University of Liège

Faculty of Applied Sciences

MATH0471 - Multiphysics Integrated Computational Project

Final Report

Fluid Simulation with Particle-in-Cell methods

Professors
Boman Romain
Geuzaine Christophe

Students
Binstok Quentin
Collin Néo



Academic Year 2025 - 2026

Contents

1 Introduction **2**

2 Methods and Implementation **3**
2.1 Considered physics . . . 3
2.2 Pressure update . . . 4
2.2.1 Boundary conditions . . . 4
2.2.2 Implementation . . . 5
2.3 Thermal update . . . 6
2.4 Gravity . . . 6
2.5 Semi-Lagrangian method . . . 7
2.6 PIC and FLIP . . . 8
2.6.1 Advection . . . 8
2.6.2 Grid-particles interaction . . . 9
2.6.3 Creation and deletion of particles . . . 9
2.6.4 Handling of free surfaces . . . 10
2.6.5 Main loop . . . 10
2.7 APIC . . . 10
2.7.1 Grid-particles interactions . . . 10
2.7.2 Particle-grid interaction loop . . . 11
2.7.3 Handling of degeneracies . . . 12

3 Results **13**
3.1 Taylor–Green vortex . . . 13
3.1.1 Impacts of simulation parameters . . . 20
3.2 Cylindrical in an uniform flow . . . 26
3.3 Dam break . . . 29
3.4 Sloshing . . . 33
3.5 Rayleigh-Bénard convection . . . 36
3.6 Heat exchanger . . . 38

4 Discussion and perspectives **41**

5 Conclusion **43**

A Development of the temperature equation **44**

B Simulation parameters **46**
B.1 Taylor-Green vortex . . . 46
B.2 Cylindrical in an uniform flow . . . 46
B.3 Dam break . . . 47
B.4 Sloshing . . . 47
B.5 Rayleigh-Bénard convection . . . 47
B.6 Heat exchanger . . . 48

1 Introduction

This project focuses on the quantitative and qualitative analysis of several Particle-in-Cell (PIC) methods applied to fluid mechanics problems. These numerical methods provide a powerful framework for simulating continuum media, such as fluids, plasmas, and deformable solids, by combining Lagrangian and Eulerian descriptions. The material is represented by a set of computational particles that carry physical properties and whose motion is explicitly tracked over time, while field quantities such as velocity, pressure, density, or stress are defined on a fixed spatial grid. Information is exchanged between particles and the grid through interpolation and deposition steps, enabling accurate and stable coupling between both representations.

As a first step, a numerical solver based on a semi-Lagrangian formulation was developed. This approach serves as a robust foundation for implementing three variants of Particles-in-Cell method : the standard (or "vanilla") PIC method, the Fluid-Implicit Particle (FLIP) method, and the Affine Particle-in-Cell (APIC) method.

In order to assess the validity of our implementation, a comparison of each methods was made against a set of carefully selected and well-documented benchmark test cases. Although these cases are relatively standard in computational fluid dynamics, they are sufficiently complex to provide meaningful insight into the behaviour and specific characteristics of each method. It means that a systematic comparison between each method could be conducted, using the same cases, in order to evaluate the strengths and limitations of each approach.

In addition to comparing the different methods, particular attention is given to the influence of simulation parameters. Parameters such as grid resolution, time step size and particle density can significantly impact the accuracy, stability, and computational cost of the simulations. Their effects are therefore analyzed in detail to better understand the trade-offs involved and to identify their general impact.

In a second phase of the project, a thermal coupling mechanism was introduced into the solver. The objective of this extension is to model fluid motion induced by temperature gradients, such as buoyancy-driven flows. As with the previous developments, the thermal model is validated using appropriate test cases that capture key thermally driven phenomena. The sensitivity of the coupled system to simulation parameters is also investigated, providing further insight into the robustness and applicability of the approach.

Overall, this project aims to provide a comprehensive evaluation of PIC-based methods for fluid simulation, along with an analysis of their extensions to thermally coupled flows.

2 Methods and Implementation

In this section the equations solved are described. Afterwards, the pressure update is discussed, then the four implemented methods are explained, which are a semi-Lagrangian method, the particle-in-cell (PIC) method, the fluid-implicit-particle (FLIP) method, and the affine-particle-in-cell (APIC) method. Moreover, the thermal coupling is detailed.

