A levelset-based cut-cell method for interfacial flows: part 1-Navier–Stokes equations and Rayleigh–Bénard instability with melting boundary

Alejandro Quirós Rodríguez, Tomas Fullana, Vincent Le Chenadec, Taraneh Sayadi

▶ To cite this version:

Alejandro Quirós Rodríguez, Tomas Fullana, Vincent Le Chenadec, Taraneh Sayadi. A levelset-based cut-cell method for interfacial flows: part 1-Navier–Stokes equations and Rayleigh–Bénard instability with melting boundary. Acta Mechanica, 2024, 236 (9), pp.5763-5786. ⟨10.1007/s00707-024-04133-4⟩. ⟨hal-05452481⟩

HAL Id: hal-05452481
https://hal.science/hal-05452481v1

Submitted on 10 Jan 2026

A levelset based cut-cell method for interfacial flows. Part 1: Navier-Stokes equations and Rayleigh-Bénard instability with melting boundary

Alejandro Quirós Rodríguez[1*], Tomas Fullana[1,2],
Vincent Le Chenadec[3], Taraneh Sayadi[1,4]

[1*]Institut Jean le Rond d'Alembert, Sorbonne Université, CNRS, Paris, F-75005, France.
[2]Laboratory of Fluid Mechanics and Instabilities, EPFL, Laussane, CH-1015, Switzerland.
[3]MSME, Université Gustave Eiffel, UPEC, CNRS, Marne-la-Vallée, F-77454, France.
[4]Mathematical and Numerical Modelling Laboratory, Conservatoire National des Arts et Métiers, Paris, F-75003, France.

*Corresponding author(s). E-mail(s):
alejandro.quiros_rodriguez@etu.sorbonne-universite.fr;
Contributing authors: tomas.fullana@epfl.ch;
vincent.le-chenadec@univ-eiffel.fr; taraneh.sayadi@lecnam.net;

Abstract

The treatment of complex geometries in Computational Fluid Dynamics applications is a challenging endeavor, particularly when they couple the dynamics of two flow fields, as in multi-phase applications. Immersed boundary and cut-cell techniques can significantly simplify the solution workflow by alleviating the meshing process required by body-fitted meshes. However, these methods introduce new challenges, as the formulation of accurate and well-posed discrete operators becomes nontrivial. In this work, a conservative Cartesian cut-cell method is proposed for the solution of scalar transport equations with mixed boundary conditions, as well as the incompressible Navier-Stokes equations on staggered Cartesian grids. Given that they require constant updates for unsteady boundaries/interfaces and subsequent matrix assembly for the solution of linear systems, emphasis is placed on the simplicity of the operators by introducing a set of geometric moments easily computed from a levelset method and retaining a nearest-neighbor stencil. For stability and robustness, the discrete operators are designed to mimic the structure of the continuous ones. These properties are first exemplified on a Poisson equation with mixed boundary conditions: in the pure Dirichlet (Neumann, respectively) case, the discrete problem is shown to be symmetric positive definite (semi-definite, respectively), just like the continuous problem. An advection equation for a vector-valued field is also studied to highlight the skew-symmetry of the proposed discretization of the convective term. This property is required to achieve discrete kinetic equation conservation in the inviscid and incompressible Navier-Stokes equations. The accuracy and robustness of the method are then demonstrated with flows past a circular cylinder, as well as the numerical simulation of Rayleigh-Bénard instability in the presence of a melting boundary.

Keywords: Immersed Boundary Method, Cut-Cell Method, Incompressible Navier-Stokes Equations

1 Introduction

A vast range of flow phenomena are dominated by the dynamics that occur within the vicinity of fluid/solid and fluid/fluid interfaces. These include the viscous and pressure drag observed in external flows, the conjugate heat transfer blades are subjected to in gas turbines, the generation of vorticity in boundary layers and its subsequent impact on the turbulent mixing, and the heat and mass transfers in phase change applications, to name a few examples. The effect of the dynamics in the vicinity of boundaries/interfaces on the overall flow singles out the treatment of boundary conditions, a critical aspect that also represents a significant challenge for many Computational Fluid Dynamics (CFD) applications.

Many numerical methods have therefore been developed to address the treatment of boundary conditions on complex geometries. Unstructured techniques, as the name suggests, leverage meshes with arbitrary polyhedral elements that at least for piecewise planar cases conform to the geometry, at the cost of explicitly storing connectivity information. They are very effective and powerful to represent arbitrary geometries, and can even represent curved surface exactly [1], but the generation of high-quality unstructured meshes continues to be a challenging and time-consuming task. In addition, the design of efficient and robust numerical algorithms targeting such meshes remains an active area of research. Finally, explicit element connectivity effectively introduces an overhead that does not exist on structured meshes, and consequently increases the computational cost per grid point.

These limitations are one of two compelling arguments for the use of structured meshes, the second being the simplicity of the implementation of many algorithms on such meshes and their efficiency (often in constant time). The connectivity is implicit, which restricts their use to simple mesh topologies, including cylindrical or curvilinear ones. To circumvent this limitation, dedicated discretization techniques, referred to as immersed boundary methods (IBM), have been devised [2]. There exists various approaches to represent the boundary (diffuse or sharp) and to account for the mass and momentum transfers that occur along the solid boundary. The original IBM [3], which targeted cardiovascular flows, represented the boundary as a flexible elastic membrane, which enabled the explicit expression of the force exerted onto the flow. This approach however is not valid for rigid boundaries, for which a variety of techniques including Lagrange multiplier methods [4] were developed. A salient feature of these techniques is that they require no description of the boundary orientation. As a result, the traction exerted onto the boundary lumps the contributions from the fluid on either side. As recently highlighted in [5], this lack of distinction is reasonable when the body is an infinitely thin structure, but in the case of thick bodies the inability to single out the contributions from the intentional and superfluous regions can be a deal breaker.

A widespread flavor of the IBM, referred to as the cut-cell method, has been developed specifically for thick bodies and does not suffer from the above-mentioned flaw. Cut-cell methods for the simulation of scalar transport equations [6] and for the solution of viscous compressible flows on collocated Cartesian grids [7, 8], as well as for the solution of two-dimensional incompressible flows on staggered Cartesian grids [9], have been proposed. The combination of the cut-cell method with staggered arrangement (also referred to as Arakawa C grid [10]), adopted in the two-dimensional cut-cell method of Cheny and Botella [9], is a sensible choice for incompressible flows: it guarantees a strong coupling between the pressure and velocity variables [11], and can potentially conserve important physical invariants such as kinetic energy in the inviscid limit of the incompressible Navier-Stokes equations [12]. Preserving such properties in the presence of complex boundaries is however a challenge which, to the best of the authors' knowledge, is yet to be fulfilled. The formulation of a fully conservative (for primary and secondary integral quantities) cut-cell method for staggered mesh arrangements remains an outstanding issue, as recently highlighted by specialists in [13]:

> A few extensions of the MAC method to cut-cell meshes have been presented in the literature. In these methods different cut-cell configurations are treated case by case and a mass, momentum and energy conserving extension is deriving using, in 2D, a 5-point stencil and a finite-volume rationale. However, achieving a fully conservative method using the 5-point stencil is impossible. Moreover, an extension of the method to 3D is problematic due to the many possible cut-cell configurations. To our knowledge only a quasi-3D extension, where the immersed boundary is parallel to one of the Cartesian coordinate axes, has been published so far.

The proposed work attempts to fill this void, namely the formulation of a fully conservative cut-cell method that preserves the stencil structure of the MAC method. This formulation is intended to simplify the treatment of complex jump or boundary conditions such as the rate or slip laws encountered in the modelling of heat, mass and momentum transfers in the vicinity of boundaries or interfaces. These goals are achieved as follows.

Firstly, to simplify the treatment of the many possible cut-cell configurations referred to in the quote above, a minimal set of geometric moments that can be computed from any boundary representation (STL, CSG, levelset...) is identified. The rationale behind this identification is simply to guarantee first order accuracy in the vicinity of the boundary when the latter is mesh aligned. The introduction of this minimal set of geometric moments (presented in Sec. 2) simplifies the implementation: they are the only information required when modifying classical formulas based on second order interpolation and differentiation formulas (forward or backward) to account for the presence of the boundary. The proposed formulation also accommodates stretching, and away from boundaries well-known second-order formulas are also recovered.

Secondly, to accommodate arbitrary boundary conditions (*e.g.* Robin or Neumann), boundary values are included as part of the solution process. The closure is performed by discretising the boundary conditions themselves. This provides a flexible way to mix boundary conditions types, without modifying the spatial operators themselves. The downside of this approach is of course an increase in the size of the linear systems to be inverted, that are themselves not necessarily diagonally dominant, even in the simple case of a scalar transport equation. The resulting matrices have however full rank (with the exception of constant null spaces in specific but well-known cases, such as purely periodic or Neumann boundary conditions) and direct or iterative methods can be used. For the latter, appropriate preconditioning techniques can be used to accelerate the convergence [14, 15]).

This first segment of a two-part contribution [16] underscores the following methodological aspects. A set of geometric moments, computed from a levelset function, and linear operators are introduced (Sec. 2) and employed to discretize both scalar transport (Sec. 3) and incompressible Navier-Stokes equations (Sec. 4 and 5) in the presence of a boundary (an iso-contour of the levelset function). The exposition of the levelset update and reinitialization methodology is briefly mentioned in Sec. 2.1, but the full exposition is deferred to Part 2. The methodology is demonstrated to accommodate mixed boundary conditions for scalar transport equations without the need for additional operators. It is first validated on an aerodynamics application (flow around a cylinder), which is standard in the development of IBMs. Following this, Rayleigh-Bénard instability in the presence of a melting boundary is numerically investigated in order to highlight the benefits of the method as a standalone candidate to discretize both bulk equations (two-phase heat and single-phase fluid flows) and boundary conditions (including Stefan condition).

The methodology presented in this segment will be expanded to two levelset functions for interfacial flows around arbitrarily solid surfaces in Part 2. This subsequent segment will further elucidate how this extension is employed to generalize the Navier-Stokes solver to the mixed boundary conditions required to accurately solve free surface and dynamic contact line dynamics.

2 Definition and notation

In this section, an introduction to the notation and a definition of the geometric moments and the elementary operators that will be used to define the discrete cut-cell operators in Secs. 3 and 4 is given.

2.1 The levelset Method

In this work, a levelset function ϕ [17] is defined on the computational domain Ω to map the locus of one of its iso-levels $(\{\boldsymbol{x} \in \Omega | \, \phi(\boldsymbol{x}, t) = \phi_0\})$ to an interface $\Gamma(t)$ that separates two non-overlapping domains, $\Omega_1(t)$ and $\Omega_2(t)$, each occupied by a different phase. The value ϕ is defined as the signed distance to the interface,

$$\phi(\boldsymbol{x}, t) = \begin{cases} -d(\boldsymbol{x}, \Gamma(t)), \, \boldsymbol{x} \in \Omega_1(t) \\ 0, \, \boldsymbol{x} \in \Gamma(t) \\ d(\boldsymbol{x}, \Gamma(t)), \, \boldsymbol{x} \in \Omega_2(t) \end{cases}, \tag{1}$$

where $d(\boldsymbol{x}, \Gamma(t))$ denotes the minimal distance between the point $\boldsymbol{x}$ and the interface $\Gamma(t)$,

$$d(\boldsymbol{x}, \Gamma(t)) = \operatorname{argmin}_{\boldsymbol{y} \in \Gamma(t)} \|\boldsymbol{x} - \boldsymbol{y}\|, \tag{2}$$

with $\|\cdot\|$ denoting the Euclidean distance. Fig. 1 shows an example of an arbitrary interface Γ defined by a levelset function ϕ.

