ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Semi-Lagrangian Finite-Element Exterior Calculus for Incompressible Flows

W. Tonnon and R. Hiptmair

Research Report No. 2023-07
January 2023

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Semi-Lagrangian Finite-Element Exterior Calculus for Incompressible Flows

Wouter Tonnon* Ralf Hiptmair†

January 12, 2023

1 Incompressible Navier-Stokes Equations

We consider the incompressible Navier-Stokes equations—a standard hydrodynamic model for the motion of an incompressible, potentially-viscous fluid—in a container with rigid walls, where we impose so-called "free boundary conditions" in the parlance of [31, p. 346] and [43, p. 502], see the latter article for further references. We search the fluid velocity field $\boldsymbol{u}(t, \boldsymbol{x})$ and the pressure $p(t, \boldsymbol{x})$ as functions of time t and space $\boldsymbol{x}$ on a bounded, Lipschitz domain $\Omega \subset \mathbb{R}^d$ such that they solve the evolution boundary-value problem

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \varepsilon \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f}, \qquad \text{on }]0, T[\times\Omega, \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \qquad \text{on }]0, T[\times\Omega, \tag{1b}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0, \qquad \text{on }]0, T[\times\partial\Omega, \tag{1c}$$

$$\varepsilon \boldsymbol{n} \times \nabla \times \boldsymbol{u} = \boldsymbol{0}, \qquad \text{on }]0, T[\times\partial\Omega, \tag{1d}$$

$$\boldsymbol{u} = \boldsymbol{u_0}, \qquad \text{on } \{0\} \times \Omega, \tag{1e}$$

where $\varepsilon \geq 0$ denotes a (non-dimensional) viscosity, $\boldsymbol{f}$ a given source term, $T > 0$ the final time, $\partial\Omega$ the boundary of Ω, and $\boldsymbol{n}(\boldsymbol{x})$ the outward normal vector at $\boldsymbol{x} \in \partial\Omega$. The initial condition $\boldsymbol{u_0}$ is to satisfy $\nabla \cdot \boldsymbol{u_0} = 0$ in Ω and $\boldsymbol{u_0} \cdot \boldsymbol{n} = 0$, $\varepsilon \boldsymbol{n} \times \nabla \times \boldsymbol{u_0} = \boldsymbol{0}$ on $\partial\Omega$. Based on the variational description of the Navier-Stokes equations as described in [5], $\boldsymbol{u}$ can be interpreted as a differential 1-form [33] and we can recast system (1) in the following way. Let $\Lambda^k(\Omega)$ for $k \in \mathbb{N}$ denote the space of differential k-forms on

*SAM, ETH Zürich, CH-8092 Zürich, wouter.tonnon@sam.math.ethz.ch
†SAM, ETH Zürich, CH-8092 Zürich, ralf.hiptmair@sam.math.ethz.ch

Ω. Then we search $\omega \in \Lambda^1(\Omega)$ and $p \in \Lambda^0(\Omega)$ such that

$$D_{\boldsymbol{u}}\omega + \varepsilon\delta\mathrm{d}\omega + \mathrm{d}p = f, \qquad \text{on }]0,T[\times\Omega, \tag{2a}$$
$$\delta\omega = 0, \qquad \text{on }]0,T[\times\Omega, \tag{2b}$$
$$\operatorname{tr}\star\omega = 0, \qquad \text{on }]0,T[\times\partial\Omega, \tag{2c}$$
$$\varepsilon \operatorname{tr}\star\mathrm{d}\omega = 0, \qquad \text{on }]0,T[\times\partial\Omega, \tag{2d}$$
$$\omega = \omega_0, \qquad \text{on} \{0\}\times\Omega, \tag{2e}$$

where $D_{\boldsymbol{u}}\omega$ denotes the material derivative of ω with respect to $\boldsymbol{u}$, $\mathrm{d} : \Lambda^{k-1}(\Omega) \mapsto \Lambda^k(\Omega)$ the exterior derivative, $\delta : \Lambda^k(\Omega) \mapsto \Lambda^{k-1}(\Omega)$ the exterior coderivative, and the trace tr is the pullback under the embedding $\partial\Omega \subset \bar{\Omega}$. Here, $\boldsymbol{u}$ is related to ω through $\omega \coloneqq \boldsymbol{u}^\flat$, i.e. $\boldsymbol{u}$ is the vector proxy of ω w.r.t. the Euclidean metric. Similarly, we have that $\Lambda^1(\Omega) \ni \omega_0 \coloneqq \boldsymbol{u_0}^\flat$ and $\Lambda^1(\Omega) \ni f \coloneqq \boldsymbol{f}^\flat$. Note that (2) can be derived through classical vector calculus for vector proxies as shown in Appendix A for $d = 3$.

As shown in [18, 19], sufficiently-smooth solutions $\omega :]0,T[\mapsto \Lambda^1(\Omega)$ of the incompressible Navier-Stokes equations as given in system (2) satisfy an energy relation, that is,

$$\frac{dE}{dt}(t) \coloneqq \frac{d}{dt}\frac{1}{2}\int_\Omega \omega(t)\wedge\star\omega(t) = -\varepsilon\int_\Omega \mathrm{d}\omega(t)\wedge\star\mathrm{d}\omega(t). \tag{3}$$

This relation implies energy conservation for $\varepsilon = 0$. In the case of $\varepsilon = 0$, we also have helicity conservation, that is,

$$\varepsilon = 0 \implies \frac{dH}{dt}(t) \coloneqq \frac{d}{dt}\int_\Omega \mathrm{d}\omega(t)\wedge\omega(t) = 0. \tag{4}$$

Note that the Onsager conjecture tells us that in the case $\varepsilon = 0$ the solutions need to be at least Hölder regular with exponent $\frac{1}{3}$ for energy conservation to hold [28].

Remark 1 *We acknowledge that the boundary condition* (1d) *is non-standard. This boundary condition was chosen because it is the natural boundary condition associated to system* (2)*. To enforce the standard no-slip boundary conditions,* (1d) *could be replaced by* $\varepsilon\boldsymbol{u}\times\boldsymbol{n} = \boldsymbol{u}$ *on* $]0,T[\times\partial\Omega$*, that is, we impose an essential instead of natural boundary condition to system* (2)*. Unfortunately, in this case, the scheme presented in this work leads to an ill-posed system. In the case* $\varepsilon = 0$*, the only imposed boundary condition* (1c) *is standard.*

Remark 2 *Boundary conditions* (1c),(1d) *can be interpreted as slip boundary conditions. However, on smooth domains* Ω*, they are only equivalent to Navier's slip boundary conditions if the Weingarten map related to* $\partial\Omega$ *vanishes [31, section 2].*

2 Outline and Related Work

We propose a semi-Lagrangian approach to the discretization of the reformulated Navier-Stokes boundary value problem (2). This method revolves around the discretization of the material derivative $D_{\boldsymbol{u}}\omega$ in the framework of a finite-element Galerkin

discretization on a fixed spatial mesh. The main idea is to approximate $D_{\boldsymbol{u}}\omega$ by backward difference quotients involving transported snapshots of the 1-form ω, which can be computed via the pullback induced by the flow of the velocity vector field $\boldsymbol{u}$.

Semi-Lagrangian methods for transient transport equations like (2) are well-established for the linear case when $\boldsymbol{u}$ is a given Lipschitz-continuous velocity field. In particular, for ω a 0-form, that is, a plain scalar-valued function, plenty of semi-Lagrangian approaches have been proposed and investigated, see, for instance, [8, 7, 20, 21, 23, 35, 39, 40, 6, 12, 45]. We refer to [24, Chapter 5] for a comprehensive pre-2013 literature review on the analysis of general semi-Lagrangian schemes. Most of these methods focus on mapping point values under the flow, with the exception of a particularly interesting class of semi-Lagrangian methods known as Lagrange-Galerkin methods. Lagrange-Galerkin methods do not transport point values, but rather triangles (in 2D) or tetrahedra (in 3D). Refer to [10] for a review of those methods.

Meanwhile semi-Lagrangian methods for transport problems for differential forms of any order have been developed [26, 25, 24]. The next section will review these semi-Lagrangian methods for linear transport problems with emphasis on 1-forms. We will also introduce a new scheme which is second-order in space and time based on so-called "small edges", see section 3.1.2 for details.

Semi-Lagrangian schemes for the incompressible Navier-Stokes equations are also well-documented in literature, with emphasis on the Lagrange-Galerkin method [10, 14, 15, 30, 1, 11, 9]. A survey of the application of Lagrange-Galerkin methods to the incompressible Navier-Stokes equations is given in [10]. It is important to note that these methods require the evaluation of integrals of transported quantities and, in case these integrals cannot be computed exactly, instabilities can occur [12, 32]. A possible remedy is to add an additional stabilization term that includes artificial diffusion [10]. Other semi-Lagrangian methods for incompressible Navier-Stokes equations directly transport point values with the nodes of a mesh instead of evaluating integrals of transported quantities, see [34, 29, 47, 46, 13] and [16], where the last work makes use of exponential integrators [17]. Most authors employ spectral elements for the discretization in space [34, 29], but any type of finite-element space with degrees-of-freedom relying on point evaluations can be used. The methods proposed in [47, 46] are also based on finite-element spaces with degrees-of-freedom on nodes, but employ backward-difference approximations for the material derivative. The work [13] proposes an explicit semi-Lagrangian method still built around the transport of point values in the nodes of the mesh. The diffusion term is also taken into account in a semi-Lagrangian fashion and the incompressibility constraint is enforced by means of a Chorin projection. Also [13] proposes an explicit semi-Lagrangian scheme using the same principles, but based on the vorticity-streamfunction form of the incompressible Navier-Stokes equations.

All the mentioned semi-Lagrangian schemes rely on the transport of point values of continuous vector fields, which is the perspective embraced in formulation (1). However, we believe that, in particular in the case of free boundary conditions (1c)

and (1d), the semi-Lagrangian method based on (2) offers benefits similar to the benefits bestowed by the use of discrete differential forms (finite-element exterior calculus, FEEC [3, 4]) for the discretization of electromagnetic fields. Section 4 will convey that the boundary conditions (2c), (2d), and the incompressiblity constraint can very naturally be incorporated into a variational formulation of (2) posed in spaces of 1-forms. This has been the main motivation for pursuing the *new idea of a semi-Lagrangian method for* (2) *that employs discrete 1-forms*. Another motivation has been the expected excellent robustness of the semi-Lagrangian discretization in the limit $\varepsilon \mapsto 0$. Numerical tests reported in section 5 will confirm this.

Two more aspects of our method are worth noting: Firstly, a discrete 1-form ω_h will not immediately spawn a continuous velocity field $\boldsymbol{u}_h = \omega_h^\flat$, However, continuity is essential for defining a meaningful flow. We need an additional averaging step, which we present in section 4.1. Secondly, since semi-Lagrangian methods fail to respect the decay/conservation laws (3) and (4) exactly, we present a way how to enforce them in section 4.3.

3 Semi-Lagrangian Advection of differential forms

3.1 Discrete differential forms

We start from a simplicial triangulation $\mathcal{K}_h(\Omega)$ of Ω, which may rely on a piecewise linear approximation of $\partial\Omega$ so that it covers a slightly perturbed domain.

3.1.1 Lowest-order case: Whitney forms

For $\Lambda^0(\Omega)$—the space of 0-forms on Ω, which is just a space of real-valued functions—the usual (Lagrange) finite-element space of continuous, piecewise-linear, polynomial functions provides the space $\Lambda^0_{h,1}(\Omega)$ of lowest-order discrete 0-forms.

Let $d \in \{2, 3\}$, K a d-simplex with edges $\{e_1, .., e_{3(d-1)}\}$. To construct lowest-order discrete 1-forms on K, we associate to every edge e_i a local shape function w^{e_i}. Let the edge e_i be directed from vertex v_i^1 to v_i^2, then the local shape function $w^{e_i} \in \Lambda^1(K)$ associated with edge e_i is

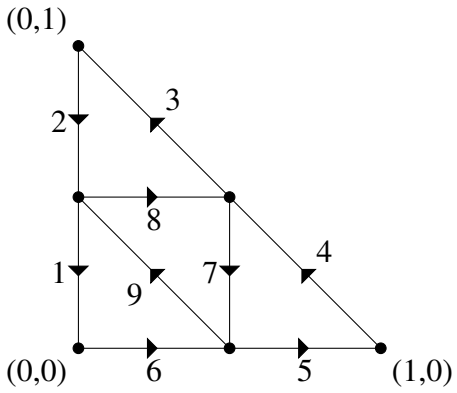
$$w^{e_i} \coloneqq \lambda_{\boldsymbol{v}_i^1} \mathrm{d}\lambda_{\boldsymbol{v}_i^2} - \lambda_{\boldsymbol{v}_i^2} \mathrm{d}\lambda_{\boldsymbol{v}_i^1}, \tag{5}$$

where $\lambda_{\boldsymbol{v}}$ represents the barycentric coordinate associated with vertex $\boldsymbol{v}$. We define the lowest-order, local space of discrete 1-forms

$$\Lambda^1_{h,1}(K) \coloneqq \operatorname{span}\{w^e; e \text{ an edge of } K\}. \tag{6}$$

Using these local spaces, we can define the global space of lowest-order, discrete 1-forms

$$\Lambda^1_{h,1}(\Omega) \coloneqq \{\omega \in \Lambda^1(\Omega); \forall K \in \mathcal{K}_h(\Omega) : \omega|_K \in \Lambda^1_{h,1}(K)\}, \tag{7}$$



(a) 9 small edges of a second-order element in 2D. All the edges between the different connection points are small edges. In 3D, we simply have all these small edges on the faces of the simplex.

edge no.	l.s.f.	edge no.	l.s.f.
1	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} x(x+y-1) \\ -(x-1)(x+y-1) \end{smallmatrix}\right]$	6	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} (y-1)(x+y-1) \\ x(1-x-y) \end{smallmatrix}\right]$
2	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} -y^2 \\ y(x-1) \end{smallmatrix}\right]$	7	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} -xy \\ x(x-1) \end{smallmatrix}\right]$
3	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} -y^2 \\ xy \end{smallmatrix}\right]$	8	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} y(1-y) \\ xy \end{smallmatrix}\right]$
4	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} -xy \\ x^2 \end{smallmatrix}\right]$	9	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} y(x+y-1) \\ x(1-x-y) \end{smallmatrix}\right]$
5	$\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \mapsto \left[\begin{smallmatrix} x(1-y) \\ x^2 \end{smallmatrix}\right]$		

(b) Local shape functions (l.s.f.) for the unit triangle associated with second-order, discrete differential forms in 2D as proposed in [38]. Each shape function corresponds to the small edge in (a) with the same numbering.