2.1 Considered physics

For an inviscid, incompressible, and adiabatic flow, the Navier-Stokes equations simplify into the Euler equations, which are

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \frac{1}{\rho} \boldsymbol{\nabla} p = \boldsymbol{g} \tag{1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{2}$$

with $\boldsymbol{u}(\boldsymbol{x}, t)$ the velocity of the fluid at position $\boldsymbol{x}$ and time t, ρ the density of the fluid, $p(\boldsymbol{x}, t)$ the pressure, and $\boldsymbol{g}$ the gravitational acceleration [1].

It is necessary to introduce now the biggest hypothesis used in the numerical methods discussed in this report, which is the splitting of the operator in Eq. 1. This consists in solving kinematic updates separately from the pressure update. As the latter is solved via finite differences on a regular grid, further developments of the equations are needed. In the following, the kinematic parts will be removed from the equations, which consists in ignoring the terms $\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}$ and $\boldsymbol{g}$. Subsequent sections will describe how those are treated.

The equations thus become

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\frac{1}{\rho} \boldsymbol{\nabla} p \tag{3}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0. \tag{4}$$

Furthermore, it is possible to easily take into account thermal effects, if one assumes the Boussinesq approximation. The equation for the temperature[1] is

$$\frac{\partial T}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} T = \frac{r}{\rho c} + \frac{1}{\rho c} \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T). \tag{5}$$

In all generality, the density of matter changes with the temperature. This would imply that the density used in the solver for the pressure, as well as the one present in Eq. 5 would need to be computed at each cell of the grid.

The Boussinesq approximation allows to handle this more easily, when dealing with fluids. It consists in keeping the density constant everywhere in the resolution, except for the gravitational force. In that case, it is approximated by a linear law,

$$\rho(T) = \rho_0 (1 - \beta (T - T_0)), \tag{6}$$

with β and T_0 material parameters. Newton's law, when only considering the gravity, then becomes

$$\rho_0 \dot{\boldsymbol{v}} = \rho_0 (1 - \beta (T - T_0)) \boldsymbol{g} \tag{7}$$

$$\Rightarrow \dot{\boldsymbol{v}} = (1 - \beta (T - T_0)) \boldsymbol{g}. \tag{8}$$

[1] The mathematical developments leading to this equation are provided in Appendix A.

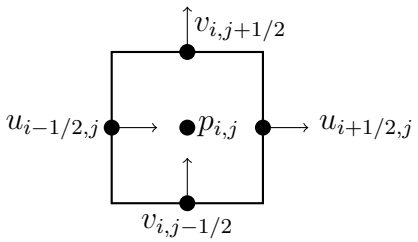


Figure 1: A cell of the staggered grid, with the components (u, v) of the velocity represented, as well as the pressure p.

2.2 Pressure update

The discretization of Eq. 3 and Eq. 4 is realized on a staggered grid, with the pressure nodes located at the centre of cells (in positions (i, j)), the nodes associated with the x component of the speed located at the centre-right of the cells (positions $(i + 1/2, j)$), and y components being at the top-centre of the cells (positions $(i, j + 1/2)$). This is represented in Fig. 1.

Thus, Eq. 3 can be discretized using forward-difference derivatives in both time and space, which gives

$$u^{n+1}_{i+1/2,j} = u^n_{i+1/2,j} - \frac{\Delta t}{\rho \Delta x}(p_{i+1,j} - p_{i,j}) \tag{9}$$

$$v^{n+1}_{i,j+1/2} = v^n_{i,j+1/2} - \frac{\Delta t}{\rho \Delta x}(p_{i,j+1} - p_{i,j}) \tag{10}$$

$$u^{n+1}_{i-1/2,j} = u^n_{i-1/2,j} - \frac{\Delta t}{\rho \Delta x}(p_{i,j} - p_{i-1,j}) \tag{11}$$

$$v^{n+1}_{i,j-1/2} = v^n_{i,j-1/2} - \frac{\Delta t}{\rho \Delta x}(p_{i,j} - p_{i,j-1}). \tag{12}$$