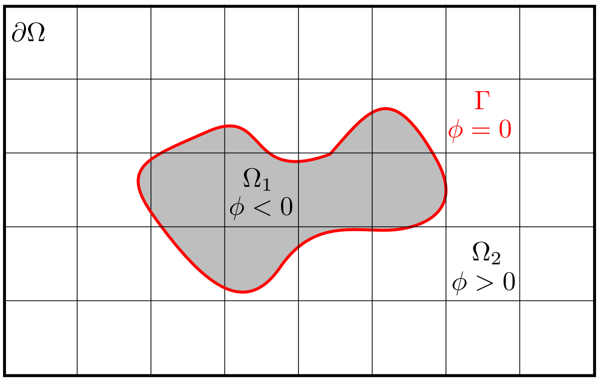


Fig. 1: Levelset defining an interface Γ separating two phases Ω_1 and Ω_2.

The main advantage of the levelset method is that its evolution is governed by a simple evolution equation (Eq. (62)), provided a velocity field can be smoothly extended from $\Gamma(t)$. This evolution equation however does not preserve the signed distance property, which can be restored by periodically employing a reinitialization procedure.

As highlighted in the introduction, the numerical details of the levelset update and reinitialization methodologies are exposed in Part 2 [16].

2.2 Geometric moments

The mesh is assumed to be rectilinear with n_x points along x and n_y along y. For wet and partially wet cells numbered i along x and j along y, the coordinates of the fluid center of mass are denoted $x^{\omega}_{i,j}$ and $y^{\omega}_{i,j}$ and represent the components of two

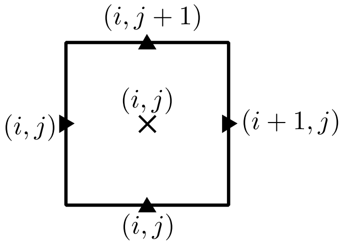


Fig. 2: Numbering convention.

vectors of size $n_x n_y$ (denoted x^ω and y^ω). Likewise, the vectors x^γ and y^γ, of size $n_x n_y$, store the coordinates of the center of mass of the boundary of partially wet cells (referred to as boundary cells).

In the following, discrete values can be cell-, face- or node-centered, therefore a numbering convention must be adopted for the indices (i, j). Fig. 2 showcases the convention used in this work for both cell and face-centered quantities (the (i, j) node-centered quantities is located at the bottom-left of the cell).

Cut-cell methods are firmly grounded in the Finite Volume Method, which defines the primary discrete variables as cell-wise averages over mesh elements. The design of the Finite Volume operators is then based on the application of the Divergence theorem. For example, given a scalar field p, this theorem states that in a Cartesian coordinate system, the x component of the gradient $\boldsymbol{q} \equiv \nabla p$ averaged over a cell Ω may be computed as

$$\Omega q_x = \int_\Omega \frac{\partial p}{\partial x} \mathrm{d}V = \oint_{\partial\Omega} p \boldsymbol{e}_x \cdot \mathrm{d}\boldsymbol{S} \tag{3}$$

where $\mathrm{d}\boldsymbol{S}$ denotes the outward-pointing surface element and $\boldsymbol{e}_x$ the unit vector along the x direction.

For the sake of presentation, the case displayed in Fig. 3 is considered where $\Omega \equiv \mathcal{V}_{i,j}$ consists of the intersection of a phase domain and a computational cell (a right hexahedron). The contour $\partial\Omega$ then consists of the union of the four planar faces $\mathcal{A}_{x,i,j}$, $\mathcal{A}_{x,i+1,j}$, $\mathcal{A}_{y,i,j}$ and $\mathcal{A}_{y,i,j+1}$ as well as the boundary surface $\Gamma_{i,j}$. A piece-wise linear approximation of $\Gamma_{i,j}$, denoted $\widetilde{\Gamma}_{i,j}$ and of unit normal $(n_{x,i,j}, n_{y,i,j})$, can be defined by applying Eq. (3) to $\mathcal{V}_{i,j}$ with $p = 1$, yielding

$$\int_{\widetilde{\Omega}} \frac{\partial 1}{\partial x} \mathrm{d}V = \mathcal{A}_{x,i+1,j} - \mathcal{A}_{x,i,j} + n_{x,i,j} \widetilde{\Gamma}_{i,j} = 0$$

which highlights the existence of a fundamental relation

$$\mathcal{A}_{x,i+1,j} - \mathcal{A}_{x,i,j} = -n_{x,i,j} \widetilde{\Gamma}_{i,j} \tag{4}$$

sometimes referred to as a Geometric Surface Conservation Law. The same can be done in the y-direction in order to obtain a relation between $\mathcal{A}_{y,i,j+1}, \mathcal{A}_{y,i,j}, n_{y,i,j}$ and $\widetilde{\Gamma}_{i,j}$. Note that to simplify the exposition, the notations ($\mathcal{A}_{x,i,j}$, $\mathcal{V}_{i,j}$...) are used to denote both a region in space and its measure (length or area, in 2D).

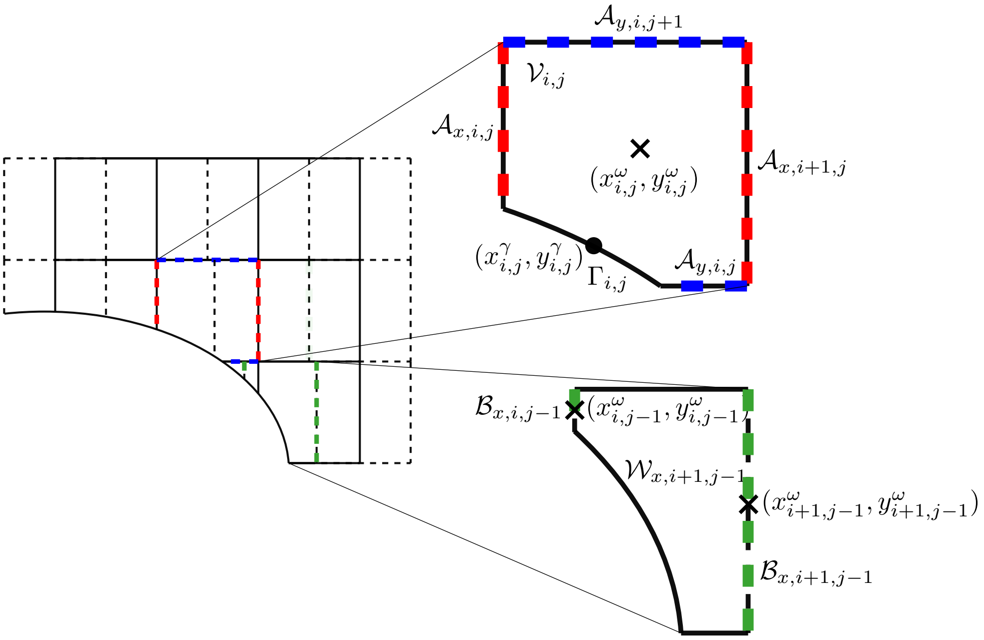


Fig. 3: Geometric moments. $\mathcal{A}_x$ in red dashed lines, $\mathcal{A}_y$ in blue dashed lines, $\mathcal{B}_x$ in green dashed lines. $\mathcal{B}_y$ are not shown.

The one-dimensional vectors containing the geometric information, $\mathcal{A}_{x,i,j}$, $\mathcal{A}_{y,i,j}$ and $\mathcal{V}_{i,j}$ are row based, meaning they are structured as $\mathcal{A}_x = \left[\mathcal{A}_{x,1,1}, \dots, \mathcal{A}_{x,n_x+1,1}, \dots, \mathcal{A}_{x,1,n_y}, \dots, \mathcal{A}_{x,n_x+1,n_y}\right]$, $\mathcal{A}_y = \left[\mathcal{A}_{y,1,1}, \dots, \mathcal{A}_{y,n_x,1}, \dots, \mathcal{A}_{y,1,n_y+1}, \dots, \mathcal{A}_{y,n_x,n_y+1}\right]$ and $\mathcal{V} = \left[\mathcal{V}_{1,1}, \dots, \mathcal{V}_{n_x,1}, \dots, \mathcal{V}_{1,n_y}, \dots, \mathcal{V}_{n_x,n_y}\right]$. Since the surface moments $\mathcal{A}_x$ and $\mathcal{A}_y$ are staggered, the size of the corresponding vectors is $n_x + 1 \times n_y$ and $n_x \times n_y + 1$ respectively, whereas the size of the volume moment vector is $n_x \times n_y$. The vectors are used to define three diagonal matrices as follows:

$$A_x = \operatorname{diag}\left(\mathcal{A}_x\right), \quad A_y = \operatorname{diag}\left(\mathcal{A}_y\right), \quad V = \operatorname{diag}\left(\mathcal{V}\right). \tag{5}$$

At each cell centroid $\left(x^{\omega}_{i,j}, y^{\omega}_{i,j}\right)$, two additional planar faces $\mathcal{B}_{x,i,j}$ and $\mathcal{B}_{y,i,j}$ are defined. The planar faces $\mathcal{B}_x$, $\mathcal{A}_y$ and the boundary surface Γ define the staggered volumes $\mathcal{W}_x$. Likewise, the planar faces $\mathcal{B}_y$, $\mathcal{A}_x$ and the boundary surface Γ define the staggered volumes $\mathcal{W}_y$. For the sake of presentation, the staggered volume $\mathcal{W}_{x,i+1,j-1}$ is shown in Fig. 3. These additional geometric moments define an additional set of diagonal matrices

$$B_x = \operatorname{diag}\left(\mathcal{B}_x\right), \quad B_y = \operatorname{diag}\left(\mathcal{B}_y\right) \quad \text{and} \quad W = \begin{bmatrix} \operatorname{diag}\left(\mathcal{W}_x\right) & \\ & \operatorname{diag}\left(\mathcal{W}_y\right) \end{bmatrix}. \tag{6}$$

All the geometric information, such as surface and volume moments and cell centroids can be computed from the levelset function. To than end, the marching squares algorithm [18] can be used. Depending on the relative position of the interface, sixteen possible cases (corresponding to sixteen different iso-values) can be identified in two-dimensions using the levelset values at the cell corners. Once the case is identified, the position where the interface intersects the cell faces can be determined, allowing all the vertices defining the wet cell to be known. The position of the vertices suffices to compute all the geometric information.

2.3 Elementary operators

To formulate non-local operations, the proposed method leverages the classical backward and forward differentiation matrices

$$\Delta^- = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & 0 \end{bmatrix} \quad \text{and} \quad \Delta^+ = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & 0 \end{bmatrix}$$

as well as backward and forward interpolation matrices

$$\Sigma^- = \frac{1}{2} \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & 1 & 1 & \\ & & & 1 & 0 \end{bmatrix} \quad \text{and} \quad \Sigma^+ = \frac{1}{2} \begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ & & & & 0 \end{bmatrix}.$$

All of these matrices can be made circulant to accommodate periodic boundary conditions (see A).

Note that the following identities also hold in the circulant case

$$(\Delta^-)^\top = -\Delta^+ \quad \text{and} \quad (\Sigma^-)^\top = \Sigma^+, \tag{7}$$

where $\top$ denotes the transpose.

Let I_x denote the $n_x \times n_x$ identity matrix, Δ_x^- and Δ_x^+ the $n_x \times n_x$ backward and forward differentiation matrices, and Σ_x^- and Σ_x^+ the $n_x \times n_x$ backward and forward interpolation matrices, with extensions to the y direction following. The multi-dimensional extension is performed using the Kronecker product

$$D_x^\pm = I_y \otimes \Delta_x^\pm \quad \text{and} \quad D_y^\pm = \Delta_y^\pm \otimes I_x \tag{8}$$

and

$$S_x^\pm = I_y \otimes \Sigma_x^\pm \quad \text{and} \quad S_y^\pm = \Sigma_y^\pm \otimes I_x. \tag{9}$$

Extension is to three spatial dimensions is straightforward, see B. As done previously for the unidimensional operators (Eq. (7)),

$$\left(D_x^-\right)^\top = -D_x^+ \quad \text{and} \quad \left(S_x^-\right)^\top = S_x^+, \tag{10}$$

with similar properties holding for the direction y.