Figure 1: Illustration of small edges (a) and corresponding local shape functions (b) for the unit triangle.

where $\Lambda^1(\Omega)$ again denotes the space of differential 1-forms on Ω. We demand that for every $\omega \in \Lambda^1(\Omega)$ integration along any smooth oriented path yields a unique value. Thus, the requirement $\omega \in \Lambda^1(\Omega)$ imposes tangential continuity on the vector proxy of ω.

3.1.2 Second-order discrete forms

Similar to the lowest-order case, the space $\Lambda^0_{h,2}(\Omega)$ of second-order discrete 0-forms is spawned by the usual (Lagrange) finite-element space of continuous, piecewise-quadratic, polynomial functions.

Let $d \in \{2, 3\}$, K a d-simplex with edges $\{e_1, .., e_{3(d-1)}\}$ and vertices $\{\boldsymbol{v}_1, .., \boldsymbol{v}_{d+1}\}$. To construct second-order discrete 1-forms, we associate local shape functions to "small edges". We can construct $3(d + 1)(d - 1)$ small edges [38, Definition 3.2] by defining $\forall i \in \{1, .., d + 1\}$ and $\forall j \in \{1, .., 3(d - 1)\}$

$$\{\boldsymbol{v}_i, e_j\} \coloneqq \{\boldsymbol{v}_i + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{v}_i); \boldsymbol{x} \in e_j\},$$

where $\{\boldsymbol{v}_i, e_j\}$ denotes the small edge. In Figure 1a we illustrate the 9 small edges of a 2-simplex. For example, we see that small edge 9 can be written as $\{(0, 0), [(1, 0), (0, 1)]\}$. To make the difference between small edges and edges of the mesh explicit, we will sometimes refer to the latter as "big edges".

The local shape function [38, Definition 3.3] associated with $\{\boldsymbol{v}_i, e_j\}$ is given by

$$w^{\{\boldsymbol{v}_i, e_j\}} \coloneqq \lambda_{\boldsymbol{v}_i} w^{e_j},$$

where w^{e_j} denotes the Whitney 1-form associated with the big edge e_j as defined in (5). In Figure 1b we give explicit expressions for the shape functions associated with the small edges in Figure 1a. Note that the local shape functions of the form $w^{\{\boldsymbol{v}, e\}}$ associated with small edges in the interior ($d = 2$) or on the same face ($d = 3$) of the form $\{\boldsymbol{v}, e\}$ such that $\boldsymbol{v} \notin \partial e$ (example: small edge 7, 8, and 9 in Figure 1a) are *linearly dependent*. We define the second-order, local space of discrete 1-forms [38, Definition 3.3]

$$\Lambda^1_{h,2}(K) \coloneqq \operatorname{span}\{w^{\{\boldsymbol{v}, e\}}; \boldsymbol{v} \text{ a vertex of } K, e \text{ a (big) edge of } K\}. \tag{8}$$

Using these local spaces, we can define the global space of second-order, discrete 1-forms

$$\Lambda^1_{h,2}(\Omega) \coloneqq \{\omega \in \Lambda^1(\Omega); \forall K \in \mathcal{K}_h(\Omega) : \omega|_K \in \Lambda^1_{h,2}(K)\}, \tag{9}$$

where again we have tangential continuity by a similar argument as in section 3.1.1.

3.1.3 Projection operators

We denote by $\mathcal{E}_{h,p}(\Omega)$ the global set of big edges ($p = 1$) or small edges ($p = 2$) associated with $\mathcal{K}_h(\Omega)$. We will define the projection operator $\mathcal{I}_{h,p} : \Lambda^1(\Omega) \mapsto \Lambda^1_{h,p}(\Omega)$

as the unique operator that maps $\omega \in \Lambda^1(\Omega)$ to $\omega_h \in \Lambda^1_{h,p}(\Omega)$ such that the mismatch

$$\sum_{e \in \mathcal{E}_{h,p}(\Omega)} \left(\int_e \omega - \int_e \omega_h \right)^2 \tag{10}$$

is minimized. Note that for $p = 1$, this mismatch can be made to vanish. In this case, $\mathcal{I}_{h,1}$ agrees with the usual edge-based nodal projection operator [27, Eq. (3.11)].

In practice, we can compute the projection locally as follows. Let $K \in \mathcal{K}_h(\Omega)$ be a d-simplex, $d \in \{2, 3\}$, and let $\{s_1, .., s_{N_{p,d}}\}$ and $\{w^{s_1}, .., w^{s_{N_{p,d}}}\}$ denote the corresponding big ($p = 1$) or small ($p = 2$) edges and corresponding shape functions as introduced above. Specifically, we have $N_{1,2} = 3$, $N_{1,3} = 6$, $N_{2,2} = 9$, and $N_{2,3} = 24$. We can define the matrix

$$(\mathcal{M})_{i,j} = \int_{s_i} w^{s_j}, \qquad 1 \leq i, j \leq N_{p,d}. \tag{11}$$

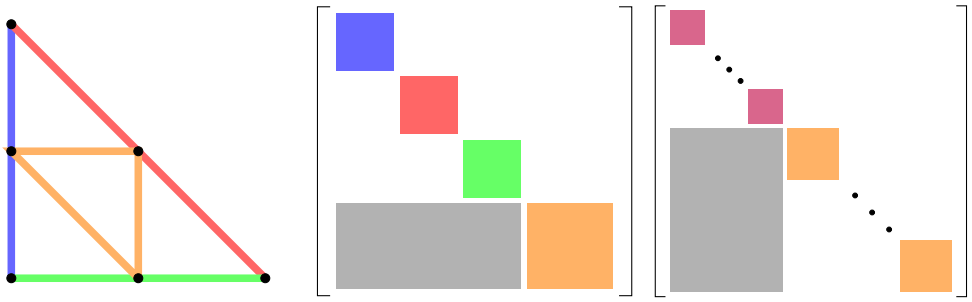
We will say that there is an interaction from edge s_j to s_i if $(\mathcal{M})_{i,j} \neq 0$. Note that for $p = 1$, $\mathcal{M}$ is the identity matrix. For $p = 2$ the local shape functions are *linearly dependent* and, thus, the above matrix is not invertible. However, we can decompose $\mathcal{M}$ into invertible and singular sub-matrices. For illustrative purposes we display for $p = 2$ and $d = 2$ the decomposition of $\mathcal{M}$ in Figure 2b. The three top-left sub-matrices in Figure 2b are invertible 2×2 matrices that describe the interaction between the two small edges that lie on the same big edge, that is, the blue, red, and green sub-matrix in Figure 2b correspond to the blue, red, and green small edges in Figure 2a, respectively. The orange sub-matrix in Figure 2b is a 3×3 matrix with rank 2 that describes the interaction between the three small edges that lie in the interior of the simplex in Figure 2a, that is, the orange small edges. The gray sub-matrix encodes the one-directional interaction from the the small edges that lie on a big edge to the small edges in the interior. Note that the decomposition of $\mathcal{M}$ as given in Figure 2b is not limited to $d = 2$. The idea can be extended to $d = 3$ by considering each face of a 3-simplex as a 2-simplex. This is sufficient, since for $d = 3$ we have no small edges in the interior and there is no interaction between small edges that do not lie on the same face. We give the general structure of $\mathcal{M}$ in Figure 2c. Note that the small, purple sub-matrices represent invertible 2×2 matrices and the bigger, orange sub-matrices represent 3×3 matrices with rank 2.

In order to find $\omega_h\big|_K \in \Lambda^1_{h,p}(K)$ such that $\omega_h\big|_K = \mathcal{I}_{h,p}\omega\big|_K$, let $\vec{\eta}_K$ be a vector of coefficients $\eta^1_K, .., \eta^{N_{p,d}}_K$ such that

$$\omega_h|_K = \sum_{i=1}^{N_{p,d}} \eta^i_K w^{s_i}. \tag{12}$$

We can then compute $\vec{\eta}_K$ as a least-squares solution of

$$\mathcal{M}\vec{\eta}_K = \left(\int_{s_i} \omega \right)_{1 \leq i \leq N_{p,d}}. \tag{13}$$



(a) 2-simplex K (b) matrix $\mathcal{M}$ ($d = 2$) (c) matrix $\mathcal{M}$ ($d = 3$)

Figure 2: For $p = 2$ and $d = 2$ the matrix $\mathcal{M}$ corresponding to the 2-simplex K in (a) has the form given in (b). Each row and column in $\mathcal{M}$ is associated to a small edge in (a). Each sub-matrix in (b) describes the interactions between edges with the same color in (a). The gray sub-matrix is an exception as it describes the one-directional interaction between the small edges that lie on a big edge and the small edges that lie in the interior. For $d = 3$, we $\mathcal{M}$ has the structure as shown in Figure 2c, where the purple sub-matrixs are 2×2 invertible matrices and the orange sub-matrixs are 3×3 matrices of rank 2.

Without loss of generality we assume that $\mathcal{M}$ has the form as given in Figure 2c. Then, we solve (13) as follows:

1. The local shape functions related to small edges that lie on a big edge of the simplex are linearly independent. We solve for their coefficients first, that is, we solve the system corresponding to the invertible blue sub-matrices in Figure 2c first.

2. Using the results from step 1, we can move the gray sub-matrix in Figure 2c to the right-hand side. Then, we solve the matrix-system corresponding to the orange sub-matrices in Figure 2c in a least-squares sense.

If we perform the above steps for all $K \in \mathcal{K}_h(\Omega)$, we find $\omega_h = \mathcal{I}_{h,p}\omega \in \Lambda^1_{h,p}(\Omega)$. Note that only the shape functions associated to small edges on a face contribute to the tangential fields on that face. Therefore, the above procedure yields tangential continuity.

Remark 3 *For $p = 1$, (13) reduces to*

$$\eta^i_K = \int_{s_i} \omega, \qquad \forall i \in \{1, .., 3(d-1)\} \tag{14}$$

with s_i a big edge of the 3-simplex K for all $i \in \{1, .., 3(d-1)\}$. This yields the standard nodal interpolation operator of [27, Eq. (3.11)].

3.2 Semi-Lagrangian material derivative

The method described in this section is largely based on [24, 26]. Throughout this section, unless stated otherwise, we fix the stationary, Lipschitz-continuous velocity field $\boldsymbol{u} \in W^{1,\infty}(\Omega)$ with $\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{0}$ on $\partial\Omega$. This means that we consider a linear transport problem and our main concern will be the discretization of the material derivative $D_{\boldsymbol{u}}\omega$ for a 1-form ω. We can define the flow $]0, T[\times\Omega \ni (\tau, \boldsymbol{x}) \mapsto X_\tau(\boldsymbol{x}) \in \mathbb{R}^d$ as the solution of the initial value problems

$$\frac{\partial}{\partial\tau} X_{t,t+\tau}(\boldsymbol{x}) = \boldsymbol{u}(X_{t,t+\tau}(\boldsymbol{x})), \qquad X_t(\boldsymbol{x}) = \boldsymbol{x}. \tag{15}$$

Given that flow we can define the material derivative for a time-dependent differential 1-form ω

$$D_{\boldsymbol{u}}\omega(t) \coloneqq \frac{\partial}{\partial\tau} X^*_{t,t+\tau}\omega(t+\tau)\bigg|_{\tau=0}. \tag{16}$$

We employ a first- or second-order, backward-difference method to approximate the derivative. Writing $X^*_{t,t-\tau}$ for the pullback of forms under the flow, we obtain for sufficiently-smooth $t \mapsto \omega(t)$ and a timestep $0 < \tau \to 0$

$$D_{\boldsymbol{u}}\omega(t) = \frac{1}{\tau}\left[\omega(t) - X^*_{t,t-\tau}\omega(t-\tau)\right] + \mathcal{O}(\tau^2) \tag{17}$$

or

$$D_{\boldsymbol{u}}\omega(t) = \frac{1}{2\tau}\left[3\omega(t) - 4X^*_{t,t-\tau}\omega(t-\tau) + X^*_{t,t-2\tau}\omega(t-2\tau)\right] + \mathcal{O}(\tau^3), \tag{18}$$

respectively. Note that both backward-difference methods are A-stable [41]. In the remainder of this section we restrict ourselves to (17), but exactly the same considerations apply to (18).

Given a temporal mesh $.. < t^n < t^{n+1} < ..$, we approximate $\omega(t^n, \cdot) \in \Lambda^1(\Omega)$ by a discrete differential form $\omega_h^n \in \Lambda^1_{h,p}(\Omega)$ with $p \in \{1, 2\}$. Using the backward-difference quotient (17), we can define the discrete material derivative for fixed timestep $\tau > 0$

$$(D_{\boldsymbol{\beta}}\omega)(t^n) \approx \frac{1}{\tau}\left[\omega_h^n - \mathcal{I}_{h,p} X^*_{t,t-\tau}\omega_h^{n-1}\right] \in \Lambda^1_{h,p}(\Omega), \tag{19}$$

where we need to use the projection operator $\mathcal{I}_{h,p} : \Lambda^1(\Omega) \mapsto \Lambda_{h,p}(\Omega)$ since $X^*_{t,t-\tau}\omega_h^{n-1} \notin \Lambda^1_{h,p}(\Omega)$ in general. Recall from section 3.1 that the degrees of freedom for discrete 1-forms are associated to small ($p = 2$) or big ($p = 1$) edges. As discussed in section 3.1.3, evaluating the interpolation operator entails integrating $X^*_{t,t-\tau}\omega_h^{n-1}$ over small ($p = 2$) or big ($p = 1$) edges. We can approximate these integrals as follows

$$\int_e X^*_{t,t-\tau}\omega_h^{n-1} = \int_{X_{t,t-\tau}(e)} \omega_h^{n-1} \approx \int_{\bar{X}_{t,t-\tau}(e)} \omega_h^{n-1}, \tag{20}$$

where e is a small or big edge and

$$\bar{X}_{t,t-\tau}(e) = \left\{(1-\xi)X_{t,t-\tau}(\boldsymbol{v}^1) + \xi X_{t,t-\tau}(\boldsymbol{v}^2); 0 \le \xi \le 1\right\} \tag{21}$$

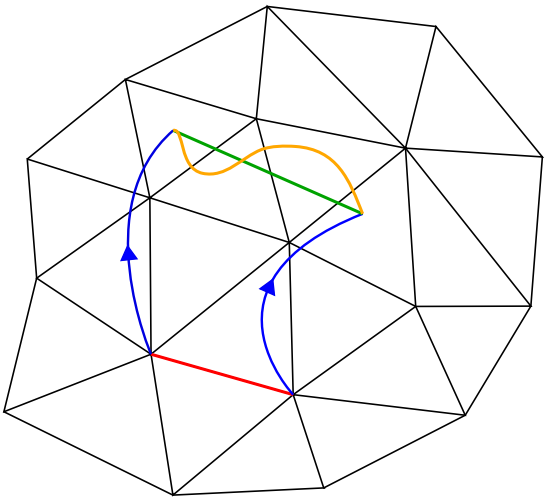


Figure 3: Edge e (in red) is transported using the flow $\boldsymbol{\beta}$ (in blue). The exact transported edge $X_\tau(e)$ and the approximate transported edge $\bar{X}_\tau(e)$ are given in orange and green.

with $\boldsymbol{v}^1, \boldsymbol{v}^2$ the vertices of e. Instead of transporting the edge e using the exact flow $X_{t,t-\tau}$, we follow [12, 24, 26] and only transport the vertices of the small edges ($p = 2$) or big edges ($p = 1$) and obtain a piecewise linear (second-order) approximation $\bar{X}_{t,t-\tau}(e)$ of the transported edge $X_{t,t-\tau}(e)$ as illustrated in Figure 3. We can approximate the movement of the endpoints of e under the flow as defined by (15) by solving (15) using the explicit Euler method or Heun's method for the first- and second-order case, respectively. We will elaborate on this further in section 4.1.

