Received: 9 May 2024 | Revised: 1 August 2024 | Accepted: 1 September 2024

DOI: 10.1002/fld.5339

RESEARCH ARTICLE

WILEY

Semi-implicit Lagrangian Voronoi approximation for the incompressible Navier–Stokes equations

Ondřej Kincl[1] | **Ilya Peshkov**[1] | **Walter Boscheri**[2,3]

[1] Laboratory of Applied Mathematics, DICAM, University of Trento, Trento, Italy

[2] Laboratoire de Mathématiques UMR 5127 CNRS, Université Savoie Mont Blanc, Le Bourget du Lac, France

[3] Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy

Correspondence
Ondřej Kincl, Laboratory of Applied Mathematics DICAM, University of Trento, 38123 Trento, Italy.
Email: ondrej.kincl@unitn.it

Funding information
Fondazione Cariplo, Grant/Award Number: 2022-1895; Ministero dell'Università e della Ricerca, Grant/Award Number: 2022N9BM3N

Abstract

We introduce semi-implicit Lagrangian Voronoi approximation (SILVA), a novel numerical method for the solution of the incompressible Euler and Navier–Stokes equations, which combines the efficiency of semi-implicit time marching schemes with the robustness of time-dependent Voronoi tessellations. In SILVA, the numerical solution is stored at particles, which move with the fluid velocity and also play the role of the generators of the computational mesh. The Voronoi mesh is rapidly regenerated at each time step, allowing large deformations with topology changes. As opposed to the reconnection-based Arbitrary-Lagrangian-Eulerian schemes, we need no remapping stage. A semi-implicit scheme is devised in the context of moving Voronoi meshes to project the velocity field onto a divergence-free manifold. We validate SILVA by illustrative benchmarks, including viscous, inviscid, and multi-phase flows. Compared to its closest competitor, the Incompressible Smoothed Particle Hydrodynamics method, SILVA offers a sparser stiffness matrix and facilitates the implementation of no-slip and free-slip boundary conditions.

KEYWORDS

incompressible flows, Lagrangian Voronoi meshes, mesh regeneration with topology changes, particle hydrodynamics, semi-implicit schemes

1 | INTRODUCTION

In computational fluid dynamics, Lagrangian methods are invaluable for their ability to handle complex deformations and material interfaces. The defining feature of Lagrangian methods is that the degrees of freedom of the spatial approximation follow the motion of the material, rather than being static with respect to a selected Eulerian frame of reference. Consequently, the (often problematic) nonlinear convective terms are absent, thus obtaining a family of numerical schemes which exhibit virtually no numerical dissipation at contact discontinuities and material interfaces. Moreover, individual particles or cells preserve their identity, facilitating the mass conservation and the implementation of sharp material interfaces.

Developing a mesh-based Lagrangian method is notoriously difficult because of the large deformations inherent to the fluid motion. This problem becomes particularly evident when considering flows with high vorticity or large velocity gradients, that stretch, twist or compress the control volumes until they get invalid, that is, with negative volumes. To overcome this problem, mesh optimization algorithms have been developed,[1–4] aiming at improving the mesh quality by performing a sort of regularization of the shape of each single cell. If the moving mesh scheme belongs to the category of Arbitrary-Lagrangian-Eulerian (ALE) methods, in which the mesh motion can be driven independently from the fluid velocity, the stage of mesh optimization is referred to as *rezoning*. The rezoning step is directly embedded in the numerical scheme for direct ALE methods,[5,6] otherwise it must be followed by a *remapping* step in indirect ALE methods.[7–9] These techniques are all based on the assumption that the optimization process does not change the topology of the underlying computational grid, hence posing a severe limit to the possibility of improving the mesh quality for complex configurations. Moreover, the mesh optimization stage has to be compatible with some physical requirements on the numerical solution, for example, contact discontinuities must remain untouched to ensure the sharp interface resolution of Lagrangian schemes. The most general solution to this difficulty, which allows to combine mesh optimization and Lagrangian mesh motion, is to allow for *topology changes*. This approach is also known as reconnection, that has been originally presented in Reference 10. The idea was to use a set of Lagrangian particles, that are used to define the computational tessellation without assuming a fix connectivity: indeed, the topology of the mesh was time dependent, so that at each time step the connectivity was established by defining the surrounding domains (or control volumes) in accordance to the current particle positions. This mesh was then used to discretize the equations written in Lagrangian form. More recent developments of this seminal idea are given by Free-Lagrange methods,[11,12] Re-ALE schemes,[13,14] and finite volume ALE methods.[15,16]

Among many others, a well-established mesh-free Lagrangian method is the Smoothed Particle Hydrodynamics (SPH), developed in 1970s: see Reference 17 for a review. In the SPH framework, the continuum is replaced by a finite number of material points (particles), and the gradients of physical variables (like density and velocity) are approximated using convolution and a convenient set of *kernel functions*. No computational grid is used in the SPH framework, hence yielding high versatility in the simulation of large deformations of the continuum. Due to its particle nature, the SPH method avoids the mesh-related issues and it is exceptionally robust. Owing to this advantage, together with the simplicity of implementation, SPH boasts with wide range of applications in science and industry.[18,19] Nevertheless, compared to mesh-based approaches, the computational cost of SPH schemes associated to each degree of freedom is typically larger, since every particle needs to communicate with every neighbor within a certain radius (as specified by the smoothing length).[20] Moreover, converging to the exact solution is challenging because the spatial step must shrunk faster than the smoothing length,[21] leading to an unfavorable effective convergence rate. To remain competitive in this regard, the accuracy of SPH methods needs to be improved by relying on a set of renormalized operators,[22,23] typically at the expense of discrete energy conservation.[24] Lastly, let us mention the problematic of boundary conditions. While free surfaces in fluids and solids are automatically handled, implementing Dirichlet or Neumann conditions in the SPH framework is less trivial (in comparison to e.g., Finite Element Method). As it turns out, having a mesh is not completely without merit and this is where Lagrangian Voronoi Methods (LVM) enter the game.

The study of Lagrangian methods based on Voronoi tessellations dates back to the 70s and 80s, for example, see Reference 11, and received further attention in the 90s, for example, see References 25–27, and in the light of recent publications.[15,16,28,29] By definition, a Voronoi cell Ω_i is the set of points which is closer to the generating particle (seed) $\boldsymbol{x}_i$ than to any other particle $\boldsymbol{x}_j$. The Voronoi mesh generated by moving particles has the advantage of keeping reasonably good quality even in case of large deformations. Furthermore, the local topology of Voronoi meshes can be arbitrary, thus this kind of mesh permits to remove some mesh imprinting that might arise from the adoption of the Lagrangian frame of reference.

While the aforementioned studies revolve around compressible fluids and use explicit time steps, in this paper, we introduce some implicitness to obtain a LVM variant for the incompressible Euler and Navier–Stokes equations. With the aim of avoiding the solution of globally implicit systems defined on the entire computational grid, the implicit-explicit (IMEX) approach has emerged in the literature[30–36] as a powerful time integration technique which permits to split explicit and implicit fluxes.[32,33,37] In particular, we rely on a semi-implicit time discretization,[38–41] in which the mesh motion is treated explicitly while the pressure terms are taken into account implicitly. This is well suited to enforce the divergence-free condition on the velocity field, which is nothing but the energy equation for incompressible fluids.

To the best of our knowledge, such studies on semi-implicit schemes on Lagrangian Voronoi meshes did not receive sufficient attention in the literature. Yet, we found out that our method is conceptually similar to the method by Borgers and Peskin.[42,43] Namely, the spatial and time discretization of the governing equations is similar to Reference 43, however, the key novelties lay in the fast computation of the Voronoi mesh, and a novel stabilized pressure gradient allowing to efficiently deal with the tensile instabilities known for particle methods. Also, our method is linked with the recent work of Després,[29] and it relies on the improved gradient approximation by Serrano and Espanol.[44]

The scheme presented here is termed Semi-Implicit Lagrangian Voronoi Approximation (SILVA), since the mathematical model is given by the incompressible Navier–Stokes equations. We will show that SILVA arises quite naturally from basic assumptions. The correctness and accuracy of the scheme is verified in benchmarks, including the lid-driven cavity problem and a Rayleigh–Taylor instability test for multi-phase fluids. Not only engineers and researchers interested in new Lagrangian numerical methods for fluid mechanics may find this paper valuable, but also SPH specialists because of the deep analogies between SILVA and incompressible SPH (ISPH). Both methods involve moving particles, are based on Helmholtz decomposition, and the cell-list structure from ISPH is a perfect tool for constructing a Voronoi grid for SILVA. To summarize, the two methods are highly inter-compatible and an implementation of a physical phenomenon in one method can be painlessly imported to the other one.

The structure of the paper is as follows. In Section 2, we present the governing equations of incompressible fluids, while in Section 3 we introduce the definition and notation of the moving Voronoi mesh, including its generation. Section 4 is devoted to the description of the numerical scheme, namely our LVM, with details on the explicit and the implicit time discretization as well as on the discrete spatial operators. The numerical validation of the accuracy and robustness of SILVA is presented in Section 5 by running several benchmarks for incompressible fluids. Finally, Section 6 finalizes this article by summarizing the work and giving an outlook to future investigations.

2 | GOVERNING EQUATIONS

We consider a d-dimensional domain $\Omega \subset \mathbb{R}^d$ closed by the boundary $\partial\Omega \subset \mathbb{R}^{d-1}$, with $\boldsymbol{x}$ being the spatial coordinate vector and $t \in \mathbb{R}_0^+$ representing the time coordinate. In the Lagrangian frame, the mathematical model which describes the phenomena of incompressible viscous flows writes

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} + \frac{\nabla p}{\rho} - \nu \Delta \boldsymbol{v} = \boldsymbol{0}, \tag{1a}$$

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

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{v}. \tag{1c}$$

Here, $\boldsymbol{v}(\boldsymbol{x}, t)$ identifies the velocity vector, $p(\boldsymbol{x}, t)$ denotes the pressure, and $\rho(\boldsymbol{x}, t)$ is the fluid density, ν is the kinematic viscosity of the fluid and

$$\frac{\mathrm{d}g}{\mathrm{d}t} = \frac{\partial g}{\partial t} + \boldsymbol{v} \cdot \nabla g. \tag{2}$$

is the material derivative of a quantity $g(\boldsymbol{x}, t)$. Equation (1c) is the so-called trajectory equation, which drives the motion of the coordinates and is the defining feature of a Lagrangian method.

The flow is incompressible, which implies that ρ is constant along the flow trajectories, although it is not necessarily constant in space. The velocity field is subjected to the divergence constraint (1b), which can be reformulated in a weak sense. Indeed, by multiplying Equation (1b) by a smooth test function $\varphi(\boldsymbol{x}) \in C^\infty$ and integrating by parts over the domain Ω, we have that

$$\int_\Omega \varphi \, \nabla \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \int_{\partial\Omega} \phi \, \boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}S - \int_\Omega \nabla\varphi \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}. \tag{3}$$

We will assume that the no-penetration condition is satisfied at the boundary

$$\boldsymbol{v} \cdot \boldsymbol{n} = 0, \quad \forall \boldsymbol{x} \in \partial\Omega, \tag{4}$$

so that Equation (3) reduces to

$$\int_\Omega \nabla\varphi \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = 0, \qquad \forall \varphi \in C^\infty. \tag{5}$$

The condition (4) is not sufficient for a viscous fluid ($\nu \neq 0$) which also requires a condition for the tangential velocity. Two possibilities will be covered in this manuscript: the free-slip and no-slip condition. We return to this topic in Section 4.5. It would be also possible to consider periodic boundaries, which would imply the use of a periodic Voronoi mesh generator. We do not foresee any substantial obstacle in this regard. The implementation of other conditions, like inflows and outflows, are left for future investigations.

3 | MOVING VORONOI MESH WITH TOPOLOGY CHANGES

3.1 | Notation and definitions

Let assume $\Omega \subset \mathbb{R}^d$ to be an open convex and bounded polytope. We consider N *distinct* sample points $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N \in \Omega$, which will be treated as material points. The points $\boldsymbol{x}_i$ are called *seeds*, for $i = 1, 2, \dots, N$. A Voronoi cell ω_i corresponding to the seed $\boldsymbol{x}_i$ is defined for every $i, j = 1, 2, \dots, N$ as

$$\omega_i = \bigcap_{j \neq i} \left\{ \boldsymbol{x} \in \Omega : \quad |\boldsymbol{x} - \boldsymbol{x}_i| < |\boldsymbol{x} - \boldsymbol{x}_j| \right\}.$$

To make notation easier throughout the entire paper, we assume that the cell indexes are always running in the interval $i, j = 1, 2, \dots, N$. In the Lagrangian framework, every cell contains a portion of fluid with mass M_i, which is constant in time. The Voronoi cell is a convex polytope and its boundary $\partial\omega_i$ is composed of the set of facets

$$\Gamma_{ij} = \Gamma_{ji} = \left\{ \boldsymbol{x} \in \partial\omega_i : \quad |\boldsymbol{x} - \boldsymbol{x}_i| = |\boldsymbol{x} - \boldsymbol{x}_j| \right\}. \tag{6}$$

Furthermore, the cell boundary $\partial\omega_i$ might share a portion of the domain boundary $\partial\Omega$ when $\partial\Omega \cap \partial\omega_i \neq 0$. In such case, we refer to ω_i as a *boundary-touching cell*, or an *interior cell* otherwise. We also define the surface element:

$$\boldsymbol{S}_i = \int_{\partial\omega_i} \boldsymbol{n} \, \mathrm{d}S, \tag{7}$$

where $\boldsymbol{n}(\boldsymbol{x})$ is the outward pointing unit normal vector defined on $\partial\omega_i$.

The golden property of Voronoi meshes is the orthogonality between the facets Γ_{ij} and the segments joining two seeds, namely $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$. Hence, the outer normal vector $\boldsymbol{n}_{ij}$ is simply given for each facet Γ_{ij} by

$$\boldsymbol{n}_{ij} = -\frac{\boldsymbol{x}_{ij}}{r_{ij}}, \tag{8}$$

where $r_{ij} = |\boldsymbol{x}_{ij}|$ represents the inter-seed distance. The d-dimensional Lebesgue measure of each cell is denoted by $|\omega_i|$, while $|\partial\omega_i|$ is the $(d-1)$-dimensional Hausdorff measure of its boundary. Likewise, $|\Gamma_{ij}|$ represents the length of the facet Γ_{ij}. In general, it is necessary to distinguish the midpoint of a facet

$$\boldsymbol{m}_{ij} = \frac{1}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} \boldsymbol{x} \mathrm{d}S, \tag{9}$$

and the inter-seed midpoint

$$\overline{\boldsymbol{x}}_{ij} = \frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2}. \tag{10}$$

Indeed, for a general Voronoi cell, it is possible that $\overline{\boldsymbol{x}}_{ij} \notin \Gamma_{ij}$.

To evaluate partial derivatives in closed form for a Voronoi mesh, we refer to the recent developments forwarded in Reference 29. There is indeed a simple formula for computing the derivative of a Voronoi volume with respect to the other seeds,[29] which is recalled in the following theorem.