Rescaling of the operators using volume moments is sometimes required. Unfortunately, these diagonal matrices have zero diagonal coefficients in the cells that are empty. To reflect this, the Moore-Penrose pseudoinverse notation (†) is used instead of the usual -1 superscript. In practice, the empty coefficients are simply overwritten with ones.

3 Poisson problem

This section presents the definitions of the discrete cut-cell operators required to solve Poisson-like problems with generalized boundary conditions. The convergence order of the constructed discrete operators are then demonstrated using the method of manufactured solutions.

3.1 First-order operators

The expressions presented below adopt the numbering convention delineated in Sec. 2.2. The formulas will need updating if a different choice is made. All the discrete operators defined in this section preserve constant states, or so-called free-streaming condition [19].

3.1.1 Gradient of a cell-centered scalar field

Let p^ω denote a discrete scalar field, and p^γ its values along the immersed boundary. Let its gradient be defined as the following bilinear map

$$\operatorname{grad}\left(p^\omega, p^\gamma\right) = W^\dagger\left(G p^\omega + H p^\gamma\right), \tag{11}$$

where

$$G = \begin{bmatrix} D_x^- B_x \\ D_y^- B_y \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} A_x D_x^- - D_x^- B_x \\ A_y D_y^- - D_y^- B_y \end{bmatrix}. \tag{12}$$

As previously stated, the discrete gradient operator preserves constant states, therefore for $p^\omega = p^\gamma = c$, $\operatorname{grad}(c, c) = \mathbf{0}$ in the whole domain.

3.1.2 Divergence of a face-centered vector field

Let $\boldsymbol{q}^\omega = \left(q_x^\omega, q_y^\omega\right)$ denote a face centered vector field in the fluid domain, and likewise $\boldsymbol{q}^\gamma = \left(q_x^\gamma, q_y^\gamma\right)$ its values along the immersed boundary. Let the volume-integrated divergence operator be defined the as the bilinear map

$$\operatorname{div}\left(\boldsymbol{q}^\omega, \boldsymbol{q}^\gamma\right) = -\left(G^\top + H^\top\right)\boldsymbol{q}^\omega + H^\top \boldsymbol{q}^\gamma, \tag{13}$$

where

$$-\left(G^{\top}+H^{\top}\right)=\left[\begin{array}{cc} D_x^{+}A_x & D_y^{+}A_y \end{array}\right]$$

$$H^{\top}=\left[\begin{array}{cc} B_xD_x^{+}-D_x^{+}A_x & B_yD_y^{+}-D_y^{+}A_y \end{array}\right].$$

It should be noted that if the vector field represents a gradient, which is a higher order quantity, no distinction is made between its values in the fluid and at the boundary, i.e. $\boldsymbol{q}^{\gamma}$ is simply set to $\boldsymbol{q}^{\omega}$, simplifying the divergence operator to

$$\operatorname{div}\left(\boldsymbol{q}^{\omega}\right)=-G^{\top}\boldsymbol{q}^{\omega}. \tag{14}$$

Eq. (10) can then be used to show that

$$-G^{\top}=\left[\begin{array}{cc} B_xD_x^{+} & B_yD_y^{+} \end{array}\right].$$

Nevertheless, both terms are retained in the definition of the divergence (Eq. (13)) to separate the fluid and boundary contributions, as exemplified in the solution of the Poisson problem with Robin boundary conditions below.

As in the case of the gradient, the discrete divergence operator also preserves constant states, therefore for $\boldsymbol{q}^{\omega}=\boldsymbol{q}^{\gamma}=\boldsymbol{c}$, $\operatorname{div}(\boldsymbol{c},\boldsymbol{c})=0$ in the whole domain.

3.2 Second-order operators

The objective of the following section is to specify the treatment of boundary conditions for second-order operators (*e.g.* Laplacian) by composing the first-order operators (Eqs. (11) and (13)) and leveraging the distinction between fluid and boundary values (ω and γ superscripts, respectively). The Poisson equation is considered for this purpose

$$-\nabla\cdot\nabla p=f \quad \text{in} \quad \Omega. \tag{15}$$

It will be demonstrated that the proposed method flexibly accommodates both Neumann and Robin boundary conditions.

3.2.1 Laplacian with Robin boundary conditions

Let us consider the Robin boundary condition shown below

$$ap+b\partial_n p=g \quad \text{on} \quad \Gamma, \tag{16}$$

where a and b are two scalar coefficients. The scalar value along the immersed boundary (p^{γ}) is not explicitly known, and has to be inferred from the values of the normal component of the gradient along the immersed boundary and g. The discretization of this equation requires the definition of the boundary vectors g^{γ}, a^{γ} and b^{γ} for boundary cells, whose components are set to,

$$\begin{aligned} g_{ij}^{\gamma}&=g\left(x_{ij}^{\gamma},y_{ij}^{\gamma}\right),\\ a_{ij}^{\gamma}&=a\left(x_{ij}^{\gamma},y_{ij}^{\gamma}\right),\\ b_{ij}^{\gamma}&=b\left(x_{ij}^{\gamma},y_{ij}^{\gamma}\right), \end{aligned} \tag{17}$$

where a^γ and b^γ define the diagonal matrices I_a and I_b,

$$I_a = \operatorname{diag}\left(a^\gamma\right) \quad \text{and} \quad I_b = \operatorname{diag}\left(b^\gamma\right). \tag{18}$$

Likewise, the components of the discrete load f^ω are defined as

$$f_{ij}^\omega = f\left(x_{ij}^\omega, y_{ij}^\omega\right) \tag{19}$$

in fluid cells and zero elsewhere.

In order to discretize Eqs. (15) and (16), the term with the derivative in the normal direction ($\partial_n p$) is set as the boundary contribution to the divergence. The resulting system is given by

$$\begin{cases} -\operatorname{div}\left(\boldsymbol{q}^\omega,\, \boldsymbol{q}^\gamma\right) = V f^\omega, \\ I_a I_\Gamma p^\gamma - I_b \operatorname{div}\left(\boldsymbol{0},\, \boldsymbol{q}^\omega\right) = I_\Gamma g^\gamma, \\ \boldsymbol{q}^\omega = \operatorname{grad}\left(p^\omega,\, p^\gamma\right), \\ \boldsymbol{q}^\gamma = \boldsymbol{q}^\omega, \end{cases} \tag{20}$$

where I_Γ measures the boundary within a given cell and is defined as

$$I_\Gamma = \operatorname{diag}\left(\left|H^\top \boldsymbol{1}\right|\right). \tag{21}$$

I_Γ is needed in order to have a dimensionally consistent equation for the boundary conditions (Eq. (16)). The intermediate variables $\boldsymbol{q}^\omega$ and $\boldsymbol{q}^\gamma$ can be eliminated from the above linear system, which can then be expressed in terms of the previously defined matrices as

$$\begin{bmatrix} G^\top W^\dagger G & G^\top W^\dagger H \\ I_b H^\top W^\dagger G & I_b H^\top W^\dagger H + I_a I_\Gamma \end{bmatrix} \begin{bmatrix} p^\omega \\ p^\gamma \end{bmatrix} \simeq \begin{bmatrix} V f^\omega \\ I_\Gamma g^\gamma \end{bmatrix}, \tag{22}$$

by eliminating all intermediate variables and using Eqs. (11) and (14). The matrix on the left-hand side is symmetric if the coefficients b are equal to one and it is straightforward to obtain a symmetric matrix if they are not. App. C presents the resulting expression for a given cell of the one-dimensional Poisson's equation, for the sake of clarity.

The proposed method defines the boundary values p^γ implicitly: as highlighted in the introduction, the downside is the increase in matrix size and potential deterioration in conditioning, the upside being that the boundary condition is satisfied discretely and no interpolation is needed from the bulk field to infer values along the boundaries. As stated in [20], the application of Robin-like boundary conditions for vector-valued fields is more complicated than its application to scalar fields. In Part 2, we present its implementation to the velocity field in the context of moving contact lines, where a Navier-slip boundary condition is imposed.

The Neumann and Dirichlet boundary conditions can then be considered as a special case of this generalized boundary condition. There are explicitly presented below.

Dirichlet boundary condition

In the case of Dirichlet boundary condition, Eq. (16) is simplified to

$$p = g \quad \text{on} \quad \Gamma, \tag{23}$$

by setting $a = 1$ and $b = 0$.

Eqs. (15) and (23) are then discretized as

$$\begin{cases} -\operatorname{div}(\boldsymbol{q}^\omega, \boldsymbol{q}^\gamma) = V f^\omega, \\ p^\gamma = g^\gamma, \\ \boldsymbol{q}^\omega = \operatorname{grad}(p^\omega, p^\gamma), \\ \boldsymbol{q}^\gamma = \boldsymbol{q}^\omega, \end{cases} \tag{24}$$

which in terms of the linear operators previously defined yields,

$$G^\top W^\dagger G p^\omega = V f^\omega - G^\top W^\dagger H g^\gamma. \tag{25}$$

Eq. (25) therefore represents the discretization of the Poisson equation (15) with Dirichlet condition (23). The matrix on the left-hand side is symmetric and the two terms on the right-hand side correspond to the load and boundary contributions, respectively. Since the boundary values are explicitly given by the boundary conditions, there is no need to solve the coupled system (22).

From Eq. (25), it is clear that a cell is (partially) filled if the corresponding columns of V and G are non-zero. This algebraic definition is consistent with the geometric one. Furthermore, the first condition is sufficient: if an element of V is non-zero, then geometrically (Sec. 2.2), the corresponding coefficients of B_x and B_y are also non-zero. But as seen from the definition (12), B_x and B_y are precisely the right-most factors of the G-blocks, and the corresponding column will therefore necessarily vanish.

Neumann boundary conditions

Neumann boundary conditions are obtained by setting $a = 0$ and $b = 1$ in Eq. (16), which gives

$$\partial_n p = g \quad \text{on} \quad \Gamma. \tag{26}$$

Eqs. (15) and (26) are discretized following

$$\begin{cases} -\operatorname{div}(\boldsymbol{q}^\omega, \boldsymbol{q}^\gamma) = V f^\omega, \\ -\operatorname{div}(0, \boldsymbol{q}^\gamma) = I_\Gamma g^\gamma, \\ \boldsymbol{q}^\omega = \operatorname{grad}(p^\omega, p^\gamma), \\ \boldsymbol{q}^\gamma = \boldsymbol{q}^\omega \end{cases} \tag{27}$$

and in terms of the linear operators

$$\begin{bmatrix} G^\top W^\dagger G & G^\top W^\dagger H \\ H^\top W^\dagger G & H^\top W^\dagger H \end{bmatrix} \begin{bmatrix} p^\omega \\ p^\gamma \end{bmatrix} \simeq \begin{bmatrix} V f^\omega \\ I_\Gamma g^\gamma \end{bmatrix}. \tag{28}$$

Eq. (28) therefore represents the discretization of the Poisson equation (15) with Neumann boundary conditions (26). The matrix on the left-hand side is also symmetric, and the two terms on the right-hand side correspond to the load and boundary contributions, respectively.

3.3 Treatment of small cells

An inherent problem of cut-cell methods is the presence of small cells that increase the condition number of the system matrix, which need to be removed. Several methods can be found in the literature that tackle this problem, including cell merging [21], cell linking [22] and mixed approaches [23].