In Figure 3, we can also see that the approximate transported edge may intersect several different elements of the mesh. When we evaluate the integral in (20), it can happen that there are discontinuities of ω_h^{n-1} along $\bar{X}_{t,t-\tau}(e)$. Therefore, we cannot apply a global quadrature rule to the entire integral. Instead, we split $\bar{X}_{t,t-\tau}(e)$ into several segments defined by the intersection of $\bar{X}_{t,t-\tau}(e)$ with cells of the mesh. In our implementation, for the sake of stability, we find the intersection points by transforming back to a reference element as visualised in Figure 4. Algorithm 1 gives all details. Note that we can forgo the treament of any special cases (e.g. intersection with vertices) without jeopardizing stability. After we split the transported edge into segments, we can evaluate the integrals over these individual pieces exactly, because we know that ω_h^{n-1} is of polynomial form when restricted to individual elements of the mesh (see section 3.1).

When simulating the fluid model (2), we will not have access to an exact velocity field. Instead we only have access to an approximation of the velocity field. This approximation may not satisfy exact vanishing normal boundary conditions. Therefore, a part of $\bar{X}_{t,t-\tau}(e)$ may end up outside the domain. This can also happen due to an approximation of the flow by explicit timestepping. Since ω_h^{n-1} is not defined outside

Algorithm 1 Splitting 1-simplex over mesh elements (see Figure 4 for illustration). Here, K_{ref} denotes the reference simplex.

Input: $x_0 \in K_0 \in \mathcal{K}_h(\Omega)$ and x_1 vertices of a 1-simplex e.
Output: Number of elements N, elements $\{K_0, .., K_{N-1}\} \in \mathcal{K}_h(\Omega)^N$.

$K \leftarrow K_0$
$F_{\text{old}} \leftarrow$ NULL
$K_{\text{old}} \leftarrow$ NULL
$N \leftarrow 1$
$E \leftarrow \{K\}$
while $x_1 \notin K$ **do**
    Find the isoparametric mapping $\varphi_K : K_{\text{ref}} \mapsto K$
    Find face $F \subset \partial K$ s.t. $F \neq F_{\text{old}}$ and $\varphi_K^{-1}(e) \cap \varphi_K^{-1}(F) \neq \emptyset$
    $K \leftarrow K \in \mathcal{K}_h(\Omega)$ s.t. $F \subset \partial K$ and $K \neq K_{\text{old}}$ (K on the other side of face F)
    $F_{\text{old}} \leftarrow F$
    $N \leftarrow N + 1$
    $E \leftarrow E \cup \{K\}$
end while

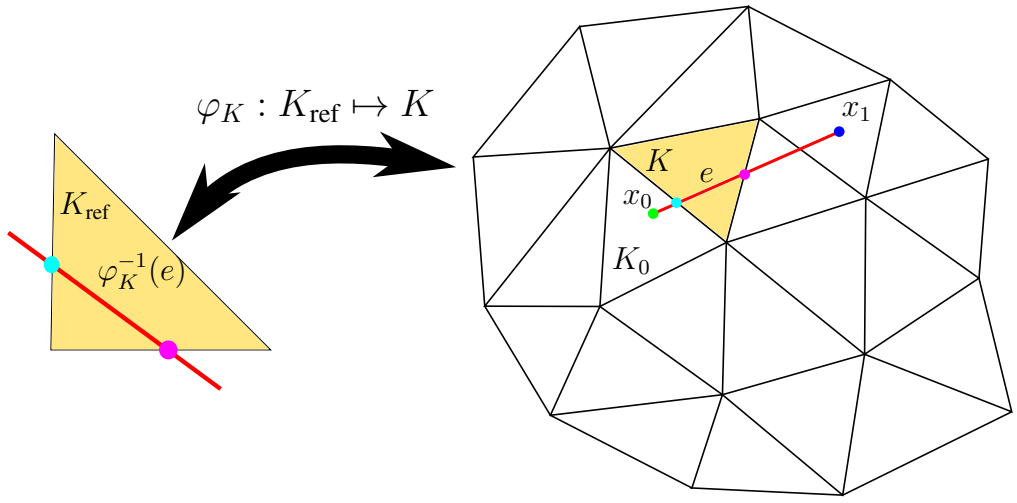


Figure 4: The red line indicates the line that spans multiple elements. On the left we see the reference triangle associated with the yellow element in the mesh on the right.

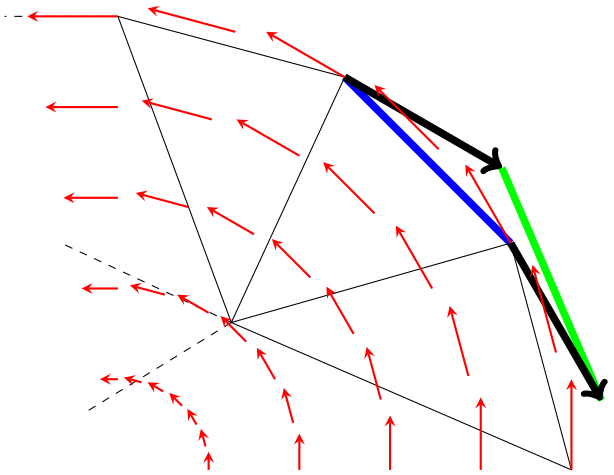


Figure 5: A coarse triangulation of $\Omega = \{\boldsymbol{x} \in \mathbb{R}^2; ||x|| < 1\}$ with the velocity field $\boldsymbol{u} = [-y, x]^T$ satisfying $\boldsymbol{u} \cdot \boldsymbol{n} = 0$. Despite the vanishing normal components of the velocity, the blue edge gets transported out of the domain to the green edge.

the domain, we set

$$\int_{\bar{X}_{t,t-\tau}(e)\big|_{\mathbb{R}^d \setminus \Omega}} \omega_h^{n-1} := \frac{\operatorname{len}\left(\bar{X}_{t,t-\tau}(e)\big|_{\mathbb{R}^d \setminus \Omega} \right)}{\operatorname{len}\left(\bar{X}_{t,t-\tau}(e) \right)} \int_e \omega_h^{n-1}, \tag{22}$$

where $\bar{X}_{t,t-\tau}(e)\big|_{\mathbb{R}^d \setminus \Omega}$ is the part of $\bar{X}_{t,t-\tau}(e)$ that lies outside the domain Ω and $\operatorname{len}(\cdot)$ gives the arclength of the argument. This is motivated by the situation displayed in Figure 5—a case where an edge gets transported out of the domain due to the use of approximate flow maps despite vanishing normal components of the velocity. If we set the value defined in (22) to zero in this case, it would be equivalent to applying vanishing tangential boundary conditions, which is inconsistent with (1). Instead, (22) just takes the tangential components from the previous timestep.

We arrive at the following approximation of the material derivative

$$(D_{\boldsymbol{\beta}}\omega)(t^n) \approx \frac{1}{\tau} \left[\omega_h^n - \mathcal{I}_{h,p} \bar{X}^*_{t,t-\tau} \omega_h^{n-1} \right], \tag{23}$$

where the only difference between (19) and (23) is that $X_{t,t-\tau}$ was replaced by $\bar{X}_{t,t-\tau}$ and $\mathcal{I}_{h,p}$ is implemented based on (22). Note that in our scheme $\bar{X}^*_{t,t-\tau}$ is always evaluated in conjunction with $\mathcal{I}_{h,p}$, which means that we need define $\bar{X}_{t,t-\tau}$ only on small ($p = 2$) or big ($p = 1$) edges. In fact, $\bar{X}_{t,t-\tau}$ is defined through (21) for all points that lie on small ($p = 2$) or big ($p = 1$) edges.

Given a velocity field $\boldsymbol{u} \in W^{1,\infty}(\Omega)$ with $\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{0}$, it was shown in [26, section 4] that using a first-order backward difference scheme and lowest-order elements for the spatial discretization, we can approximate a smooth solution $\omega \in \Lambda^1(\Omega)$ of

$$D_{\boldsymbol{u}}\omega = 0, \tag{24}$$

with an L^2-error of $\mathcal{O}(\tau^{-\frac{1}{2}}h)$, where h is the spatial meshwidth and $\tau > 0$ is the timestep size. However, numerical experiments [26, section 6] performed with $\tau = \mathcal{O}(h)$ show an error of $\mathcal{O}(h)$—a slight improvement over the a-priori estimates. This motivates us to link the timestep to the mesh width as $\tau = \mathcal{O}(h)$.

4 Semi-Lagrangian Advection applied to the Incompressible Navier-Stokes Equations

Given a temporal mesh $t_0 < t_1 < ... < t_{N-1} < t_N$, we elaborate a single timestep $t_{n-1} \mapsto t_n$ of size $\tau \coloneqq t_n - t_{n-1}$, $n \leq N$. We assume that approximations $\omega_h^k \in \Lambda^1_{h,p}(\Omega)$ of $\omega(t_k, \cdot) \in \Lambda^1(\Omega)$ are available for $k < n$ with $[0, T] \mapsto \omega \in \Lambda^1(\Omega)$ a solution of (2).

4.1 Approximation of the flow map

In the Navier-Stokes equations, the flow is induced by the unknown, time-dependent velocity field $\boldsymbol{u}(t, \boldsymbol{x})$. Therefore, (15) becomes

$$\frac{\partial}{\partial \tau} X_{t,t+\tau}(\boldsymbol{x}) = \boldsymbol{u}(t + \tau, X_{t,t+\tau}(\boldsymbol{x})), \qquad X_t(\boldsymbol{x}) = \boldsymbol{x}\ , \quad t \in (0, T). \tag{25}$$

The discretization of the material derivative requires us to approximate the flow map $X_{t,t-\tau}$ in order to evaluate (21).

4.1.1 A first-order scheme

We use the explicit Euler method to approximate the (backward) flow according to

$$X_{t_n,t_n-\tau}(\boldsymbol{x}) \approx \boldsymbol{x} - \tau \boldsymbol{u}(t_n - \tau, \boldsymbol{x}), \tag{26}$$

where t_n is a node in the temporal mesh and τ denotes the timestep size. We only have access to an approximation $\boldsymbol{u}_h^{n-1} \coloneqq (\omega_h^{n-1})^\sharp$ of $\boldsymbol{u}$ at time t_{n-1}, which gives

$$X_{t_n,t_n-\tau}(\boldsymbol{x}) \approx \boldsymbol{x} - \tau \boldsymbol{u}_h^{n-1}(\boldsymbol{x}). \tag{27}$$

Note that a direct application of the explicit Euler method would require an evaluation of the velocity field at t_n. Instead, we perform a constant extrapolation and evaluate the velocity field at t_{n-1}, that is, we use $\boldsymbol{u}_h^{n-1}$ in (27).

The approximation $\boldsymbol{u}_h^{n-1}$ resides in the space of vector proxies for discrete differential 1-forms as discussed in section 3.1. This means that only tangential continuity of $\boldsymbol{u}_h^{n-1}$ across faces of elements of the mesh is guaranteed, while discontinuities may appear in the normal direction of the faces. Therefore, $\boldsymbol{u}_h^{n-1}$ is not defined point-wise—even though (27) requires point-wise evaluation. For that reason, we will

replace $\boldsymbol{u}_h^{n-1}$ by a globally-continuous, smoothened velocity field $\bar{\boldsymbol{u}}_h^{n-1}$ that approximates $\boldsymbol{u}_h^{n-1}$ (see section 4.1.3 for the construction). We then have

$$X_{t_n,t_n-\tau}(\boldsymbol{x}) \approx \boldsymbol{x} - \tau \bar{\boldsymbol{u}}_h^{n-1}(\boldsymbol{x}) \tag{28}$$

which yields a first-order-in-time approximation of $X_{t_n,t_n-\tau}(\boldsymbol{x})$, provided that $\bar{\boldsymbol{u}}_h^{n-1}$ is a first-order approximation of $\boldsymbol{u}(t_{n-1},\cdot)$.

4.1.2 A second-order scheme

A second-order approximation can be achieved by using Heun's method [44] instead of explicit Euler. We find the following second-order in time approximations

$$X_{t_n,t_n-\tau}(\boldsymbol{x}) \approx \boldsymbol{x} - \frac{\tau}{2}\left[\boldsymbol{u}_h^*(\boldsymbol{x}) + \boldsymbol{u}_h^{n-1}(\boldsymbol{x} - \tau \boldsymbol{u}_h^*(\boldsymbol{x}))\right], \tag{29}$$

$$X_{t_n t_n-2\tau}(\boldsymbol{x}) \approx \boldsymbol{x} - \tau\left[\boldsymbol{u}_h^*(\boldsymbol{x}) + \boldsymbol{u}_h^{n-2}(\boldsymbol{x} - 2\tau \boldsymbol{u}_h^*(\boldsymbol{x}))\right], \tag{30}$$

where we approximate the velocity field at t_n by the linear extrapolation $\boldsymbol{u}_h^* = 2\boldsymbol{u}_h^{n-1} - \boldsymbol{u}_h^{n-2}$. As described in section 4.1.1, we replace the velocity fields by suitable smooth approximations. We obtain

$$X_{t_n,t_n-\tau}(\boldsymbol{x}) \approx \bar{X}_{t-\tau}(\boldsymbol{x}) \coloneqq \boldsymbol{x} - \frac{\tau}{2}\left[\bar{\boldsymbol{u}}_h^*(\boldsymbol{x}) + \bar{\boldsymbol{u}}_h^{n-1}(\boldsymbol{x} - \tau \bar{\boldsymbol{u}}_h^*(\boldsymbol{x}))\right], \tag{31}$$

$$X_{t_n,t_n-2\tau}(\boldsymbol{x}) \approx \bar{X}_{t-2\tau}(\boldsymbol{x}) \coloneqq \boldsymbol{x} - \tau\left[\bar{\boldsymbol{u}}_h^*(\boldsymbol{x}) + \bar{\boldsymbol{u}}_h^{n-2}(\boldsymbol{x} - 2\tau \bar{\boldsymbol{u}}_h^*(\boldsymbol{x}))\right], \tag{32}$$

where $\bar{\boldsymbol{u}}_h^\bullet$ with $\bullet = *, n-1, n-2$ denotes the smoothened version of $\boldsymbol{u}_h^\bullet$ as it will be constructed in the next section.