Theorem 1. *For $j \neq i$ we have the closed form formulae*:

$$\frac{\partial |\omega_i|}{\partial \boldsymbol{x}_j} = -\frac{|\Gamma_{ij}|}{r_{ij}}\left(\boldsymbol{m}_{ij} - \boldsymbol{x}_j\right), \tag{11}$$

and

$$\frac{\partial |\omega_i|}{\partial \boldsymbol{x}_i} = -\sum_{j\neq i}\frac{\partial |\omega_j|}{\partial \boldsymbol{x}_i} = -\sum_{j\neq i}\frac{\partial |\omega_i|}{\partial \boldsymbol{x}_j} - \boldsymbol{S}_i. \tag{12}$$

Proof. We present a two page proof using the Lebesgue (dominated convergence) theorem in A. Other proofs can be found in References 29 and 42, which rely on a smooth function to approximate the characteristic function of ω_i. Another derivation restricted to the two-dimensional case can also be found in Reference 43. ■

Remark 1. It is possible to get an intuitive understanding of (11) by evaluating the rate of change of volume when the seeds are moving. By the transport theorem we have

$$\frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} = \sum_j \int_{\Gamma_{ij}} v_{n,ij}(\boldsymbol{x})\mathrm{d}S, \tag{13}$$

where $v_{n,ij}$ is the speed of the interface Γ_{ij} in the outward normal direction with respect to ω_i. Naively, one would expect

$$v_{n,ij} = \frac{1}{2}\boldsymbol{n}_{ij}\cdot\left(\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{x}_j}{\mathrm{d}t}\right). \tag{14}$$

Therefore, in the case when only $\boldsymbol{x}_j$ is moving one obtains

$$\frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} = -\frac{|\Gamma_{ij}|}{2r_{ij}}\boldsymbol{x}_{ij}\cdot\frac{\mathrm{d}\boldsymbol{x}_j}{\mathrm{d}t} = -\frac{|\Gamma_{ij}|}{r_{ij}}(\overline{\boldsymbol{x}}_{ij} - \boldsymbol{x}_j)\cdot\frac{\mathrm{d}\boldsymbol{x}_j}{\mathrm{d}t}. \tag{15}$$

As it turns out, this is true only when $\boldsymbol{x}_j$ moves in a direction parallel to the vector $\boldsymbol{x}_{ij}$, like during a linear stretching (and it is always true for a one-dimensional Voronoi mesh, where $\overline{\boldsymbol{x}}_{ij} = \boldsymbol{m}_{ij}$). Formula (11) can be understood as a correction of (15) for displacements of $\boldsymbol{x}_j$ that also contain transversal components to $\boldsymbol{x}_{ij}$. To see this, take for example a rotational motion

$$\frac{\mathrm{d}\boldsymbol{x}_j}{\mathrm{d}t} = \Theta(\boldsymbol{x}_j - \boldsymbol{x}_i), \tag{16}$$

where Θ is a skew-symmetric matrix (with other seeds, including $\boldsymbol{x}_i$, being all fixed). It is easy to see that the plane P_{ij} of equidistance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ will also rotate with the velocity

$$\boldsymbol{v}(\boldsymbol{x}) = \Theta(\boldsymbol{x} - \boldsymbol{x}_i). \tag{17}$$

Substituting this velocity field into the transport theorem (13), we find

$$\begin{aligned}\frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} &= -\frac{1}{r_{ij}}\int_{\Gamma_{ij}}\Theta(\boldsymbol{x} - \boldsymbol{x}_i)\cdot\boldsymbol{x}_{ij}\mathrm{d}S(\boldsymbol{x}) \\ &= -\frac{|\Gamma_{ij}|}{r_{ij}}\Theta(\boldsymbol{m}_{ij} - \boldsymbol{x}_i)\cdot\boldsymbol{x}_{ij} \\ &= \frac{|\Gamma_{ij}|}{r_{ij}}(\boldsymbol{m}_{ij} - \boldsymbol{x}_i)\cdot\Theta\boldsymbol{x}_{ij}\end{aligned}$$

$$
\begin{aligned}
&= -\frac{|\Gamma_{ij}|}{r_{ij}}(\boldsymbol{m}_{ij} - \boldsymbol{x}_j) \cdot \Theta \boldsymbol{x}_{ij} \\
&= -\frac{|\Gamma_{ij}|}{r_{ij}}(\boldsymbol{m}_{ij} - \boldsymbol{x}_j) \cdot \frac{\mathrm{d}\boldsymbol{x}_j}{\mathrm{d}t}.
\end{aligned}
\tag{18}
$$

Remark 2. We note that formula (11) fails when Ω is nonconvex. This is quite a concerning limitation, which will require a special treatment in future studies. In this paper, all domains are rectangular.

3.2 | Voronoi mesh generation

The problem of generating Voronoi meshes is well studied in the literature[45] and Voronoi mesh generators have been implemented in a variety of open-source libraries, such as Qhull,[46] Voro++[47] or CGAL.[48] Most of them rely on constructing the Delaunay triangulation, which is a topological dual of a Voronoi grid. However, for our LVM, it is important to have a fast algorithm for computing the Voronoi grid from the moving seeds at every time step. We found that a direct approach to compute Voronoi cells (i.e., avoiding the computation of the dual mesh) is preferable.[49,50] Such a method only involves the data from the nearest neighbors, that is, it can be considered as being *local*, and hence suits well to parallelization. We illustrate this iterative technique in Figure 1 for the first five iterations $k = \{1, 2, 3, 4, 5\}$. In the first step, that is, for $k = 0$, every cell ω_i is initialized as $\omega_i^0 = \Omega$. We then construct a cell list, which divides the computational domain Ω into a finite number of partitions (squares in Figure 1) of size comparable to the spatial resolution δr (in this paper, we set the side length of those partitions to be $2\delta r$). For a given $\boldsymbol{x}_i$ (green particle in Figure 1), we find the containing partition $P_{i,0} \ni \boldsymbol{x}$ and proceed by generating an iterator through a sequence $P_{i,k}$, $k = 1, 2, 3, \dots$ of all other partitions in the cell list, sorted in ascending order by their distance to $P_{i,0}$. Then, the polytopes

$$
\omega_i^k = \bigcap_{\boldsymbol{x}_j \in P_{i,k}} \left\{ \boldsymbol{x} \in \omega_i^{k-1} : |\boldsymbol{x} - \boldsymbol{x}_i| < |\boldsymbol{x} - \boldsymbol{x}_j| \right\}, \quad k = 1, 2, 3, \dots \tag{19}
$$

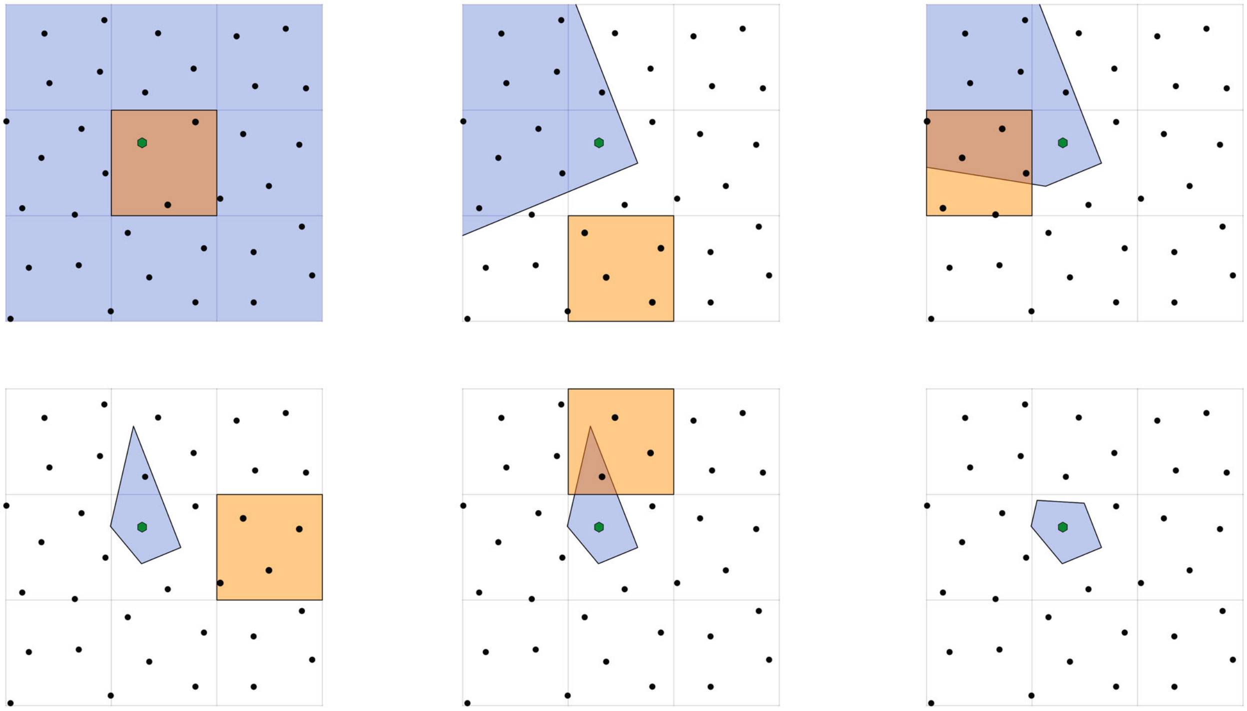


FIGURE 1 A Voronoi cell for the green particle in the center can be obtained by intersecting $N - 1$ half-planes, one for each other seed. The cell list structure optimizes the computation by focusing on nearby seed only. The orange and blue colors highlight the partition $P_{i,k}$ and the incomplete Voronoi cell ω_i^k, respectively, for $k = 0, 1, \dots, 5$. [Colour figure can be viewed at wileyonlinelibrary.com]

are constructed in an iterative process. Each ω_i^k has a radius R_i^k, which is the distance between $\boldsymbol{x}_i$ and the furthest vertex of ω_i^k. The iterative procedure stops when the Hausdorff distance between $P_{i,0}$ and $P_{i,k+1}$ satisfies

$$R_i^k < \frac{1}{2}\text{dist}\left(P_{i,0}, P_{i,k+1}\right), \tag{20}$$

thus we set $\omega_i = \omega_i^k$, as all remaining points are provably too remote and can be excluded from the computation.

In the Lagrangian approach, the seeds are displaced with the fluid velocity. Therefore, the Voronoi mesh needs to be *regenerated* at each time step with possible changes in the mesh connectivity. Starting from the second time step, the process of mesh generation is further accelerated by initializing

$$\omega_i^0 = \bigcap_{j \in N_i^{\text{old}}} \left\{ \boldsymbol{x} \in \Omega : |\boldsymbol{x} - \boldsymbol{x}_i| < |\boldsymbol{x} - \boldsymbol{x}_j| \right\}, \tag{21}$$

where N_i^{old} is the set of neighbors of ω_i from the previous mesh configuration. With this heuristics, the number of non-trivial half-plane cuts is reduced to bare minimum. In the context of numerical simulations on moving Voronoi tessellations, the method has the following advantages.

- If $\text{diam}(\omega_i/\delta r)$ is bounded, $O(N)$ complexity is guaranteed in the mesh generation process.
- Since the cell constructions are independent processes, the method can be easily parallelized.
- This technique works both in two and three dimensions.

4 | LAGRANGIAN VORONOI METHOD

In this section, we describe and analyze the discrete gradient operator on moving Voronoi grids for a generic smooth function. Then, we present the semi-discrete in space numerical scheme, and finally we provide the details of the semi-implicit time marching algorithm. For the sake of clarity, we first assume no viscous terms in the velocity equation (1a), hence focusing on the incompressible Euler model. Viscous forces will be consistently considered in the numerical scheme and presented at the end of this section.

4.1 | Discrete gradient operator

Let us assume that $p(\boldsymbol{x})$ is a smooth function defined on Ω, and $p_i = p(\boldsymbol{x}_i)$ are its point values. In order to approximate $\nabla p(\boldsymbol{x}_i)$, we use the following representation of the identity matrix:

$$\mathbb{I} = \frac{1}{|\omega_i|} \int_{\partial \omega_i} (\boldsymbol{x} - \boldsymbol{x}_i) \otimes \boldsymbol{n} \, \mathrm{d}S, \tag{22}$$

which holds for every cell ω_i by virtue of the divergence theorem. For any interior cell ω_i, using the orthogonality between $\boldsymbol{x}_{ij}$ and the facet Γ_{ij} (see Equation 8) we obtain:

$$\begin{aligned} \nabla p(\boldsymbol{x}_i) &= \frac{1}{|\omega_i|} \int_{\partial \omega_i} (\boldsymbol{x} - \boldsymbol{x}_i)(\nabla p(\boldsymbol{x}_i) \cdot \boldsymbol{n}) \, \mathrm{d}S \\ &= \frac{1}{|\omega_i|} \sum_{j \neq i} \int_{\Gamma_{ij}} (\boldsymbol{x} - \boldsymbol{x}_i)\left(\nabla p(\boldsymbol{x}_i) \cdot \boldsymbol{n}_{ij}\right) \, \mathrm{d}S \\ &= -\frac{1}{|\omega_i|} \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} (\boldsymbol{m}_{ij} - \boldsymbol{x}_i)\left(\nabla p(\boldsymbol{x}_i) \cdot \boldsymbol{x}_{ij}\right) \\ &\approx -\frac{1}{|\omega_i|} \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} p_{ij} (\boldsymbol{m}_{ij} - \boldsymbol{x}_i) =: \langle \nabla p \rangle_i, \end{aligned} \tag{23}$$

where $p_{ij} = p_i - p_j$. This is the definition of the discrete gradient operator $\langle \nabla p \rangle_i$. The only approximation occurs in the step

$$\nabla p(\boldsymbol{x}_i) \cdot \boldsymbol{x}_{ij} \approx p_{ij}, \tag{24}$$

which is exact only for polynomials up to first degree. The following error estimate is from Reference 51, p. 804.

Theorem 2. *Let $p(\boldsymbol{x}) \in C^2$, that is, p is twice continuously differentiable, and let assume $d = 2$ in two space dimensions. Then for any interior cell ω_i we have that*

$$|\nabla p(\boldsymbol{x}_i) - \langle \nabla p \rangle_i| \leq 2d \; \|\nabla^2 p\|_\infty \mathrm{diam}(\omega_i). \tag{25}$$

Proof. Using the idea of (23), we have

$$|\nabla p(\boldsymbol{x}_i) - \langle \nabla p \rangle_i| \leq \frac{1}{|\omega_i|} \sum_j |\Gamma_{ij}| \; r_{ij} \; \|\nabla^2 p\|_\infty \mathrm{diam}(\omega_i), \tag{26}$$

and we get the desired result by combining this with a formula for the Voronoi cell volume:

$$|\omega|_i = \frac{1}{2d} \sum_j |\Gamma_{ij}| \; r_{ij}. \tag{27}$$

■

Remark 3. Intriguingly, the error term does not depend on the mesh quality. We do not require to bound the number of facets or the ratios $\mathrm{diam}(\omega_i)/r_{ij}$.

Remark 4. If the Voronoi mesh is rectangular with the sides Δx and Δy, then (23) becomes equivalent to the central finite difference operator

$$\langle \nabla p \rangle_i = \begin{pmatrix} \frac{p(x+\Delta x,y)-p(x-\Delta x,y)}{2\Delta x} \\ \frac{p(x,y+\Delta y)-p(x,y-\Delta y)}{2\Delta y} \end{pmatrix}, \tag{28}$$

which has second order of accuracy. As we shall demonstrate in Section 5.1, it is possible to obtain a super-linear convergence in velocity if a rectangular grid is used in the initialization. This is understandable, because with decreasing spatial step, the mesh eventually becomes "locally rectangular" throughout the entire simulation time.