In this work, a simpler approach which consists in the clipping of the small/large cells is employed. To that end, a threshold ϵ is defined and the volume and surface moments are clipped according to

$$\mathcal{V}_{i,j} = \begin{cases} 0 & \text{if } \mathcal{V}_{i,j} < \epsilon, \\ \Delta x \Delta y & \text{if } \mathcal{V}_{i,j} > \Delta x \Delta y - \epsilon, \\ \mathcal{V}_{i,j} & \text{else}, \end{cases} \tag{29}$$

$$\mathcal{A}_{\alpha,i,j} = \begin{cases} 0 & \text{if } \mathcal{A}_{\alpha,i,j} < \sqrt{\epsilon}, \\ \Delta \alpha & \text{if } \mathcal{A}_{\alpha,i,j} > \Delta \alpha - \sqrt{\epsilon}, \quad \forall \alpha \in \{x, y\}\,, \\ \mathcal{A}_{\alpha,i,j} & \text{else}, \end{cases} \tag{30}$$

where Δx and Δy are the cell width and height, respectively. Whenever a cell is clipped, the corresponding volume and surface moments are clipped accordingly, depending on which condition has been satisfied.

The clipping procedure induces changes in the volumes of the phases under study. In the case of stationary boundaries, the piece-wise linear approximation of the interface has a larger effect than the clipping procedure. In the case of moving boundaries, small oscillations appear in the volume as the relative position of the interface with respect to the cells changes. However, it has been demonstrated that the leading source of volume conservation errors is the interface representation and errors related to clipping are negligible. Further comments on this can be found in the companion paper (Part 2) [16].

3.4 Numerical tests

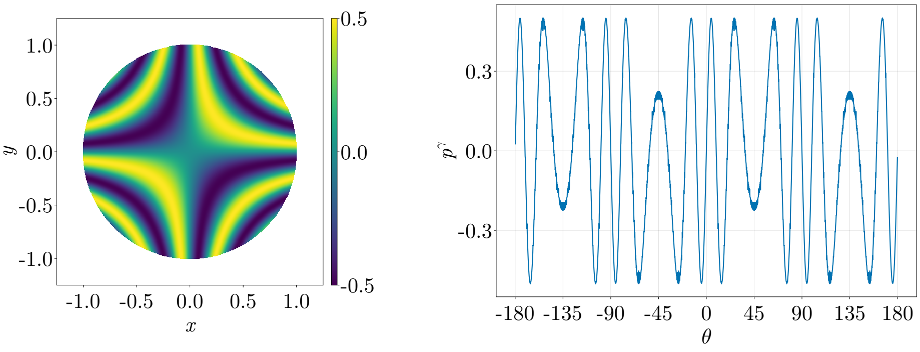
The method of manufactured solutions is employed in this section to compare analytical solutions of the Poisson equation (15) with the results obtained using the discrete cut-cell operators. The scalar function p is chosen as follows

$$p\,(x, y) = \cos\left(\pi^2 xy\right) \sin\left(\pi^2 xy\right),$$

and its Laplacian is given by

$$\Delta p\,(x, y) = -4\pi^4 \cos\left(\pi^2 xy\right) \sin\left(\pi^2 xy\right) \left(x^2 + y^2\right).$$

Eq. (15) is solved inside a unit circle by setting f^ω to the exact values of the Laplacian calculated at the cell centroids $(x^\omega_{ij}, y^\omega_{ij})$ and g^γ to the respective analytical value at the boundary centers $(x^\gamma_{ij}, y^\gamma_{ij})$. The L_1, L_2 and L_∞ norms of the errors using Dirichlet, Neumann and Robin boundary conditions for $n = [64, 128, 256, 512]$ points per dimension are computed in order to obtain the convergence rates for each case. Fig. 4 shows an example of the result obtained with Robin boundary conditions and 512 points. As explained before, both the discrete bulk and boundary fields are obtained as part of the solution and both are displayed. Fig. 4 (b) shows small oscillations in the boundary field p^γ, which are expected since the stencils employed for the definition of the gradient and divergence operators only use two points. They have also been observed in the vicinity of the boundary in previous works using cut-cell methods, see for instance [24]. In our case, the oscillations are confined to the regions where the orientations of the cut-cells are close to $\{\pm 45°, \pm 135°\}$, which are the most adverse regions, with a maximum relative error of a 3%. Far from these regions the oscillations vanish. This is a promising result for phase-change problems where accurate representations of the boundary values are essential. Since no interpolation is needed in our methodology to obtain the boundary values, one source of error is removed. To assure that these oscillations are due to the cell orientations, the Poisson equation has also been solved for an axisymmetric problem ($p(x, y) = x^2 + y^2$) applying Robin boundary conditions. The boundary field p^γ is shown in Fig. 5, where one quarter of the boundary field is presented. The oscillations vanish for mesh oriented cells and increase as they get close to ±45°, as in the previous example, with a maximum relative error of a 0.4%. The relative error is larger in the more complex case because of the curvature of the solution.



(a) Bulk field p^ω.

(b) Boundary field p^γ.

Fig. 4: Numerical result using Robin boundary conditions and 512 points per dimension.

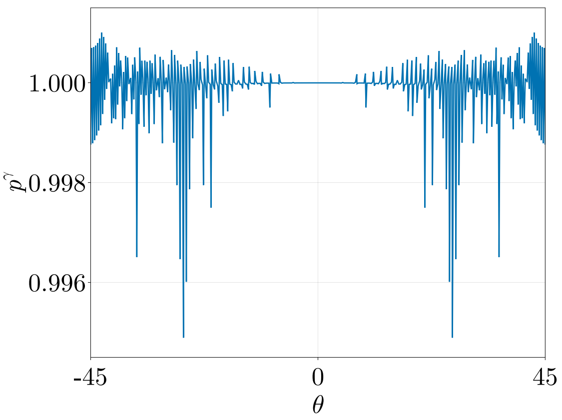


Fig. 5: Boundary field p^γ for an axisymmetric problem with Robin boundary conditions.

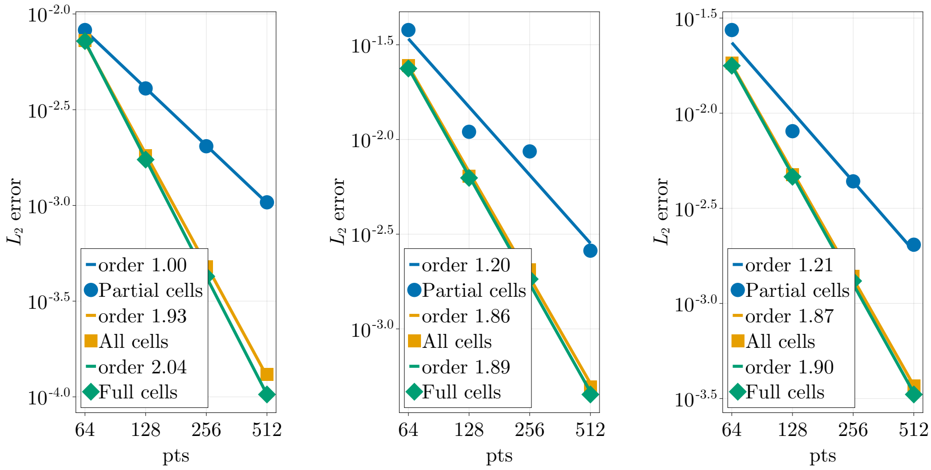
The volume-integrated p-norm of the error e^ω is defined as

$$\|e\|_p = \left(\int_\Omega |e^\omega|^p \mathrm{d}\Omega \middle/ \Omega \right)^{1/p}, \tag{31}$$

where $e^\omega_{ij} = p^\omega_{ij} - p\left(x^\omega_{ij}, y^\omega_{ij}\right)$. Tab. 1 shows the convergence rates for all the norms and boundary conditions for partial, full and all cells; whereas Fig. 6 presents the L_2 error for all the boundary conditions analyzed. It can be seen that the convergence rate for the L_1 and L_2 norm in partial cells is slightly higher in the Neumann and Robin cases whereas for the L_∞ is roughly the same in all the cases. In full cells, all the convergence rates stay close to order 2 except in the L_∞ norm with Neumann boundary conditions. In the whole domain, the convergence rates for L_1 and L_2 are close to those of the full cells, given that the amount of mixed cells is small. In the case of the L_∞, the convergence rates are the same than for the mixed cells. The second-order convergence in all cells is similar to the convergence rates found in other works using cut-cell methods.

	L_1 norm			L_2 norm			L_∞ norm		
BC	Partial	Full	All	Partial	Full	All	Partial	Full	All
Dirichlet	0.92	2.04	2.01	1.00	2.04	1.93	0.88	1.84	0.88
Neumann	1.54	1.93	1.93	1.20	1.89	1.86	0.84	1.62	0.84
Robin	1.47	1.94	1.94	1.21	1.90	1.87	0.78	1.72	0.78

Table 1: Convergence rates for different norms and types of boundary conditions.



(a) Dirichlet boundary conditions.

(b) Neumann boundary conditions.

(c) Robin boundary conditions.

Fig. 6: L_2 norm convergence.

While the results presented here are all for a fixed geometry, the method is easily extendable to moving geometries. This extension has been discussed and validated for the heat equation with moving geometries in the context of the Stefan problem by [25].

4 Advection equation of a vector-valued field

Towards the Navier-Stokes equation, the following section focuses on the discretization of the convective transport of a vector field $\boldsymbol{q}$ by a flow velocity field $\boldsymbol{u}$, which can be rewritten in conservative form if $\boldsymbol{u}$ is divergence-free as

$$(\boldsymbol{u} \cdot \nabla)\, \boldsymbol{q} = \nabla \cdot (\boldsymbol{q} \otimes \boldsymbol{u}) - (\nabla \cdot \boldsymbol{u})\, \boldsymbol{q} = \nabla \cdot (\boldsymbol{q} \otimes \boldsymbol{u})\,, \tag{32}$$

which in discrete form can be defined as

$$\begin{aligned} \mathrm{conv}_x\left(\boldsymbol{u}^{\omega_i}, \boldsymbol{u}^{\gamma}, \boldsymbol{q}^{\omega_i}, \boldsymbol{q}^{\gamma}\right) &= C_x\left(\boldsymbol{u}^{\omega_i}\right) q_x^{\omega_i} - \frac{K_x\left(\boldsymbol{u}^{\gamma}\right) q_x^{\omega_i} + K_x\left(\boldsymbol{q}^{\gamma}\right) u_x^{\omega_i}}{2}, \\ \mathrm{conv}_y\left(\boldsymbol{u}^{\omega_i}, \boldsymbol{u}^{\gamma}, \boldsymbol{q}^{\omega_i}, \boldsymbol{q}^{\gamma}\right) &= C_y\left(\boldsymbol{u}^{\omega_i}\right) q_y^{\omega_i} - \frac{K_y\left(\boldsymbol{u}^{\gamma}\right) q_y^{\omega_i} + K_y\left(\boldsymbol{q}^{\gamma}\right) u_y^{\omega_i}}{2}, \end{aligned} \tag{33}$$

where $\boldsymbol{q}^\omega$, $\boldsymbol{q}^\gamma$, $\boldsymbol{u}^\omega$ and $\boldsymbol{u}^\gamma$ denote face-centered discrete vector fields and

$$\begin{aligned}
C_x\left(\boldsymbol{u}^{\omega_i}\right) &= D^+_{x,x}\operatorname{diag}\left(S^-_{x,x}A_x u^{\omega_i}_x\right)S^-_{x,x} + D^+_{x,y}\operatorname{diag}\left(S^-_{x,x}A_y u^{\omega_i}_y\right)S^-_{x,y},\\
K_x\left(\boldsymbol{u}^\gamma\right) &= \operatorname{diag}\left(S^+_{x,x}H^\top \boldsymbol{u}^\gamma\right),\\
C_y\left(\boldsymbol{u}^{\omega_i}\right) &= D^+_{y,x}\operatorname{diag}\left(S^-_{y,y}A_x u^{\omega_i}_x\right)S^-_{y,x} + D^+_{y,y}\operatorname{diag}\left(S^-_{y,y}A_y u^{\omega_i}_y\right)S^-_{y,y},\\
K_y\left(\boldsymbol{u}^\gamma\right) &= \operatorname{diag}\left(S^+_{y,y}H^\top \boldsymbol{u}^\gamma\right).
\end{aligned} \tag{34}$$