4.1.3 Smooth reconstruction of the velocity field

Given a discrete velocity field $\boldsymbol{u}_h^\flat \in \Lambda^1_{h,p}(\Omega)$, we can define a smoothened version $\bar{\boldsymbol{u}}_h$ of $\boldsymbol{u}_h$ that is

- Lipschitz continuous to ensure stable evaluation of (28),
- well-defined on every point of the meshed domain,
- practically computable, and
- second-order accurate.

We introduce $\bar{\boldsymbol{u}}_h$ as follows. Let $h_{\min}$ denote the length of the shortest edge of the mesh and $(u_h^i)_{i=1,..,d}$ the components of $\boldsymbol{u}_h$. Then,

$$\bar{u}_h^i(\boldsymbol{x}) = \frac{1}{h_{\min}} \int_{x_i - \frac{1}{2}h_{\min}}^{x_i + \frac{1}{2}h_{\min}} u_h^i([x_1, \ldots, x_{i-1}, \xi, x_{i+1}, \ldots, x_d]^T) d\xi \tag{33}$$

provides a second-order, Lipschitz-continuous approximation of $\boldsymbol{u}_h$. In the above definition, we can also replace $h_{\min}$ by a localized parameter that scales as $\mathcal{O}(h)$ with h the length of the edges "close" to $\boldsymbol{x}$. Note that the above integral can be evaluated up to machine precision using the algorithm as described in section 3.2 for (20). The averaging (33) provides a second-order approximation of $\boldsymbol{u}_h$ on every point in the mesh.

4.2 A first- and second-order SL scheme

We are now ready to turn the ideas of section 3 into a concrete numerical scheme for the incompressible Navier-Stokes equations as given in (2). We cast (2a) and (2b) into weak form and, subsequently, do Galerkin discretization in space relying on those spaces of discrete differential forms introduced in section 3.1. For the first-order scheme, we have the following discrete variational formulation. Given $\omega_h^{n-1} \in \Lambda^1_{h,1}(\Omega)$, we search $p_h^n \in \Lambda^0_{h,1}(\Omega), \omega_h^n \in \Lambda^1_{h,1}(\Omega)$ such that

$$\left(\frac{1}{\tau}\left[\omega_h^n - \mathcal{I}_{h,1}\bar{X}^*_{t_n,t_n-\tau}\omega_h^{n-1}\right], \eta_h\right)_\Omega + \varepsilon\left(\mathrm{d}\omega_h^n, \mathrm{d}\eta_h\right)_\Omega + \left(\mathrm{d}p_h^n, \eta_h\right)_\Omega = \left(f^n, \eta_h\right)_\Omega, \tag{34a}$$

$$\left(\omega_h^n, \mathrm{d}\psi_h\right)_\Omega = 0 \tag{34b}$$

for all $\eta_h \in \Lambda^1_{h,1}(\Omega)$ and $\psi_h \in \Lambda^0_{h,1}(\Omega)$. $\mathcal{I}_{h,p}$ denotes the projection operator as defined in section 3.1. For the second-order scheme, we use second-order timestepping and second-order discrete differential forms. Given $\omega_h^{n-2}, \omega_h^{n-1} \in \Lambda^1_{h,2}(\Omega)$, we search $p_h^n \in \Lambda^0_{h,2}(\Omega), \omega_h^n \in \Lambda^1_{h,2}(\Omega)$ such that

$$\left(\frac{1}{2\tau}\left[3\omega_h^n - 4\mathcal{I}_{h,2}\bar{X}^*_{t_n,t_n-\tau}\omega_h^{n-1} + \mathcal{I}_{h,2}\bar{X}^*_{t_n,t_n-2\tau}\omega_h^{n-2}\right], \eta_h\right)_\Omega + \varepsilon\left(\mathrm{d}\omega_h^n, \mathrm{d}\eta_h\right)_\Omega + \left(\mathrm{d}p_h^n, \eta_h\right)_\Omega = \left(f^n, \eta_h\right)_\Omega, \tag{35a}$$

$$\left(\omega_h^n, \mathrm{d}\psi_h\right)_\Omega = 0 \tag{35b}$$

for all $\eta_h \in \Lambda^1_{h,2}(\Omega)$ and $\psi_h \in \Lambda^0_{h,2}(\Omega)$. Numerical experiments reported in section 5 give evidence that these schemes indeed do provide first- and second-order convergence for smooth solutions. Note that the schemes presented in this section only require solving symmetric, linear systems of equations at every time-step.

4.3 Conservative SL schemes

In order to enforce energy-tracking—the correct behavior of the total energy $E(t)$ over time as expressed in (3)—we add a suitable constraint plus a Lagrange multiplier to

the discrete variational problems proposed in section 4.2. Given $\omega_h^{n-1} \in \Lambda_{h,1}^1(\Omega)$, we search $p_h^n \in \Lambda_{h,1}^0(\Omega)$, $\omega_h^n \in \Lambda_{h,1}^1(\Omega)$, and $\mu^n \in \mathbb{R}$ such that

$$\left(\frac{1}{\tau}\left[\omega_h^n - \mathcal{I}_{h,1}\bar{X}^*_{t_n,t_n-\tau}\omega_h^{n-1}\right],\eta_h\right)_\Omega + (\mathrm{d}p_h^n,\eta_h)_\Omega + \varepsilon(\mathrm{d}\omega_h^n,\mathrm{d}\eta_h)_\Omega$$
$$+\mu^n\left[(\omega_h^n,\eta_h)_\Omega + 2\varepsilon\tau(\mathrm{d}\omega_h^n,\mathrm{d}\eta_h)_\Omega - \tau(f^n,\eta_h)_\Omega\right] = (f^n,\eta_h)_\Omega\,, \tag{36a}$$

$$(\omega_h^n,\mathrm{d}\psi_h)_\Omega = 0, \tag{36b}$$

$$(\omega_h^n,\omega_h^n)_\Omega + 2\varepsilon\tau(\mathrm{d}\omega_h^n,\mathrm{d}\omega_h^n)_\Omega - \tau(f^n,\omega_h^n)_\Omega = \left(\omega_h^{n-1},\omega_h^{n-1}\right)_\Omega \tag{36c}$$

for all $\eta_h \in \Lambda_{h,1}^1(\Omega)$ and $\psi_h \in \Lambda_{h,1}^0(\Omega)$. Note that the last scalar equation enforces energy conservation for $\varepsilon = 0$ and $f = 0$. To solve the nonlinear system (36) for ω_h^n, p_h^n, μ^n, we propose the following iterative scheme. Assume that we have a sequence $(\omega_{h,k}^n)_{k\in\mathbb{N}}$ with $\omega_{h,k}^n \to \omega_h^n (k \to \infty)$. Then we can employ the Newton-like linearization

$$\begin{aligned}(\omega_h^n,\omega_h^n)_\Omega &\leftarrow \left(\omega_{h,k}^n,\omega_{h,k}^n\right)_\Omega \\ &= \left(\omega_{h,k-1}^n,\omega_{h,k-1}^n\right)_\Omega + 2\left(\omega_{h,k-1}^n,\omega_{h,k}^n-\omega_{h,k-1}^n\right)_\Omega + \mathcal{O}\left(||\omega_{h,k}^n-\omega_{h,k-1}^n||_\Omega^2\right).\end{aligned} \tag{37}$$

We use the above expansion to replace the quadratic terms $(\omega^n,\omega^n)_\Omega$ and $(\mathrm{d}\omega^n,\mathrm{d}\omega^n)_\Omega$ and arrive at the following linear variational problem to be solved in every step of the inner iteration. Given $\omega_h^{n-1}, \omega_{h,k-1}^n \in \Lambda_{h,1}^1(\Omega)$, we search $p_{h,k}^n \in \Lambda_{h,1}^0(\Omega)$, $\omega_{h,k}^n \in \Lambda_{h,1}^1(\Omega)$, and $\mu_k^n \in \mathbb{R}$ such that

$$\left(\frac{1}{\tau}\left[\omega_{h,k}^n - \mathcal{I}_{h,1}\bar{X}^*_{t_n,t_n-\tau}\omega_h^{n-1}\right],\eta_h\right)_\Omega + \left(\mathrm{d}p_{h,k}^n,\eta_h\right)_\Omega + \varepsilon(\mathrm{d}\omega_{h,k}^n,\mathrm{d}\eta_h)_\Omega$$
$$+\mu_k^n\left[(\omega_{h,k-1}^n,\eta_h)_\Omega + 2\varepsilon\tau(\mathrm{d}\omega_{h,k-1}^n,\mathrm{d}\eta_h)_\Omega - \tau(f^n,\eta_h)_\Omega\right] = (f^n,\eta_h)_\Omega\,, \tag{38a}$$

$$\left(\omega_{h,k}^n,\mathrm{d}\psi_h\right)_\Omega = 0, \tag{38b}$$

$$\begin{aligned}&\left(\omega_{h,k-1}^n,\omega_{h,k-1}^n\right)_\Omega + 2\left(\omega_{h,k-1}^n,\omega_{h,k}^n-\omega_{h,k-1}^n\right) \\ &+2\varepsilon\tau[(\mathrm{d}\omega_{h,k-1}^n,\mathrm{d}\omega_{h,k-1}^n)_\Omega + 2(\mathrm{d}\omega_{h,k-1}^n,\mathrm{d}\omega_{h,k}^n-\mathrm{d}\omega_{h,k-1}^n)] \\ &-\tau(f^n,\omega_{h,k-1}^n)_\Omega = \left(\omega_h^{n-1},\omega_h^{n-1}\right)_\Omega\end{aligned} \tag{38c}$$

for all $\eta_h \in \Lambda_{h,1}^1(\Omega)$ and $\psi_h \in \Lambda_{h,1}^0(\Omega)$. This is a symmetric, linear system that is equivalent to the original system in the limit $(\omega_{h,k}^n, p_{h,k}^n, \mu_k^n) \to (\omega_h^n, p_h^n, \mu^n)$. In numerical experiments we observe that it takes around 2-3 steps of the inner iteration to converge to machine precision using an initial guess $\omega_{h,0}^n = \omega_h^{n-1}$. We can apply the same idea for energy-tracking to our second-order scheme as proposed in section 4.2.

Remark 4 *For the case $\varepsilon = 0$ and $f = 0$, we can also enforce helicity conservation by adding a suitable Lagrange multiplier to the discrete system. Given $\omega_h^{n-1} \in \Lambda_{h,1}^1(\Omega)$, we search $p_h^n \in \Lambda_{h,1}^0$, $\omega_h^n \in \Lambda_{h,1}^1(\Omega)$, and $\lambda^n \in \mathbb{R}$ such that*

$$\left(\frac{1}{\tau}\left[\omega_h^n - \mathcal{I}_{h,1}\bar{X}^*_{t_n,t_n-\tau}\omega_h^{n-1}\right], \eta_h\right)_\Omega + (dp_h^n, \eta_h)_\Omega + \varepsilon(d\omega_h^n, d\eta_h)_\Omega$$
$$+\lambda^n(\omega_h^n, d\eta_h)_\Omega + \lambda^n(d\omega_h^n, \eta_h)_\Omega + \mu^n(\omega_h^n, \eta_h)_\Omega = 0, \tag{39a}$$

$$(\omega_h^n, d\psi_h)_\Omega = 0, \tag{39b}$$

$$(\omega_h^n, \omega_h^n)_\Omega = \left(\omega_h^{n-1}, \omega_h^{n-1}\right)_\Omega, \tag{39c}$$

$$(\omega_h^n, d\omega_h^n)_\Omega = \left(\omega_h^{n-1}, d\omega_h^{n-1}\right)_\Omega \tag{39d}$$

for all $\eta_h \in \Lambda_{h,1}^1(\Omega)$ and $\psi_h \in \Lambda_{h,1}^0(\Omega)$. By linearization as for energy-tracking, we obtain the following system. Given $\omega^{n-1}, \omega_{h,k-1}^n \in \Lambda_{h,1}^1(\Omega)$, we search $p_{h,k}^n \in \Lambda_{h,1}^0(\Omega)$, $\omega_{h,k}^n \in \Lambda_{h,1}^1(\Omega)$, and $\lambda_k^n \in \mathbb{R}$ such that

$$\left(\frac{1}{\tau}\left[\omega_{h,k}^n - \mathcal{I}_{h,1}\bar{X}^*_{t_n,t_n-\tau}\omega^{n-1}\right], \eta_h\right)_\Omega + \left(dp_{h,k}^n, \eta_h\right)_\Omega + \varepsilon(d\omega_{h,k}^n, d\eta_h)_\Omega$$
$$+\lambda_k^n(\omega_{h,k-1}^n, d\eta_h)_\Omega + \lambda_k^n(d\omega_{h,k-1}^n, \eta_h)_\Omega + \mu_k^n(\omega_{h,k-1}^n, \eta_h)_\Omega = 0, \tag{40a}$$

$$\left(\omega_{h,k}^n, d\psi_h\right)_\Omega = 0, \tag{40b}$$

$$\left(\omega_{h,k-1}^n, \omega_{h,k-1}^n\right)_\Omega + 2\left(\omega_{h,k-1}^n, \omega_{h,k}^n - \omega_{h,k-1}^n\right) = \left(\omega_h^{n-1}, \omega_h^{n-1}\right)_\Omega, \tag{40c}$$

$$\left(\omega_{h,k-1}^n, d\omega_{h,k}^n\right)_\Omega + \left(\omega_{h,k}^n, d\omega_{h,k-1}^n\right)_\Omega - \left(\omega_{h,k-1}^n, d\omega_{h,k-1}^n\right)_\Omega = \left(\omega_h^{n-1}, d\omega_h^{n-1}\right)_\Omega \tag{40d}$$

for all $\eta_h \in \Lambda_{h,1}^1(\Omega)$ and $\psi_h \in \Lambda_{h,1}^0(\Omega)$. Again, this is a symmetric, linear system that is equivalent to the original system in the limit $(\omega_{h,k}^n, p_{h,k}^n, \mu_k^n) \to (\omega_h^n, p_h^n, \mu^n)$. Also, numerical experiments hint that it takes around 2-3 steps of the inner iteration to converge to machine precision using an initial guess $\omega_{h,0}^n = \omega_h^{n-1}$.