Remark 5. By adding a "clever zero", we can obtain an alternative formulation (valid for interior cells):

$$\begin{aligned} \langle \nabla p \rangle_i = \langle \nabla p \rangle_i - \int_{\partial\omega_i} p_i \boldsymbol{n} \, \mathrm{d}S &= -\frac{1}{|\omega_i|} \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} \left(p_{ij}(\boldsymbol{m}_{ij} - \boldsymbol{x}_i) + p_i \boldsymbol{x}_{ij} \right) \\ &= -\frac{1}{|\omega_i|} \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} \left(p_{ij}(\boldsymbol{m}_{ij} - \overline{\boldsymbol{x}}_{ij}) + \overline{p}_{ij} \boldsymbol{x}_{ij} \right), \end{aligned} \tag{29}$$

where $\overline{p}_{ij} = \frac{1}{2}(p_i + p_j)$ and $\overline{\boldsymbol{x}}_{ij}$ is given by (10). This form of the gradient approximation is used in the work of Serrano et al.[44] and Springel.[15]

Remark 6. For a boundary-touching cell, with the facet $\Gamma_{ij} \subset \partial\Omega$, an additional term must be considered:

$$\nabla p(\boldsymbol{x}_i) = \langle \nabla p \rangle_i + \frac{1}{|\omega_i|} \int_{\Gamma_{ij}} (\boldsymbol{x} - \boldsymbol{x}_i)(\nabla p(\boldsymbol{x}_i) \cdot \boldsymbol{n}) \, \mathrm{d}S + O(\mathrm{diam}(\omega_i)). \tag{30}$$

For a Robin boundary condition of the type

$$ap + b\frac{\partial p}{\partial n} = 0, \tag{31}$$

it is reasonable to approximate the gradient as

$$\nabla p(\boldsymbol{x}_i) = \langle \nabla p \rangle_i - \frac{|\Gamma_i|}{|\omega_i|}\frac{a p_i}{b}(\boldsymbol{m}_{ij} - \boldsymbol{x}_i) + O(\mathrm{diam}(\omega_i)). \tag{32}$$

Note that the extra term in (32) disappears for a homogeneous Neumann condition.

Next, we consider a different discretization of the gradient operator. This takes the form

$$\langle \widetilde{\nabla} p \rangle_i = \frac{1}{|\omega_i|}\sum_j \frac{|\Gamma_{ij}|}{r_{ij}} p_{ij}(\boldsymbol{m}_{ij} - \boldsymbol{x}_j), \tag{33}$$

which is motivated by the following “discrete integration by parts.”

Theorem 3. *Let $p(\boldsymbol{x}) \in C^2$ and $q(\boldsymbol{x}) \in C^2$, and let us consider the discrete gradient operators (23) and (33). The following identity holds true for $i = 1, 2, \dots, N$:*

$$\sum_i |\omega_i| (q_i \langle \nabla p \rangle_i + p_i \langle \widetilde{\nabla} q \rangle_i) = \sum_i p_i q_i \boldsymbol{S}_i. \tag{34}$$

Proof. Using the previously introduced definitions and some algebra, we find:

$$\begin{aligned}
&\sum_i |\omega_i| (q_i \langle \nabla p \rangle_i + p_i \langle \widetilde{\nabla} q \rangle_i) \\
&\quad = \sum_i \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} \left(-q_i p_{ij}(\boldsymbol{m}_{ij} - \boldsymbol{x}_i) + p_i q_{ij}(\boldsymbol{m}_{ij} - \boldsymbol{x}_j) \right) \\
&\quad = \sum_i \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} p_i q_i \boldsymbol{x}_{ij} + \sum_i \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} \left((\boldsymbol{m}_{ij} - \boldsymbol{x}_i) q_i p_j - (\boldsymbol{m}_{ij} - \boldsymbol{x}_j) p_i q_j \right).
\end{aligned} \tag{35}$$

The second term on the right-hand side vanishes by virtue of anti-symmetry. The first term can be simplified using

$$\boldsymbol{S}_i = \int_{\Gamma_i} \boldsymbol{n}_i \mathrm{d}S = -\sum_j \int_{\Gamma_{ij}} \boldsymbol{n}_i \mathrm{d}S = \sum_j \frac{|\Gamma_{ij}|}{r_{ij}} \boldsymbol{x}_{ij}, \tag{36}$$

hence the result. ■

We shall refer to (23) and (33) as *strong* and *weak* gradient, respectively. Unfortunately, our investigation reveals that the weak gradient is no longer first-order exact, but it exhibits an oscillatory nature and cannot be used to obtain reliable point-wise approximations of the continuous gradient operator, especially on highly irregular meshes. See Figure 2 for a comparison between the two gradient operators. The situation is similar to SPH, where a pair of gradient operators is defined, such that they satisfy a discrete integration by parts. Enforcing some degrees of exactness on one operator does not carry over to the other one.[20]

Nonetheless, the weak gradient is useful to derive the variation of the cell volume $\delta|\omega_i|$. By virtue of (12), we get

$$\begin{aligned}
\delta|\omega_i| &= \sum_j \frac{\partial |\omega_i|}{\partial \boldsymbol{x}_j} \cdot \delta \boldsymbol{x}_j \\
&= \sum_{j \neq i} \frac{\partial |\omega_i|}{\partial \boldsymbol{x}_j} \cdot \delta \boldsymbol{x}_j + \frac{\partial |\omega_i|}{\partial \boldsymbol{x}_i} \cdot \delta \boldsymbol{x}_i \\
&= -\sum_{j \neq i} \frac{\partial |\omega_i|}{\partial \boldsymbol{x}_j} \cdot \delta \boldsymbol{x}_{ij} - \boldsymbol{S}_i \cdot \delta \boldsymbol{x}_i.
\end{aligned} \tag{37}$$

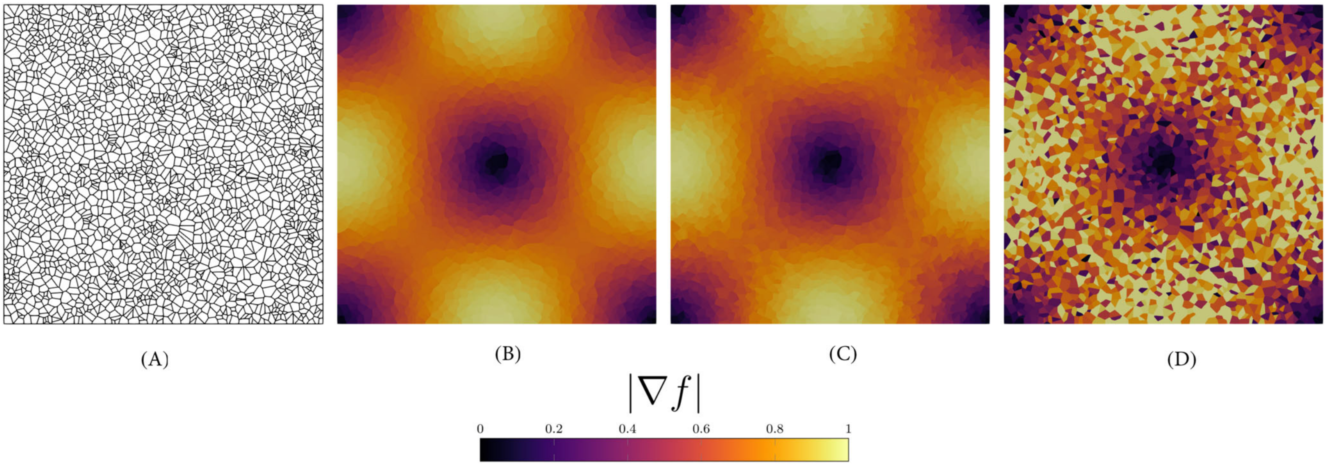


FIGURE 2 The magnitude of ∇f on a square $\Omega = [0,1]^2$ with an irregular Voronoi grid of 6400 cells (A) for $f = \frac{1}{\pi}\cos(\pi x)\cos(\pi y)$. The image (B) and (C) show exact solution and the strong gradient (23), respectively, and they are barely distinguishable. The image (D) uses the weak gradient (33) and is infested with spurious noise. (A) Mesh; (B) Exact; (C) Strong; (D) Weak. [Colour figure can be viewed at wileyonlinelibrary.com]

Next, we substitute (11), to find

$$\delta|\omega_i| = \sum_{j\neq i} \frac{|\Gamma_{ij}|}{r_{ij}} (\boldsymbol{m}_{ij} - \boldsymbol{x}_j)\cdot \delta \boldsymbol{x}_{ij} - \boldsymbol{S}_i \cdot \delta \boldsymbol{x}_i, \tag{38}$$

and so, we recognize the definition (33):

$$\delta|\omega_i| = \langle \widetilde{\nabla} \cdot \delta \boldsymbol{x}\rangle_i - \boldsymbol{S}_i \cdot \delta \boldsymbol{x}_i. \tag{39}$$

4.2 | Semi-discrete space discretization

We consider the seeds of a moving Voronoi mesh $\boldsymbol{x}_i$ being the material particles with point-values of velocity $\boldsymbol{v}_i$, density ρ_i and pressure p_i. The mathematical model is given by the velocity equation (1a), the divergence constraint in weak form (5), and the trajectory equation (1c). For the first exposition of the method we shall ignore the viscous terms and only include them later.

For each cell ω_i, a natural semi-discrete space discretization writes

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = -\frac{1}{\rho_i}\langle \nabla p\rangle_i \tag{40a}$$

$$\sum_i |\omega_i|\ \boldsymbol{v}_i \cdot \langle \nabla \varphi\rangle_i = 0, \quad \forall\ \varphi(\boldsymbol{x}), \tag{40b}$$

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \boldsymbol{v}_i. \tag{40c}$$

Theorem 4. *Any solution of the semi-discrete scheme (40) satisfies the following properties.*

- *The individual cell volumes $|\omega_i|$ are conserved:*

$$\frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} = 0. \tag{41}$$

- *The energy*

$$E = \frac{1}{2}\sum_i |\omega_i| \ \rho_i \ \|\boldsymbol{v}_i\|^2, \tag{42}$$

is conserved.

Proof. Using the variation (39), the time evolution of ω_i is given by

$$\frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} = |\omega_i| \langle \widetilde{\nabla} \cdot \boldsymbol{v} \rangle_i - \boldsymbol{S}_i \cdot \boldsymbol{v}_i. \tag{43}$$

Multiplication by a smooth test function $\varphi(\boldsymbol{x})$ and integration over all cells, leads to

$$\sum_i \frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} \ \varphi_i = \sum_i |\omega_i| \langle \widetilde{\nabla} \cdot \boldsymbol{v} \rangle_i \varphi_i - \sum_i \varphi_i \boldsymbol{v}_i \cdot \boldsymbol{S}_i. \tag{44}$$

Using the identity (34) and the semi-discrete divergence constraint (40b), from the above relation we find that

$$\sum_i \frac{\mathrm{d}|\omega_i|}{\mathrm{d}t} \ \varphi_i = -\sum_i |\omega_i| \boldsymbol{v}_i \cdot \langle \nabla \varphi \rangle_i = 0, \tag{45}$$

which implies (41) after localization.

In light of the previous result we deduce that the cell volume does not depend on time, thus the time evolution of the total energy (42) yields

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \sum_i |\omega_i| \ \rho_i \boldsymbol{v}_i \cdot \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}. \tag{46}$$

Inserting the balance of momentum (40a) and using the incompressibility constraint (40b), we obtain

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\sum_i |\omega_i| \ \boldsymbol{v}_i \cdot \langle \nabla p \rangle_i = 0. \tag{47}$$

■

4.3 | Semi-implicit time discretization

The idea behind the semi-implicit time discretization for incompressible fluid is fairly standard, although it is not frequently applied in the context of LVM (we are aware of only[43]). It can be understood as Chorin iterations,[52] or equivalently, the projection of the velocity field onto the solenoidal space in the sense of Helmholtz decomposition. This requires the solution of an implicit system on the pressure, globally defined on the entire computational domain, which is obtained by combining the velocity equation (1a) with divergence-free constraint (5). On the other hand, the mesh motion is explicit, using the trajectory equation (1c).

Let us consider the time interval $T = [0, t_f]$, with $t \in T$ and t_f being the final time of the simulation. The time interval is discretized by a sequence of time steps $\Delta t = t^{n+1} - t^n$, with t^n denoting the current time. We proceed by introducing a semi-implicit time discretization for the semi-discrete scheme (40).

The Voronoi seed positions are advected explicitly by the velocity field, that is

$$\boldsymbol{x}_i^{n+1} = \boldsymbol{x}_i^n + \Delta t \ \boldsymbol{v}_i^n, \tag{48}$$

and the Voronoi mesh is regenerated at the new time level t^{n+1}, hence obtaining the new tessellation. All geometry related quantities can now be computed, for example, facet lengths $|\Gamma_{ij}|^{n+1}$, inter-seed distances r_{ij}^{n+1} and facet midpoints $\boldsymbol{m}_{ij}^{n+1}$.

As proven in Theorem 4, cell volumes are instead constructed to be constant in time, hence we simply have $|\omega_i|^{n+1} = |\omega_i|^n = |\omega_i|$.

Even though this step does not modify the velocity $\boldsymbol{v}_i^n$ per se, it introduces a non-vanishing divergence of the velocity field through the deformation of the mesh. Therefore, the velocity needs to be updated at the next time level by the *correct* pressure gradient, which can be interpreted as a Lagrangian multiplier enforcing the incompressibility constraint. To this aim, we impose that the new velocity $\boldsymbol{v}_i^{n+1}$ and pressure p_i^{n+1} satisfy

$$\boldsymbol{v}_i^{n+1} = \boldsymbol{v}_i^n - \frac{\Delta t}{\rho_i}\langle\nabla p\rangle_i^{n+1}, \tag{49a}$$

$$\sum_i |\omega_i|\ \boldsymbol{v}_i^{n+1} \cdot \langle\nabla\varphi\rangle_i = 0, \quad \forall\ \varphi(\boldsymbol{x}^{n+1}). \tag{49b}$$

Formal substitution of the velocity equation (49a) into the divergence-free constraint (49b), yields an elliptic problem for the pressure

$$\sum_i \frac{|\omega_i|}{\rho_i}\langle\nabla p\rangle_i^{n+1} \cdot \langle\nabla\varphi\rangle_i = \frac{1}{\Delta t}\sum_i |\omega_i|\ \boldsymbol{v}_i^n \cdot \langle\nabla\varphi\rangle_i. \tag{50}$$

The associated $N \times N$ system matrix $\mathbf{A}$ is defined for any function p and φ to satisfy the following relation:

$$\sum_i \frac{|\omega_i|}{\rho_i}\langle\nabla p\rangle_i^{n+1} \cdot \langle\nabla\varphi\rangle_i = \sum_i\sum_j \varphi_i\ \mathbf{A}_{ij}\ p_j^{n+1}, \tag{51}$$

thus the matrix $\mathbf{A}$ is clearly symmetric. The right-hand side of (50) is an N-element vector representing the linear functional

$$\begin{aligned}\sum_i \mathbf{b}_i\ \varphi_i &= \frac{1}{\Delta t}\sum_i |\omega_i|\ \boldsymbol{v}_i^n \cdot \langle\nabla\varphi\rangle_i \\ &= \frac{|\omega_i|}{\Delta t}\langle\widetilde{\nabla} \cdot \boldsymbol{v}\rangle_i^n.\end{aligned} \tag{52}$$

The coefficients of the system can be evaluated by setting $\varphi_i = \delta_{ik}$ and $p_j = \delta_{j\ell}$ for fixed indexes (k, ℓ). Unfortunately, the matrix is not very sparse because $\mathbf{A}_{ij} \neq 0$ not only when i, j are neighbors, but also when j is a neighbor of i.