The definition of the convective term is based on the skew-symmetric convective operator given in [12] for finite differences, which has been modified to work with the cut-cell method. The operators acting on the bulk field are skew-symmetric, as required to have a conservative discretization of the Navier-Stokes equation. Eq. (33) can be rearranged to show the skew-symmetric operator

$$\begin{aligned}
\operatorname{conv}_x\left(\boldsymbol{u}^{\omega_i},\boldsymbol{u}^\gamma,\boldsymbol{q}^{\omega_i},\boldsymbol{q}^\gamma\right) &= \underbrace{\left[C_x\left(\boldsymbol{u}^{\omega_i}\right)-\frac{1}{2}K_x\left(\boldsymbol{u}^\gamma\right)\right]}_{\text{Skew-symmetric}} v^{\omega_i}_x - \frac{1}{2}K_x\left(\boldsymbol{q}^\gamma\right)u^{\omega_i}_x,\\
\operatorname{conv}_y\left(\boldsymbol{u}^{\omega_i},\boldsymbol{u}^\gamma,\boldsymbol{q}^{\omega_i},\boldsymbol{q}^\gamma\right) &= \underbrace{\left[C_y\left(\boldsymbol{u}^{\omega_i}\right)-\frac{1}{2}K_y\left(\boldsymbol{u}^\gamma\right)\right]}_{\text{Skew-symmetric}} v^{\omega_i}_y - \frac{1}{2}K_y\left(\boldsymbol{q}^\gamma\right)u^{\omega_i}_y.
\end{aligned} \tag{35}$$

In the following, whenever $\boldsymbol{q} = \boldsymbol{u}$, the convective term will be referred as $\operatorname{conv}_\alpha\left(\boldsymbol{u}^\omega,\boldsymbol{u}^\gamma\right)\ \forall\ \alpha \in \{x,y\}$. The discrete convective operator also exhibits the free-streaming condition, hence for $\boldsymbol{u}^\omega = \boldsymbol{u}^\gamma = \boldsymbol{c}$, $\operatorname{conv}_x\left(\boldsymbol{c},\boldsymbol{c}\right) = 0$ and $\operatorname{conv}_y\left(\boldsymbol{c},\boldsymbol{c}\right) = 0$.

4.1 Momentum and kinetic energy conservation

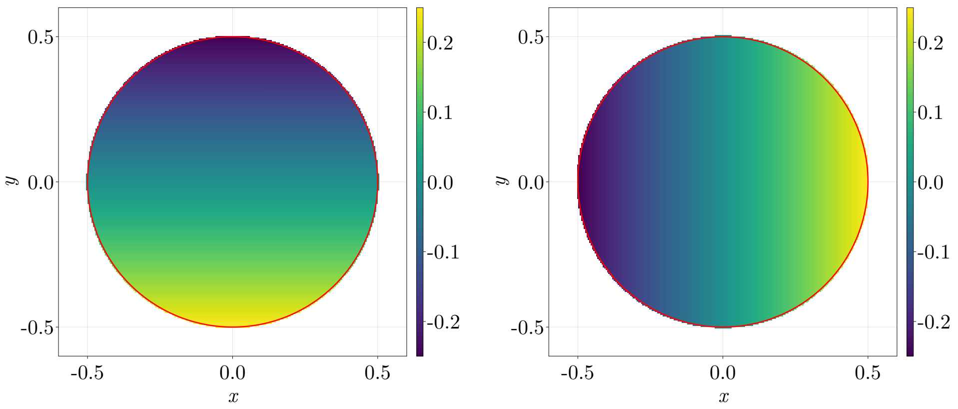
To assess the conservation properties of the proposed method, the advection equation

$$\frac{\partial \boldsymbol{u}}{\partial t} + \left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u} = \boldsymbol{0}. \tag{36}$$

is solved inside a cylinder of non-dimensionalized radius 0.5. The vector field $\boldsymbol{u}$ is initialized with an angular velocity $\omega_0 = 1$, representing the rotation of a rigid body, and slip boundary conditions at the wall, resulting in velocity components expressed as

$$\begin{aligned}
u_x &= -\frac{1}{2}\omega_0 y,\\
u_y &= \frac{1}{2}\omega_0 x.
\end{aligned} \tag{37}$$

Fig. 7 shows the horizontal and vertical components of the velocity for the initial conditions.



(a) Horizontal component of the velocity in x-direction.

(b) Vertical component of the velocity.

Fig. 7: Initial conditions.

The time derivative is discretized using the implicit midpoint method. The discrete momentum and kinetic energy can be defined as

$$\begin{aligned} \mathrm{Mom}_x &= \sum_{ij} u_{x,ij} V_{x,ij}, \\ \mathrm{Mom}_y &= \sum_{ij} u_{y,ij} V_{y,ij}, \end{aligned} \tag{38}$$

and

$$\mathrm{K} = \frac{1}{2} \sum_{ij} \left[\left(\left[S_x^+ V_x u_x^2 \right]_{ij} + \left[S_y^+ V_y u_y^2 \right]_{ij} \right) \right]. \tag{39}$$

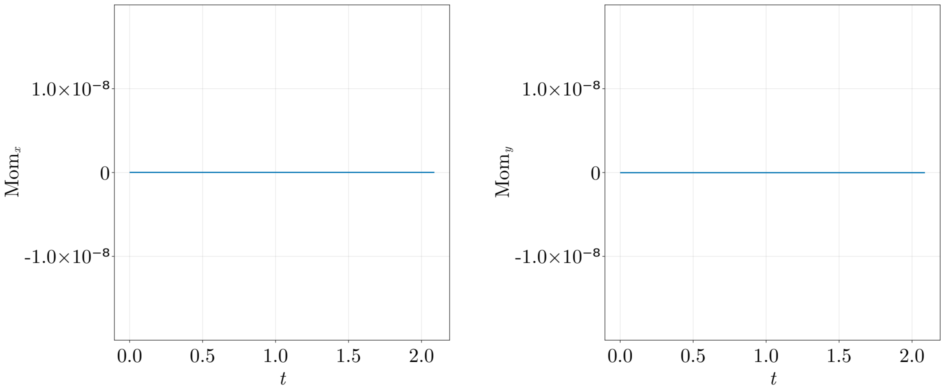
Since a staggered grid is used, the velocity components must be interpolated to the cell centers in order to calculate the kinetic energy.

Figs. 8 and 9 show the temporal evolution of the momentum and the kinetic energy ratio with respect to the initial kinetic energy, respectively, using 256 points per dimension. The shift in both the momentum and the kinetic energy is negligible.

5 Incompressible Navier-Stokes equations

5.1 Discrete operators for staggered quantities

In the incompressible Navier-Stokes equations the viscous term is function of the velocity Laplacian, which is yet to be defined. A staggered grid arrangement is employed as stated in the introduction, meaning that the Laplacian presented in Sec. 3.2 cannot be directly used. To solve this problem two new staggered grids are defined, each corresponding to a velocity component, one displaced half a cell width and the other one displaced half a cell height. Based on these two new staggered grids, two new sets of



(a) Momentum in x-direction.

(b) Momentum in y-direction.

Fig. 8: Temporal evolution of the momentum.

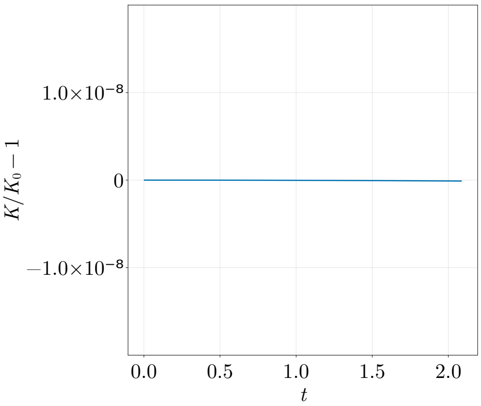


Fig. 9: Temporal evolution of the kinetic energy ratio.

geometric moments $\{\mathcal{A}_{\alpha,x}, \mathcal{A}_{\alpha,y}, \mathcal{B}_{\alpha,x}, \mathcal{B}_{\alpha,y}, \mathcal{V}_{\alpha}, \mathcal{W}_{\alpha,x}, \mathcal{W}_{\alpha,y}\}$ $\forall \alpha \in \{x, y\}$ are defined. These new geometric moments can be used to define the gradient of a staggered quantity u_α

$$\operatorname{grad}_\alpha \left(u_\alpha^\omega,\, u_\alpha^\gamma\right) = W_\alpha^\dagger \left(G_\alpha u_\alpha^\omega + H_\alpha u_\alpha^\gamma\right), \tag{40}$$

where W_α, G_α and H_α are built using the same expressions as in Eqs. (5), (6) and (12) but with the new geometric moments. Similarly, the volume-integrated divergence of a certain quantity $\boldsymbol{q}_\alpha$ can be defined following

$$\operatorname{div}_\alpha \left(\boldsymbol{q}_\alpha^\omega,\, \boldsymbol{q}_\alpha^\gamma\right) = -\left(G_\alpha^\top + H_\alpha^\top\right) \boldsymbol{q}_\alpha^\omega + H_\alpha^\top \boldsymbol{q}_\alpha^\gamma \tag{41}$$

which again can be simplified if $\boldsymbol{q}_\alpha$ represents a gradient, yielding

$$\operatorname{div}_\alpha\left(\boldsymbol{q}_\alpha^\omega\right) = -G_\alpha^\top \boldsymbol{q}_\alpha^\omega. \tag{42}$$

The resulting Laplacian of a staggered quantity u_α reads

$$\operatorname{div}_\alpha\left(\operatorname{grad}_\alpha\left(u_\alpha^\omega, u_\alpha^\gamma\right)\right), \tag{43}$$

which can be employed to define the viscous term in the discretized incompressible Navier-Stokes equations.

5.2 Pressure-projection method

This section presents the proposed discretization of the incompressible Navier-Stokes equations for an isotropic Newtonian fluid

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + \left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u} = -\dfrac{\nabla p}{\rho} + \dfrac{1}{\mathrm{Re}}\nabla\cdot\nabla\boldsymbol{u}, \\ \nabla\cdot\boldsymbol{u} = 0, \end{cases} \tag{44}$$

where $\boldsymbol{u}$ and p respectively denote the fluid's velocity and pressure fields, ρ its constant density and Re denotes the Reynolds number. Eq. (44) can be rewritten in semi-discrete form using the linear operators previously defined, yielding the momentum equation in direction α

$$V_\alpha \frac{\mathrm{d}u_\alpha^\omega}{\mathrm{d}t} + \operatorname{conv}_\alpha\left(\boldsymbol{u}^\omega, \boldsymbol{u}^\gamma\right) = -\frac{1}{\rho}\left[\operatorname{grad}\left(p^\omega, p^\gamma\right)\right]_\alpha + \frac{1}{\mathrm{Re}}\operatorname{div}_\alpha\left(\operatorname{grad}_\alpha\left(u_\alpha^\omega, u_\alpha^\gamma\right)\right), \tag{45}$$

and the divergence-free condition

$$\operatorname{div}\left(\boldsymbol{u}^\omega, \boldsymbol{u}^\gamma\right) = 0, \tag{46}$$

where $\boldsymbol{u}^\omega$ and $\boldsymbol{u}^\gamma$ are two face-centered vector fields and p^ω and p^γ are two cell-centered scalar fields. The discretization of the aforementioned incompressible Navier-Stokes equations results in a saddle point system of equations [26], sometimes also called Karush-Kuhn-Tucker (KKT) system in optimization [27]. A wide range of algorithms have been devised to efficiently solve saddle point systems (or approximation thereof). In the field of fluid mechanics, a common approach is the fractional step method [28]. In the present work, the method referred to as projection method II (PmII) by [29], which ensures a second order discretization of the equations, is employed.