5 Numerical Results

In this section, we present multiple numerical experiments to validate the new scheme. In the following, we will always consider schemes that include energy-tracking as introduced in section 4.3 unless stated otherwise. We only include helicity-conservation as introduced in Remark 4 for domains in $\mathbb{R}^3$ when $\varepsilon = 0$ and $f = 0$. The experiments are based on a C++ code that heavily relies on MFEM [2]. The source code is published under the GNU General Public License in the online code repository `https://gitlab.com/WouterTonnon/semi-lagrangian-tools`.

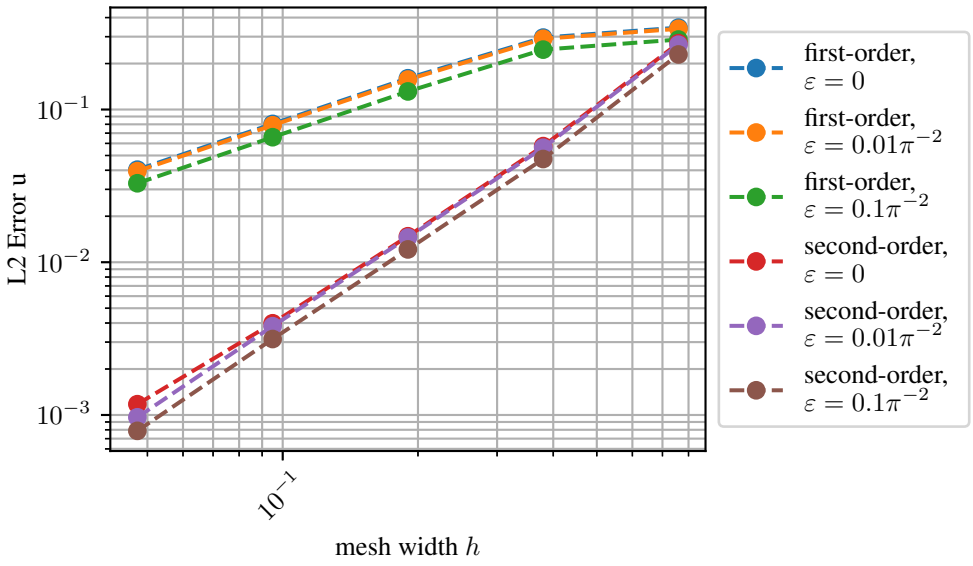


Figure 6: Convergence results for Experiment 1 using the first- and second-order schemes on simplicial meshes with mesh width h, timestep $\tau = 0.065804h$. We observe first- and second-order algebraic convergence for all values of ε.

5.1 Experiment 1: Decaying Taylor-Green Vortex

We consider the incompressible Navier-Stokes equations with $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$, $T = 1$, varying $\varepsilon \geq 0$, $f = 0$, and vanishing boundary conditions. An exact, classical solution is the following Taylor-Green vortex [42]

$$\boldsymbol{u}(t, \boldsymbol{x}) = \begin{bmatrix} \cos(\pi x_1)\sin(\pi x_2) \\ -\sin(\pi x_1)\cos(\pi x_2) \end{bmatrix} e^{-2\pi^2 \varepsilon t}. \tag{41}$$

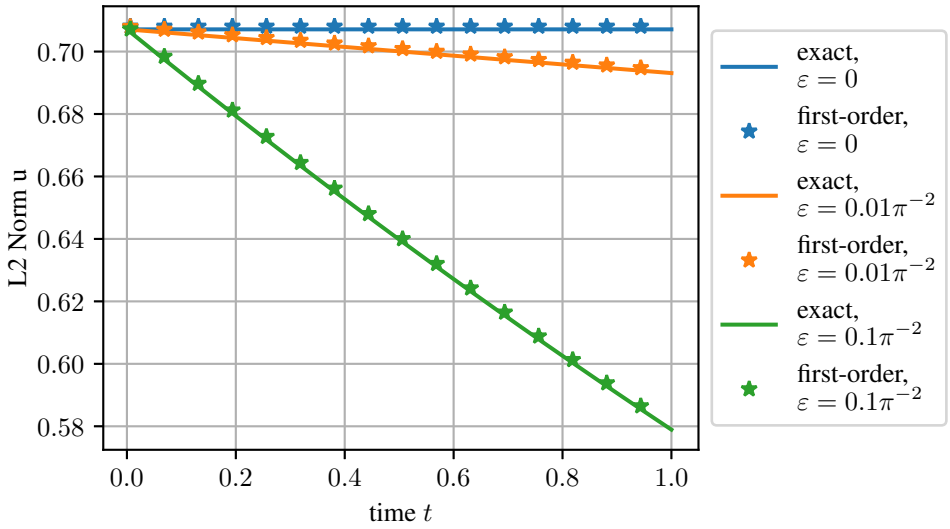
We ran a h-convergence analysis for different values of $\varepsilon \geq 0$ and summarize the results in Figure 6. We also track the energy for different values of ε and compare the energy to the exact solution in Figure 7.

5.2 Experiment 2: Taylor-Green Vortex

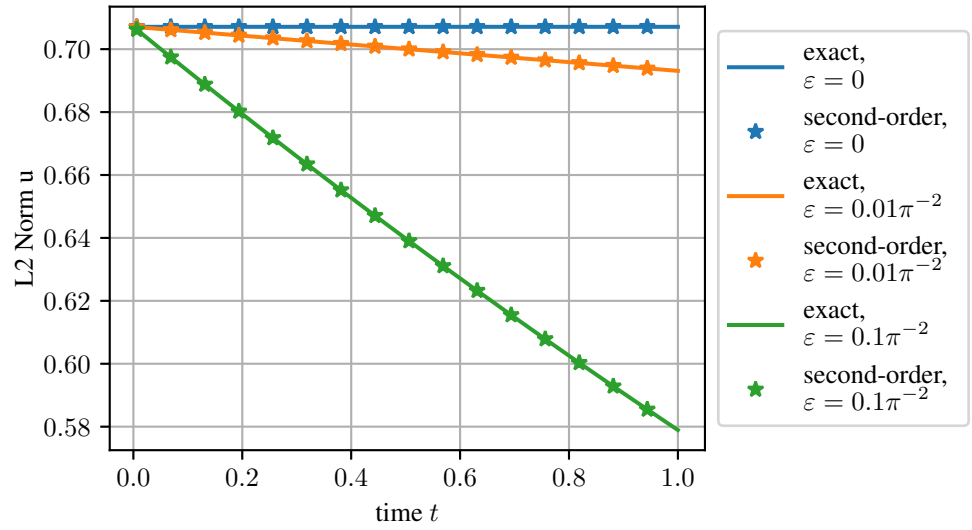
We consider the incompressible Navier-Stokes equations with $\Omega = [-1, 1]^2$, $T = 1$, varying $\varepsilon \geq 0$, f and the boundary conditions chosen such that

$$\boldsymbol{u}(t, \boldsymbol{x}) = \begin{bmatrix} \cos(\pi x_1)\sin(\pi x_2) \\ -\sin(\pi x_1)\cos(\pi x_2) \end{bmatrix} \tag{42}$$

is an exact, classical solution. We ran a h-convergence analysis for all parameters and summarize the results in Figure 8. We observe first- and second-order algebraic convergence for the corresponding schemes. Note that the error of the scheme is stable



(a) first-order



(b) second-order

Figure 7: Energy of the discrete and exact solution for Experiment 1 using the first- and second-order, energy-tracking schemes on a simplicial mesh with mesh width $h = 0.0949795$, timestep $\tau = 0.00625$.

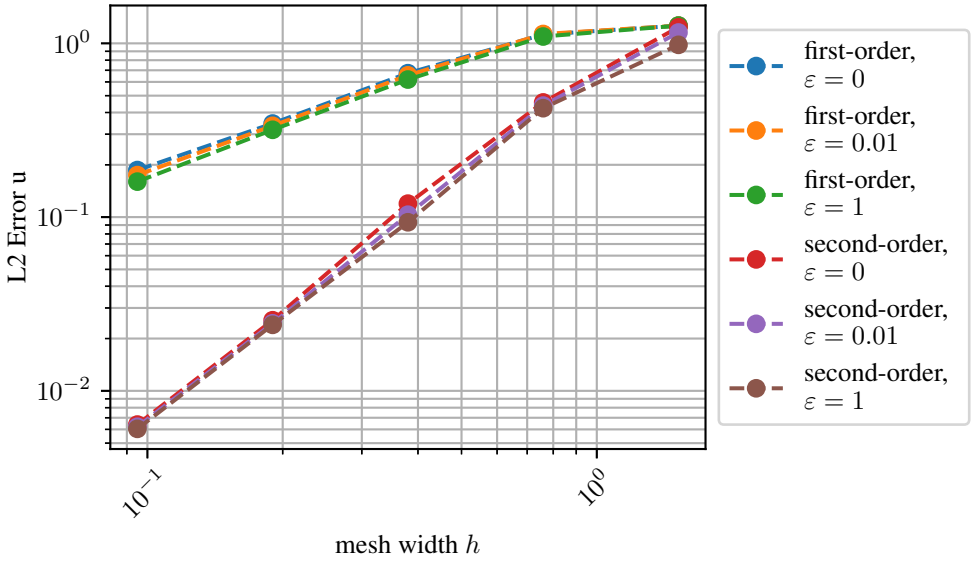


Figure 8: Convergence results for Experiment 2 using the first- and second-order, non-conservative schemes on simplicial meshes with mesh width h, timestep $\tau = 0.032902h$. As $\varepsilon \to 0$ the error remains bounded.

as $\varepsilon \to 0$. This is in agreement with the analysis performed on the vectorial advection equations presented in [24]. This experiment thus suggests that this analysis can be extended to the scheme presented in this work.

5.3 Experiment 3: A rotating hump problem

The Taylor-Green vortices provide exact solutions to the incompressible Navier-Stokes equations, but they are rather "static" solutions. In this experiment, we consider a more dynamic solution. Let us consider the incompressible Navier-Stokes equations with $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$, $T = 1$, $\varepsilon = 0$, $f = 0$, and vanishing normal boundary conditions. We consider the following initial condition

$$\boldsymbol{u}_0(\boldsymbol{x}) = \begin{bmatrix} -\pi e^{x_1} \cos(\pi x_1) \sin(\pi x_2) \\ \pi e^{x_1} \sin(\pi x_1) \cos(\pi x_2) - e^{x_1} \cos(\pi x_1) \cos(\pi x_2) \end{bmatrix}. \tag{43}$$

The exact solution to this problem is unknown, so we compare the solution computed by our scheme to the solution produced by the incompressible Euler solver Gerris [37]. The algorithm used in this solver is described in [36]. We computed the solution to this problem using the second-order, energy-tracking scheme presented in this work. Then, we plotted the magnitude of the computed velocity vector field for different mesh-sizes and time-steps at different time instances in Figures 10 to 13. Note that different visualisation tools were used to visualize the fields computed using the different solvers, but we observe that the solution computed by the semi-Lagrangian scheme comes visually closer to the solution computed by Gerris as we decrease the mesh width and

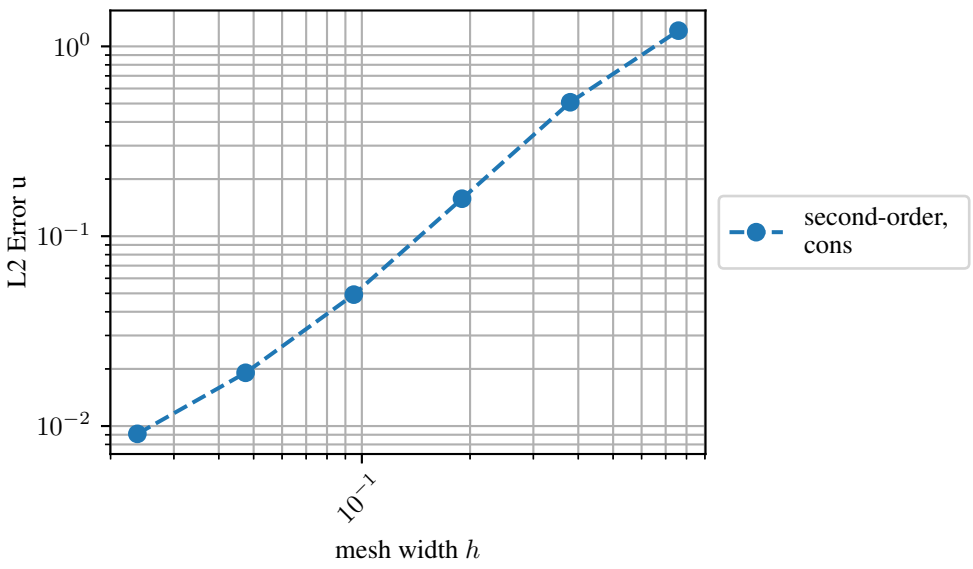


Figure 9: Convergence results for Experiment 2 using the second-order, conservative scheme on simplicial meshes with mesh width h, timestep $\tau = 0.06580h$, and final time $T = 1$. The reference solution is a solution computed by Gerris [36]

time step. This is confirmed by Figure 9, where we display the L2 error between the solution computed using the semi-Lagrangian scheme and the solution computed using Gerris. In Figure 15, we display the vector field as computed using the second-order, conservative semi-Lagrangian scheme.

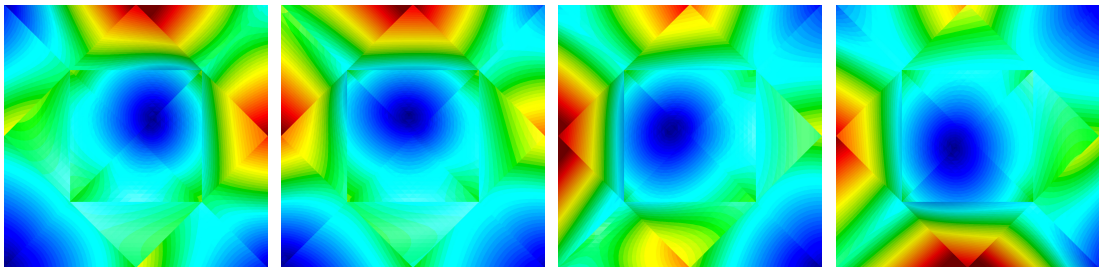
Also, in Figure 16 we display the values of the L2 norm over time of the solutions produced using our first- and second-, energy-tracking and non-energy-tracking schemes. Note that the energy-tracking schemes preserve the L2 norm as expected. The first-order, non-conservative scheme seems unstable at first, but in reality the ordinate axis spans a very small range and it turns out that the L2 norm converges to a bounded value for longer run-times. Note that the helicity has no meaning in $\mathbb{R}^2$.