Afterwards, it is wanted that Eq. 4 be discretized such that the velocity field obtained at the next time step be incompressible. Hence, the discretization is

$$u^{n+1}_{i+1/2,j} + v^{n+1}_{i,j+1/2} = u^{n+1}_{i-1/2,j} + v^{n+1}_{i,j-1/2}. \tag{13}$$

Introducing Eq. 9, Eq. 10, Eq. 11 and Eq. 12 in the equation above yields an implicit scheme for the pressure. It is more convenient to write it in a form suitable for iterative solvers, giving

$$p_{i,j} = \frac{1}{4}\left[p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - \frac{\rho \Delta x^2}{\Delta t} \text{div } \boldsymbol{u}_{i,j}\right] \tag{14}$$

where div $\boldsymbol{u}_{i,j}$ is the divergence of $\boldsymbol{u}$ computed on the (i, j) cell.

2.2.1 Boundary conditions

Two types of boundary conditions are considered here. The first represents a solid boundary, which then enforces a condition of impermeability. The second corresponds to an area without fluid, or free surface, and thus to a pressure of zero.

Impermeability is equivalent to requiring that $\boldsymbol{u} \cdot \boldsymbol{n} = 0$, with $\boldsymbol{n}$ the unit normal to the boundary. However, given that the discretization considered here is very simple,

enforcing impermeability reduces to setting a component of the velocity to zero in the cell on the left of the boundary (if it is vertical), or in the one on the bottom of the boundary (if it is horizontal).

To introduce this in Eq. 14, one sets Eq. 9, Eq. 10, Eq. 11 and Eq. 12 to zero, leading to

$$p_{i+1,j} = p_{i,j} + \frac{\rho \Delta x}{\Delta t} u^n_{i+1/2,j} \tag{15}$$

$$p_{i,j+1} = p_{i,j} + \frac{\rho \Delta x}{\Delta t} v^n_{i,j+1/2} \tag{16}$$

$$p_{i-1,j} = p_{i,j} - \frac{\rho \Delta x}{\Delta t} u^n_{i-1/2,j} \tag{17}$$

$$p_{i,j-1} = p_{i,j} - \frac{\rho \Delta x}{\Delta t} v^n_{i,j-1/2}. \tag{18}$$

Then, if a cell adjacent to (i, j) is a solid cell, one simply replaces the corresponding pressure by the corresponding expression listed above.

Moreover, if an adjacent cell corresponds to air, one just has to replace the corresponding pressure by zero.

2.2.2 Implementation

Eq. 14 is solved here via iterative methods. Two of such methods have been implemented: the Jacobi method, and the successive over-relaxation (SOR) method.

The Jacobi method is straightforward. At each iteration, it updates the pressure field by applying Eq. 14 [2]. As all the values of p used in the formula must correspond to the same iteration, it was necessary to use two instances of the pressure field, one having the values used for the computation, and the other used to write the newly computed values of the pressure field. At each iteration, those two instances are exchanged. An advantage of the Jacobi method is that it can be massively parallelized.

This SOR algorithm updates the pressure in a different way than Jacobi. Where Jacobi replaces the value of the pressure $p^n_{i,j}$ by the newly computed one $p^{new}_{i,j}$, the SOR algorithm updates the pressure with $\omega p^{new}_{i,j} + (1 - \omega) p^n_{i,j}$, with $\omega \in [1, 2]$. When used to solve a Poisson equation, the optimal value for ω is given by

$$\omega_{\text{opt}} = \frac{2}{1 + \sin(\pi / N)} \tag{19}$$

when N is large, with N the smallest number of cells on a side of the domain [2]. As this method becomes unstable when $\omega > 2$, ω is computed here as

$$\omega = \min(1.95, \omega_{\text{opt}}). \tag{20}$$

It was also possible to parallelize the SOR algorithm. This is done in a "red-black" way. The grid is divided in two in the manner of a check-board, and each half is computed after the other. Thus, each iteration is subdivided in two. However, as in Eq. 14, the cells containing the values of the pressure required at each cell form a diamond, the computation of a the values in a given check-board can be parallelized. If such a division of the domain was not used, memory races would occur. This is due to the fact that the SOR algorithm re-uses in a given iteration the values already computed.

