^{n-1})\Delta\tau$
$\mathbf{u}(t^{n-1})\Delta\tau$
$\mathbf{X}(\tau)$
$\Delta\tau_2$
exact departure point (“foot” at $\tau = t^{n-1}$)
$\mathbf{x}_{**}$
$\mathbf{x}_d^{n-1}$
$\tau = t^{n-1}$
trajectory error at $\tau = t^{n-1}$

FIGURE 3 Illustration of the ALE/FEM semi-lagrangian integration occurring over a space-time domain. The arrival point is the “head” of the characteristics $\mathbf{X}(\tau)$, which describes the particle motion under the Lagrangian viewpoint. The Eulerian locus (area shaded in gray) is maintained fixed with time, but it is abstractly distorted at each time before the next advection is updated (areas shaded in red and yellow). In a purely Lagrangian motion, the characteristic carries “foot” points to the arrival point by forming abstract meshes to which x_* and x_{**} belong. However, such points are approximated by off-mesh departure points $x^n_{d,L}$ and $x^{n-1}_{d,L}$ found from backward vector shifts.

$\Delta\tau_1$. The arrival point $\mathbf{x}_a$ lies in $\tau = t^{n+1}$ and it is the “head” of the characteristic line $\mathbf{X}(\tau)$ (green solid line) of a fluid particle that visited the points $\mathbf{x}_*$, at t_n, and $\mathbf{x}_{**}$, at t_{n-1}. In order to integrate the trajectory, a first-level backward search tries to pursue $\mathbf{x}_*$ exactly, but this is hampered by numerical errors that leads the integration to an approximated departure point $\mathbf{x}^n_d$.

In the case of a pure Lagrangian motion ($\hat{\mathbf{u}} = \mathbf{0}$), it is necessary only to have a unique displacement $\boldsymbol{\alpha} = \mathbf{u}(t^n)\Delta\tau_1$ from an Eulerian locus (gray-shaded area) to reach the approximated “foot” point one “step” behind. In reality, the Eulerian locus is required to distort back onto an abstract locus at t^n (omitted in the figure) to which $\mathbf{x}_*$ belongs. However, in virtue of the arbitrary mesh motion ($\hat{\mathbf{u}} \neq \mathbf{0}$), before this distortion occurs, an intermediary displacement abstractly moves $\mathbf{x}_a$ to $\hat{\mathbf{x}}^{n+1}$, so that the computed departure point is a composition of displacement vectors. By writing $\boldsymbol{\alpha}_1 = \hat{\mathbf{u}}(t^n)\Delta\tau_1$, and $\boldsymbol{\alpha}_2 = \mathbf{c}(t^n)\Delta\tau_1$, we find that

$$\mathbf{x}_* \approx \mathbf{x}^n_d = \mathbf{x}_a - (\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2). \tag{12}$$

This way, the mesh velocity is embedded into the material derivative approximation by considering the arbitrary geometrical locus (red-shaded area) which is connected to the ALE approach. Depending on the flow configuration, $\hat{\mathbf{x}}^{n+1}$ can be aligned toward the projection of $\mathbf{x}^n_d$ (at the slice t^{n+1}) to bring $\boldsymbol{\alpha}_2$ closer to $\boldsymbol{\alpha}$. This takes place, for instance, when bubbles, drops or dispersed bodies are simulated, in which the mesh motion follows the flow direction and motion trends tightly. Elsewhere, $\hat{\mathbf{x}}^{n+1}$ moves arbitrarily to any other direction within acceptable limits that do not damage the mesh construction.

Departure points will generally occupy different positions. When moving back to previous time instants, we can have as many slices as desired. However, as said before, the true departure points are numerically approximated by off-mesh points resulting from backward displacements. A second-level backward search is also represented in Fig. 3. We consider a two-time integration that can be explained alike. Now, to have an approximation $\mathbf{x}_d^{n-1}$ to $\mathbf{x}_{**}$, the mesh motion occurring at $\tau = t^n$ (yellow-shaded area) must be embodied into $\mathbf{x}_d^n$ for a posterior longer search from $\mathbf{x}_a$. The possibilities to devise multi-step schemes here are vast and may cover all family of Runge-Kutta, Adams-like, or BDF-like methods, for example. At our convenience, we adopted an illustration that approaches the SL2 scheme introduced next. The vector algebra for a two-step scheme reads as

$$\mathbf{x}_{**} \approx \mathbf{x}_d^{n-1} = \mathbf{x}_a - (\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_3), \tag{13}$$

with $\boldsymbol{\alpha}_3 = \mathbf{c}(t^{n-1})\Delta\tau = \mathbf{c}(t^{n-1})(\Delta\tau_1 + \Delta\tau_2)$.

In this paper, we consider the ALE viewpoint. Hence, the arrival points move arbitrarily with the mesh velocity $\hat{\mathbf{u}}$ — details on the dynamic meshing and the equation for $\hat{\mathbf{u}}$ are given in?. As seen, the SL/ALE integration is similar to the traditional interpretation of the SL method, but differs as for persistent searches for past locations that advance forward in time to match points that were dynamically displaced due to remeshing operations.

2.5 | Semi-Lagrangian discrete splitting

In the next two subsections, we will derive expressions of SL integrators for the velocity field that govern the advection term in Eq. (1a), but it should be clear that equivalent reasoning holds for every scalar function ψ being transported through the flow field.

2.5.1 | One-step scheme (SL1)

The one-step integrator (SL1) relies on a one-step backward displacement written as

$$\boldsymbol{\alpha} = \int_{t^n}^{t^{n+1}} \mathbf{u}(\mathbf{X}(\tau), \tau)\, d\tau \approx \mathbf{u}_i^n + \Delta t_1 \approx (\hat{\mathbf{u}}_i^n + \mathbf{c}_i^n)\Delta t_1, \tag{14}$$

where $\mathbf{c}_i^n = \mathbf{u}_i^n - \hat{\mathbf{u}}_i^n$. Whenever it is clear, the subscript i – which is related to a mesh node – will be omitted. Using Eq. (11), the discrete version of the material derivative becomes

$$\frac{D\mathbf{u}}{D\tau} \approx \frac{\mathbf{u}^{n+1} - \mathbf{u}_d^n}{\Delta t_1}. \tag{15}$$

Eqs. (8), (10), and (14) can be put together to integrate the characteristics, so that

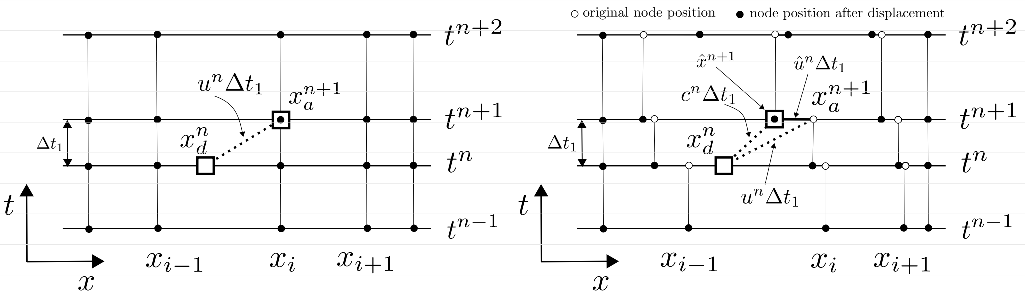
$$\frac{d\mathbf{X}(\tau)}{d\tau} \approx \frac{\mathbf{x}_a^{n+1} - \mathbf{x}_d^n}{\Delta t_1} = \mathbf{u}^n \Rightarrow \mathbf{x}_d^n = \mathbf{x}_a^{n+1} - (\hat{\mathbf{u}}^n + \mathbf{c}^n)\Delta t_1 \tag{16}$$

is a first-order scheme. The current velocity $\mathbf{u}^{n+1}$ becomes the unknown field and $\mathbf{u}_d^n$ is the departure velocity, which must be computed with a search and interpolation procedure. In fact, the computation of $\mathbf{u}_d^n$ is the most costly operation. Additionally, we should note that the time step Δt_1 may vary arbitrarily due to the mesh motion, since the characteristic mesh size may reduces and, consequently, affecting the time restriction imposed by the explicitly treatment of surface tension. Once all unique departure points are computed, the flow variables must be interpolated at these points. In particular, the velocity field is found by using the $\mathbf{u}$-interpolant defined in Eq. (7a) as

$$\mathbf{u}_d^n = \mathcal{I}[\mathbf{u}(\mathbf{x}_j^n)], \tag{17}$$

or $\mathbf{u}(\mathbf{x}_d, t^n) = \mathcal{I}[\mathbf{u}(\mathbf{x}_j, t^n)]$, in extended notation (see Fig. 5).