5.4 Experiment 4: Taylor-Green Vortex in 3D

To observe conservation of helicity, we need to consider a problem in 3D. We consider the incompressible Navier-Stokes equations with $\Omega = [-\frac{1}{2}, \frac{1}{2}]^3$, $T = 1$, $\varepsilon = 0$, vanishing normal boundary conditions, and f chosen such that

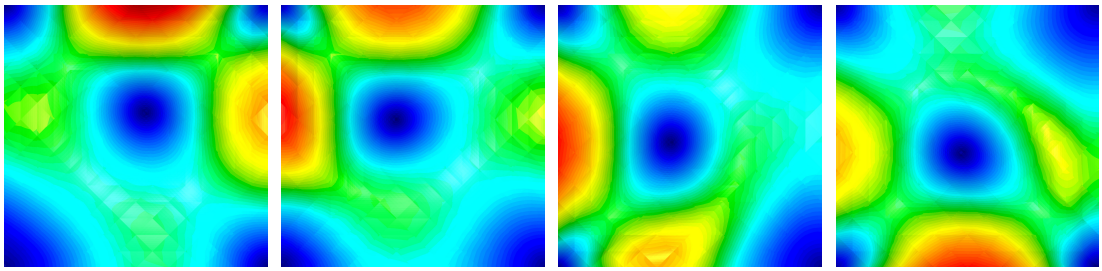
$$\boldsymbol{u}(t, \boldsymbol{x}) = \begin{bmatrix} \cos(\pi x_1)\sin(\pi x_2)\sin(\pi x_3) \\ -\frac{1}{2}\sin(\pi x_1)\cos(\pi x_2)\sin(\pi x_3) \\ -\frac{1}{2}\sin(\pi x_1)\sin(\pi x_2)\cos(\pi x_3) \end{bmatrix} \tag{44}$$

is a solution. Note that since the solution is static, we can enforce helicity conservation despite $f \neq 0$. We run several experiments using the first- and second-order,



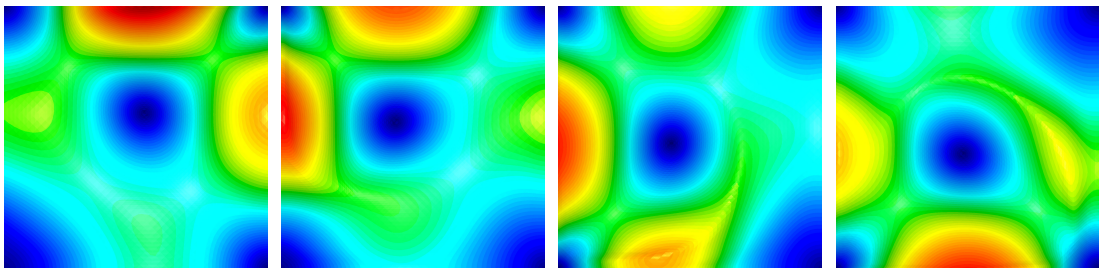
(a) $t = 0.25$ (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1$

Figure 10: Experiment 3: mesh width $h = 0.379918$ and time-step $\tau = 0.025$.



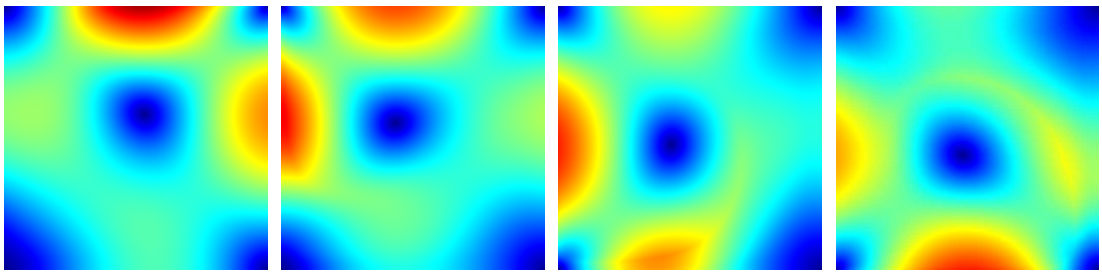
(a) $t = 0.25$ (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1$

Figure 11: Experiment 3: mesh width $h = 0.0949795$ and time-step $\tau = 0.00625$.



(a) $t = 0.25$ (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1$

Figure 12: Experiment 3: mesh width $h = 0.023744875$ and time-step $\tau = 0.0015625$.



(a) $t = 0.25$ (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1$

Figure 13: Reference solution Experiment 3 computed using [37].

Figure 14: Colorbar associated with Figures 10 to 13

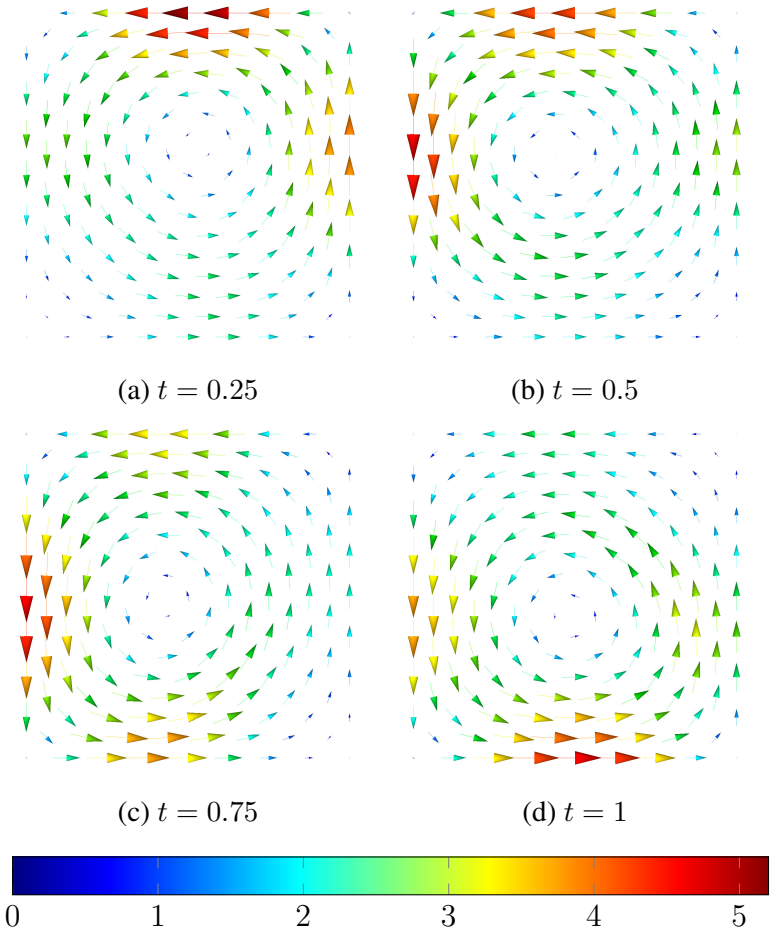


Figure 15: Velocity field for Experiment 3 computed using the second-order, conservative semi-Lagrangian scheme on a simplicial mesh with mesh width $h = 0.189959$ and time-step $\tau = 0.0125$. The colors indicate the magnitude of the vector.

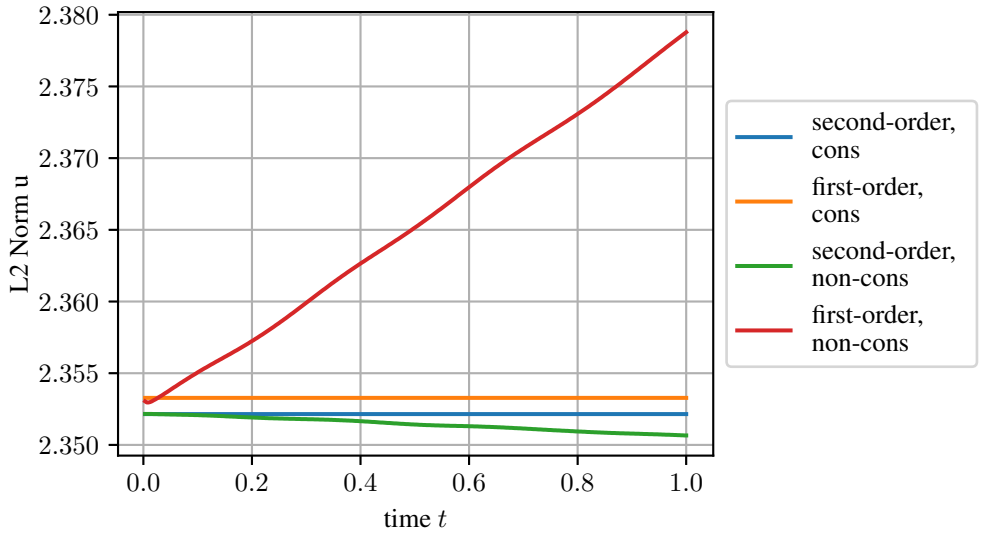


Figure 16: The L2 norm of the computed solutions for Experiment 3 using different variants of the semi-Lagrangian scheme on a simplicial mesh with mesh width $h = 0.04748975$ and time-step $\tau = 0.003125$. In the legend, 'cons' is short for 'conservative' and refers to energy-tracking schemes. We use $\varepsilon = 0$ and $f = 0$.

conservative semi-Lagrangian schemes. We summarize the results in Figure 17 and we observe first- and second-order algebraic convergence for the corresponding schemes. In Figure 18 and Figure 19 we plot the L2 norm and helicity over time of the discrete solution for both the first- and second-order scheme. We observe that both quantities are conserved up to machine precision.

5.5 Experiment 5: A transient solution in 3D

To verify the scheme for transient solutions in 3D, we consider the incompressible Navier-Stokes equations with $\Omega = [-\frac{1}{2}, \frac{1}{2}]^3$, $T = 1$, $\varepsilon = 0$, vanishing normal boundary conditions and f is chosen such that

$$\boldsymbol{u}(t, \boldsymbol{x}) = \begin{bmatrix} -x_2\pi \cos(\frac{t}{4} + \pi x_2 x_3) \cos(\pi x_2) \\ -x_3\pi \cos(\frac{t}{4} + \pi x_1 x_3) \cos(\pi x_3) \\ -x_1\pi \cos(\frac{t}{4} + \pi x_1 x_2) \cos(\pi x_1) \end{bmatrix} \tag{45}$$

is a solution. We ran a simulation for different mesh-sizes with time-steps determined by a suitable CFL condition. We summarize the results in Figure 20. We observe second-order convergence for the second-order scheme. The first-order scheme seems to achieve an order of convergence that is between first- and second-order, but this may be pre-asymptotic behaviour.

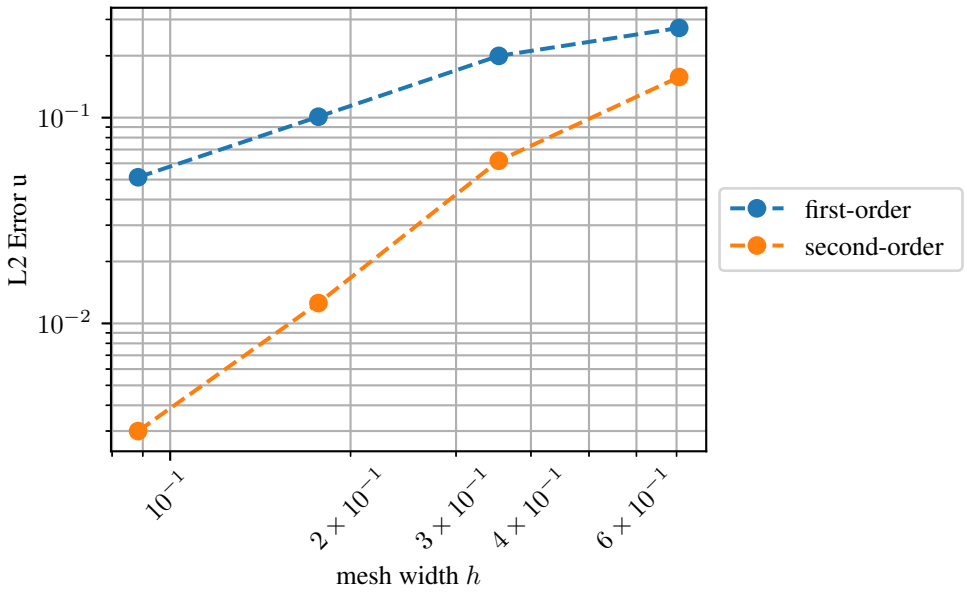


Figure 17: Convergence results for Experiment 4 using the first- and second-order, conservative schemes on simplicial meshes with mesh width h, timestep $\tau = \frac{1}{\sqrt{2}}h$.

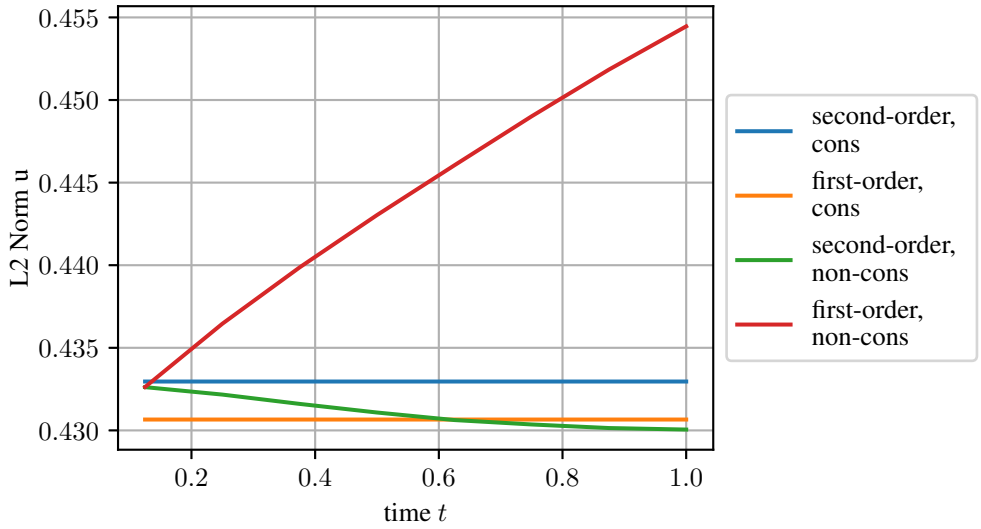


Figure 18: The L2 norm of the computed solutions for Experiment 4 using different variants of the semi-Lagrangian scheme on a simplicial mesh with mesh width $h = 0.08838834764$ and time-step $\tau = 0.0625$. In the legend, 'cons' is short for 'conservative' and refers to energy-tracking schemes and helicity-conserving schemes. We use $\varepsilon = 0$ and $f = 0$.