Both methods use the same convergence criterion. It consists in evaluating the 2-norm of the residual of the linear system, and comparing this to a user-defined tolerance.

Once the new pressure field has been computed, Eq. 3 is used to compute the new velocity field. To that end, it is discretized as in Eq. 9 and Eq. 10.

2.3 Thermal update

Eq. 5 is simplified in a manner similar to what is done for the momentum equation, meaning that the advection part is handled separately from the rest.

The equation solved on the grid is thus reduced to

$$\frac{\partial T}{\partial t} = \frac{r}{\rho c} + \frac{1}{\rho c} k \boldsymbol{\nabla}^2 T. \tag{21}$$

This equation is then discretized using an implicit scheme, which yields

$$T_{i,j}^{n+1} = \frac{1}{1+4\alpha}\left[T_{i,j}^{n} + \gamma + \alpha\left(T_{i+1,j}^{n+1} + T_{i-1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1}\right)\right] \tag{22}$$

with $\alpha = \frac{k\Delta t}{(\Delta x)^2 \rho c}$ and $\gamma = \frac{r\Delta t}{\rho c}$.

To solve this, a successive over-relaxation (SOR) solver has been implemented. It supports both Dirichlet and Neumann boundary conditions. The former is trivial to implement, while the second necessitates some mathematical developments. This condition consists in imposing a heat flux q, which, via Fourier's law, gives

$$q = k\frac{T_{BC}^{n+1} - T_{i,j}^{n+1}}{\Delta x}. \tag{23}$$

Isolating T_{BC}^{n+1} then gives

$$T_{BC}^{n+1} = \frac{q\Delta x}{k} + T_{i,j}^{n+1}. \tag{24}$$

It should be noted that there is no need to take into account the direction of the heat flux. Indeed, as heat flows from hot to cold, T_{BC}^{n+1} should be higher than $T_{i,j}^{n+1}$ if $q > 0$, and lower in the opposite case. This is verified by the formula obtained above.

2.4 Gravity

Gravity being a kinematic update, it is applied separately from the pressure update. From Newton's law, one has

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{g} \tag{25}$$

as the advection is treated separately. Thus, the vertical component of the speed is updated such that

$$v \mapsto v - g\Delta t. \tag{26}$$

For the PIC, FLIP, and APIC methods, this update is directly applied on the particles. For the semi-Lagrangian method, this is not possible, and thus it is applied on the grid.

When taking into account thermal effects, a corrective factor is added, as described in Eq. 8. This changes the gravitational update such that

$$v \mapsto v - (1 - \beta(T - T_0))g\Delta t. \tag{27}$$

2.5 Semi-Lagrangian method

As introduced in the subsection 2.1, when the hypothesis of the splitting of the operator is introduced, the advection term $\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}$ is treated separately,

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} = 0 \tag{28}$$

Projecting this equation on both axis, the vector field $\boldsymbol{u}$ becomes the scalar fields u and v.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{29}$$

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial y} = 0 \tag{30}$$

Generalizing to any scalar field q, a crucial step of the fluid simulation is solving the advection equation,

$$\frac{Dq}{Dt} = 0 \tag{31}$$

This equation can be encapsulated in the following numerical routine,

$$q^{n+1} = \text{advect}(\boldsymbol{u}, \Delta t, q^n) \tag{32}$$

which returns an approximation to the result of advecting q through the velocity field over the time interval Δt. Interestingly, the velocity is transported using the velocity itself.

Bridson [1] introduced a physically motivated advection scheme called the semi-Lagrangian method, based on the fact that Eq. 31 is always satisfied in the Lagrangian framework. The idea behind the method is, that it is possible to determine the new value of q at a given point $\boldsymbol{x}$ and at a given time t by conceptually tracing backward the trajectory of the hypothetical fluid particle that arrives at $\boldsymbol{x}$ in t. Let's note that the velocity field used for the advection should be the divergence free one.

The implementation of the numerical routine on a staggered MAC grid is relatively straightforward. The objective is to compute the updated scalar field q^{n+1} at every cell center (x, y) of the computational domain.

First, the updated divergence-free velocity field $\boldsymbol{u}^{n+1}$ is interpolated at each cell center using bilinear interpolation, while taking into account that the velocity components are stored on the cell faces of the staggered grid. Although the iteration over the domain is designed to avoid any out-of-bounds access, additional safety checks were implemented in order to prevent potential crashes or invalid memory accesses.