In the case when the fluid is incompressible and homogeneous ($\rho_i = \rho = const$), it is possible to sparsify the problem (51) through the following "approximation of approximation," in a direct analogy to incompressible SPH. Recalling the strong gradient definition (23) and using the homogeneous condition $\rho_i = \rho$, the system matrix in (51) is rewritten as follows:

$$\begin{aligned}\sum_i \frac{|\omega_i|}{\rho_i}\langle\nabla p\rangle_i^{n+1} \cdot \langle\nabla\varphi\rangle_i &= -\frac{1}{\rho}\sum_i\sum_j \frac{|\Gamma_{ij}|^{n+1}}{r_{ij}^{n+1}}\ p_{ij}^{n+1}\ (\boldsymbol{m}_{ij}^{n+1} - \boldsymbol{x}_i) \cdot \langle\nabla\varphi\rangle_i \\ &\approx -\frac{1}{\rho}\sum_i\sum_j \frac{|\Gamma_{ij}|^{n+1}}{r_{ij}^{n+1}}\ p_{ij}^{n+1}\ (\varphi(\boldsymbol{m}_{ij}^{n+1}) - \varphi_i) \\ &= \frac{1}{\rho}\sum_i\sum_j \frac{|\Gamma_{ij}|^{n+1}}{r_{ij}^{n+1}}\ p_{ij}^{n+1}\varphi_i.\end{aligned} \tag{53}$$

The term involving $\varphi(\boldsymbol{m}_{ij}^{n+1})$ disappears because of the anti-symmetry of the addends. The new matrix $\mathbf{B}_{ij}$ is thus defined such as

$$\sum_j \mathbf{B}_{ij} p_j^{n+1} = \frac{1}{\rho}\sum_j \frac{|\Gamma_{ij}|^{n+1}}{r_{ij}^{n+1}}\ p_{ij}^{n+1}, \tag{54}$$

and it is much simpler, more sparse and still symmetric compared to the matrix $\mathbf{A}$ in (51). Figure 3 depicts an illustration of the sparsity of $\mathbf{B}$ when compared to incompressible SPH. SILVA produces much sparser matrices, which is a great

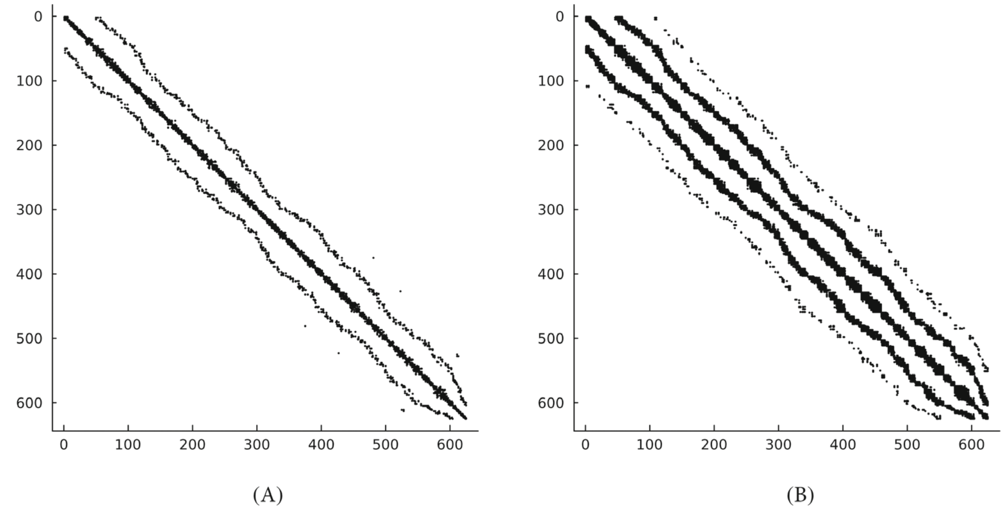


FIGURE 3 Example of a sparsity pattern of the left-hand-side matrix in semi-implicit Lagrangian Voronoi approximation (SILVA) and incompressible smoothed particle hydrodynamics (ISPH) on uniformly random distribution of 625 points (Voronoi seeds) in two dimensions. The smoothing length for ISPH is $1.5\delta r$ and the Wendland quintic kernel is used. For the sake of readability, the points are sorted by their occurrence in the cell list. (A) SILVA; (B) ISPH.

advantage compared to SPH-based methods: for SILVA, the number of non-zeros per node is less than 7, but for SPH it is about 24. We expect that the difference becomes even more pronounced in three space dimensions. In the definition (54), we can recognize the approximation of the Laplacian operator as used in the Finite Volume Method on Voronoi meshes, which generalizes the standard central finite difference operator on Cartesian meshes.[53] Since

$$\sum_i \sum_j p_i^{n+1}\, \mathbf{B}_{ij}\, p_j^{n+1} = \frac{1}{2\rho} \sum_i \sum_j \frac{|\Gamma_{ij}|^{n+1}}{r_{ij}^{n+1}} (p_{ij}^{n+1})^2, \tag{55}$$

we see that the matrix $\mathbf{B}$ is positive semi-definite and singular, with kernel generated by a constant vector $p_i^{n+1} \equiv 1$ (since Ω is connected). This implies that an additive constant on the pressure must be added to solve the Poisson problem with Neumann boundary conditions. Luckily, when we use a Krylov solver, like the Conjugate Gradient (CG) or the minimal residual (MINRES) method, this never becomes a real issue, since the matrix $\mathbf{B}$ is regular in the invariant subspace of vectors with zero average. The right-hand side $\mathbf{b}$ defined by (52) clearly belongs to this vector space, since

$$\sum_i \mathbf{b}_i = \frac{1}{\Delta t} \sum_i |\omega_i|\, \boldsymbol{v}_i^n \cdot \langle \nabla 1 \rangle_i = 0. \tag{56}$$

Once the new pressure field p_i^{n+1} is determined, the divergence-free velocity field $\boldsymbol{v}_i^{n+1}$ is readily updated by means of (49a).

Remark 7. As an alternative to (53) it could be possible to obtain a sparse matrix by adopting the staggered grid approach, along the lines of the semi-implicit finite volume schemes proposed in.[38] We leave this option for future investigations.

4.4 | Vortex core instability: stabilized pressure gradient

In the SPH method, a well-known numerical artifact, called *tensile instability*, appears in regions with negative pressure and leads to undesirable clustering of particles.[54] In a naive implementation of a LVM, a similar problem arises. Even if every sample point occupies the same area in the sense of the Voronoi mesh, this does not forbid the seeds to approach each other arbitrarily close, leading to stability issues. This problem is particularly evident near pressure minima (such as vortex cores). To shed some light on this problem, let us suppose that close to a seed $\boldsymbol{x}_i$, the pressure is given by

$$p = \lambda\, |\boldsymbol{x}-\boldsymbol{x}_i|^2, \tag{57}$$

where λ is a positive constant. Then $\nabla p(\boldsymbol{x}_i) = 0$ and the particle should not be accelerated. Unfortunately, this is not what we observe because at the discrete level, the strong gradient operator (23) applied to the above pressure filed yields

$$\langle \nabla p\rangle_i = \lambda \langle \nabla |\boldsymbol{x}-\boldsymbol{x}_i|^2\rangle_i = \frac{\lambda}{|\omega_i|}\sum_j |\Gamma_{ij}|\; r_{ij}\;(\boldsymbol{m}_{ij}-\boldsymbol{x}_i) \neq 0. \tag{58}$$

Indeed, a Voronoi cell ω_i can be divided into subcells (triangles for $d = 2$), each of which is a convex hull of $\{\boldsymbol{x}_i\} \cup \Gamma_{ij}$ and has volume

$$V_{ij} = \frac{1}{d}|\Gamma_{ij}|r_{ij}, \tag{59}$$

and a centroid

$$\boldsymbol{c}_{ij} = \frac{d}{d+1}\boldsymbol{m}_{ij} + \frac{1}{d+1}\boldsymbol{x}_i. \tag{60}$$

Substitution of (59) and (60) into (58) leads to

$$\langle \nabla p\rangle_i = \frac{\lambda(d+1)}{|\omega_i|}\sum_j V_{ij}\left(\boldsymbol{c}_{ij}-\boldsymbol{x}_i\right) = \lambda(d+1)(\boldsymbol{c}_i-\boldsymbol{x}_i), \tag{61}$$

where $\boldsymbol{c}_i$ is the centroid of ω_i. Therefore, unless $\boldsymbol{x}_i = \boldsymbol{c}_i$, the seed $\boldsymbol{x}_i$ will be accelerated away from the centroid, hence distorting the cell. This means that we have to remove the potentially nonzero term given by $\lambda(d+1)(\boldsymbol{c}_i-\boldsymbol{x}_i)$. To determine the positive coefficient λ, let us compute the Laplacian of the pressure field (57), which leads to

$$\lambda = \frac{\Delta p(\boldsymbol{x}_i)}{2d}. \tag{62}$$

The above expression is then inserted in (61) to obtain the following definition of a *stabilized pressure gradient*:

$$\langle \nabla p\rangle_i^{\mathrm{s}} = \langle \nabla p\rangle_i - \frac{d+1}{2d}[\langle \Delta p\rangle_i]^{+}(\boldsymbol{c}_i-\boldsymbol{x}_i), \tag{63}$$

where $\langle \Delta p\rangle_i$ is evaluated relying on the discrete Laplace operator (64), which will be introduced in the next section. Here, $[f]^{+} = \max(f, 0)$ denotes the positive part of a function. We shall use (63) instead of $\langle \nabla p\rangle_i$ in (49a) for the update of the velocity field. Note that this stabilizer does not introduce any numerical parameter nor does it interfere with the exactness of the discrete gradient on linear functions.

Other stabilization techniques for LVM are suggested in the literature and can be used also in our setting. In Reference 15, the clustering is prevented in the spirit of ALE method, by introducing a velocity of coordinates in the direction defined by the vector $\boldsymbol{c}_i - \boldsymbol{x}_i$. In,[29] a stabilizer is designed which adds a repulsion force nearby particles and is proven to be weakly consistent. We prefer to adopt the stabilizer (63) because it is relatively simple to implement and it preserves the fully Lagrangian nature of SILVA.

4.5 | Discrete viscous operator

It remains to discretize the viscous terms in the velocity equation (1a). In a finite volume sense, the Laplacian of a twice differentiable vector field can be estimated as follows:

$$\begin{aligned} \Delta \boldsymbol{v}(\boldsymbol{x}_i) &\approx \frac{1}{|\omega_i|}\int_{\omega_i} \Delta \boldsymbol{v}\ \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{|\omega_i|}\int_{\partial\omega_i} \nabla \boldsymbol{v}\boldsymbol{n}\ \mathrm{d}S \\ &\approx -\frac{1}{|\omega_i|}\sum_j \frac{|\Gamma_{ij}|}{r_{ij}}\boldsymbol{v}_{ij} =: \langle \Delta \boldsymbol{v}\rangle_i. \end{aligned} \tag{64}$$

Using the above definition and assuming a constant kinematic viscosity coefficient ν, the viscous force in the incompressible Navier–Stokes model (1) can be easily implemented within each cell ω_i as

$$\boldsymbol{f}_{\mathrm{visc},i} = \nu\Delta\boldsymbol{v}(\boldsymbol{x}_i) \approx \nu\langle\Delta\boldsymbol{v}\rangle_i = -\frac{\nu}{|\omega_i|}\sum_j \frac{|\Gamma_{ij}|}{r_{ij}}\boldsymbol{v}_{ij}, \tag{65}$$

with $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$. Naturally, if viscosity is included, we can no longer expect the conservation of energy E in (42), since E represents only the mechanical component of energy and does not take into account entropy production. However, the velocity field is still divergence-free and the cell areas $|\omega_i|$ are still preserved. Regarding the time discretization, we include the viscous force right after the advection step (48) in an explicit manner. In this way, we obtain an intermediate velocity field

$$\boldsymbol{v}_i^{n*} = \boldsymbol{v}_i^n + \Delta t\, \boldsymbol{f}_{\mathrm{visc},i}(\boldsymbol{x}_i^{n+1}, \boldsymbol{v}_i^n), \tag{66}$$

which is then projected to a solenoidal space by the implicit pressure step (49) with $\boldsymbol{v}^{n*}$ in place of $\boldsymbol{v}^n$.

Remark 8. To prescribe a Dirichlet condition for the velocity vector of the type

$$\boldsymbol{v} = \boldsymbol{v}_D, \quad \boldsymbol{x} \in \Gamma_D, \tag{67}$$

on a linear boundary Γ_D, we need to take into account the viscous force by which walls act onto Voronoi cells. The force can be introduced by means of a fictitious mirror polygon ω_j', which is the reflection of ω_i with respect to Γ_D and which has velocity

$$\boldsymbol{v}_j' = 2\boldsymbol{v}_D - \boldsymbol{v}_i. \tag{68}$$

Including these reflected cells, Equation (65) can be used without modification by setting $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j'$. Following the same reasoning, no-slip boundary conditions are obtained by setting $\boldsymbol{v}_j' = -\boldsymbol{v}_i$, which means $\boldsymbol{v}_D = \boldsymbol{0}$.

4.6 | Summary of the SILVA method

Finally, let us briefly summarize what one time step of SILVA looks like.

1. Update of the seed positions $\boldsymbol{x}_i^{n+1}$ with (48), hence by solving explicitly the trajectory equation (1c).
2. Re-generation of the Voronoi mesh according to the procedure detailed in Section 3. Update of all geometry related quantities (facet lengths, midpoints and inter-seed distances).
3. Explicit computation of the viscous forces using (65).
4. Computation of the new pressure p_i^{n+1} by implicitly solving the discrete pressure Poisson equation (50) with the matrix $\mathbf{B}$ given by (54) and the right-hand-side vector $\mathbf{b}$ given by (52) using the MINRES method.
5. Update of the divergence-free velocity field $\boldsymbol{v}_i^{n+1}$ by means of (49a) with the stabilized pressure gradient (63).

5 | NUMERICAL RESULTS

5.1 | Taylor–Green vortex

We consider the Taylor–Green vortex benchmark on the computational domain $\Omega = [-0.5, 0.5]^2$. This test case has the analytical solution in the form of an exponentially decaying vortex with the velocity field given by

$$\boldsymbol{v}(t,x,y) = \begin{pmatrix} \cos \pi x \sin \pi y \\ -\sin \pi x \cos \pi y, \end{pmatrix} \exp\left(-\frac{2t\pi^2}{\mathrm{Re}}\right), \tag{69}$$

while the pressure field is

$$p(t,x,y) = \frac{\sin^2 \pi x + \sin^2 \pi x - 1}{2} \exp\left(-\frac{2t\pi^2}{\mathrm{Re}}\right). \tag{70}$$

The total kinetic energy is

$$E = \frac{1}{2} \exp\left(-\frac{2t\pi^2}{\mathrm{Re}}\right). \tag{71}$$

$\mathrm{Re} = U_0\, L_0/\nu$ is the Reynolds number and it represents the ratio between inertial and viscous forces in the studied phenomenon. $U_0 = 1$ and $L_0 = 1$ are the reference velocity and domain size, respectively. A free-slip boundary and a compatible initial condition is prescribed. In accordance with Reference 41, the simulation is terminated at time $t_f = 0.2$, and we measure the energy conservation error as well as the L^2 error of pressure, velocity and the divergence of velocity, evaluated using the discrete operator (33). The mesh convergence is tested on two types of Voronoi meshes: one is initiated as a structured Cartesian mesh, while the other one as an unstructured mesh, see Figure 4. Three different values of Reynolds number are considered, namely $\mathrm{Re} = \{400, 1000, \infty\}$, and the obtained convergence rates are depicted in Figure 5. The convergence tables to these graphs can be found in B. Compared to the results by Borgers & Peskin,[43] we report a super-linear, rather than linear, convergence in velocity and we do not suffer any major issues in the inviscid case $\mathrm{Re} = \infty$. However, we also observe a relatively larger error in the pressure field. The divergence of the velocity field is not conserved up to the machine precision, due to the adoption of a sparser matrix and algebraic errors, but has a very fast convergence rate.