In this projection method, the convective term is discretized using the explicit second-order Adams-Bashforth scheme and the viscous term is discretized using the implicit Crank-Nicolson scheme. The first step of the method consists of obtaining an intermediate velocity field $\boldsymbol{u}^\star$ by solving

$$V_\alpha \frac{u_\alpha^{\omega,\star} - u_\alpha^{\omega,n}}{\tau} + \frac{3}{2}\operatorname{conv}_\alpha\left(\boldsymbol{u}^{\omega,n}, \boldsymbol{u}^{\gamma,n}\right)$$

$$-\frac{1}{2}\operatorname{conv}_\alpha\left(\boldsymbol{u}^{\omega,n-1},\boldsymbol{u}^{\gamma,n-1}\right)=-\frac{1}{\rho}\left[\operatorname{grad}\left(p^{\omega,n-1/2},p^{\gamma,n-1/2}\right)\right]_\alpha$$
$$+\frac{1}{2\,\mathrm{Re}}\left[\operatorname{div}_\alpha\left(\operatorname{grad}_\alpha\left(u_\alpha^{\omega,\star},u_\alpha^{\gamma,\star}\right)\right)+\operatorname{div}_\alpha\left(\operatorname{grad}_\alpha\left(u_\alpha^{\omega,n},u_\alpha^{\gamma,n}\right)\right)\right]\ \forall\,\alpha\in\{x,y\},\quad(47)$$

where τ denotes the time step and the superscript n the iteration number. The boundary conditions applicable to $\boldsymbol{u}^\star$ (the predicted velocity field) are those of the velocity field at the next time step. The Dirichlet boundary conditions applied for example to impose the no-slip condition or the inflow conditions read

$$u_\alpha^{\gamma,\star}=u_{\alpha,\mathrm{ref}},\quad\forall\alpha\in\{x,y\},\tag{48}$$

where $u_{\alpha,\mathrm{ref}}$ is the value of the velocity to be imposed at the boundary in the α-direction. The homogeneous Neumann boundary condition applied to impose outflow conditions is given by

$$-\operatorname{div}_\alpha\left(0,\,\operatorname{grad}_\alpha\left(u_\alpha^{\omega,\star},u_\alpha^{\gamma,\star}\right)\right)=0,\quad\forall\alpha\in\{x,y\},\tag{49}$$

In the projection step, the velocity field is updated by projecting $\boldsymbol{u}^\star$ using the intermediate pressure field ψ^{n+1}, which is obtained by solving the following Poisson equation

$$\operatorname{div}\left(\operatorname{grad}\left(\psi^{\omega,n+1},\psi^{\gamma,n+1}\right)\right)=\frac{\rho}{\tau}\operatorname{div}\left(\boldsymbol{u}^{\omega,\star},\boldsymbol{u}^{\gamma,\star}\right).\tag{50}$$

Whenever no-slip or inflow conditions are to be imposed, a homogeneous Neumann boundary condition is applied to the intermediate pressure field following

$$-\operatorname{div}\left(0,\,\operatorname{grad}\left(\psi^{\omega,n+1},\psi^{\gamma,n+1}\right)\right)=0.\tag{51}$$

With outflow boundary conditions, a Dirichlet boundary condition is imposed following

$$\psi^{\gamma,n+1}=\psi_{\mathrm{ref}},\tag{52}$$

The velocity field is ultimately corrected as

$$u_\alpha^{\omega,n+1}=u_\alpha^{\omega,\star}-\frac{\tau}{\rho}\left[\operatorname{grad}\left(\psi^{\omega,n+1},\psi^{\gamma,n+1}\right)\right]_\alpha\ \forall\,\alpha\in\{x,y\}.\tag{53}$$

and the pressure is finally updated as

$$p^{\omega,n+1/2}=p^{\omega,n-1/2}+\psi^{n+1}-\frac{\tau}{2\,\mathrm{Re}}\operatorname{div}\left(\operatorname{grad}\left(\psi^{\omega,n+1},\psi^{\gamma,n+1}\right)\right),\tag{54}$$

where the last term ensures the second order accuracy of the pressure field.

6 Applications

6.1 Flow around a cylinder

The viscous flow around a cylinder at Re = 100 is used to test the accuracy of the proposed method. Four different grids have been tested with a domain size $\mathcal{D}=$

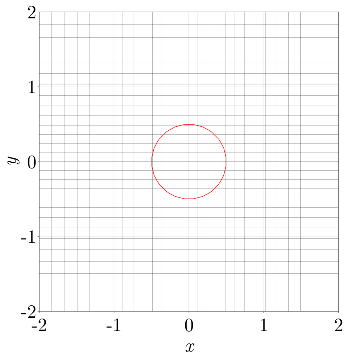


Fig. 10: Close-up view of grid G1.

Grid	$n_x \times n_y$	$\Delta x_{\min}$	$\Delta x_{\max}$
G1	170×110	0.12	0.4
G2	320×200	0.06	0.2
G3	600×350	0.03	0.1
G4	1150×500	0.015	0.075

Fig. 11: Grid parameters for the cylinder.

$[-15, 30] \times [-15, 15]$ with varying resolutions, labelled G1 (coarsest) to G4 (finest), in order to assess the accuracy of the method in a canonical configuration and to highlight its convergence properties. Fig. 10 shows a close-up view of the grid G1, whereas the number of points in each direction and the minimum and maximum cell size of each grid are shown in Tab. 11.

Dirichlet boundary condition is applied on the left border of the domain on the velocity field whereas homogeneous Neumann boundary conditions are applied on the bottom, right and top borders as outflow boundary conditions. On the pressure field, homogeneous Neumann is applied on the left border and Dirichlet on the bottom, right and top borders. A no-slip Dirichlet boundary condition is used at the wall for the velocity and homogeneous Neumann for the pressure. The CFL number is set to 0.5 in all the simulations. The horizontal and vertical components of the velocity field are initialized to 1 and 0, respectively. The simulations are advanced 150 time units in order to reach the periodic state.

Fig. 12 depicts the error and the order of convergence of the proposed methodology by measuring the error as the difference in the mean drag coefficient between the values obtained using grids G1, G2 and G3 and the value obtained using grid G4, which is used as reference. A convergence rate of 1.604 is observed. The results obtained for the Strouhal number (St), the root mean square lift coefficient (r.m.s. C_l) and the drag coefficient (C_d) are presented in Tab. 2 for the finest grid G4 and compared with several reference solutions, showing a good agreement.

A snapshot of the streamwise and vertical velocity fields at $t = 150$ is displayed in Fig. 13 showing the expected von Kármán vortex street.

6.2 Rayleigh-Bénard instability with a melting boundary

Finally, we test our new numerical method following the recent studies on the Rayleigh-Bénard (RB) instability in the presence of a melting boundary [34, 35]. Convection

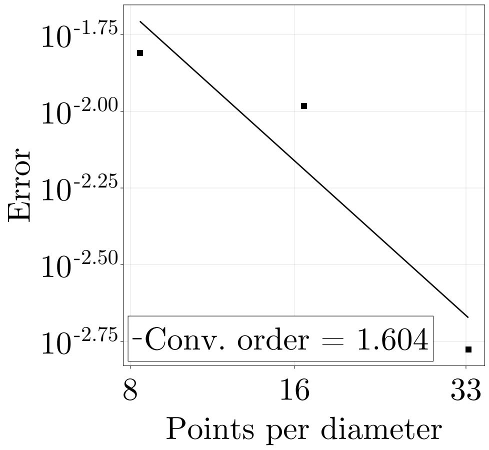
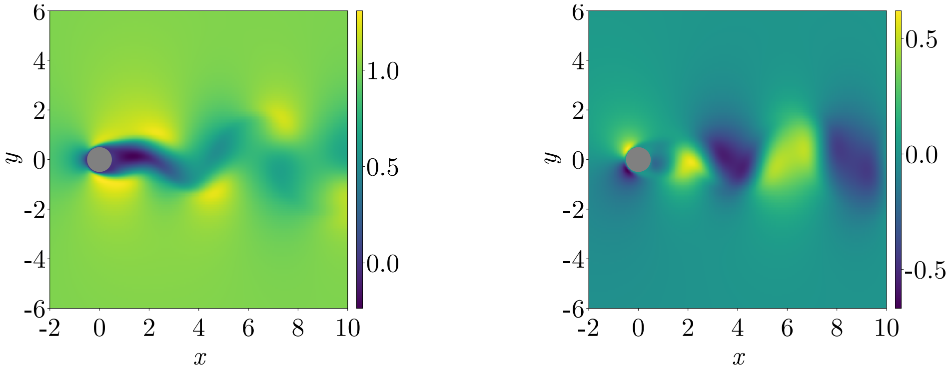


Fig. 12: Convergence of the average drag coefficient.

Study	St	r.m.s. C_l	C_d
Current (Grid G4)	0.167	0.241	1.352 ± 0.009
Norberg [30]	0.164	0.265	-
Henderson [31]	0.164	-	1.350
He *et al.* [32]	0.167	-	1.353
Linnick and Fasel [33]	0.166	-	1.38 ± 0.009

Table 2: Comparison of Strouhal number, r.m.s. lift coefficient and drag coefficient for the cylinder case at Re = 100.



(a) Horizontal velocity component.

(b) Vertical velocity component.

Fig. 13: Colormap of the velocity components using grid G4 at $t = 150$.

with a fixed topography is usually studied in the classical RB setup [36]. However, many natural phenomena such as erosion or melting involve a coupling between the flow and the moving boundary. Among the recent work on this topic, the phase-field model is the numerical method of choice when considering very high Rayleigh numbers [37] or turbulent underlying flows [38]. In this simplified test case, we will show that the presented methodology is able to accurately simulate this complex phenomenon.

To model this phenomena, we consider two immiscible phases (liquid and solid) of equal densities and thermal conductivities. In the fluid phase, we solve the Navier-Stokes equations in the Boussinesq approximation coupled to the convection-diffusion equation for the temperature field

$$\begin{aligned} \mathrm{Pr}^{-1}\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) =& -\nabla p + \mathrm{Ra}\; T_L\, \boldsymbol{e}_y + \nabla^2\boldsymbol{u}, \\ \nabla\cdot\boldsymbol{u} =& 0, \\ \frac{\partial T_L}{\partial t} + \boldsymbol{u}\cdot\nabla T_L =& \nabla^2 T_L, \end{aligned} \tag{55}$$

where T_L is the dimensionless temperature field in the liquid phase. The dimensionless numbers governing this equation are the Prandtl number

$$\mathrm{Pr} = \frac{\nu}{k}, \tag{56}$$

defined by the ratio of the liquid kinematic viscosity ν and thermal diffusivity k, and the Rayleigh number

$$\mathrm{Ra} = \frac{g\,\alpha_t \Delta T\, H^3}{\nu\, k}, \tag{57}$$

where g is the constant gravitational acceleration, α_t the coefficient of thermal expansion, H a characteristic length of the system and $\Delta T = \Theta_1 - T_M$ where Θ_1 comes from the boundary condition and T_M is the melting temperature. In the solid phase, the heat equation applies

$$\frac{\partial T_S}{\partial t} = \nabla^2 T_S, \tag{58}$$

where T_S is the dimensionless temperature field in the solid phase. At the interface, both temperature fields are equal to the melting temperature

$$T_L(\boldsymbol{x},t) = T_S(\boldsymbol{x},t) = T_M \quad \boldsymbol{x}\in\Gamma, \tag{59}$$

where Γ denotes the front position. In the present example, we disregard surface tension effects in the thermodynamic boundary condition at the front (ie. the Gibbs-Thomson relation).