Figure 4 shows the behavior of discrete trajectories over one-dimensional space-time stencils for the conventional fixed mesh Eulerian framework (left) and the dynamic mesh ALE framework (right). The key differences between both rely on the characteristic velocity used to find the departure points and on the distribution of the mesh nodes. While the nodes may change with time in the ALE framework, i.e. $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}$, they are maintained fixed in the Eulerian framework provided that $\mathbf{c} = \mathbf{u}$.



(a) SL1: Eulerian framework.

(b) SL1: ALE framework.

FIGURE 4 One-dimensional space-time stencils representing the first-order SL scheme. The point x_d is found by integrating the fluid trajectories backward-in-time. Afterward, the flow variables are interpolated by considering the degrees of freedom placed at the nodes x_{i-1} and x_i.

Figure 5 represents, for a single node, the methodology used to solve Eq. (16) along with the search and interpolation procedure based on the neighbor-to-neighbor technique. As said before, the departure point $\mathbf{x}_d$ does not match a mesh node. Therefore, the algorithm should be able to detect points belonging to the elements. The strategy of such algorithm follows the nearest node direction in which each vertex at the neighbor element is evaluated to check the distance from the known points x_i^n.

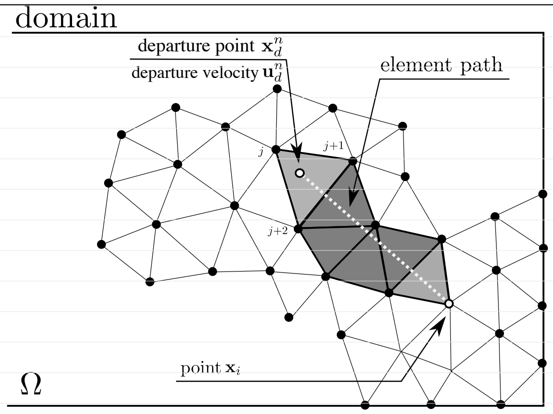

domain
departure point $\mathbf{x}_d^n$
departure velocity $\mathbf{u}_d^n$
element path
j
$j+1$
$j+2$
point $\mathbf{x}_i$
Ω

FIGURE 5 Methodology of nearest node direction illustrated in an unstructured mesh of triangular elements. In the SL1 scheme, the search algorithm finds departure points over a unique element path patch. Elements in light gray contain either the arrival node $\mathbf{x}_a^{n+1}$ (fixed mesh) or $\hat{\mathbf{x}}^{n+1}$ (moving mesh), which matches the mesh node $\mathbf{x}_i$ mandatory, or the departure node $\mathbf{x}_d^n$. In this illustration, the velocity $\mathbf{u}_d^n$ at the departure point $\mathbf{x}_d^n$ is interpolated by velocities at the nodal indices j, $j+1$, and $j+2$.

The final form of the discrete finite element matrix equations embedding the first-order approximation provided by the SL1 scheme can be described by

$$\left(\frac{\mathbf{M}_\rho}{\Delta t_1} + \frac{1}{Ar^{1/2}}\mathbf{K}\right)\mathbf{u}^{n+1} + \mathbf{G}p^{n+1} = \frac{\mathbf{M}_\rho}{\Delta t_1}\mathbf{u}_d^n + \frac{1}{Eo}\mathbf{f} + \mathbf{M}_\rho\mathbf{g}, \tag{18}$$

where $\mathbf{M}_\rho$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix and the other parameters are the same found in Eq. (1a).

2.5.2 | Two-step scheme (SL2)

One of the objectives of the SL method is to improve the integration accuracy. This can be done by developing high-order formulae to compute $\boldsymbol{\alpha}$ in Eq. (10) and assure that the same order of accuracy in the material derivative is respected at the update. As opposed to the previous SL1 scheme, we propose a two-step scheme (from now on SL2) that performs an extra backward integration of the fluid trajectories. The SL2 integrator requires two displacement vectors, namely

$$\boldsymbol{\alpha}_{2,1} = \int_{t^n}^{t^{n+1}} \mathbf{u}(\mathbf{X}(\tau),\tau)\,d\tau \approx [\hat{\mathbf{u}}^{n+\frac{1}{2}} + \mathbf{c}^{n+\frac{1}{2}}]\Delta t_2 \text{ and} \tag{19a}$$

$$\boldsymbol{\alpha}_{2,2} = \int_{t^{n-1}}^{t^{n+1}} \mathbf{u}(\mathbf{X}(\tau),\tau)\,d\tau \approx 2(\hat{\mathbf{u}}^n + \mathbf{c}^n)\Delta t_1. \tag{19b}$$

(19c)

Likewise Eq. (15), we define a second-order approximation for the material derivative as:

$$\frac{D\mathbf{u}}{D\tau} \approx \frac{d_1\mathbf{u}^{n+1} - d_2\mathbf{u}_d^n + d_3\mathbf{u}_d^{n-1}}{\Delta t_1}, \tag{20}$$

where $\mathbf{u}^{n+1}$ continues to be the current velocity, $\mathbf{u}_d^n$ is the departure velocity at t^n, and $\mathbf{u}_d^{n-1}$, ditto at t^{n-1}, which should be the ultimate velocity responsible for updating the advection. Both departure velocities are unknown at this moment. Since Eq. (20) is approximated by a second-order time finite difference, $\mathbf{u}_d^n$ and $\mathbf{u}_d^{n-1}$ depend on a two-stage time splitting procedure of search and interpolation within Δt_1. Each stage divides Δt_1 into the first half, from t^n to $t^{n+\frac{1}{2}}$, and the second half, from $t^{n+\frac{1}{2}}$ to t^{n+1}. It is usual to have varying time steps when simulating realistic cases of multiphase flows. In this case, the coefficients appearing in Eq. (20) are defined as

$$d_1 = \frac{\left[(\Delta t_1 + \Delta t_2)^2 - \Delta t_1^2\right]}{\Delta t_2(\Delta t_1 + \Delta t_2)} \tag{21a}$$

$$d_2 = \frac{(\Delta t_1 + \Delta t_2)^2}{\Delta t_2(\Delta t_1 + \Delta t_2)} \tag{21b}$$

$$d_3 = \frac{\Delta t_1^2}{\Delta t_2(\Delta t_1 + \Delta t_2)}. \tag{21c}$$

In case of constant time steps, $d_1 = 3/2$, $d_2 = 2$, and $d_3 = 1/2$.

For the first backward integration in SL2, Eqs. (8), (10), (14), and (19a) lead to

$$\begin{aligned} \frac{d\mathbf{X}(\tau)}{d\tau} &\approx \frac{\mathbf{x}_a^{n+1} - \mathbf{x}_d^n}{\Delta t_2} = \hat{\mathbf{u}}^{n+\frac{1}{2}} + \mathbf{c}^{n+\frac{1}{2}} \\ &\Rightarrow \mathbf{x}_d^n = \mathbf{x}_a^{n+1} - \boldsymbol{\alpha}_{2,1} \\ &\Rightarrow \mathbf{x}_d^n = \mathbf{x}_a^{n+1} - [\hat{\mathbf{u}}^{n+\frac{1}{2}} + \mathbf{c}^{n+\frac{1}{2}}]\Delta t_2. \end{aligned} \tag{22}$$

During the first stage of integration, we expect that the departure velocity $\mathbf{u}_d^n$ is estimated through a second-order approximator. However, before computing it, we should note that the relative velocity $\mathbf{c}^{n+\frac{1}{2}}$ where the dynamics of the characteristic is embedded is undetermined at $t^{n+\frac{1}{2}}$, thereby requiring extrapolation. To maintain the order of accuracy, we used a second order extrapolation that estimates

$$\mathbf{c}^{n+1/2} = \left(\frac{\Delta t_1 + 2\Delta t_2}{2\Delta t_1}\right)\mathbf{c}^n - \frac{1}{2}\mathbf{c}^{n-1} \tag{23}$$

for varying time steps, or

$$\mathbf{c}^{n+1/2} = \frac{3}{2}\mathbf{c}^n - \frac{1}{2}\mathbf{c}^{n-1} \tag{24}$$

for constant time steps. The first stage is concluded while updating the departure velocity at $\mathbf{u}_d^n$ by applying the $\mathbf{u}$-interpolant in the same way as in Eq. (17).