5.2 | Gresho vortex

The Gresho vortex problem is a standard test case for inviscid fluids. The initial conditions impose a vortex with a triangular velocity profile. With no viscosity, the flow field is stable under evolution, but this behavior is difficult to recover numerically. For the initial setup of this test, we refer to Reference 55. The computational domain is initially discretized with three different resolutions, namely with $N = 50^2$, $N = 100^2$ and $N = 200^2$ Voronoi seeds. The results of the

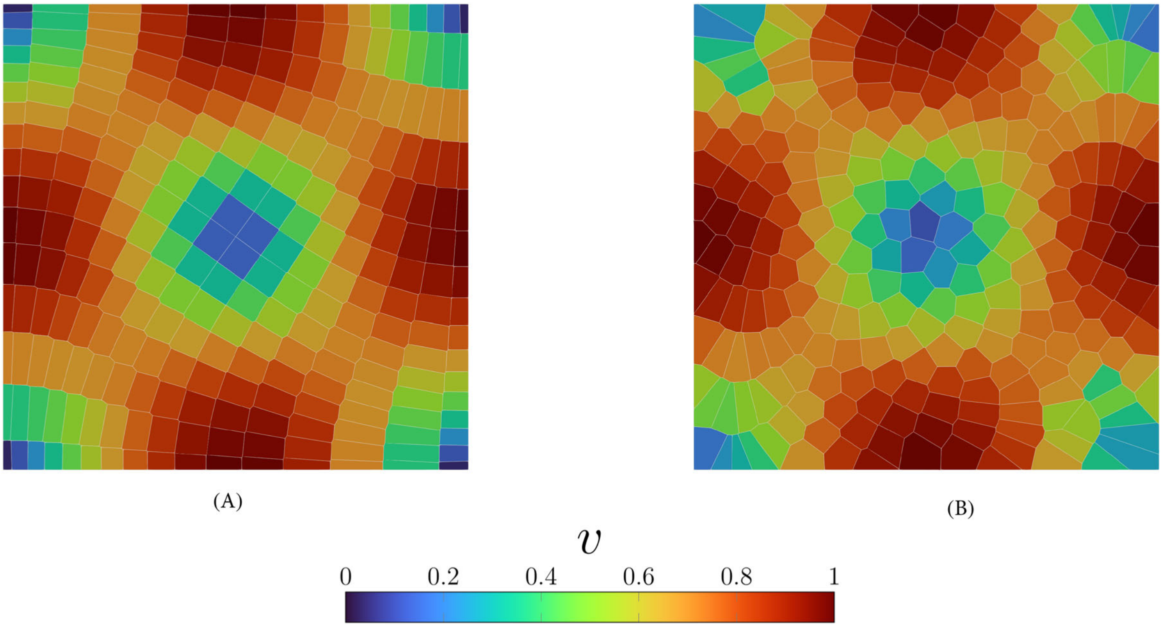
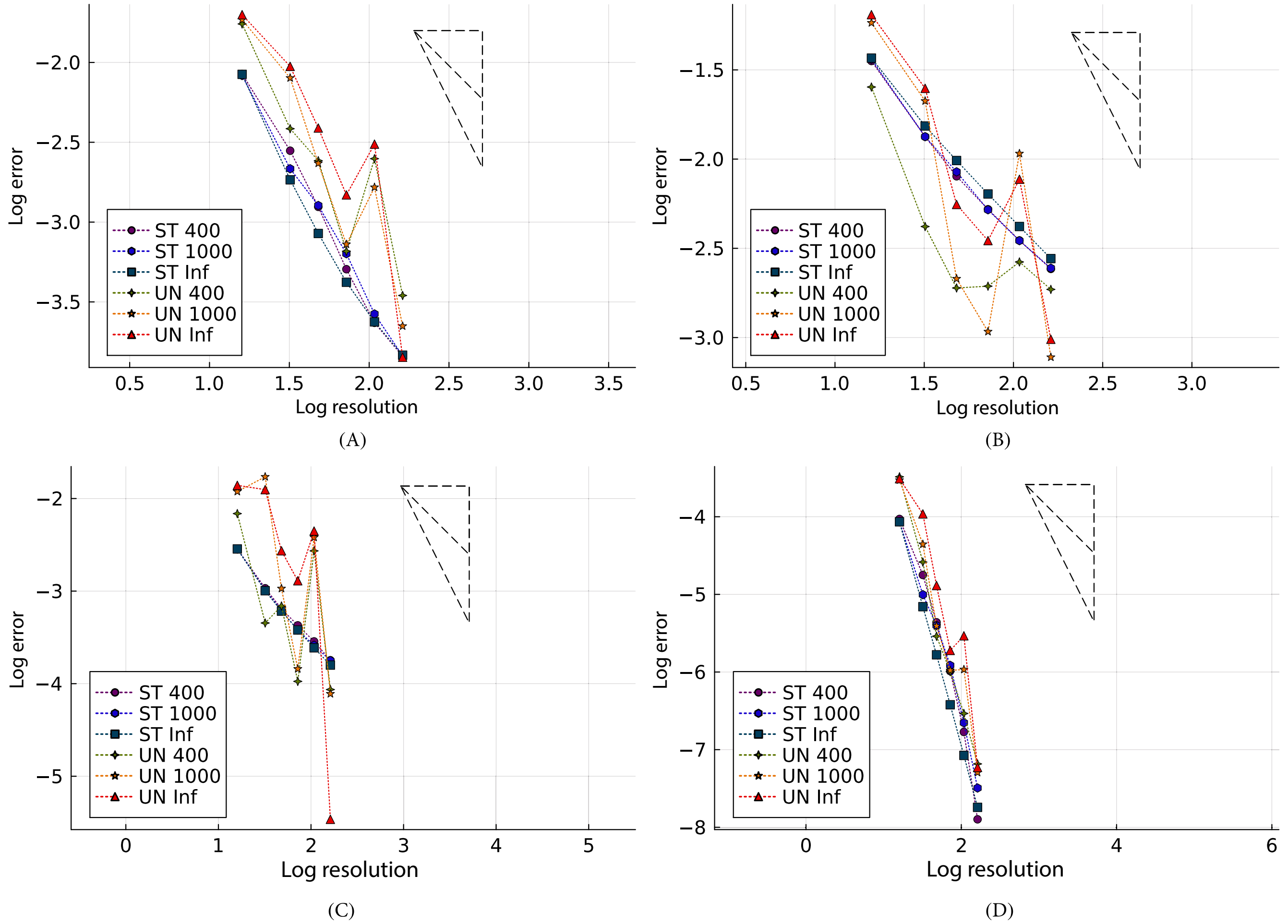


FIGURE 4 The final time step of Taylor–Green Vortex for $N = 16$, $\mathrm{Re} = \infty$. (A) Rectangular grid; (B) Unstructured grid. [Colour figure can be viewed at wileyonlinelibrary.com]



(A)

(B)

(C)

(D)

FIGURE 5 Errors of velocity, pressure, energy and velocity divergence in the Taylor–Green vortex benchmark. The horizontal axis corresponds to $\log_{10} N$, where N is the number of cells per side of the computational domain. The vertical axis indicates the logarithm of error. We also compare the result for a structured (ST) rectangular and an unstructured (UN) Vogel grid for $\text{Re} = \{400, 1000, \infty\}$. The dashed triangles indicate what the reference linear and quadratic convergence slopes. The convergence curves on unstructured grids are more erratic for unknown reasons. Nonetheless, the overall convergence rate is comparable. (A) Velocity error; (B) Pressure error; (C) Energy error; (D) Divergence error. [Colour figure can be viewed at wileyonlinelibrary.com]

simulation are depicted in Figure 6, and they indicate that SILVA preserves the triangular profile with reasonable accuracy, while exhibiting some diffusive behavior. Furthermore, mesh convergence with the respect to the loss of energy is shown. The color map of the velocity field is shown in Figure 7. Note that the final time $t_f = 3$ corresponds to more than three full revolutions of the vortex, which would be clearly impossible to simulate on a Lagrangian mesh with fixed topology. Without using the stabilizer introduced in Section 4.4, we observe the well-known vortex core instability, as illustrated in Figure 8.

5.3 | Lid-driven cavity

To test the correct simulation of the viscous forces, we consider the lid-driven cavity problem for an incompressible fluid. The computational domain is given by $\Omega = [-0.5, 0.5]^2$, and the fluid is initially at rest, that is, $p = 0$ and $\boldsymbol{v} = \boldsymbol{0}$. No-slip wall boundary conditions are defined on the vertical sides ($x = \pm 0.5$) and at the bottom ($y = -0.5$) of the domain, while on the top side ($y = 0.5$) the velocity field $\boldsymbol{v} = (1, 0)$ is imposed. The simulations are performed as time-dependent problems on a Voronoi grid of 10,000 sample points, and four different Reynolds numbers are consider, namely $\text{Re} = \{100, 400, 1000, 3200\}$. An excellent agreement with the notorious referential solution by Ghia et al.[56] is achieved and

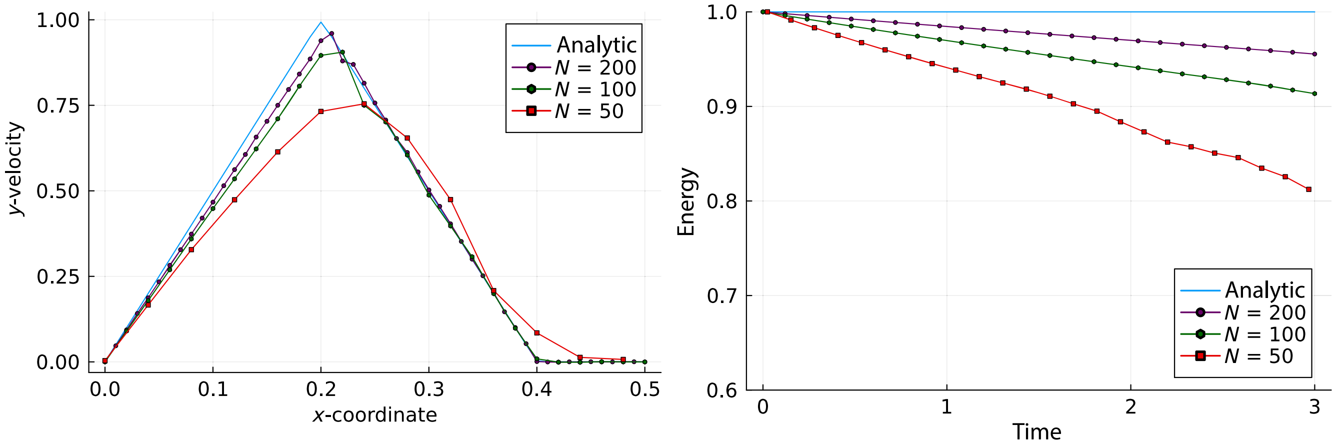


FIGURE 6 Left: comparison of the tangential velocity along the x-axis with the exact solution in the Gresho vortex benchmark. The result is at $t = 3$ and compares three different resolutions and an analytical solution. Right: time evolution of energy (relative to the initial state). The graph can be compared to the high-order semi-implicit method presented in Reference 33. [Colour figure can be viewed at wileyonlinelibrary.com]

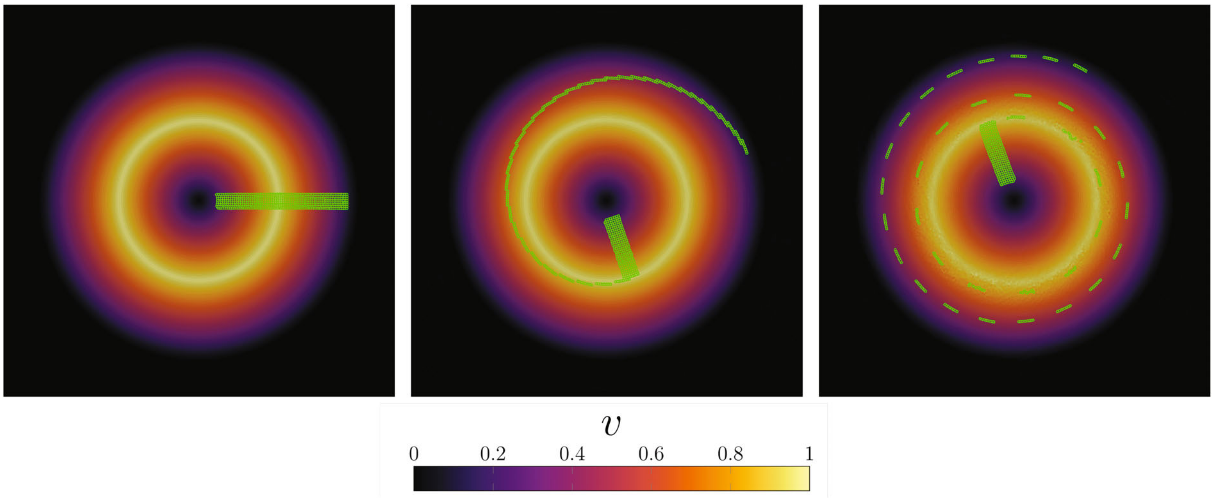


FIGURE 7 Gresho vortex at $t = \{0, 1, 3\}$. The color map indicates magnitude of velocity. A group of Voronoi cells is highlighted in green. In the last frame, the vortex structure is starting to deteriorate. The resolution in this figure is 200 × 200. [Colour figure can be viewed at wileyonlinelibrary.com]

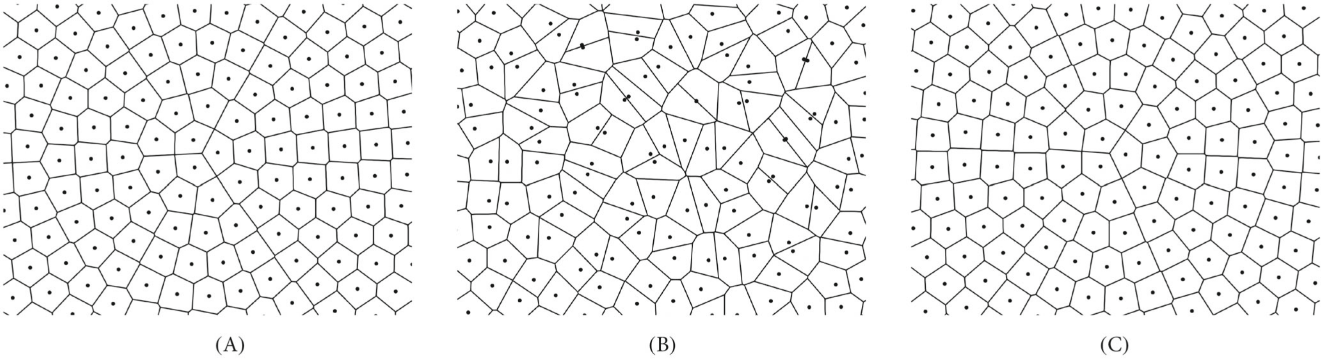


FIGURE 8 Zoom at the vortex core at $t = 0$ and $t = 0.8$ with and without stabilization. The Voronoi mesh becomes distorted as seeds are propagated away from their respective centroids. (A) Initial state; (B) Unstable ($t = 0.8$). (C) Stabilized ($t = 0.8$).

plot in Figure 9. More precisely, the agreement for Reynolds numbers Re = 100, 400, 1000 is almost perfect. At Re = 3200, we observe some discrepancy, presumably because of the numerical diffusion and insufficient resolution near boundary layers. However, SILVA is sufficiently precise to indicate subtle features of the flow, such as the left and right bottom corner vortices as well as the emergence of top left corner vortex for Re = 3200, see Figure 10. To emphasize the Lagrangian character of SILVA, Figure 11 shows the trajectories of some Voronoi cells.