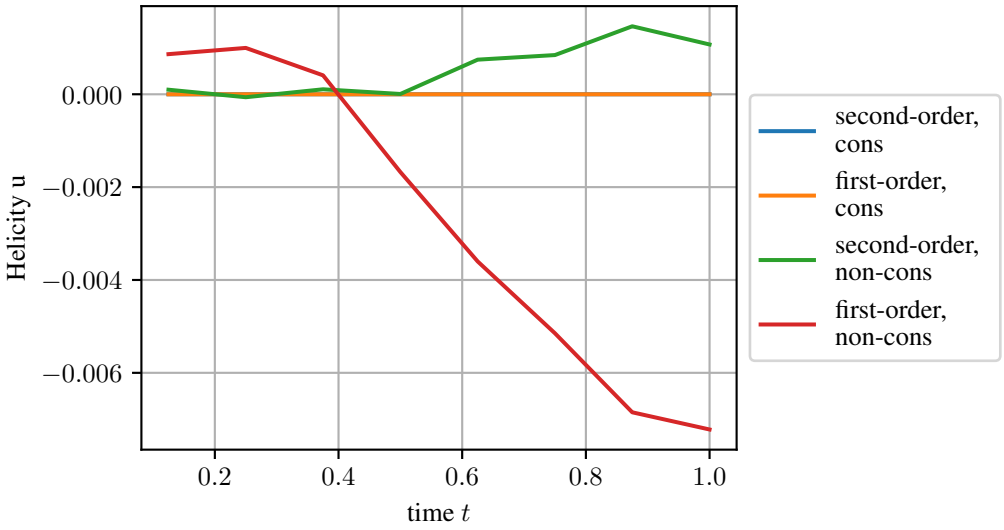


Figure 19: The helicity of the computed solutions for Experiment 4 using different variants of the semi-Lagrangian scheme on a simplicial mesh with mesh width $h = 0.08838834764$ and time-step $\tau = 0.0625$. In the legend, 'cons' is short for 'conservative' and refers to energy-tracking and helicity-conserving schemes. We use $\varepsilon = 0$ and $f = 0$.

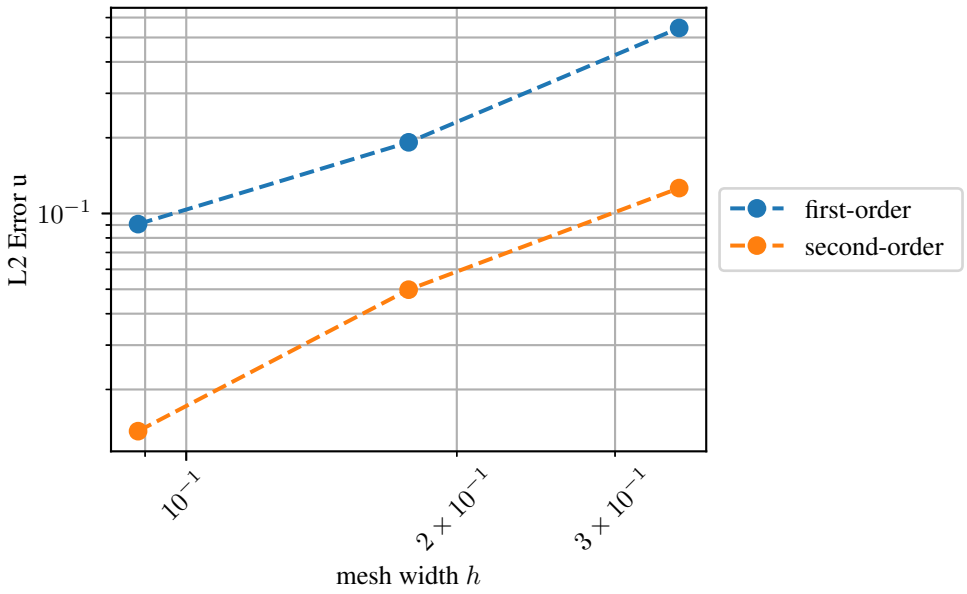


Figure 20: Convergence results for Experiment 5 using the first- and second-order schemes without energy-tracking and helicity-conservation on simplicial meshes with mesh width h, timestep $\tau = \frac{1}{\sqrt{2}}h$.

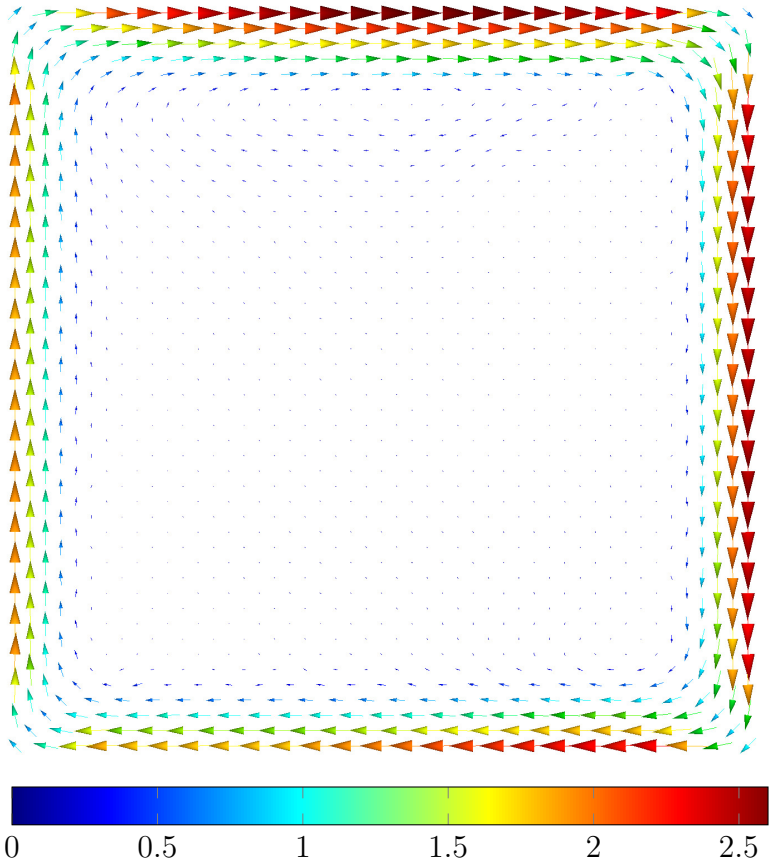


Figure 21: Velocity field at $T = 7.93s$ of Experiment 6 computed using the second-order, non-conservative semi-Lagrangian scheme on a simplicial mesh with mesh width $h = 0.189959$ and $\tau = 0.01$.

5.6 Experiment 6: Lid-driven cavity with slippery walls

In this section, we simulate a situation that resembles a lid-driven cavity problem. Consider the incompressible Navier-Stokes with $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$, $T = 7.93$, $\varepsilon = 0$, vanishing normal boundary conditions and the initial velocity field is set equal to zero. Then, to simulate a moving lid at the top, we apply the force-field $\boldsymbol{f}(t, \boldsymbol{x}) = [v(\boldsymbol{x}), 0]^T$ with

$$v(\boldsymbol{x}) = \begin{cases} \exp\left(1 - \frac{1}{1-100(0.5-x_2)^2}\right), & \text{if } 1 - 100(0.5 - x_2)^2 > 0, \\ 0, & \text{else.} \end{cases} \tag{46}$$

This force field gives a strong force in the x_1-direction close to the top lid, but quickly tapers off to zero as we go further from the top lid. In Figure 21, we display the computed velocity field. Note that, because we apply slip boundary conditions, we do not expect to observe vortices. The numerical solution reproduces this expectation.

5.7 Experiment 7: A more complicated domain

The numerical experiments given above, show the convergence and conservative properties of the introduced schemes. However, these experiments are all performed on very simple, rectangular domains. In this experiment, we consider a more complicated domain and mesh (generated using [22]) as shown in Figure 22.

We consider the case of the incompressible Navier-Stokes equations on the domain as given in Figure 22, $T = 100$, $\varepsilon = 0$, $f = 0$ and vanishing normal boundary conditions. We need to construct an initial condition that is divergence-free with vanishing normal boundary conditions. Following an approach close to a Chorin projection, we start with

$$\boldsymbol{w}(x, y) = \begin{bmatrix} \sin\left(2\cos(\sqrt{x^2+y^2}) - \text{atan2}(y, x)\right) \\ \sin\left(\cos(\sqrt{x^2+y^2}) - 2\,\text{atan2}(y, x)\right) \end{bmatrix}. \tag{47}$$

We use this definition to define a scalar function, φ, as

$$\Delta\varphi = \nabla \cdot \boldsymbol{w} \qquad \text{in } \Omega, \tag{48}$$

$$\nabla\varphi \cdot \hat{n} = \boldsymbol{w} \cdot \hat{n} \qquad \text{on } \partial\Omega. \tag{49}$$

We can define our initial condition, $\boldsymbol{u}_0$, as

$$\boldsymbol{u}_0 = \boldsymbol{w} - \nabla\varphi \tag{50}$$

Note that $\boldsymbol{u}_0$ is divergence-free and has vanishing normal boundary conditions. The above system of equations can be solved using an appropriate finite-element implementation.

Note that in this experiment, the field outside the domain is unknown. This is well-defined on a continuous level, since vanishing boundary conditions imply that no particle will flow in from outside the domain. However, on the discrete level we cannot guarantee that the same will happen. It could happen that a part of a transported edge (as discussed in section 3.2) ends up outside the domain. In this case, we will assume that the average of the vector field along the part of the edge that lies outside the domain, will have the same value as the average of the corresponding edge in its original location (before transport) at the previous timestep.

The first ten seconds were simulated and a video of the results can be found at `https://youtu.be/Eica8XHLtxY`. For the different schemes, we also tracked the energy in Figure 23.

6 Conclusion

We have developed a mesh-based semi-Lagrangian discretization of the time-dependent incompressible Navier-Stokes equations with free boundary conditions recast as a nonlinear transport problem for a momentum 1-form. A linearly implicit fully discrete version of the scheme enjoys excellent stability properties in the vanishing-viscosity

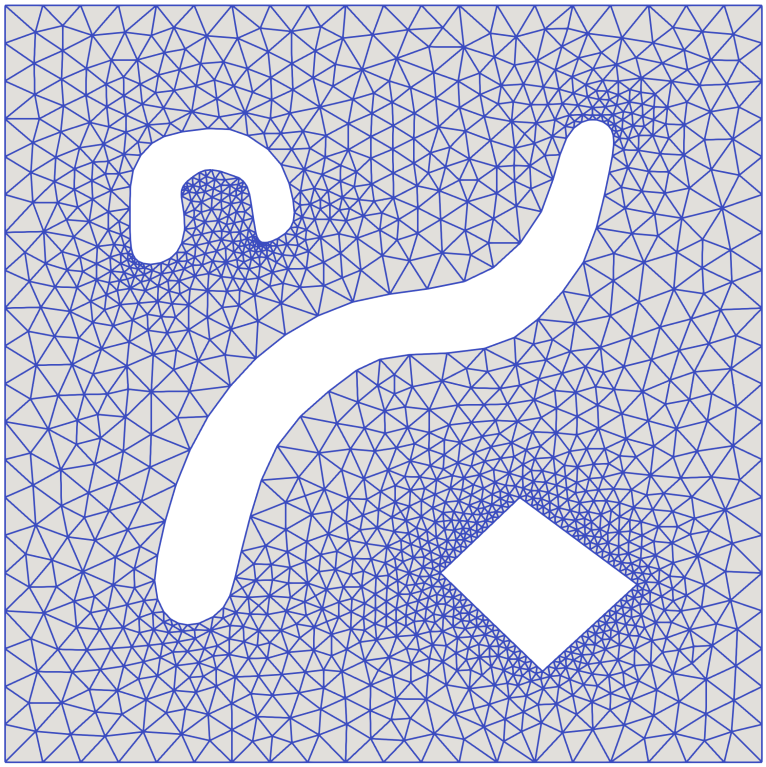


Figure 22: Domain and mesh associated with Experiment 7.

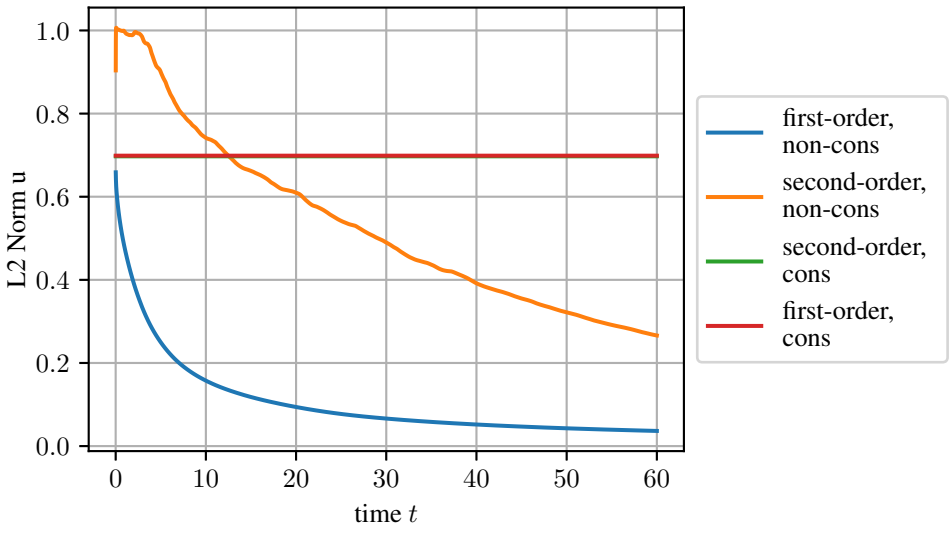


Figure 23: The L2 norm of the computed solutions for Experiment 7 using different variants of the semi-Lagrangian scheme on a simplicial mesh as given in Figure 22 and time-step $\tau = 0.01$. In the legend, 'cons' is short for 'conservative' and refers to energy-tracking schemes.

limit and is applicable to inviscid incompressible Euler flows. However, in this case conservation of energy and helicity have to be enforced separately. Making the reasonable choice of a time-step size proportional to the mesh width, the algorithm involves only local computations. Yet, these are significantly more expensive compared to those required for purely Eulerian finite-element and finite-volume methods. At this point the verdict on the competitiveness of our semi-Lagrangian scheme is still open.