The second stage consists in finding the location of the departure point $\mathbf{x}_d^{n-1}$ by using a BDF2 scheme for the characteristics whose accuracy is second-order time??. Then, for the second backward integration in SL2, an equation with similar form to

Eq. (22) can be written as

$$\begin{aligned} \frac{d\mathbf{X}(\tau)}{d\tau} &\approx \frac{\mathbf{x}_a^{n+1} - \mathbf{x}_d^{n-1}}{2\Delta t_1} = \hat{\mathbf{u}}^n + \mathbf{c}^n \\ &\Rightarrow \mathbf{x}_d^{n-1} = \mathbf{x}_a^{n+1} - \boldsymbol{\alpha}_{2,2} \\ &\Rightarrow \mathbf{x}_d^{n-1} = \mathbf{x}_a^{n+1} - 2(\hat{\mathbf{u}}^n + \mathbf{c}^n)\Delta t_1. \end{aligned} \tag{25}$$

However, Eq. (25) corresponds to discretizations with constant time step. For varying time steps, it must be modified[?] to

$$\begin{aligned} \frac{d\mathbf{X}(\tau)}{d\tau} &\approx \frac{\mathbf{x}_a^{n+1} + \left[(\Delta t_1/\Delta t_2)^2 - 1\right]\mathbf{x}^n - (\Delta t_1/\Delta t_2)^2 \mathbf{x}_d^{n-1}}{\left[(\Delta t_1/\Delta t_2) + 1\right]\Delta t_1} = \hat{\mathbf{u}}^n + \mathbf{c}^n \\ &\Rightarrow \mathbf{x}_d^{n-1} = \left(\frac{1}{\theta^2}\right)\mathbf{x}_a^{n+1} + \left(\frac{\theta^2 - 1}{\theta^2}\right)\mathbf{x}_d^n - \left(\frac{\theta + 1}{\theta^2}\right)\Delta t_1[\hat{\mathbf{u}}^n + \mathbf{c}^n], \\ &\text{with } \theta = \frac{\Delta t_1}{\Delta t_2}. \end{aligned} \tag{26}$$

Once all the secondary departure points are computed, the flow variables must be interpolated at these points. In particular, the velocity field is updated by using the **u**-interpolant defined in Eq. (7a). Points at t^{n-1} are reached with nodal velocities evaluated at t^n.

$$\mathbf{u}_d^{n-1} = \mathcal{I}[\mathbf{u}(\mathbf{x}_i^n)], \tag{27}$$

or $\mathbf{u}(\mathbf{x}_d, t^{n-1}) = \mathcal{I}[\mathbf{u}(\mathbf{x}_i, t^n)]$, in extended notation.

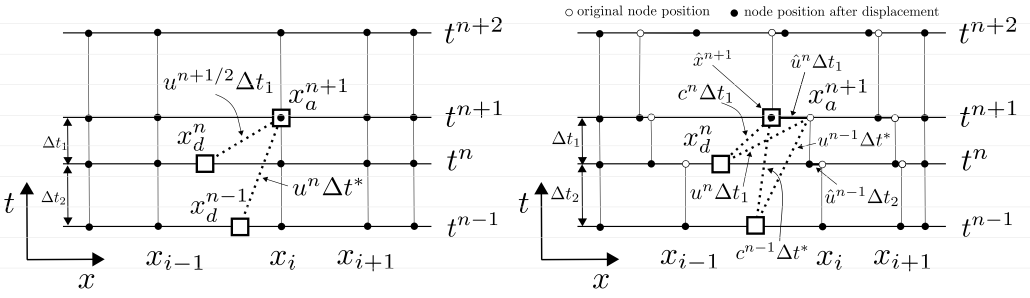
Likewise Fig. 4, Fig. 6 shows the behavior of discrete trajectories over one-dimensional space-time stencils for the conventional fixed mesh Eulerian framework (left) and the dynamic mesh ALE framework (right) now for the second-order time scheme. As seen, the core difference between SL1 and SL2 is an extra search required to find the ultimate departure point $\mathbf{x}_d^{n-1}$ and, consequently, the interpolated velocity $\mathbf{u}_d^{n-1}$ used to update the material derivative.

Figure 7 is an extension of Fig. 5 that depicts the methodology used to solve Eq. (20) with the additional search and interpolation procedure based on the neighbor-to-neighbor technique. Once again, the departure point $\mathbf{x}_d^{n-1}$ does not necessarily match a mesh node. Therefore, the algorithm should be able to detect inner points to the elements.

In this scheme, the discrete finite element matrix equations embed the second-order approximation for the general case of varying time step. Their final form is given by

$$\left(d_1 \frac{\mathbf{M}_\rho}{\Delta t_1} + \frac{1}{Ar^{1/2}}\mathbf{K}\right)\mathbf{u}^{n+1} + \mathbf{G}p^{n+1} = d_2 \frac{\mathbf{M}_\rho}{\Delta t_1}\mathbf{u}_d^n - d_3 \frac{\mathbf{M}_\rho}{\Delta t_1}\mathbf{u}_d^{n-1} + \frac{1}{Eo}\mathbf{f} + \mathbf{M}_\rho \mathbf{g} \tag{28}$$

where the parameters are the same defined in Eqs. (1a) and (18).



(a) SL2: Eulerian framework.

(b) SL2: ALE framework.

FIGURE 6 One-dimensional space-time stencils representing the second-order SL scheme for varying time step in two frameworks: Eulerian and ALE. The point x_d^n is found with a backward integration that uses the extrapolated ALE velocity $\mathbf{c}^{n+1/2}$ to determine the first displacement vector at t^n and complete the first stage of integration. For the second stage, this process is repeated, but now x_d^{n-1} is found with a backward integration whose displacement vector moves back over two time steps. In this latter situation, the velocity used to step back is the ALE velocity $\mathbf{c}^n$. The update of the advective transport occurs in two steps as well. Firstly, u_d^n is interpolated by using the velocity field at t^n; secondly, u_d^{n-1} is interpolated at t^{n-1} but with those nodal velocities stored at t^n. This way, the ultimate advection is updated toward t^{n+1} by averaging u_d^n and u_d^{n-1} as shown in Eq. (20).

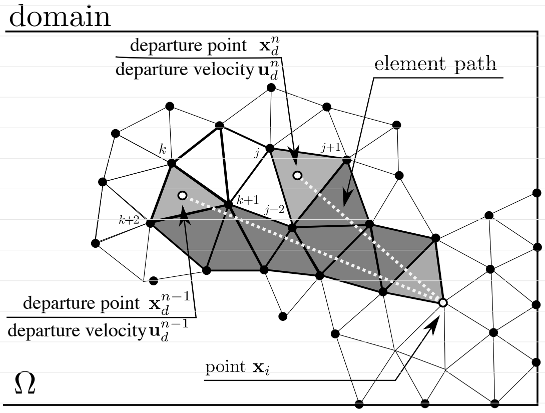

domain
departure point $\mathbf{x}_d^n$
departure velocity $\mathbf{u}_d^n$
element path
k
j
$j+1$
$k+1$
$j+2$
$k+2$
departure point $\mathbf{x}_d^{n-1}$
departure velocity $\mathbf{u}_d^{n-1}$
point $\mathbf{x}_i$
Ω

FIGURE 7 Methodology of nearest node direction illustrated in an unstructured mesh of triangular elements. In the SL2 scheme, the algorithm searches for departure points over two element path patches. Elements shaded in light gray contain either the arrival node $\mathbf{x}_a^{n+1}$ (fixed mesh) or $\hat{\mathbf{x}}^{n+1}$ (moving mesh), which matches the mesh node $\mathbf{x}_i$, or the departure node. In this illustration, the velocity $\mathbf{u}_d^n$ at the departure point $\mathbf{x}_d^n$ is interpolated by velocities at the nodal indices j, $j+1$, and $j+2$., whereas the velocity $\mathbf{u}_d^{n-1}$ at the departure point $\mathbf{x}_d^{n-1}$ is interpolated by velocities at the nodal indices k, $k+1$, and $k+2$.