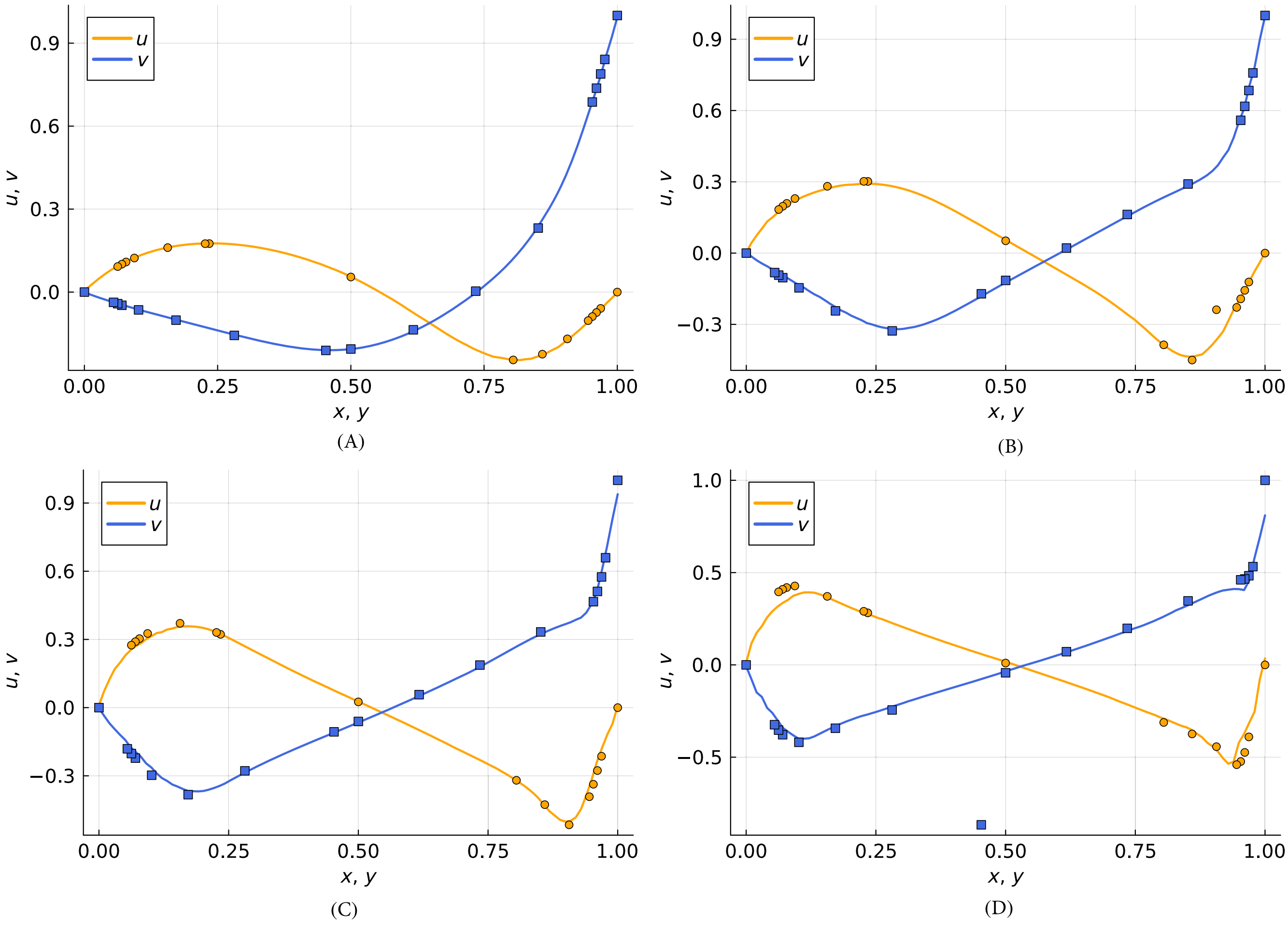
5.4 | Rayleigh–Taylor instability

Finally, we present the simulation of a multiphase problem with SILVA, namely the Rayleigh–Taylor instability[57] of an inviscid fluid without surface tension. Our setup is the same as in the SPH study presented in Reference 58. The domain is the rectangle $\Omega = [0, 1] \times [0, 2]$ which is discretized with 80,000 Voronoi seeds. The initial density field is given by

$$\rho(x, y) = \begin{cases} 1.8 & y > \phi(x) \\ 1 & y < \phi(x) \end{cases}. \tag{72}$$

Two immiscible fluids of different densities are involved, which are separated by a curve described as

$$\phi(x) = 1 - 0.15\cos(2\pi x). \tag{73}$$



(A)
(B)
(C)
(D)

FIGURE 9 Horizontal velocity u along the vertical center-line (orange) and vertical velocity v along the horizontal center-line (blue) in the lid-driven cavity benchmark computed using semi-implicit Lagrangian Voronoi approximation (SILVA) for various Reynolds numbers. We compare the result with the referential solution of Ghia et al.[56] (A) Re = 100, $t = 10$; (B) Re = 400, $t = 40$; (C) Re = 1000, $t = 100$ (D) Re = 3200, $t = 320$. [Colour figure can be viewed at wileyonlinelibrary.com]

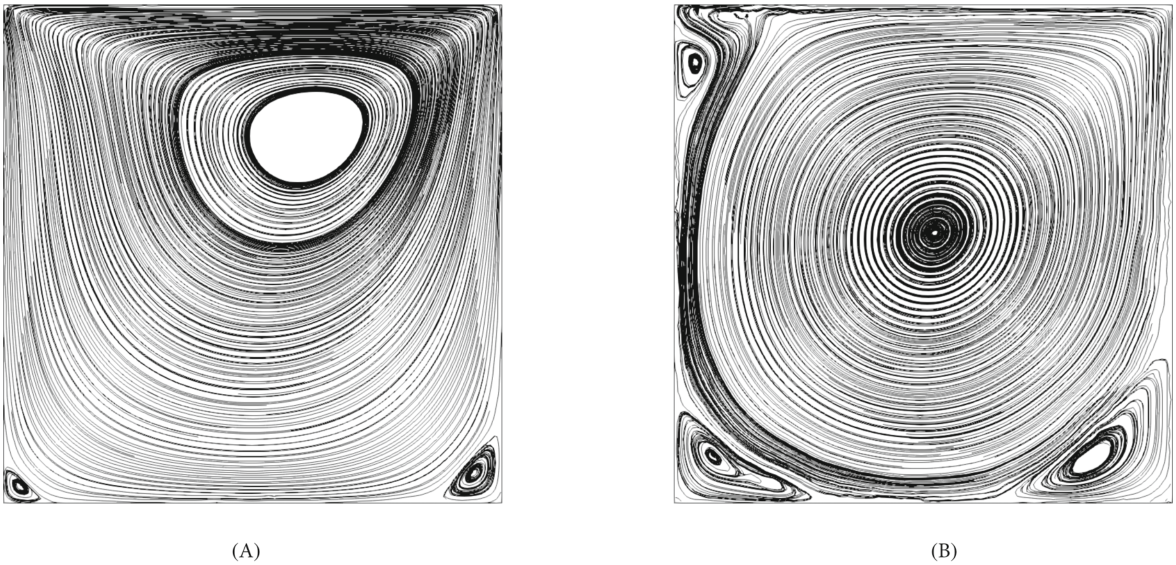


FIGURE 10 Streamline plots in the lid-driven cavity benchmark. (A) Re = 100, t = 10; (B) Re = 3200, t = 320.

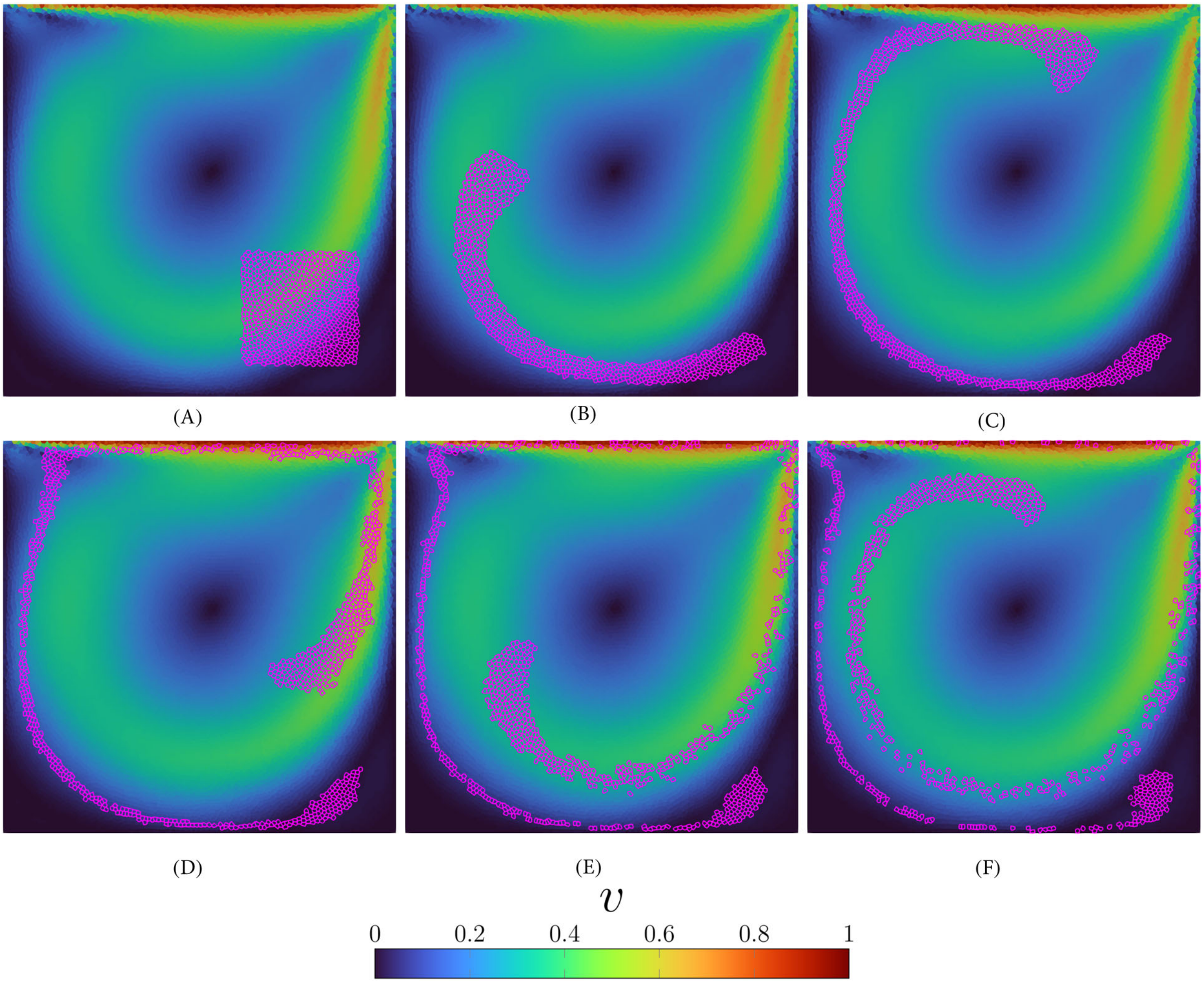


FIGURE 11 Color-map of the velocity magnitude for Re = 1000. A group of Voronoi cells is highlighted in magenta. Some of these cells remain trapped in the corner vortex. (A) t = 90; (B) t = 92; (C) t = 94; (D) t = 96; (E) t = 98; (F) t = 100. [Colour figure can be viewed at wileyonlinelibrary.com]

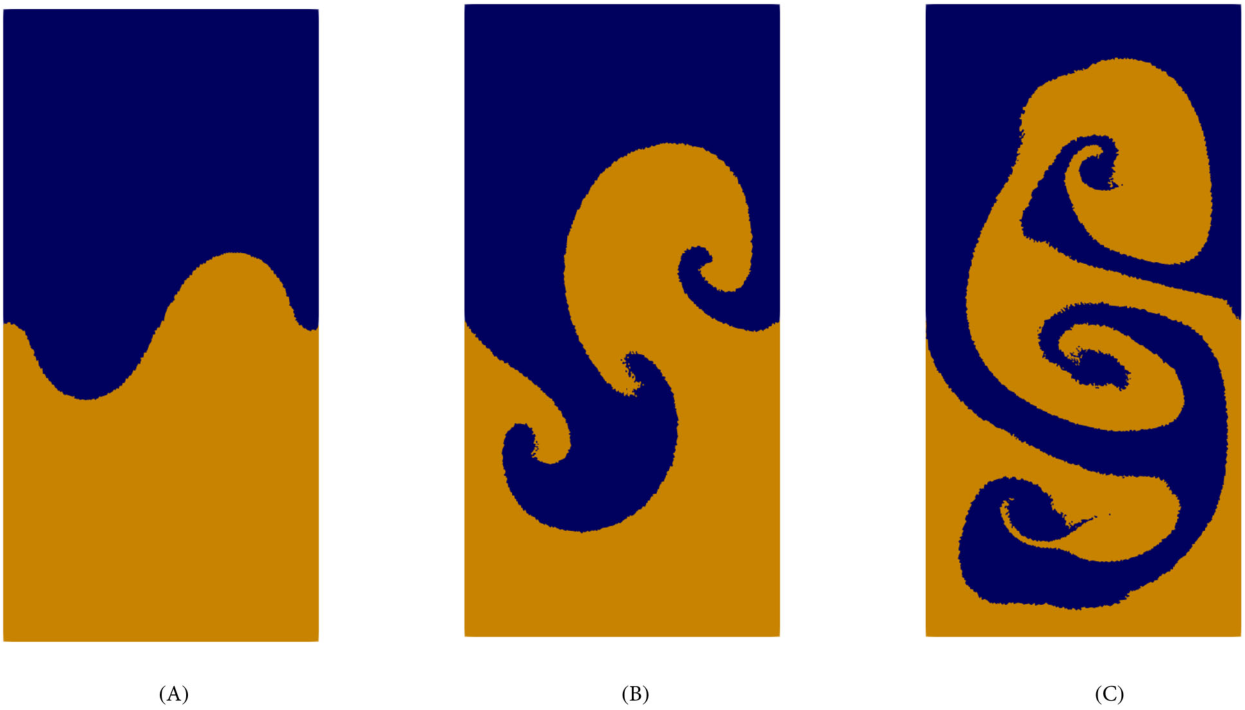


FIGURE 12 Snapshots of the Rayleigh–Taylor instability at output times $t = \{1, 3, 5\}$. The lighter fluid is orange and the heavier is dark blue. The simulation contained 80,000 Voronoi cells. (A) $t = 1$; (B) $t = 3$; (C) $t = 5$. [Colour figure can be viewed at wileyonlinelibrary.com]