Finally, as phase change occurs, there will appear a latent heat which is either absorbed or released. The condition of heat conservation at a given point on the moving interface corresponds to the rate at which heat is generated at the boundary, balanced by the rate at which this heat flows in either phase. Along the interface, the Stefan condition states that

$$\boldsymbol{v} = \mathrm{St}\,[\nabla T]_S^L \cdot \boldsymbol{n}, \quad \boldsymbol{x} \in \Gamma, \tag{60}$$

where v is the interface velocity, $\boldsymbol{n}$ is the outward normal unit vector at the interface and $[\nabla T]_S^L = \partial T_L/\partial \boldsymbol{n} - \partial T_S/\partial \boldsymbol{n}$ is the jump in normal components of the temperature gradients from the liquid phase to the solid phase. The Stefan number St is defined as the ratio between available heat in the system and the latent heat

$$\mathrm{St} = \frac{k\,\Delta T}{L_H}, \tag{61}$$

where L_H is the latent heat. The moving interface is captured using the levelset method as defined in Section 2.1. The interface is moved through the levelset advection equation

$$\frac{\partial \phi}{\partial t} + \boldsymbol{v} \cdot \nabla \phi = 0. \tag{62}$$

The levelset method coupled to the present cut-cell method has already been used in the context of shape optimization of two-dimensional Stefan problems [25, 39]. Moreover, the reader is referred to the companion paper [16] for further details on the numerical implementation of the levelset advection equation.

In this validation case, we consider the evolution of an initially flat liquid layer heated from below and comprised between a wall and a solid layer. As the solid melts, the liquid layer will grow vertically until a critical height where convection kicks in. If the solid layer is cooled, an equilibrium position will be reached. A schematic of this configuration is shown in Fig. 14. The gravity is pointing downwards $\boldsymbol{g} = -g\boldsymbol{e}_y$ and the horizontal size of the domain is bH, where b is the aspect ratio and where $H = 1$ is the height of the system. The temperature at the bottom wall is $\Theta_1 = 0.7$, the one at the top wall is $\Theta_0 = -0.3$, such that $\Theta_0 < T_M < \Theta_1$ where $T_M = 0$ is the melting temperature at the front. The initial flat interface is located at $h_0 = 0.05$ and the initial temperature field is set to zero.

The global Rayleigh number in Eq. (57) will control the onset of the Rayleigh-Bénard instability. We define an effective Rayleigh number as

$$\mathrm{Ra}_e = \mathrm{Ra}\,(1 - T_M)\,\bar{h}^3, \tag{63}$$

where $\bar{h}$ is the average fluid height, defined as

$$\bar{h}(t) = \frac{1}{b} \int_0^b h(x,t)dx. \tag{64}$$

When the effective Rayleigh number reaches the critical value $\mathrm{Ra}_c = 1707.76$, the initial diffusion-driven motion is transformed into a convection-driven one. We carry out simulations for different $\mathrm{Ra} = 5 \times 10^3$, 5×10^4, 2×10^5 and 4×10^5 with a Stefan

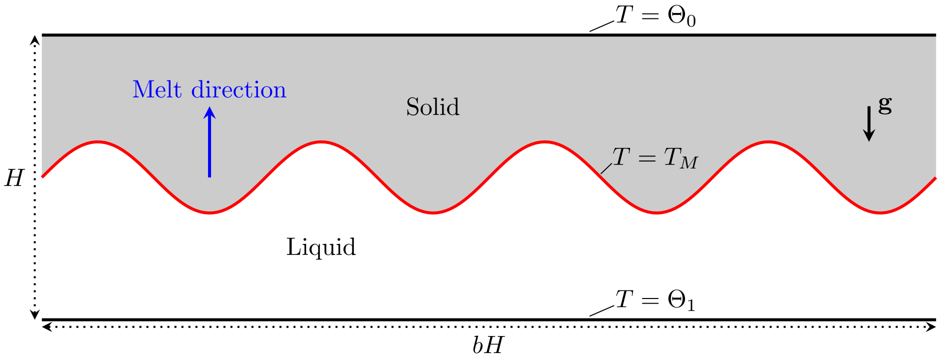


Fig. 14: Schematic of the melting boundary problem.

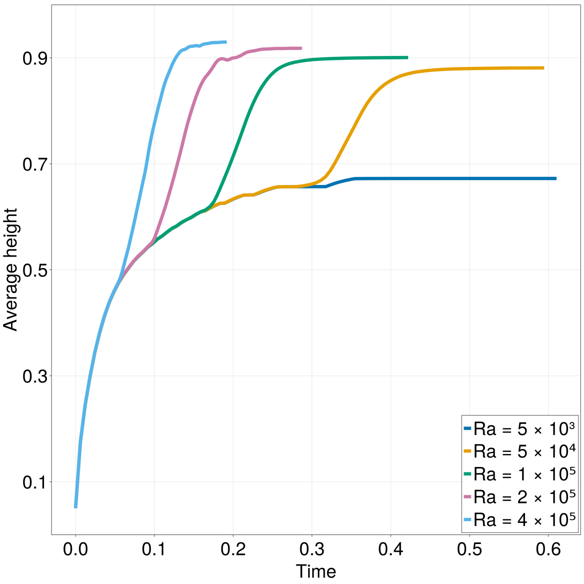


Fig. 15: Average height as a function of time for varying global Rayleigh numbers.

number set to St $= 0.1$. The aspect ratio is set to $b = 8$ and the grid used in our simulation is 512×64.

Figure 15 shows the average height as function of time for the different cases. In the $\mathrm{Ra} = 5 \times 10^3$ case, the critical Rayleigh number is not reached and the motion of the fluid layer is not affected, thus remaining a diffusion-driven one, similarly to the planar motion. In the rest of the cases, the effective Rayleigh number Ra_e crosses the threshold indicating the onset of the instability, characterized by a sharp increase in interface speed.

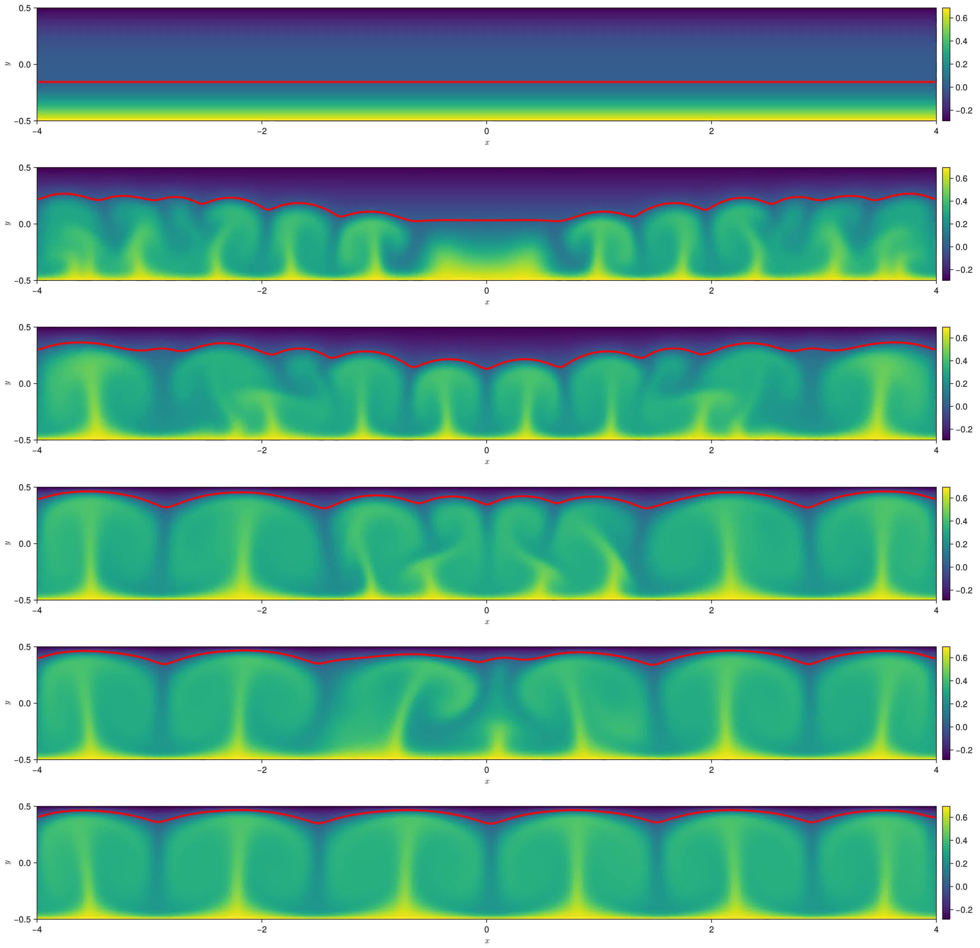


Fig. 16: Times series of the temperature field and interface position for the Ra $= 4 \times 10^5$ case. The color map is the temperature field in both phases and the interface corresponding to the 0-levelset is denoted in red.

Figure 16 shows a time series of the temperature field and interface position for the Ra $= 4 \times 10^5$ case. When the critical Rayleigh number is reached, the first bifurcation appears, creating the convection cells. The size of the convection cells will then vary with the secondary bifurcations mechanism. When the averaged height $\bar{h}$ matches the characteristic wavelength of the convection rolls, the convection cells have sufficient time to merge and then stabilize. We also note that the interface is deformed according to the shape of the cells.

7 Conclusions and perspectives

The proposed cut-cell methodology formulates discrete operators for the solution of the incompressible Navier-Stokes equations on staggered Cartesian grids in arbitrarily-shaped domains. Emphasis is set on both accuracy and structural properties of the first- and second-order operators. The geometric information is encapsulated in a set of surface and volume moments, designed to preserve constant states, recover classical formulas away from the boundary and in the vicinity of mesh-aligned boundaries, and retain a nearest-neighbor stencil. Boundary values are obtained as part of the solution, removing one possible source of error, since no interpolation is needed to obtain them. The accuracy of the boundary field is related to the boundary orientations. By construction, the spatial operators conserve volume and linear momenta locally and globally as well as kinetic energy in the inviscid limit. The method is shown to perform well in canonical two-dimensional flow configurations at low Reynolds number compared to previous works and in the presence of moving boundaries.

As highlighted in the introduction, the extension of the proposed solver to moving contact lines and free surfaces is presented in a companion paper in this same special issue (Part 2) [16].

Future independent work includes the replacement of the segregated approximation with a monolithic pressure-velocity solver, and an improved treatment of the small cut-cells. The latter, currently performed by removing the cut-cells below a specified threshold from the geometry (as described in Sec. 3.3), is the major bottleneck to an extension to three-dimensional configurations. In fact, with the discrete operators all having a Kronecker product structure (as seen in App. B), the only implementation effort is the computation of the geometric moments in 3D, which can be performed using the Marching Tetrahedra algorithm or special purpose libraries. The solution of the linear systems, currently performed using direct techniques, is therefore the limiting factor, since such techniques do not scale to 3D. Two directions are currently being investigated: the modification of the discrete equations in the vicinity of the small cut-cells (at the risk of a potential loss of conservation or symmetry), and a purely algebraic treatment based on dedicated preconditioning strategies to work hand in hand with classical gradient-based descent algorithms.

A Periodic boundary condition

To accommodate periodic boundary conditions, simply use

$$\Delta^- = \begin{bmatrix} 1 & & & -1 & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ 1 & & & -1 & 0 \end{bmatrix} \quad \text{and} \quad \Delta^+ = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & 0 \end{bmatrix}$$

and

$$\Sigma^- = \frac{1}{2}\begin{bmatrix} 1 & & & 1 & \\ 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & 1 & 1 & \\ 1 & & & 1 & 0 \end{bmatrix} \quad \text{and} \quad \Sigma^+ = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ & & & & 0 \end{bmatrix}$$

in the relevant directions.