2.6 | Implementation synopsis

The key differences of the schemes aforementioned can be described through an implementation synopsis as follows. To carry out the SL1 scheme, the following procedure can be performed:

In the last step, $\mathcal{F}_1(t^n, Ar, Eo, \mathbf{f}, \mathbf{g})$ is only a generic functional form that embeds all the terms resulting from the uncoupled solution of Eqs. (1a-1a) by a projection method?,?. This way, the update step becomes similar to the classical Euler's method for initial value problems.

Algorithm 1 1st. order ALE-FE Semi-Lagrangian method

```
Compute the trajectory: $\mathbf{x}_d^n \longrightarrow \mathbf{x}_d^n = \mathbf{x}_a^{n+1} - \Delta t_1 \mathbf{c}_i^n$
Interpolate the velocity: $\mathbf{u}_d^n = \mathcal{I}[\mathbf{u}(\mathbf{x}_i^n)]$
Update the advective transport: $\mathbf{u}^{n+1} = \mathbf{u}_d^n + \Delta t_1 \mathcal{F}_1(t^n, Ar, Eo, \mathbf{f}, \mathbf{g})$
```

On the other hand, to implement the SL2 scheme, we use the following steps:

Algorithm 2 2st. order ALE-FE Semi-Lagrangian method

```
Extrapolate the velocity: $\mathbf{c}_i^{n+\frac{1}{2}} = [(\Delta t_1 + 2\Delta t_2)/2\Delta t_1]\mathbf{c}_i^n - (1/2)\mathbf{c}_i^{n-1}$
Compute the 1st. trajectory: $\mathbf{x}_d^n \longrightarrow \mathbf{x}_d^n = \mathbf{x}_a^{n+1} - \Delta t_1 \mathbf{c}_i^{n+\frac{1}{2}}$
Interpolate the 1st. departure velocity: $\mathbf{u}_d^n = \mathcal{I}[\mathbf{u}(\mathbf{x}_i^n)]$
Compute the 2nd. trajectory: $\mathbf{x}_d^{n-1} = \left(\frac{1}{\theta^2}\right)\mathbf{x}_a^{n+1} + \left(\frac{\theta^2 - 1}{\theta^2}\right)\mathbf{x}_d^n - \left(\frac{\theta + 1}{\theta^2}\right)\Delta t_1 \mathbf{c}^n$
Interpolate the ultimate velocity: $\mathbf{u}_d^{n-1} = \mathcal{I}[\mathbf{u}(\mathbf{x}_i^n)]$
Update the material derivative: $d_1\mathbf{u}^{n+1} = d_2\mathbf{u}_d^n - d_3\mathbf{u}_d^{n-1} + \Delta t_1 \mathcal{F}_2(t^n, Ar, Eo, \mathbf{f}, \mathbf{g}, d_2, d_3)$
```

In the last step, $\mathcal{F}_2(t^n, Ar, Eo, \mathbf{f}, \mathbf{g}, d_2, d_3)$ is a generic functional differing from $\mathcal{F}_1$ of SL1 in a few terms. Table 1 summarizes the fundamental information used for both integrators to approximate the fluid flow particle trajectories.

TABLE 1 Summary of parameters and values used to implement the semi-lagrangian integrators SL1 and SL2 in constant and varying time steps.

semi-lagrangian scheme	displacement vector			constant time step coefficients			varying time step coefficients		
	α_1	$\alpha_{2,1}$	$\alpha_{2,2}$	d_1	d_2	d_3	d_1	d_2	d_3
SL1	$c_i^n \Delta t_1$	0	0	1	1	0	1	1	0
SL2	0	$c_i^{n+\frac{1}{2}} \Delta t_2$	$2c_i^n \Delta t_1$	3/2	2	1/2	$\frac{[(\Delta t_1 + \Delta t_2)^2 - \Delta t_1^2]}{\Delta t_2(\Delta t_1 + \Delta t_2)}$	$\frac{(\Delta t_1 + \Delta t_2)^2}{\Delta t_2(\Delta t_1 + \Delta t_2)}$	$\frac{\Delta t_1^2}{\Delta t_2(\Delta t_1 + \Delta t_2)}$

3 | RESULTS

We now present a series of numerical simulations of single- and two-phase flows based on the SL1 and SL2 schemes on the ALE framework. All the tests presented next were carried out using the same configurations for remeshing criteria, time step, geometry, and initial/boundary conditions, thus keeping consistency for the comparisons. We simulated the passive scalar transport for two distinct geometries and assessed the error curves. The time step is kept constant at each simulation and computed according to the number of iterations. Several time steps were reported.

The following test cases were focused in two-phase flow systems, where the interfacial dynamics plays an important role. For such tests, a variable time step has been chosen according to the following constraint:

$$dt < \left[\frac{\overline{\rho}h^3\mathrm{Eo}}{2\pi}\right] \tag{29}$$

where $\overline{\rho}$ is the average fluid density between both fluids, Eo is the Eötvös number as defined previously and h is the mesh characteristic length. Since both interface and background meshes move with respect to the ALE velocity, a constant time step may not be accomplished due to the deformation of the meshes and, consequently, the variation of the characteristic length h. The rising of a confined single air bubble in sucrose solution is investigated and compared to the well-know reference[?]. Next, we analyzed bubble's center of mass, liquid film thickness, and bubble shape for both integrators. Finally, the rising of a single air bubble through a periodic constricted channel is simulated. Dynamical system techniques are undertaken to evaluate stability and accuracy of the method's responses.

3.1 | 2D advection in rotating flows

The transport of a scalar field over a two-dimensional rotating flow is now considered here. The aim of this test case is to assess and compare the error between SL1 and SL2 schemes for a scalar field subjected to a known rotating flow. We solved the advection-diffusion equation by using the SL/FEM approach. The dimensionless form of the equation in Cartesian coordinates is given by:

$$\frac{\partial\psi(x,y)}{\partial\tau} + \mathbf{c}\cdot\nabla\psi(x,y) = \frac{1}{Pe}\nabla^2\psi(x,y), \tag{30}$$

where $\psi(x,y)$ is the scalar field, Pe is the Péclet number commonly defined in the literature, which is set to $Pe = 10000$, therefore reducing the diffusion effect. In this case, the convective velocity $\mathbf{c} = \mathbf{v} - \hat{\mathbf{v}}$ is not computed. Instead, it is prescribed as base flow for the transport of the scalar field and given by $\mathbf{v}(x,y) = (y,-x)$ to create a clockwise rotating flow. To accurately assess the solution produced by SL1 and SL2 schemes, the mesh velocity $\hat{\mathbf{v}} = 0$. That is to say, the solution of the convection is purely conveyed by the SL term.

For these simulations, we considered an unstructured triangular mesh having 109510 nodes and 24230 triangular (`cubic`) elements. We have also tested a finer and coarser meshes and verified that the same error trend is noted. By choosing the total simulation time $T = 2\pi n_{rev}$ – where n_{rev} is the number of revolutions – and time step size $\Delta t = T/200$, two complete rigid body rotations are carried out with $n_{rev} = 2$ within $nIter$ iterations with $nIter = 2, 5, 10, 20, 40, 80, 100, 200, 400, 800, 1200, 1600, 2000, 4000, 15000$, thereby resulting in Δt ranging from 6.28319 to 8.377×10^{-4}. To compute the trajectory error, we calculated the difference between the exact center of mass' trajectory (perfect circumference) and the trajectory of the computed center of mass (spiral). The center of mass is determined from the disk's area by using

$$\mathbf{x_c} = \frac{\int_A \Psi(x, y)\mathbf{x}\, \mathrm{dA}}{\int_A \Psi(x, y)\, \mathrm{dA}}. \tag{31}$$

3.1.1 | Gaussian function

In this test case, the initial scalar field is given by the Gaussian function

$$\psi(x, y) = A \exp\left\{ -\frac{1}{2}\left[\left(\frac{x - \mu_x}{\sigma_x} \right)^2 + \left(\frac{y - \mu_y}{\sigma_y} \right)^2 \right] \right\}, \tag{32}$$

where $A = 1$ is the scalar amplitude, $\mu_x = 0.0$ and $\mu_y = 0.5$ are the center of the Gaussian function (or expected values), and σ_x^2 and σ_y^2 are variances corresponding to the x and y direction respectively. We set $\sigma_x^2 = \sigma_y^2 = 0.64$, as depicted in the Fig. 8, resulting in $\mathbf{x_c} = \{-4.38495 \times 10^{-4}, 5.00546 \times 10^{-1}\}$ as for its center of mass. The numerical domain is a circle with diameter $\phi = 2D$, where $D = 1$ is the non-dimensional reference diameter. A homogeneous Dirichlet condition $\psi(x, y) = 0$ was imposed over the boundary Γ.