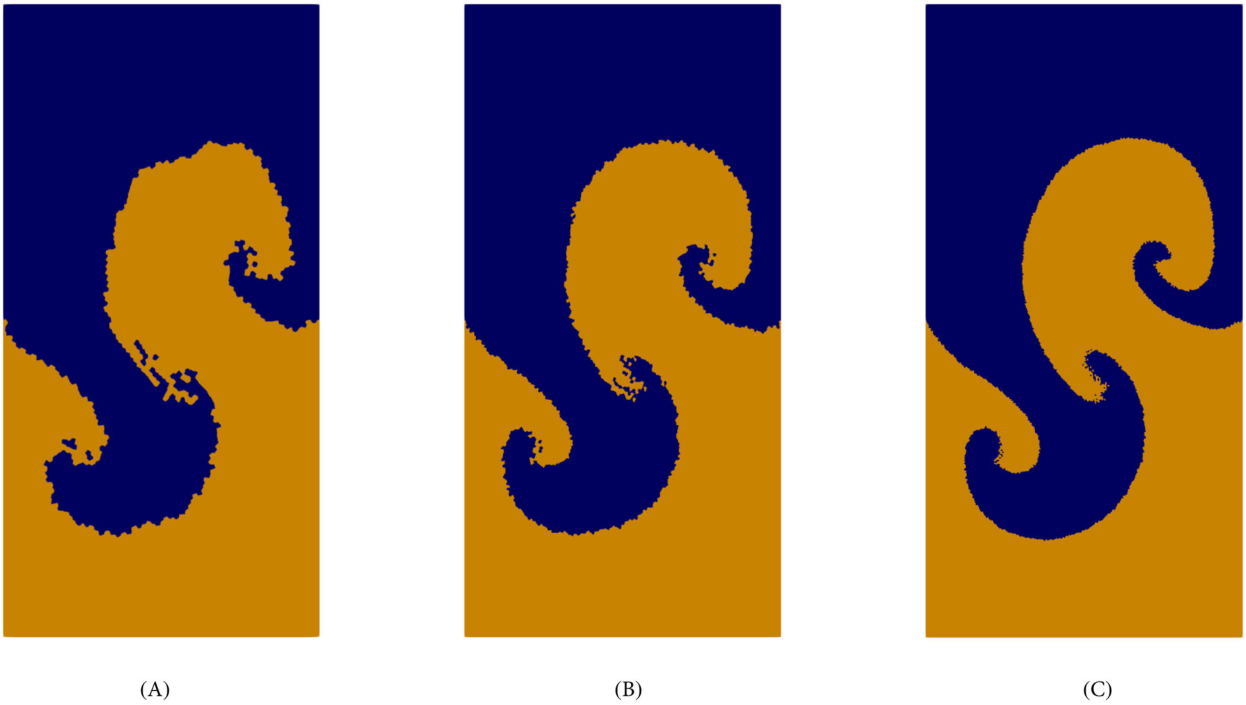


FIGURE 13 The Rayleigh–Taylor instability at $t = 3$ for three different resolutions. N is the number of Voronoi seeds per shorter side of the domain, so there are $2N^2$ Voronoi seeds in total. (A) $N = 60$; (B) $N = 100$; (C) $N = 200$. [Colour figure can be viewed at wileyonlinelibrary.com]

The densities correspond to Atwood number A = 2/7. The two fluids begin at rest, have identical viscosity and are subjected to a homogeneous gravitational field. The Reynolds number is Re = 420 and the Froude number is Fr = 1. No-slip boundary conditions are imposed on all sides of the domain.

This problem is concerned with an incompressible fluid but with a *heterogeneous* density field. Hence, the approximation (53) is no longer valid when we need to take into account the gradient of pressure. The correct discrete equation is now given by

$$-\langle \Delta p \rangle_i^{n+1} = -\frac{\rho_i}{\Delta t} \langle \widetilde{\nabla} \cdot \boldsymbol{v} \rangle_i^n - \frac{1}{\rho_i} \langle \nabla \rho \rangle_i \cdot \langle \nabla p \rangle_i^{n+1}. \tag{74}$$

The above system can be solved quite efficiently by means of fixed-point iterations, with m denoting the current iteration number:

$$-\langle \Delta p^{n+1,(m+1)} \rangle_i = -\frac{\rho_i}{\Delta t} \langle \widetilde{\nabla} \cdot \boldsymbol{v} \rangle_i^n - \frac{1}{\rho_i} \langle \nabla \rho \rangle_i \cdot \langle \nabla p^{n+1,(m)} \rangle_i, \tag{75}$$

starting with $p^{n+1,(0)} = p^n$. The iterative process stops when the condition $|p^{n+1,(m+1)} - p^{n+1,(m)}| < 10^{-12}$ is satisfied. Figure 12 displays the result of SILVA computation, which are in qualitative agreement with the ISPH solution.[58] We also perform a mesh convergence study of this test for three different mesh resolutions ($N = \{60, 100, 200\}$), and the results are depicted in Figure 13. We observe that the interface profile between the two immiscible fluids becomes increasingly more smooth, and the numerical artifacts disappear.

6 | CONCLUSION

In this paper, a new LVM for the solution of the incompressible Navier–Stokes equations has been presented. The spatial discretization is carried out by means of moving Voronoi cells that are regenerated at each time step with an efficient cell-list-based Voronoi mesh generation technique, optimized for multithreaded time-dependent problems. To overcome the problem related to the vortex-core instability associated with the established LVM gradient operator, we have proposed a new stabilizing procedure, which is notable by the lack of numerical parameters and by preserving the exactness of linear gradients. A semi-implicit time discretization has been developed for the first time in the context of LVM, which discretizes explicitly the trajectory equation, that drives the mesh motion, and the viscous terms in the velocity equation. To enforce the projection of the velocity onto the associated divergence-free manifold, the velocity equation is inserted in the divergence-free constraint, yielding an implicit linear system for the pressure that corresponds to the well-known elliptic Poisson equation. The matrix of the system is provably symmetric and can therefore be solved with very efficient iterative linear solvers like the Conjugate Gradient method or MINRES. The accuracy of the scheme has been assessed in four different test-cases, featuring both viscous and inviscid fluids as well as multiphase flow. Linear convergence has been empirically studied in a Gresho vortex benchmark. Our novel scheme, referred to as SILVA, is able to run long-time simulations without any mesh degeneration, inspired by the so-called reconnection-based algorithms on moving meshes.

SILVA is an interesting alternative to the incompressible SPH method, because of the benefits coming from a sparser matrix associated to the elliptic Poisson system, and a relatively easy implementation of free-slip and Dirichlet boundary conditions. Multiphase problems with sharp interface reconstruction and fluid–structure interaction are among the most promising applications. So far, a limitation of our scheme is the requirement of domain convexity and the inability to treat free surfaces. The benchmarks in this paper are two-dimensional, although we do not foresee major obstacles regarding the extension to three-dimensional problems. Further research could be directed toward improving the order of convergence, both in space (by means of a second-order reconstruction) and in time (using high-order IMEX schemes, such as CNAB integrator or implicit-explicit midpoint rule[41]). Compressible flows will also be addressed in the future, by solving a mildly nonlinear system for the pressure. For the sake of simplicity, we have used an explicit treatment of the viscous terms, but an implicit approach would be preferable for flows with very small Reynolds number, as presented in Reference 59.

ACKNOWLEDGMENTS

This work was financially supported by the Italian Ministry of University and Research (MUR) in the framework of the PRIN 2022 project number 2022N9BM3N. WB received financial support by Fondazione Cariplo and Fondazione CDP

(Italy) under the project number 2022-1895. IP and WB are members of the INdAM GNCS group in Italy. Open access publishing facilitated by Universita degli Studi di Trento, as part of the Wiley - CRUI-CARE agreement.

DATA AVAILABILITY STATEMENT

The algorithm presented in this paper is implemented in the code which is publicly available under the MIT license.[60] All other data are available upon reasonable request to the corresponding author.

REFERENCES

1. Winslow AM. "Equipotential" Zoning of Two-Dimensional Meshes. Technical Report. Lawrence Livermore Lab: California University. 1963.
2. Winslow AM. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J Comput Phys*. 1966;1(2):149-172.
3. Dobrev V, Knupp P, Kolev T, Mittal K, Tomov V. The target-matrix optimization paradigm for high-order meshes. *SIAM J Sci Comput*. 2019;41(1):B50-B68.
4. Galera S, Maire PH, Breil J. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *J Comput Phys*. 2010;229(16):5755-5787.
5. Boscheri W, Dumbser M. A direct arbitrary-Lagrangian–Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3D. *J Comput Phys*. 2014;275:484-523.
6. Boscheri W, Dumbser M. Arbitrary-Lagrangian–Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. *J Comput Phys*. 2017;346:449-479.
7. Dukowicz JK, Kodis JW. Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations. *SIAM J Sci Stat Comput*. 1987;8(3):305-321.
8. Loubere R, Shashkov MJ. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods. *J Comput Phys*. 2005;209(1):105-138.
9. Kucharik M, Shashkov M. Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods. *J Comput Phys*. 2014;258:268-304.
10. Pasta JR, Ulam S. Heuristic numerical work in some problems of hydrodynamics. *Math Tables Other Aid Comput*. 1959;13(65):1-12.
11. Fritts MJ, Crowley WP, Trease H. *The Free-Lagrange Method: Proceedings of the First International Conference on Free-Lagrange Methods, Held at Hilton Head Island, South Carolina, March 4–6, 1985*. Springer; 1985.
12. Crowley W. FLAG: a free-Lagrange method for numerically simulating hydrodynamic flows in two dimensions. *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics: September 15–19, 1970 University of California, Berkeley*. Springer; 2005:37-43.
13. Loubere R, Maire PH, Shashkov M, Breil J, Galera S. ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method. *J Comput Phys*. 2010;229(12):4724-4761.
14. Bo W, Shashkov M. Adaptive reconnection-based arbitrary Lagrangian Eulerian method. *J Comput Phys*. 2015;299:902-939.
15. Springel V. Hydrodynamic simulations on a moving Voronoi mesh. *arXiv preprint arXiv:1109.2218* 2011.
16. Gaburro E, Boscheri W, Chiocchetti S, Klingenberg C, Springel V, Dumbser M. High order direct arbitrary-Lagrangian-Eulerian schemes on moving Voronoi meshes with topology changes. *J Comput Phys*. 2020;407:109167.
17. Monaghan JJ. Smoothed particle hydrodynamics. *Annu Rev Astron Astrophys*. 1992;30:543-574.
18. Monaghan JJ. Smoothed particle hydrodynamics and its diverse applications. *Annu Rev Fluid Mech*. 2012;44:323-346.
19. Bašić J, Degiuli N, Blagojević B, Ban D. Lagrangian differencing dynamics for incompressible flows. *J Comput Phys*. 2022;462:111198.
20. Violeau D, Rogers BD. Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future. *J Hydraul Res*. 2016;54(1):1-26.
21. Zhu Q, Hernquist L, Li Y. Numerical convergence in smoothed particle hydrodynamics. *Astrophys J*. 2015;800(1):6.
22. Kulasegaram S, Bonet J, Lok T, Rodriguez-Paz M. Corrected smooth particle hydrodynamics-a meshless method for computational mechanics. *CD-Rom Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS-2000, Barcelona, 11-14 September*. CIMNE; 2000.
23. Oger G, Doring M, Alessandrini B, Ferrant P. An improved SPH method: towards higher order convergence. *J Comput Phys*. 2007;225(2):1472-1492.
24. Violeau D. *Fluid Mechanics and the SPH Method: Theory and Applications*. Oxford University Press; 2012.
25. Trease HE, Fritts MF, Crowley WP, eds. *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*. Lecture Notes in Physics. Vol 395. Springer Berlin Heidelberg; 1991.
26. Ball G. A free-Lagrange method for unsteady compressible flow: simulation of a confined cylindrical blast wave. *Shock Wave*. 1996;5:311-325.
27. Howell B, Ball G. A free-Lagrange augmented Godunov method for the simulation of elastic–plastic solids. *J Comput Phys*. 2002;175(1):128-167. doi:10.1006/JCPH.2001.6931
28. Heß S, Springel V. Particle hydrodynamics with tessellation techniques. *Mon Notice R Astron Soc*. 2010;406(4):2289-2311.
29. Després B. Lagrangian Voronoi meshes and particle dynamics with shocks. *Comput Methods Appl Mech Eng*. 2024;418:116427.
30. Osher S, Solomon F. A partially implicit method for large stiff systems of Ode's with only few equations introducing small time-constants. *SIAM J Numer Anal*. 1976;13:645-663.

31. Pareschi L, Russo G. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *J Sci Comput*. 2005;25:129-155.
32. Dumbser M, Casulli V. A conservative, weakly nonlinear semi-implicit finite volume scheme for the compressible Navier-Stokes equations with general equation of state. *Appl Math Comput*. 2016;272:479-497.
33. Boscheri W, Pareschi L. High order pressure-based semi-implicit IMEX schemes for the 3D Navier-stokes equations at all Mach numbers. *J Comput Phys*. 2021;434:110206.
34. Izzo G, Jackiewicz Z. Strong stability preserving IMEX methods for partitioned systems of differential equations. *Commun Appl Math Comput*. 2021;3(4):719-758.
35. Izzo G, Jackiewicz Z. Construction of SDIRK methods with dispersive stability functions. *Appl Numer Math*. 2021;160:265-280.
36. Izzo G, Jackiewicz Z. Highly stable implicit–explicit Runge–Kutta methods. *Appl Numer Math*. 2017;113:71-92.
37. Toro E, Vázquez-Cendón M. Flux splitting schemes for the Euler equations. *Comput Fluid*. 2012;70:1-12.
38. Casulli V. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *J Comput Phys*. 1990;86:56-74.
39. Park J, Munz CD. Multiple pressure variables methods for fluid flow at all Mach numbers. *Int J Num Meth Fluids*. 2005;49:905-931.
40. Boscheri W, Pisaturo G, Righetti M. High order divergence-free velocity reconstruction for free surface flows on unstructured Voronoi meshes. *Int J Num Meth Fluids*. 2019;90:296-321.
41. Carlino MG, Boscheri W. Arbitrary-Lagrangian-Eulerian finite volume IMEX schemes for the incompressible Navier-Stokes equations on evolving chimera meshes. *J Comput Phys*. 2024;501:112764.
42. Börgers C, Peskin CS. A Lagrangian method based on the Voronoi diagram for the incompressible Navier stokes equations on a periodic domain. *The Free-Lagrange Method*. Springer-Verlag; 1985:87-113.
43. Börgers C, Peskin CS. A Lagrangian fractional step method for the incompressible Navier-stokes equations on a periodic domain. *J Comput Phys*. 1987;70(2):397-438.
44. Serrano M, Espanol P. Thermodynamically consistent mesoscopic fluid particle model. *Phys Rev E*. 2001;64(4):046115.
45. Aurenhammer F, Klein R, Lee DT. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific Publishing Company; 2013.
46. Barber CB, Dobkin DP, Huhdanpaa H. Qhull: Quickhull algorithm for computing the convex hull. *Astrophys Source Code Lib*. 2013;22(4):ascl–1304.
47. Rycroft C. VORO++: A three-dimensional Voronoi cell library in C++. 2009.
48. Fabri A, Pion S. CGAL: the computational geometry algorithms library. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM; 2009:538-539.
49. Ray N, Sokolov D, Lefebvre S, Lévy B. Meshless Voronoi on the GPU. *ACM Trans Graph*. 2018;37(6):1-12.
50. Liu X, Ma L, Guo J, Yan DM. Parallel computation of 3D clipped Voronoi diagrams. *IEEE Trans vis Comput Graph*. 2020;28(2):1363-1372.
51. Springel V. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Mon Notice R Astron Soc*. 2010;401(2):791-851.
52. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *J Comput Phys*. 1997;135(2):118-125.
53. Eymard R, Gallouët T, Herbin R. Finite volume methods. *Handbook of Numerical Analysis*. Vol 7. Elsevier; 2000:713-1018.
54. Monaghan JJ. SPH without a tensile instability. *J Comput Phys*. 2000;159(2):290-311.
55. Liska R, Wendroff B. Comparison of several difference schemes on 1D and 2D test problems for the Euler equations. *SIAM J Sci Comput*. 2003;25(3):995-1017.
56. Ghia U, Ghia KN, Shin C. High-Re solutions for incompressible flow using the Navier-stokes equations and a multigrid method. *J Comput Phys*. 1982;48(3):387-411.
57. Rayleigh. Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. *Proce Lond Math Soc*. 1882;1(1):170-177.
58. Hu X, Adams NA. An incompressible multi-phase SPH method. *J Comput Phys*. 2007;227(1):264-278.
59. Boscheri W, Tavelli M. High order semi-implicit schemes for viscous compressible flows in 3D. *Appl Math Comput*. 2022;434:127457.
60. Kincl O. LagrangianVoronoi.jl. https://github.com/OndrejKincl/LagrangianVoronoi.jl
61. Bartle RG. *The Elements of Integration and Lebesgue Measure*. John Wiley & Sons; 2014.