B Three-dimensional extension

In 3D, the number of points is $n_x \times n_y \times n_z$ and the discrete velocity fields includes one more component

$$u = (u_x, u_y, u_z) \quad \text{and} \quad u^\gamma = \left(u_x^\gamma, u_y^\gamma, u_z^\gamma\right).$$

As far as discrete operators are concerned

$$D_x^\pm = I_z \otimes I_y \otimes \Delta_x^\pm, \quad D_y^\pm = I_z \otimes \Delta_y^\pm \otimes I_x \quad \text{and} \quad D_z^\pm = \Delta_z^\pm \otimes I_y \otimes I_x$$

and

$$S_x^\pm = I_z \otimes I_y \otimes \Sigma_x^\pm, \quad S_y^\pm = I_z \otimes \Sigma_y^\pm \otimes I_x \quad \text{and} \quad S_z^\pm = \Sigma_z^\pm \otimes I_y \otimes I_x.$$

C One-dimensional Poisson's equation

The discretized Poisson's equation (15) in terms of the cut-cell operators reads,

$$G^\top W^\dagger G p^\omega + G^\top W^\dagger H p^\gamma = V f^\omega, \tag{65}$$

with the one-dimensional version expanding to

$$-\left[B_x D_x^+ W_x^\dagger D_x^- B_x p^\omega + B_x D_x^+ W_x^\dagger \left(A_x D_x^- - D_x^- B_x\right) p^\gamma\right] = V f^\omega \tag{66}$$

in terms of the elementary discrete operators, if the x-direction is considered. We can now further expand this expression for a given cell i, which yields the following expression

$$\begin{aligned} -\mathcal{B}_{x,i}\left[\mathcal{W}_{x,i+1}\left(\mathcal{B}_{x,i+1}p_{i+1}^\omega - \mathcal{B}_{x,i}p_i^\omega\right) - \mathcal{W}_{x,i}\left(\mathcal{B}_{x,i}p_i^\omega - \mathcal{B}_{x,i-1}p_{i-1}^\omega\right)\right] - \\ \mathcal{B}_{x,i}\left\{\mathcal{W}_{x,i+1}\left[\left(\mathcal{B}_{x,i+1} - \mathcal{A}_{x,i+1}\right)p_{i+1}^\gamma + \left(\mathcal{A}_{x,i+1} - \mathcal{B}_{x,i}\right)p_i^\gamma\right] - \right. \\ \left. \mathcal{W}_{x,i}\left[\left(\mathcal{B}_{x,i} - \mathcal{A}_{x,i}\right)p_i^\gamma + \left(\mathcal{A}_{x,i} - \mathcal{B}_{x,i-1}\right)p_{i-1}^\gamma\right]\right\} = \mathcal{V}_i f_i^\omega. \end{aligned} \tag{67}$$

As seen in the latter equation, the discrete Laplacian yields a 3-point stencil in both p^ω and p^γ.

References

[1] Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering **194** (2005) https://doi.org/10.1016/j.cma.2004.10.008

[2] Mittal, R., Iaccarino, G.: Immersed boundary methods. Annual Review of Fluid Mechanics **37**, 239–261 (2005) https://doi.org/10.1146/annurev.fluid.37.061903.175743

[3] Peskin, C.S.: Flow Patterns Around Heart Valves: A Numerical Method. Journal of Computational Physics **10**, 252–271 (1972) https://doi.org/10.1016/0021-9991(72)90065-4

[4] Taira, K., Colonius, T.: The immersed boundary method: A projection approach. Journal of Computational Physics **225**(2), 2118–2137 (2007) https://doi.org/10.1016/j.jcp.2007.03.005

[5] Eldredge, J.D.: A method of immersed layers on Cartesian grids, with application to incompressible flows. Journal of Computational Physics **448**, 110716 (2022) https://doi.org/10.1016/j.jcp.2021.110716

[6] Calhoun, D., LeVeque, R.J.: A Cartesian Grid Finite-Volume Method for the Advection-Diffusion Equation in Irregular Geometries. Journal of Computational Physics **157**(1), 143–180 (2000) https://doi.org/10.1006/jcph.1999.6369

[7] Berger, M., Aftosmis, M.: Progress Towards a Cartesian Cut-Cell Method for Viscous Compressible Flow. In: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. American Institute of Aeronautics and Astronautics, Nashville, Tennessee (2012). https://doi.org/10.2514/6.2012-1301 . https://arc.aiaa.org/doi/10.2514/6.2012-1301

[8] Schneiders, L., Günther, C., Meinke, M., Schröder, W.: An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible flows. Journal of Computational Physics **311**, 62–86 (2016) https://doi.org/10.1016/j.jcp.2016.01.026

[9] Cheny, Y., Botella, O.: The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. Journal of Computational Physics **229**(4), 1043–1076 (2010) https://doi.org/10.1016/j.jcp.2009.10.007

[10] Arakawa, A., Lamb, V.R.: Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model. In: Methods in Computational Physics: Advances in Research and Applications. Methods in Computational Physics: Advances in Research and Applications, vol. 17, pp. 173–265. Elsevier,

Los Angeles, CA (1977). https://doi.org/10.1016/B978-0-12-460817-7.50009-4

[11] Harlow, F.H., Welch, J.E.: Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. Physics of Fluids **8**(12), 2182 (1965) https://doi.org/10.1063/1.1761178

[12] Morinishi, Y., Lund, T.S., Vasilyev, O.V., Moin, P.: Fully Conservative Higher Order Finite Difference Schemes for Incompressible Flow. Journal of Computational Physics **143**(1), 90–124 (1998) https://doi.org/10.1006/jcph.1998.5962

[13] Beltman, R., Anthonissen, M., Koren, B.: Conservative Mimetic Cut-Cell Method for Incompressible Navier-Stokes Equations. In: Radu, F.A., Kumar, K., Berre, I., Nordbotten, J.M., Pop, I.S. (eds.) Numerical Mathematics and Advanced Applications ENUMATH 2017 vol. 126, pp. 1035–1043. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-96415-7_98 . Series Title: Lecture Notes in Computational Science and Engineering. http://link.springer.com/10.1007/978-3-319-96415-7_98

[14] Wathen, A.J.: Preconditioning. Acta Numerica **24**, 329–376 (2015) https://doi.org/10.1017/S0962492915000021

[15] Nataf, F., Tournier, P.-H.: A GenEO Domain Decomposition method for Saddle Point problems. Comptes Rendus. Mécanique (2023) https://doi.org/10.5802/crmeca.175

[16] Quirós Rodríguez, A., Fullana, T., Sayadi, T., Le Chenadec, V.: A level-set based cut cell method for interfacial flows. Part 2: Free-surface flows and dynamic contact angle treatment. Acta Mechanica (2024)

[17] Sethian, J.A.: Level Set Methods and Fast Marching. Cambridge Monographs on Applied and Computational Mathematics, vol. 1. Cambridge University Press, University of California, Berkeley (1999)

[18] Maple, C.: Geometric Desing and Space Planning Using the Marching Squares and Marching Cube Algorithms. In: 2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings, pp. 90–95 (2003). https://doi.org/10.1109/GMAG.2003.1219671

[19] Saurel, R., Abgrall, R.: A multiphase godunov method for compressible multifluid and multiphase flows. Journal of Computational Physics **150**(2), 425–467 (1999) https://doi.org/10.1006/jcph.1999.6187

[20] Tavares, M., Josserand, C., Limare, A., Lopez-Herrera, J.M., Popinet, S.: A coupled VOF/embedded boundary method to model two-phase flows on arbitrary solid surfaces. Computers & Fluids **278**, 106317 (2024) https://doi.org/10.1016/j.compfluid.2024.106317

[21] Ye, T., Mittal, R., Udaykumar, H.S., Shyy, W.: An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. Journal of Computational Physics **156**(2), 209–240 (1999) https://doi.org/10.1006/jcph.1999.6356

[22] Kirkpatrick, M.P., Armfield, S.W., Kent, J.H.: A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid. Journal of Computational Physics **184**(1), 1–36 (2003) https://doi.org/10.1016/S0021-9991(02)00013-X

[23] Hartmann, D., Meinke, M., Schröder, W.: An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods. Computers & Fluids **37**(9), 1103–1125 (2008) https://doi.org/10.1016/j.compfluid.2007.06.007

[24] Johansen, H., Colella, P.: A Cartesian Grid Embedded Boundary Method for Poisson's Equation on Irregular Domains. Journal of Computational Physics **147**, 60–85 (1998) https://doi.org/10.1006/jcph.1998.5965

[25] Fullana, T., Le Chenadec, V., Sayadi, T.: Adjoint-based optimization of two-dimensional Stefan problems. Journal of Computational Physics **475**, 111875 (2023) https://doi.org/10.1016/j.jcp.2022.111875

[26] Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numerica **14**, 1–137 (2005) https://doi.org/10.1017/S0962492904000212

[27] Nocedal, J., Wright, S.J.: Numerical Optimization, Second edition edn. Springer series in operation research and financial engineering. Springer, New York, NY (2006)

[28] Chorin, A.J.: Numerical Solution of the Navier-Stokes Equations. American Mathematical Society **22**(104), 18 (1968) https://doi.org/10.2307/2004575

[29] Brown, D.L., Cortez, R., Minion, M.L.: Accurate Projection Methods for the Incompressible Navier–Stokes Equations. Journal of Computational Physics **168**(2), 464–499 (2001) https://doi.org/10.1006/jcph.2001.6715

[30] Norberg, C.: Fluctuating lift on a circular cylinder: review and new measurements. Journal of Fluids and Structures **17**(1), 57–96 (2003) https://doi.org/10.1016/S0889-9746(02)00099-3

[31] Henderson, R.D.: Nonlinear dynamics and pattern formation in turbulent wake transition. Journal of Fluid Mechanics **352**, 65–112 (1997) https://doi.org/10.1017/S0022112097007465

[32] He, J.-W., Glowinski, R., Metcalfe, R., Nordlander, A., Periaux, J.: Active Control and Drag Optimization for Flow Past a Circular Cylinder. Journal of Computational Physics **163**(1), 83–117 (2000) https://doi.org/10.1006/jcph.

2000.6556

[33] Linnick, M.N., Fasel, H.F.: A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. Journal of Computational Physics **204**(1), 157–192 (2005) https://doi.org/10.1016/j.jcp.2004.09.017

[34] Favier, B., Purseed, J., Duchemin, L.: Rayleigh–Bénard convection with a melting boundary. J. Fluid Mech. **858**, 437–473 (2019) https://doi.org/10.1017/jfm.2018.773

[35] Limare, A., Popinet, S., Josserand, C., Xue, Z., Ghigo, A.: A hybrid level-set / embedded boundary method applied to solidification-melt problems. Journal of Computational Physics **474**, 111829 (2023) https://doi.org/10.1016/j.jcp.2022.111829

[36] Bodenschatz, E., Pesch, W., Ahlers, G.: Recent Developments in Rayleigh-Bénard Convection. Annual Review of Fluid Mechanics **32**(Volume 32, 2000), 709–778 (2000) https://doi.org/10.1146/annurev.fluid.32.1.709 . Publisher: Annual Reviews Type: Journal Article

[37] Yang, R., Howland, C.J., Liu, H.-R., Verzicco, R., Lohse, D.: Morphology evolution of a melting solid layer above its melt heated from below. Journal of Fluid Mechanics **956**, 23 (2023) https://doi.org/10.1017/jfm.2023.15

[38] Perissutti, D., Marchioli, C., Soldati, A.: Morphodynamics of melting ice over turbulent warm water streams (2024). https://arxiv.org/abs/2406.12116

[39] Fullana, T.: Simulation and optimization of complex phenomena in multiphase flows. phdthesis, Sorbonne Université (December 2022). https://theses.hal.science/tel-04053531