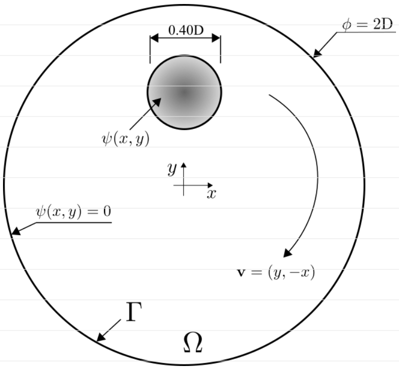


FIGURE 8 Diagram of the simulation domain for the Gaussian function transport. The velocity field is known and imposed by $\mathbf{v}(x, y) = (y, -x)$. Therefore, a clockwise rotating flow is achieved. Two full rotations are exhibited during the simulations.

Figure 9 informs the numerical error for both integrators and examines the accuracy order in time for trajectory and volume. As expected, SL1 behaves as a first-order time scheme, whereas SL2 shows a second-order behaviour.

3.1.2 | Zalezak's disk

A more challenging test case is presented next. Again, the transport of the scalar field $\psi(x, y)$ is investigated along with a complex fluid motion. The Zalesak's disk is used to evaluate the correctness of the transport mediated by SL1 and SL2 schemes. The geometry of the Zalesak's disk should ideally remain constant with time, since $Pe = 10000$. However, due to discretization errors in both space and time, artificial diffusion may appear and deteriorate the disk's original shape by deforming it, especially due to the sharp edges found inside the circle. Figure 10 illustrates the simulation domain, geometric relations, as well as the initial and

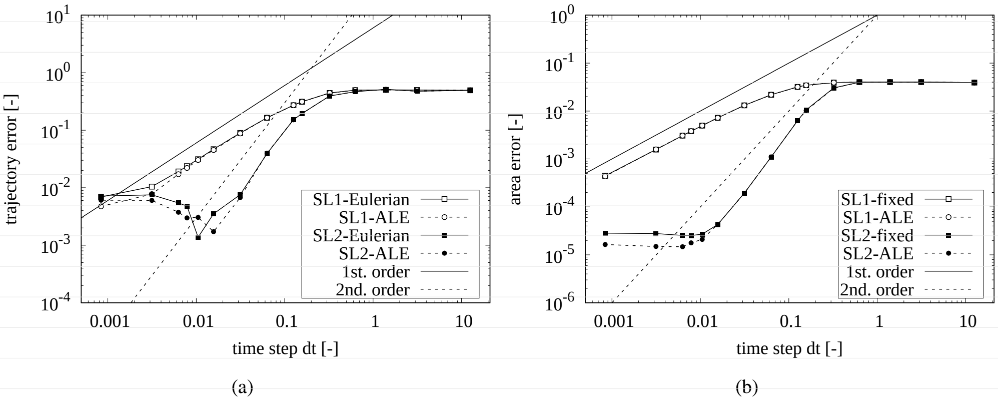


FIGURE 9 Numerical errors over time computed after 2 revolutions of the Gaussian profile test for SL1 and SL2 schemes: (a) trajectory error; (b) area error.

boundary conditions for the problem. Note that with the proposed geometric configuration, $\mathbf{x_c} = \{-4.38495 \times 10^{-4}, 5.00546 \times$ $-1\}$ is found to be the center of mass. As seen, the Zalesak's disk is defined as a circle with radius $R = 0.30D$ slotted by a rectangle measuring $0.12D \times 0.20D$. The same parameters as those for the Gaussian function are replicated here.

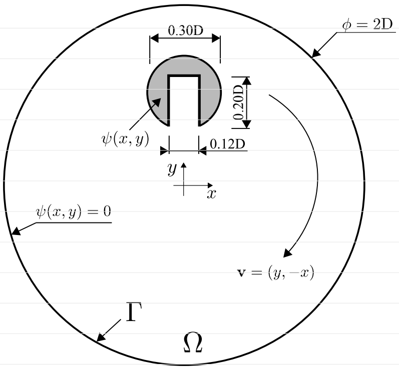


FIGURE 10 Diagram of the simulation domain for the Zalesak's disk transport. The velocity field is known and imposed by $\mathbf{v}(x, y) = (y, -x)$. Therefore, a clockwise rotating flow is achieved. Two full rotations are exhibited during the simulations.

Figure 11 presents the numerical error for both integrators for the Zalesak's disk test. Likewise, SL1 behaves as a first-order time scheme, whereas SL2 is second-order time. A qualitative comparison for the Zalesak's disk test case was also included in Fig. 12. The solution of the SL1 scheme for the final simulation time $t = 200\Delta t$ after two revolutions shows a non-conservative geometric shape (Fig. 12d) and mass loss due to numerical diffusion. Such an undesired effect stems from the approximation of the particle's trajectory by the SL1 method. On the other hand, the SL2 integrator was able to reduce efficiently the numerical dissipation (Fig. (12h) for both departure velocities $\mathbf{u}_d^n$ and $\mathbf{u}_d^{n-1}$. Consequently, the mass is better preserved in this scheme, as

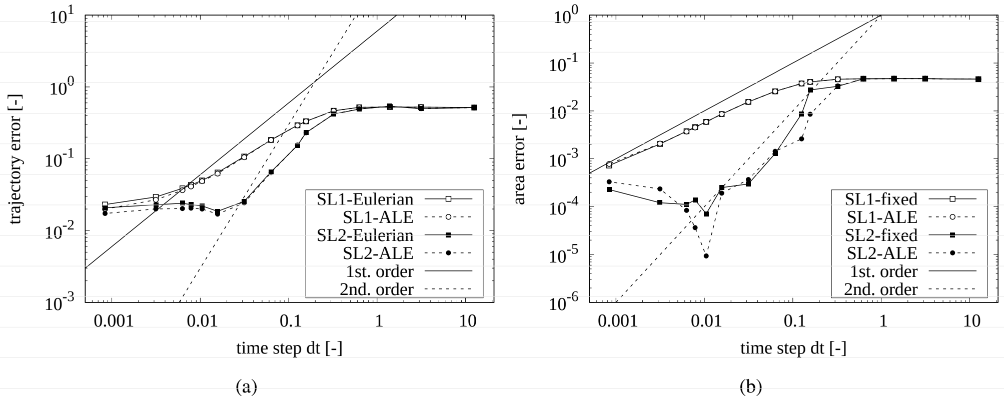
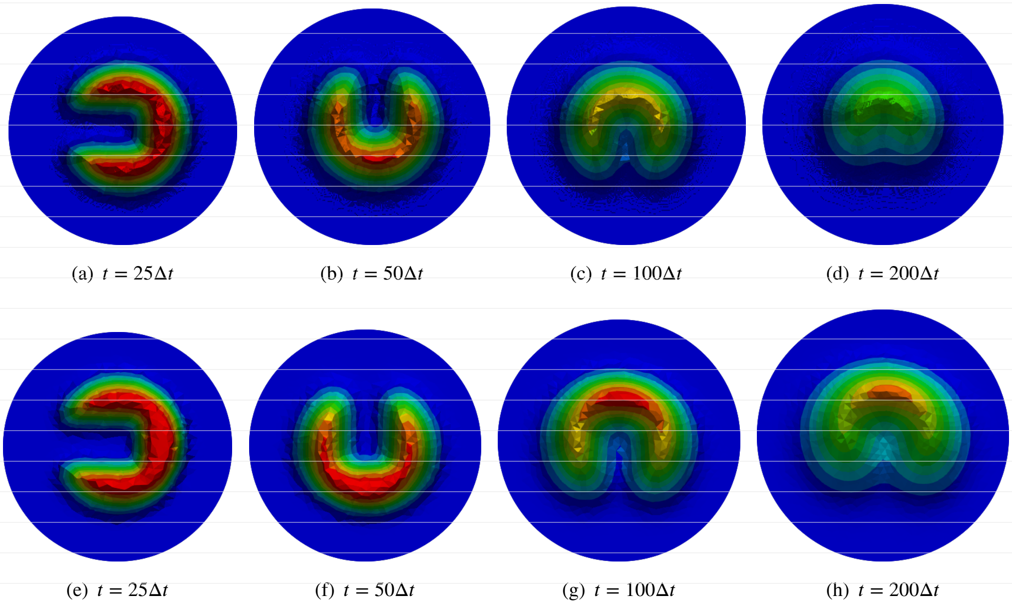