A Two formulations of the momentum equation

Consider the momentum equation in (1)

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \varepsilon \Delta \boldsymbol{u} + \nabla p = \boldsymbol{0}.$$

Note that we have by standard vector calculus identities

$$\Delta \boldsymbol{u} = \nabla(\nabla \cdot \boldsymbol{u}) - \nabla \times \nabla \times \boldsymbol{u},$$

where we can use $\nabla \cdot \boldsymbol{u} = 0$ to obtain

$$\Delta \boldsymbol{u} = -\nabla \times \nabla \times \boldsymbol{u}.$$

This allows us to rewrite the momentum equation as

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \varepsilon \nabla \times \nabla \times \boldsymbol{u} + \nabla p = \boldsymbol{0}.$$

Using the gradient of the dot-product, we find

$$\nabla(\boldsymbol{u} \cdot \boldsymbol{u}) = 2\boldsymbol{u} \cdot \nabla \boldsymbol{u} + 2\boldsymbol{u} \times (\nabla \times \boldsymbol{u}).$$

This identity allows us to rewrite the momentum equation to

$$\partial_t \boldsymbol{u} + \nabla(\boldsymbol{u} \cdot \boldsymbol{u}) - \boldsymbol{u} \times (\nabla \times \boldsymbol{u}) + \varepsilon \nabla \times \nabla \times \boldsymbol{u} + \nabla \left(-\frac{1}{2} \boldsymbol{u} \cdot \boldsymbol{u} + p \right) = \boldsymbol{0}.$$

From [27, 24], we obtain the identity

$$(\mathrm{L}_{\boldsymbol{u}} \omega)^\sharp = \nabla(\boldsymbol{u} \cdot \boldsymbol{u}) - \boldsymbol{u} \times (\nabla \times \boldsymbol{u})$$

where ω is the differential 1-form such that $\boldsymbol{u} = \omega^\sharp$. Since the material derivative for this 1-form is

$$D_{\boldsymbol{u}} \omega \coloneqq \partial_t \omega + \mathrm{L}_{\boldsymbol{u}} \omega,$$

we find that the momentum equation can be written as

$$D_{\boldsymbol{u}} \omega + \varepsilon \delta \mathrm{d} \omega + \mathrm{d} \tilde{p} = 0,$$

where $\tilde{p} = -\frac{1}{2} \boldsymbol{u} \cdot \boldsymbol{u} + p$.

References

[1] Mofdi El-Amrani and Mohammed Seaid. "An L2-projection for the Galerkin-characteristic solution of incompressible flows". In: *SIAM Journal on Scientific Computing* 33.6 (Jan. 2011), pp. 3110–3131. ISSN: 10648275. DOI: 10.1137/100805790.

[2] Robert Anderson et al. "MFEM: A modular finite element methods library". In: *Computers and Mathematics with Applications* 81 (2021). ISSN: 08981221. DOI: 10.1016/j.camwa.2020.06.009.

[3] Douglas N. Arnold. *Finite Element Exterior Calculus*. Philadelphia, PA: Society for Industrial and Applied Mathematics, Dec. 2018, p. 120. ISBN: 978-1-61197-553-6. DOI: 10.1137/1.9781611975543.

[4] Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. "Finite element exterior calculus, homological techniques, and applications". In: *Acta Numerica* 15 (May 2006), pp. 1–155. ISSN: 0962-4929. DOI: 10.1017/S0962492906210018.

[5] Vladimir Arnold. "Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications à l'hydrodynamique des fluides parfaits". In: *Annales de L'Institut Fourier* 16 (1966), pp. 319–361. URL: http://www.numdam.org/articles/10.5802/aif.233/.

[6] Markus Bause and Peter Knabner. "Uniform error analysis for Lagrange-Galerkin approximations of convection-dominated problems". In: *SIAM Journal on Numerical Analysis* 39.6 (2002), pp. 1954–1984. ISSN: 00361429. DOI: 10.1137/S0036142900367478.

[7] M. Bercovier, O. Pironneau, and V. Sastri. "Finite elements and characteristics for some parabolic-hyperbolic problems". In: *Applied Mathematical Modelling* 7.2 (1983), pp. 89–96.

[8] Michel Bercovier and Olivier Pironneau. "Characteristics and the finite element method". In: *Finite Element Flow Analysis* (1982), pp. 67–73.

[9] Rodolfo Bermejo, Pedro Galán Del Sastre, and Laura Saavedra. "A Second Order in Time Modified Lagrange–Galerkin Finite Element Method for the Incompressible Navier–Stokes Equations". In: *SIAM Journal on Numerical Analysis* 50.6 (2012), pp. 3084–3109.

[10] Rodolfo Bermejo and Laura Saavedra. "Lagrange–Galerkin methods for the incompressible Navier-Stokes equations: a review". In: *Communications in Applied and Industrial Mathematics* 7.3 (Sept. 2016), pp. 26–55. ISSN: 2038-0909. DOI: 10.1515/caim-2016-0021.

[11] Rodolfo Bermejo and Laura Saavedra. "Modified Lagrange–Galerkin Methods to Integrate Time Dependent Incompressible Navier–Stokes Equations". In: *SIAM Journal on Scientific Computing* 37.6 (Jan. 2015), B779–B803. ISSN: 1064-8275. DOI: 10.1137/140973967.

[12] Rodolfo Bermejo and Laura Saavedra. “Modified Lagrange-Galerkin methods of first and second order in time for convection-diffusion problems”. In: *Numerische Mathematik* 120.4 (2012). ISSN: 0029599X. DOI: 10.1007/s00211-011-0418-8.

[13] Luca Bonaventura, Roberto Ferretti, and Lorenzo Rocchi. “A fully semi-Lagrangian discretization for the 2D incompressible Navier–Stokes equations in the vorticity-streamfunction formulation”. In: *Applied Mathematics and Computation* 323 (Apr. 2018), pp. 132–144. ISSN: 00963003. DOI: 10.1016/j.amc.2017.11.030.

[14] Karima Boukir, Yvon Maday, and Brigitte Métivet. “A high order characteristics method for the incompressible Navier–Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 116.1-4 (Jan. 1994), pp. 211–218. ISSN: 00457825. DOI: 10.1016/S0045-7825(94)80025-1.

[15] Gustavo C. Buscaglia and Enzo A. Dari. “Implementation of the Lagrange-Galerkin method for the incompressible Navier-Stokes equations”. In: *International Journal for Numerical Methods in Fluids* 15.1 (July 1992), pp. 23–36. ISSN: 0271-2091. DOI: 10.1002/fld.1650150103.

[16] Elena Celledoni, Bawfeh Kingsley Kometa, and Olivier Verdier. “High Order Semi-Lagrangian Methods for the Incompressible Navier–Stokes Equations”. In: *Journal of Scientific Computing* 66.1 (2016), pp. 91–115. ISSN: 08857474. DOI: 10.1007/s10915-015-0015-6.

[17] Elena Celledoni, Arne Marthinsen, and Brynjulf Owren. “Commutator-free Lie group methods”. In: *Future Generation Computer Systems* 19.3 (Apr. 2003), pp. 341–352. ISSN: 0167739X. DOI: 10.1016/S0167-739X(02)00161-9.

[18] Alexandre J. Chorin and Jerrold E. Marsden. *A mathematical introduction to fluid mechanics*. Springer, 1993, p. 172. ISBN: 978-0-387-97918-2. DOI: 10.1007/978-1-4612-0883-9.

[19] Luigi De Rosa. “On the Helicity conservation for the incompressible Euler equations”. In: *Proceedings of the American Mathematical Society* 148.7 (Feb. 2020), pp. 2969–2979. ISSN: 0002-9939. DOI: 10.1090/proc/14952. URL: http://arxiv.org/abs/1812.00678%20https://www.ams.org/proc/2020-148-07/S0002-9939-2020-14952-0/.

[20] Jim Douglas Jr and Thomas F. Russell. “Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures”. In: *SIAM Journal on Numerical Analysis* 19.5 (1982), pp. 871–885.

[21] Richard E. Ewing, Thomas F. Russell, and Mary F. Wheeler. "Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics". In: *Computer Methods in Applied Mechanics and Engineering* 47.1-2 (1984), pp. 73–92.

[22] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (Sept. 2009), pp. 1309–1331. ISSN: 00295981. DOI: 10.1002/nme.2579.

[23] Y. Hasbani, E. Livne, and M. Bercovier. "Finite elements and characteristics applied to advection-diffusion equations". In: *Computers & Fluids* 11.2 (1983), pp. 71–83.

[24] Holger Heumann and Ralf Hiptmair. "Convergence of Lowest Order Semi-Lagrangian Schemes". In: *Foundations of Computational Mathematics* 13.2 (2013), pp. 187–220. ISSN: 16153383. DOI: 10.1007/s10208-012-9139-3.

[25] Holger Heumann and Ralf Hiptmair. "Eulerian and semi-lagrangian methods for convection-diffusion for differential forms". In: (2011), pp. 1–26. DOI: 10.3929/ethz-a-006506738.

[26] Holger Heumann et al. "Fully discrete semi-Lagrangian methods for advection of differential forms". In: *BIT Numerical Mathematics* 52.4 (2012), pp. 981–1007. ISSN: 00063835. DOI: 10.1007/s10543-012-0382-4.

[27] Ralf Hiptmair. "Finite elements in computational electromagnetism". In: *Acta Numerica* 11 (2002), pp. 237–339. ISSN: 14740508. DOI: 10.1017/S0962492902000041.

[28] Philip Isett. "A proof of Onsager's conjecture". In: *Annals of Mathematics* 188.3 (2018), pp. 871–963. ISSN: 19398980. DOI: 10.4007/annals.2018.188.3.4.

[29] George Karniadakis and Spencer Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. 2nd ed. Oxford University Press, 2005, p. 668. ISBN: 9780198528692. DOI: 10.1093/acprof:oso/9780198528692.001.0001.

[30] P. D. Minev and C. Ross Ethier. "A characteristic/finite element algorithm for the 3-D Navier–Stokes equations using unstructured grids". In: *Computer Methods in Applied Mechanics and Engineering* 178.1-2 (July 1999), pp. 39–50. ISSN: 00457825. DOI: 10.1016/S0045-7825(99)00003-1.

[31] Marius Mitrea and Sylvie Monniaux. "The nonlinear Hodge-Navier-Stokes equations in Lipschitz domains". In: *Differential and Integral Equations* 22.3/4 (Mar. 2009), pp. 339–356. ISSN: 0893-4983. DOI: 10.57262/die/1356019778.

[32] K.W. Morton, A. Priestley, and E. Suli. “Stability of the Lagrange-Galerkin method with non-exact Integration”. In: *Mathematical Modelling and Numerical Analysis* 22.4 (1988), pp. 625–653. URL: http://www.numdam.org/item/M2AN_1988__22_4_625_0/.

[33] Andrea Natale and Colin J. Cotter. “A variational H(div) finite-element discretization approach for perfect incompressible fluids”. In: *IMA Journal of Numerical Analysis* 38.3 (2018), pp. 1388–1419. ISSN: 14643642. DOI: 10.1093/imanum/drx033.

[34] Anthony T. Patera. “A spectral element method for fluid dynamics: Laminar flow in a channel expansion”. In: *Journal of Computational Physics* 54.3 (1984), pp. 468–488. ISSN: 10902716. DOI: 10.1016/0021-9991(84)90128-1.

[35] Olivier Pironneau. “On the transport-diffusion algorithm and its applications to the Navier-Stokes equations”. In: *Numerische Mathematik* 38.3 (Oct. 1982), pp. 309–332. ISSN: 0029-599X. DOI: 10.1007/BF01396435.

[36] Stéphane Popinet. “Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries”. In: *Journal of Computational Physics* 190.2 (Sept. 2003), pp. 572–600. ISSN: 00219991. DOI: 10.1016/S0021-9991(03)00298-5.

[37] Stéphane Popinet. *The Gerris Flow Solver*. 2007. URL: https://gfs.sourceforge.net/wiki/index.php/Main_Page.

[38] Francesca Rapetti and Alain Bossavit. “Whitney Forms of Higher Degree”. In: *SIAM Journal on Numerical Analysis* 47.3 (2009), pp. 2369–2386. DOI: 10.1137/070705489.

[39] Thomas F. Russell. “Time stepping along characteristics with incomplete iteration for a Galerkin approximation of miscible displacement in porous media”. In: *SIAM Journal on Numerical Analysis* 22.5 (1985), pp. 970–1013.

[40] Endre Süli. “Convergence and nonlinear stability of the Lagrange-Galerkin method for the Navier-Stokes equations”. In: *Numerische Mathematik* 53.4 (1988), pp. 459–483. DOI: 10.1007/BF01396329.

[41] Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Aug. 2003. ISBN: 9780511801181. DOI: 10.1017/CBO9780511801181.

[42] Geoffrey Ingram Taylor and Albert Edward Green. “Mechanism of the production of small eddies from large ones”. In: *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences* 158.895 (1937), pp. 499–521. DOI: 10.1098/rspa.1937.0036.

[43] R. Temam and M. Ziane. “Navier-Stokes equations in three-dimensional thin domains with various boundary conditions”. In: *Advances in Differential Equations* 1.4 (1996), pp. 499–546. DOI: ade/1366896027.

[44] Cornelis Vuik et al. *Numerical methods for ordinary differential equations*. 1st ed. Delft Academic Press, 2015, p. 123. ISBN: 9789065623737. DOI: s10915-004-4647-1.

[45] Hong Wang and Kaixin Wang. "Uniform Estimates of an Eulerian–Lagrangian Method for Time-Dependent Convection-Diffusion Equations in Multiple Space Dimensions". In: *SIAM Journal on Numerical Analysis* 48.4 (Jan. 2010), pp. 1444–1473. ISSN: 0036-1429. DOI: 10.1137/070682952.

[46] D. Xiu et al. "Strong and auxiliary forms of the semi-Lagrangian method for incompressible flows". In: *Journal of Scientific Computing* 25.1 (2005), pp. 323–346. ISSN: 08857474. DOI: 10.1007/s10915-004-4647-1.

[47] Dongbin Xiu and George Em Karniadakis. "A Semi-Lagrangian High-Order Method for Navier–Stokes Equations". In: *Journal of Computational Physics* 172.2 (Sept. 2001), pp. 658–684. ISSN: 00219991. DOI: 10.1006/jcph.2001.6847.