How to cite this article: Kincl O, Peshkov I, Boscheri W. Semi-implicit Lagrangian Voronoi approximation for the incompressible Navier–Stokes equations. *Int J Numer Meth Fluids*. 2024;1-28. doi: 10.1002/fld.5339

APPENDIX A. PROOF OF THEOREM 1

We remind that Ω is an open, bounded and convex polytope and the seeds $\boldsymbol{x}_i \in \Omega$ are distinct. We wish to prove that for any $j \neq i$:

$$\frac{\partial |\omega_i|}{\partial \boldsymbol{x}_j} = -\frac{|\Gamma_{ij}|}{r_{ij}} \left(\boldsymbol{m}_{ij} - \boldsymbol{x}_j \right), \tag{A1}$$

and that

$$\frac{\partial|\omega_i|}{\partial \boldsymbol{x}_i} = -\sum_{j\neq i}\frac{\partial|\omega_j|}{\partial \boldsymbol{x}_i} = -\sum_{j\neq i}\frac{\partial|\omega_i|}{\partial \boldsymbol{x}_j} - \boldsymbol{S}_i. \tag{A2}$$

Proof. Let us show (A1) for $i = 1$ and $j = 2$. Choosing an appropriate coordinate system, we can freely assume $\boldsymbol{x}_1 = \boldsymbol{0}$ and

$$\boldsymbol{x}_2 = \begin{pmatrix} \mu \\ \boldsymbol{\nu} \end{pmatrix}, \tag{A3}$$

where $\mu > 0$ and $\boldsymbol{\nu} \in \mathbb{R}^{d-1}$. Let us define

$$\tilde{\omega}_1 = \Omega \cap \bigcap_{k=3}^{N} \{\boldsymbol{x} : |\boldsymbol{x} - \boldsymbol{x}_1| \leq |\boldsymbol{x} - \boldsymbol{x}_k|\}. \tag{A4}$$

Then $\tilde{\omega}_1$ does not depend on $\boldsymbol{x}_2$ and

$$\omega_1 = \{\boldsymbol{x} \in \tilde{\omega}_1 : |\boldsymbol{x} - \boldsymbol{x}_1| \leq |\boldsymbol{x} - \boldsymbol{x}_2|\} = \left\{ \begin{pmatrix} z \\ \boldsymbol{w} \end{pmatrix} \in \tilde{\omega}_1 : z \leq \zeta(\boldsymbol{w}) \right\}, \tag{A5}$$

where

$$\zeta(\boldsymbol{w}) = \frac{\mu^2 + |\boldsymbol{\nu}|^2 - 2\boldsymbol{\nu}\cdot\boldsymbol{w}}{2\mu}. \tag{A6}$$

Having this notation in place, we can perform a formal computation:

$$\begin{aligned} \frac{\partial|\omega_i|}{\partial \mu} &= \frac{\partial}{\partial \mu}\int_{\omega_1} \mathrm{d}\boldsymbol{x} \\ &= \frac{\partial}{\partial \mu}\int_{\mathbb{R}^{d-1}}\int_{-\infty}^{\zeta(\boldsymbol{w})} 1_{\tilde{\omega}_1}(z, \boldsymbol{w})\mathrm{d}z\mathrm{d}\boldsymbol{w} \\ &\overset{*}{=} \int_{\mathbb{R}^{d-1}}\frac{\partial}{\partial \mu}\int_{-\infty}^{\zeta(\boldsymbol{w})} 1_{\tilde{\omega}_1}(z, \boldsymbol{w})\mathrm{d}z\mathrm{d}\boldsymbol{w} \\ &\overset{\dagger}{=} \int_{\mathbb{R}^{d-1}}\frac{\partial \zeta}{\partial \mu}(\boldsymbol{w})1_{\tilde{\omega}_1}(\zeta(\boldsymbol{w}), \boldsymbol{w})\mathrm{d}\boldsymbol{w} \\ &= \frac{1}{\mu}\int_{\mathbb{R}^{d-1}}(\mu - \zeta(\boldsymbol{w}))1_{\tilde{\omega}_1}(\zeta(\boldsymbol{w}), \boldsymbol{w})\mathrm{d}\boldsymbol{w}. \end{aligned} \tag{A7}$$

And similarly:

$$\frac{\partial|\omega_i|}{\partial \boldsymbol{\nu}} = \frac{1}{\mu}\int_{\mathbb{R}^{d-1}}(\boldsymbol{\nu} - \boldsymbol{w})1_{\tilde{\omega}_1}(\zeta(\boldsymbol{w}), \boldsymbol{w})\mathrm{d}\boldsymbol{w}. \tag{A8}$$

Needless to say, it is not *a priori* clear that all steps in this computation were legal. To begin with, the interchange of the integral and derivative sign $(*)$ requires a further justification. So does the usage of Leibniz rule for a discontinuous integrand $(\dagger)$. We shall address these concerns later. Now, we notice that

$$\boldsymbol{w} \mapsto \begin{pmatrix} \zeta(\boldsymbol{w}) \\ \boldsymbol{w} \end{pmatrix}, \tag{A9}$$

parametrizes a plane (or line, if $d = 2$). Let us call the plane P_{12}. Then P_{12} has a surface element

$$\mathrm{d}S = \sqrt{1 + \left|\frac{\mathrm{d}\zeta}{\mathrm{d}w}\right|^2}\,\mathrm{d}\boldsymbol{w} = \frac{r_{12}}{\mu}\mathrm{d}\boldsymbol{w}. \tag{A10}$$

and $P_{12} \cap \tilde{\omega}_1 = \Gamma_{12}$. Combining (6), (A8), and (A10) yields

$$\frac{\partial|\omega_1|}{\partial \boldsymbol{x}_2} = \frac{1}{r_{12}} \int_{\Gamma_{12}} (\boldsymbol{x}_2 - \boldsymbol{x})\mathrm{d}S(\boldsymbol{x}) = -\frac{|\Gamma_{12}|}{r_{12}}(\boldsymbol{m}_{12} - \boldsymbol{x}_2), \tag{A11}$$

which is the sought result (A1). Equation (A2) can be easily deduced from (A1) by writing

$$|\Omega| = \sum_j |\omega_j|, \tag{A12}$$

and differentiating both sides of the equation with respect to $\boldsymbol{x}_i$.

To justify the step (†), we would like to verify

$$\frac{\mathrm{d}}{\mathrm{d}\zeta} \int_{-\infty}^{\zeta} 1_{\tilde{\omega}_1}(z, \boldsymbol{w})\mathrm{d}z = 1_{\tilde{\omega}_1}(\zeta, \boldsymbol{w}), \tag{A13}$$

for all $\boldsymbol{w} \in \mathbb{R}^{d-1}$. Such claim is quite challenging to prove because it is not true. It only holds when $1_{\tilde{\omega}_1}$ is continuous at $(\zeta, \boldsymbol{w})$. Fortunately, it is enough to show (A13) for *almost every* $\boldsymbol{w} \in \mathbb{R}^{d-1}$. In other words, we need that $\partial\tilde{\omega}_1 \cap P_{12}$ is a null set with respect to the surface measure dS on P_{12}. To see this, consider that $\partial\tilde{\omega}_1$ is a subset of a finite union of planes, where each of them falls into one of two categories:

- a plane Q that extends a facet of the polytope $\partial\Omega$,
- a plane P_{1k}, which is the locus of points equidistant to $\boldsymbol{x}_1$ and some other generating seed $\boldsymbol{x}_k$, where $k > 2$.

In the former case, Ω is contained in the open half-space defined by the plane Q and the inward normal vector. Then, $Q \neq P_{12}$ because $P_{12} \cap \Omega$ is nonempty. (It is here and only here, where the convexity of Ω is required.) In the latter case, we have $P_{1k} \neq P_{12}$ since $\boldsymbol{x}_k$ is a reflection of $\boldsymbol{x}_1$ with respect to P_{1k} and $\boldsymbol{x}_k \neq \boldsymbol{x}_2$. Since the planes which compose (a superset of) the boundary of $\partial\tilde{\omega}_1$ are different from P_{12}, their intersection is either empty or a linear space of lower dimension and hence, a null set in the dS measure.

To finalize the proof, we need to justify (∗). Using the dominated convergence theorem (see, e.g., Corol. 5.7 in Reference 61), it suffices to find an integrable function g($\boldsymbol{w}$) such that

$$\mathrm{g}(\boldsymbol{w}) \geq \left|\frac{\partial}{\partial\mu} \int_{-\infty}^{\zeta(\boldsymbol{w})} 1_{\tilde{\omega}_1}(z, \boldsymbol{w})\mathrm{d}z\right| = \frac{|\mu - \zeta(\boldsymbol{w})|}{\mu} 1_{\tilde{\omega}_1}(\zeta(\boldsymbol{w}), \boldsymbol{w}), \tag{A14}$$

for almost every $\boldsymbol{w} \in \mathbb{R}^{d-1}$ and every $\boldsymbol{x}_2$ in some neighborhood. Let the neighborhood be

$$U = \left\{ \begin{pmatrix} \mu \\ \nu \end{pmatrix} \in \Omega \, : \, \mu > \epsilon \right\} \setminus \bigcup_{k>2} \{\boldsymbol{x}_k\}, \tag{A15}$$

where $\epsilon > 0$ is a fixed positive number. Denoting R the diameter of Ω, we find

$$(\zeta(\boldsymbol{w}), \boldsymbol{w}) \in \tilde{\omega}_1 \Rightarrow |\zeta(\boldsymbol{w})| \leq R \wedge |\boldsymbol{w}| \leq R, \tag{A16}$$

and so we satisfy (A14) with

$$\mathrm{g}(\boldsymbol{w}) = \frac{2R}{\epsilon} 1_{D(R)}(\boldsymbol{w}), \tag{A17}$$

where $D(R)$ is the zero-centered unit disk (segment in two dimensions) in $\mathbb{R}^{d-1}$. ■

APPENDIX B. TAYLOR–GREEN VORTEX CONVERGENCE TABLES

Tables B1–B4 show the numerical errors in the Taylor–Green vortex benchmark (see Section 5.1).

TABLE B1 L^2 velocity error per resolution N.

	Structured mesh			Unstructured mesh		
N	**Re = 400**	**Re = 1000**	**Re = Inf**	**Re = 400**	**Re = 1000**	**Re = Inf**
16	8.37E−03	8.37E−03	8.45E−03	1.74E−02	1.84E−02	1.97E−02
32	2.79E−03	2.15E−03	1.84E−03	3.84E−03	7.98E−03	9.28E−03
48	1.24E−03	1.27E−03	8.47E−04	2.42E−03	2.36E−03	3.80E−03
72	5.06E−04	6.33E−04	4.19E−04	6.58E−04	7.29E−04	1.46E−03
108	2.36E−04	2.67E−04	2.39E−04	2.49E−03	1.67E−03	3.04E−03
162	1.41E−04	1.44E−04	1.46E−04	3.51E−04	2.26E−04	1.41E−04

TABLE B2 L^2 pressure error per resolution N.

	Structured mesh			Unstructured mesh		
N	**Re = 400**	**Re = 1000**	**Re = Inf**	**Re = 400**	**Re = 1000**	**Re = Inf**
16	3.54E−02	3.63E−02	3.66E−02	2.54E−02	2.54E−02	6.38E−02
32	1.33E−02	1.33E−02	1.53E−02	4.20E−03	4.20E−03	2.45E−02
48	7.92E−03	8.55E−03	9.78E−03	1.90E−03	1.90E−03	5.47E−03
72	5.20E−03	5.16E−03	6.37E−03	1.94E−03	1.94E−03	3.44E−03
108	3.47E−03	3.47E−03	4.18E−03	2.65E−03	2.65E−03	7.56E−03
162	2.42E−03	2.44E−03	2.77E−03	1.86E−03	1.86E−03	9.59E−04

TABLE B3 Total energy error per resolution N.

	Structured mesh			Unstructured mesh		
N	**Re = 400**	**Re = 1000**	**Re = Inf**	**Re = 400**	**Re = 1000**	**Re = Inf**
16	2.87E−03	2.87E−03	2.82E−03	6.89E−03	1.18E−02	1.34E−02
32	1.08E−03	1.02E−03	1.00E−03	4.52E−04	1.71E−02	1.20E−02
48	6.15E−04	6.05E−04	6.05E−04	6.89E−04	1.07E−03	2.65E−03
72	4.30E−04	3.78E−04	3.78E−04	1.05E−04	1.45E−04	1.23E−03
108	2.87E−04	2.65E−04	2.44E−04	2.73E−03	3.84E−03	4.37E−03
162	1.79E−04	1.79E−04	1.57E−04	8.64E−05	7.71E−05	3.32E−06

TABLE B4 L^2 divergence error per resolution N.

	Structured mesh			Unstructured mesh		
N	**Re = 400**	**Re = 1000**	**Re = Inf**	**Re = 400**	**Re = 1000**	**Re = Inf**
16	9.53E−05	8.82E−05	8.65E−05	3.29E−04	3.04E−04	2.98E−04
32	1.80E−05	1.01E−05	6.99E−06	2.61E−05	4.39E−05	1.05E−04
48	4.48E−06	3.95E−06	1.67E−06	2.93E−06	3.95E−06	1.25E−05
72	1.03E−06	1.23E−06	3.76E−07	1.01E−06	1.05E−06	1.84E−06
108	1.70E−07	2.23E−07	8.32E−08	2.93E−07	1.07E−06	2.82E−06
162	1.27E−08	3.22E−08	1.80E−08	6.53E−08	5.03E−08	5.65E−08