FIGURE 11 Numerical errors over time computed after 2 revolutions of the Zalesak's disk test for SL1 and SL2 schemes: (a) trajectory error; (b) area error. As can be seen, in the SL1 scheme the convergence error is found to be linear for the area error and sub-linear for the trajectory error. On the other hand, the SL2 scheme presented second order of convergence in the area error and sub-quadractic order of convergence for the trajectory error.

expected. In the current test case, the convergence error for the SL1 scheme is found to be linear for the area error and sub-linear for the trajectory error. On the other hand, the SL2 scheme presented second order of convergence in the area error and sub-quadratic order of convergence for the trajectory error.


(a) $t = 25\Delta t$ (b) $t = 50\Delta t$ (c) $t = 100\Delta t$ (d) $t = 200\Delta t$

(e) $t = 25\Delta t$ (f) $t = 50\Delta t$ (g) $t = 100\Delta t$ (h) $t = 200\Delta t$

FIGURE 12 Comparison snapshots of Zalesak's disk profiles during revolutions impelled by the rotation flow for several simulation times within 200 iterations. First row: scalar blob approximated by the SL1 scheme. Second row: scalar blob approximated by the SL2 scheme.

In both scalar transport test cases presented here, both SL1 and SL2 schemes presented a plateau-like behaviour for the convergence of error when using large time steps $dt > 0.1$. A lower bound plateau-like behaviour is also found where very small time steps were used $dt < 0.01$, therefore indicating sup-inf limits for best usage of the proposed methodology in fluid flows according to the average mesh size $\overline{h}$ adopted in the simulations, i.e. $\overline{h} = 0.016$ for the current test.

3.2 | Rising of Taylor air bubble

The rising velocity of a single Taylor air bubble in sucrose solution is investigated according to the experimental setup by?. In this case, velocity at the bubble's center of mass and liquid film thickness are evaluated for both integrators (SL1 and SL2) to compare their performance, accuracy, and correctness. The chosen two-phase system is sucrose solution as suspending fluid with density $\rho_{out} = 1172\text{kg/m}^3$ and dynamic viscosity $\mu = 5.65 \times 10^{-6}\text{Pa} \cdot \text{s}$, and air with density $\rho_{out} = 1.789\text{kg/m}^3$, dynamic viscosity $\mu = 1.225 \times 10^{-6}\text{Pa} \cdot \text{s}$. The dimensionless numbers were set to $Eo = 30$ and $Mo = 10^{-7}$, thus resulting in the Archimedes number $Ar = 519615$. The air bubble flows along the channel while a thin liquid film fills in the gap between the wall an the bubble's lateral surface, thus preventing gas dry-out. Simulations that represent the liquid film thickness correctly are hard to be performed. Despite of that, we show that the numerical code is robust, since it delivers reasonable outcomes while respecting the mass balance, bubble shape, and terminal rising velocity.

Figure 13 depicts the domain's geometry and location of the Taylor air bubble. We assumed a circular channel with dimensionless diameter $D = 1$. The total channel length is $20D$ wide and the bubble's nose is placed at $z = 16.4D$. The initial bubble's length is $5.1D$. From the initial liquid film thickness of $\delta_0 = 0.12D$, we monitored its evolution with time for purposes of comparison. A moving frame reference is adopted here to simulate the rising motion of the bubble, i.e. its center of mass remains fixed in space, whereas the axial velocity at the upper and right boundaries are constantly updated as a function of the bubble's axial (rising) velocity.

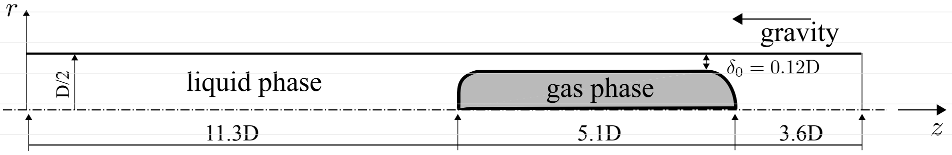


FIGURE 13 Schematic representation of the domain's geometry and initial air bubble's location for the two-phase flow of a rising Taylor air bubble.

Figure 14 shows the temporal evolution of velocity at the Taylor bubble's center of mass for both time integration schemes SL1 and SL2. In spite of its highly transient behavior from the initial bubble's shape to its terminal shape, both schemes present approximately the same solution, resting nearby the final dimensionless terminal velocity $v_{exp} = 0.305$ according to the experimental observations by? after few oscillations in time $t < 10$.

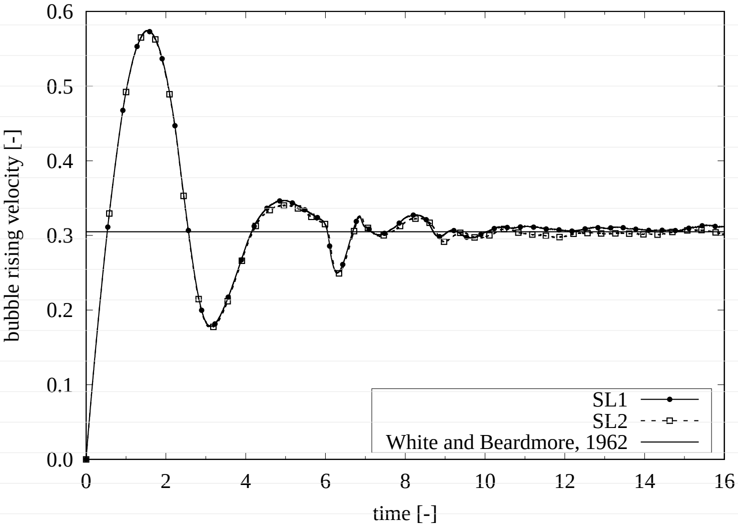


FIGURE 14 Rising velocity profile of a single Taylor air bubble immersed in a sucrose solution for $Mo = 10^{-7}$ and $Eo = 30$ Both profiles show that the bubble's center of mass velocity obtained with schemes SL1 and SL2 are in narrow agreement with experiments.

Additionally, we present the dynamic behavior of the liquid film thickness for the air/sucrose bubble flow in Fig. 15 and compare it with Brown's correlation? . Oscillations for δ are perceived from the threshold $t \approx 4$ within the range $[0.05, 0.06]$. From $t \approx 9$, the thickness practically stagnates around $\delta = 0.058$ for both integration schemes. No significant improvement was verified for the evolution of the liquid film thickness regarding the second-order scheme.

The bubble's terminal shape at $t = 14$ is presented in Fig. 16 for both time integration methods. As observed, the profiles for both time integration methods are identical with bubble length $L_b \approx 5.0$. The final number of mesh nodes and elements for this simulation was 67138 and 14179, respectively, for the SL1 scheme, but 67459 and 14250, respectively, for the SL2 scheme.

3.3 | Motion of drops in periodically constricted capillary

We also studied the rising velocity of a single air bubble immersed in a glycerol-like solution rising along a sinusoidal capillary channel. This test should capture the continuously changing motion that the bubble performs due to the periodic behavior of the channel border. The relevant fact on this case study is the absence of steady solution for both bubble's shape and velocity even for large simulation times. In turn, we can accurately examine the transient regime. Here, we chose the experiments carried out

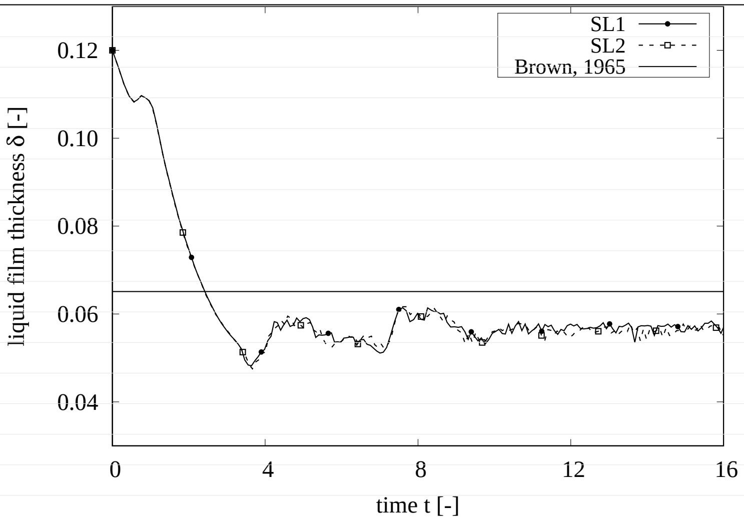


FIGURE 15 Temporal evolution of the liquid film thickness δ for the rising Taylor air bubble immersed in a sucrose solution. This simulation considered $Mo = 10^{-7}$ and $Eo = 40$. The liquid film thickness is computed by using both the SL1 and SL2 schemes and compared afterward with Brown's correlation for elongated bubbles.

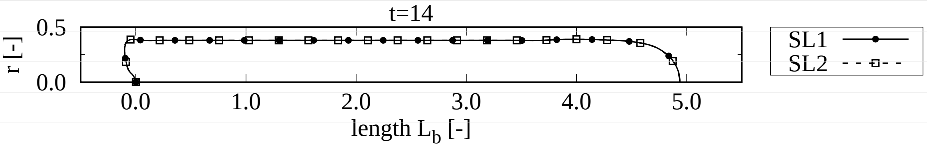


FIGURE 16 Comparison of the air bubble's terminal shape for both time integration schemes moving in a sucrose solution. This simulation considered $Mo = 10^{-7}$ and $Eo = 30$ and the snapshot was taken in dimensionless time $t = 14$.

by? as benchmark. They used the following wave equation to model the corrugations:

$$r_{wall}^{n+1} = \frac{D}{2} + A \sin\left[\frac{2\pi}{\lambda}(x + x_{ref}^{n+1}) - \phi_c\right], \tag{33}$$

where D is the channel's dimensionless diameter, A is the wave amplitude, λ is the wavelength, and ϕ_c is the wave phase. The axial coordinate is x, while x_{ref} refers to the bubble's referential position. Bubble's center of mass remains fixed in space and its referential axial coordinate x_{ref} is incremented each time step n with the axial rising velocity of the bubble u_{ref} as $x_{ref}^{n+1} = x_{ref}^{n} + u_{ref}\, dt$. The initial referential coordinate is set to $x_{ref}^{0} = 0$. The new boundary node position r_{wall}^{n+1} is computed every time step as a function of the bubble's referential coordinate. In this test case, the dimensionless numbers were set to $Eo = 16.97$ and $Ar = 11174$. The dimensionless fluid properties were set to $\rho_{in} = 1.225$, $\rho_{out} = 1350$, $\mu_{in} = 1.78 \times 10^{-5}$, $\mu_{out} = 0.04$, $\sigma = 0.078$, thus resulting in the Morton number $Mo = 3.92 \times 10^{-5}$.

Figure 17 brings a schematic representation of the domain and location of the bubble. The moving frame approach for dynamic boundaries detailed in? was used here to comply with the need for a large numerical domain. Thus, the bubble's center $\mathbf{x}_c$ is placed at the coordinates $(0.0, 5.2)$ and remains fixed in space. The top wall moves according to Eq. (33). Such approach is similar to that of a fixed frame reference where the bubble is constantly moving in the opposite gravity direction. As seen, the bubble shape was initialized as an axisymmetric shape and its center was located at the coordinates $(x_c, z_c) = (0.0, 5.2D)$ to avoid inflow disturbances. The numerical length was set to $8D$ to enable full bubble's wake description. The bubble's radius R_d was computed as a function of the κ_d parameter and the channel diameter as $R_d = \kappa_d\ D/2$. In the current simulation, $\kappa_d = 0.95$. Therefore, the bubble's diameter is 95% of the channel's diameter. In Eq. (33), we set $A = 0.18$, $\lambda = 4$, and $\phi_c = 6.28$. Practically, the phase represents the relative position of the bubble and the channel and does not affect the flow dynamics once the periodic flow is reached. The bubble's center mass axial velocity v_{z_c} is used as a reference to move the top wall by updating z_{ref}^{n+1} at every time step with the motion equation $z_{ref}^{n+1} = z_{ref}^{n} + v_{z_c}\ dt$.

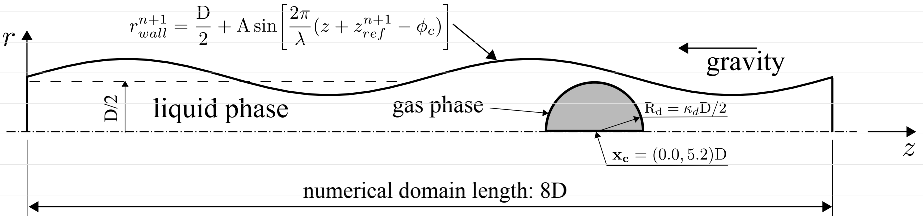

$r_{wall}^{n+1} = \frac{D}{2} + A\sin\left[\frac{2\pi}{\lambda}(z + z_{ref}^{n+1} - \phi_c)\right]$
r
gravity
D/2
liquid phase
gas phase
$R_d = \kappa_d D/2$
z
$\mathbf{x_c} = (0.0, 5.2)D$
numerical domain length: 8D

FIGURE 17 Schematic representation of the domain's geometry and initial air bubble's location for the single drop rising along a sinusoidal capillary channel.

In Fig. 18, we present a detailed comparison involving bubble shape variations. At the top-left, the evolution of the perimeter ratio P/P_{init} is measured, where P_{init} is the initial bubble's perimeter and P the bubble's perimeter along the length L_c of the corrugated channel. This plot shows that the SL2 method captures shape variations underwent by the bubble more precisely than SL1. We observe that within $0.7 < L_c < 0.9$, the growth of the perimeter ratio in SL1 is suddenly damped. Amplified fluctuations appearing in $0.9 < L_c < 1.0$ mean that bubble's interface bounces are better described by SL2 than SL1. Moreover, we note a lagging behavior of SL1 in relation to SL2 as for the curve's peak, which occurs close to $L_c = 0.705$ in the first method but around $L_c = 0.755$ in the second method.

Additionally, we observe that the growth of P/P_{init} and ensuing stabilization around the unity reflects the ability of the code in conserving mass. Since there is no disturbance strong enough to yield a breakup, the bubble undergoes the compressing effect induced by the channel's motion, slightly rebounds, and finally tends to the equilibrium while preserving its shape.

As a way of verifying the implementation of the integration methods in detail, we plotted a phase diagram in Fig. 19 that relates bubble's perimeter ratio P/P_{init} and the bubble's center of mass. Under the current test conditions, the orbits for both methods remain stable as shown. That is to say, no chaotic behavior was identified due to boundary's motion as expected. We can see that the dynamics of the SL2 method features a limit circle more expanded than SL1's, thus defining a pronounced curvature in the range $1.0 < P/P_{init} < 1.4$, when the bubble walks through the convergent section of the channel. In Fig. 20

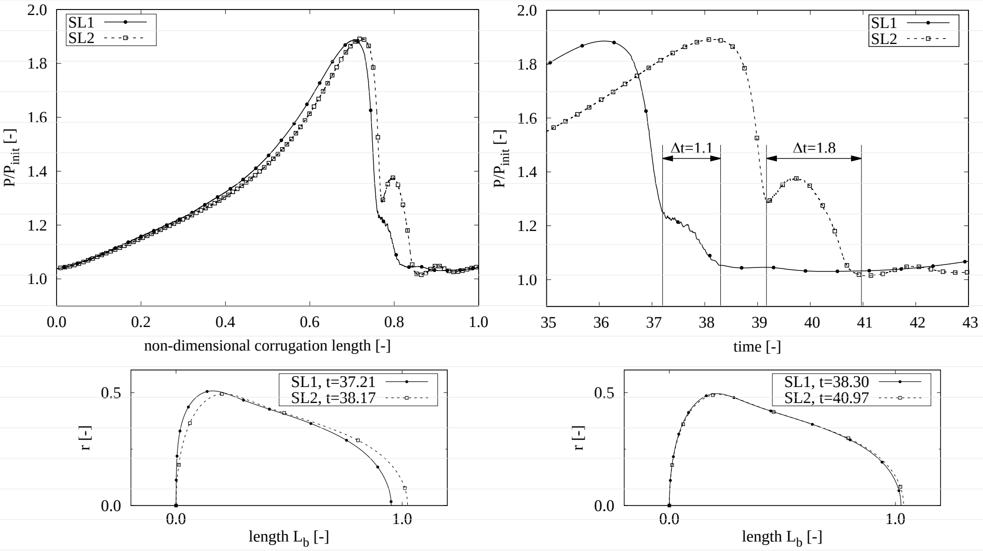


FIGURE 18 Comparative plots of the SL1 and SL2 methods for the case of the single bubble rising along a corrugated channel. Top-left: profile of the rising bubble's perimeter ratio P/P_{init} at one corrugated length L_c. Top-right: zoomed view of the corrugated length within the range $0.75 \leq L_c \leq 0.83$ highlighting the bubble shape evolution with time. Bottom figures: bubble shape at $t = 37.21$ and $t = 38.30$ for the SL1 method (left); at $t = 37.21$ and $t = 40.97$ for the SL2 method (right).

the periodic motion of the velocity profile at the bubble's center of mass is shown for both methods. As seen, the wave phase discrepancy as well as the presence of some small perturbations during bubble's deceleration captured by SL2 method point to a superior capability in representing the bubble's motion.

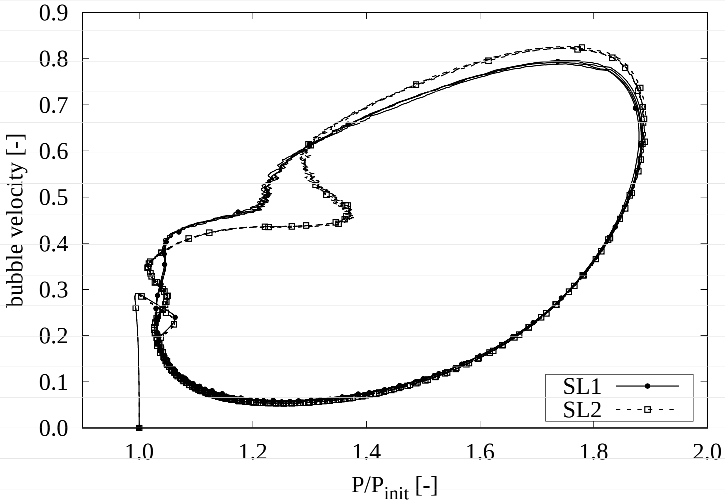


FIGURE 19 Phase diagram relating the bubble's perimeter ratio P/P_{init} and the bubble's center of mass. Stable orbits show well-defined limit cycles for both SL1 and SL2 methods and absence of chaos, thus verifying that the numerical code reproduces the physical behavior with high fidelity.

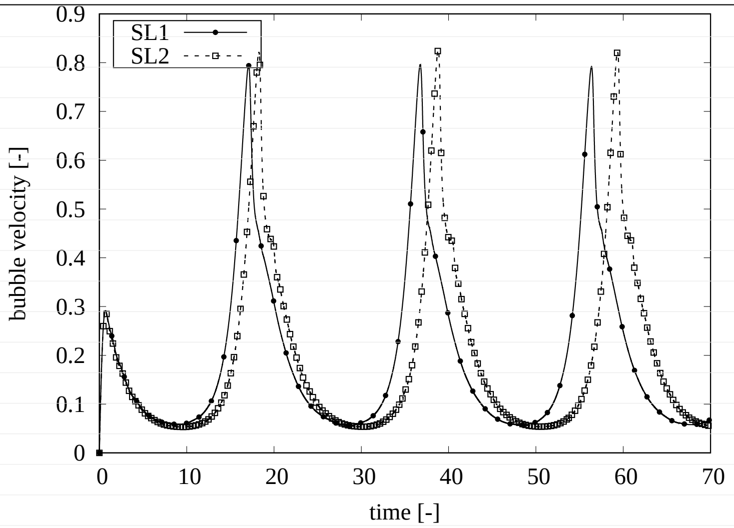


FIGURE 20 Temporal evolution of the velocity at the bubble's center of mass comparing the SL1 and SL2 methods along three corrugated lengths.

Finally, the evolution with time of the liquid film thickness along one corrugated length is presented in Fig. 21 for the SL1 and SL2 methods. The initial bubble's film thickness $\delta_0 = 0.19$ behaves in a quasi-periodic manner within $0.05 < \delta < 0.2$. In particular, the SL2 method present elevated peaks close to $t = 20, 40, 60, 80$, whereas the SL1 presented them at $t = 19, 38, 58, 77$ with lagged phases. The final number of mesh nodes and mesh triangular elements for this simulation were 89933 and 19944, respectively, for the SL1 scheme, and 92926 and 20185, respectively, for the SL2 scheme.

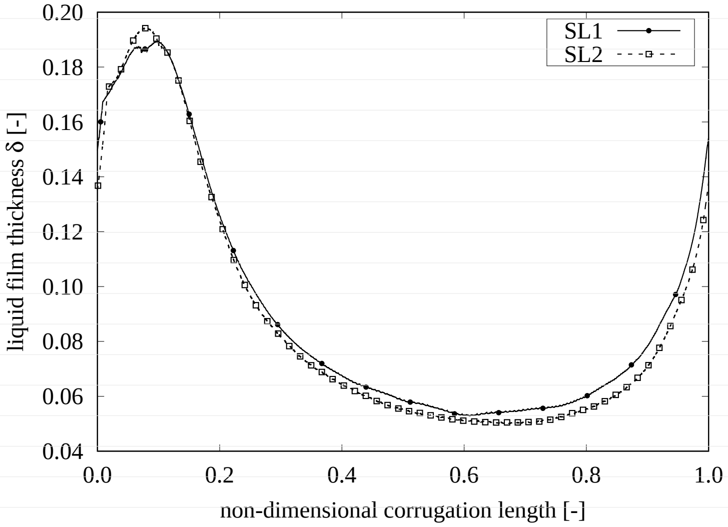


FIGURE 21 Evolution of liquid film thickness with time for both first- and second-order time Semi-Lagrangian methods for the case of the single bubble ascending through a sinusoidal channel.

4 | CONCLUSION

In this paper, we introduced a second-order time semi-lagrangian scheme to deal with advection dominated flows that works in combination with finite element discretizations and abritrary Lagrangian-Eulerian frameworks. We demonstrated the method's capability to simulate flows with constant or varying time steps.

A large set of flow conditions was implemented to validate the proposal. Firstly, the ability to conserve mass in relation to a first-order scheme was verified through the passive scalar transport mediated by rotating flows. We simulated the transport of a scalar blob distributed in the form of a Gaussian function and also in the form the well-known Zalesak's disk.

Secondly, the robustness of the method in capturing dynamic features of dispersed elements in two-phase flow configurations was verified through a series of simulations involving classical benchmarks of rising bubble and drop flows in different solutions. Liquid film thickness, terminal shape, trajectories and velocity were seen to vary in a few cases. In particular, the new second-order time method can detect bubble shape fluctuations with more precision than its first-order counterpart.

In particular, the new scheme bypasses negative damping effects yielded by the first-order scheme that deteriorate the dynamic motion by highlighting the physical behaviour of bubbles and drops with detail. We also underlined the features of the method concerning stability, thereby concluding that it reaches a good agreement with the experimental evidence.

Since the second-order semi-Lagrangian method proposed here is spatially independent, such method can be further implemented for three-dimensional flows straightforwardly with similar expected results.

ACKNOWLEDGEMENT

The authors acknowledge the support of the Brazilian National Council for Scientific and Technological Development (CNPq) and the Foundation for Research Support of the State of Rio de Janeiro (FAPERJ) to this research. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 778104.

References

How to cite this article: G. R. Anjos, Oliveira G. P., N. Mangiavacchi, and J. R. Thome (2021), One- and two-step semi-Lagrangian integrators for ALE-FE two-phase flow simulations, *Int. J. Num. Meth. in Fluids*, *XXX*.