Once both velocity components have been evaluated, the departure point associated with the current cell is computed by tracing the trajectory backward in time according to

$$x_p = x - \Delta t\, v_x, \tag{33}$$

$$y_p = y - \Delta t\, v_y. \tag{34}$$

The coordinates of the departure point are then clamped so that they remain inside the computational domain. This operation is necessary because the backtraced particle

position may leave the domain, particularly near the boundaries or in regions of high velocity. The clamping procedure is defined as

$$x_p = \max\left(0, \min\left(x_p, (n_x - 1)\Delta x\right)\right), \tag{35}$$
$$y_p = \max\left(0, \min\left(y_p, (n_y - 1)\Delta x\right)\right), \tag{36}$$

which guarantees that the departure point always lies within the valid interpolation region of the grid.

Once the departure point has been determined, the value of the scalar field at this location is reconstructed using bilinear interpolation from the four neighboring grid values. The interpolated value is then assigned to the updated field according to

$$q^{n+1}(x, y) = q^n(x_p, y_p), \tag{37}$$

which corresponds to transporting the scalar quantity along the fluid trajectory during one time step. Although semi-Lagrangian advection is very stable and allows relatively large time steps, the interpolation step introduces numerical diffusion, which tends to smooth sharp gradients in the transported quantity. This numerical diffusion has been analyzed in [1], using the backward error analysis and leads to a viscosity-like term with coefficient $u\Delta x$ in Eq. 31. It means that the Lagrangian method tries to solve the advection equation without viscosity, but the results look like we are simulating a fluid with viscosity. Moreover, the damping introduced by this numerical diffusion should only be dependant over the space step Δx but not over the time step Δt, as long as the CFL condition remains valid.

2.6 PIC and FLIP

As the particle-in-cell (PIC) and fluid-implicit-particle (FLIP) methods are very similar, they will be discussed jointly. The advection will first be considered. Afterwards, the grid-particles interaction will be discussed. Finally, the deletion and introduction of particles will be detailed.

2.6.1 Advection

For these methods, the advection is handled via the particles. Using their speed, their position is updated using a three-stage third-order Runge-Kutta scheme [1]. This means that the new position of the particles is computed as

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \frac{2}{9}\Delta t\boldsymbol{k}_1 + \frac{3}{9}\Delta t\boldsymbol{k}_2 + \frac{4}{9}\Delta t\boldsymbol{k}_3 \tag{38}$$

with

$$\boldsymbol{k}_1 = \boldsymbol{v}(\boldsymbol{x}^n) \tag{39}$$
$$\boldsymbol{k}_2 = \boldsymbol{v}(\boldsymbol{x}^n + \frac{1}{2}\Delta t\boldsymbol{k}_1) \tag{40}$$
$$\boldsymbol{k}_3 = \boldsymbol{v}(\boldsymbol{x}^n + \frac{3}{4}\Delta t\boldsymbol{k}_2) \tag{41}$$

and $\boldsymbol{x}^n$ the position of the particle before the advection, and $\boldsymbol{x}^{n+1}$ its position after.

2.6.2 Grid-particles interaction

As the speed is computed on the grid, but used by each particle in the advection, a central point of the method is the grid-particle interaction.

Going from the particles to the grid is done by iterating on the particles. The contribution of each of them to each point of the grid is weighted using a bilinear kernel that extends only over a single cell. As the grids for the vertical and horizontal velocities are staggered, it is necessary to use two different loops for this.

Sending the velocities computed on the grid to the particles is done by interpolation. In this project, bilinear interpolation is used. This step is also where the FLIP method is applied. Instead of replacing the velocity of the particle by the newly computed one (from the pressure field) as done in the PIC method, it adds to the speed of the particle the difference between the former velocity grid, and the newly computed one.

It is perfectly possible to use a mix of the PIC and FLIP methods. The formula for the update of the particle velocity then becomes

$$\boldsymbol{v}_{\text{part}}^{n+1} = (1-\alpha)\boldsymbol{v}_{\text{grid}}^{n+1} + \alpha(\boldsymbol{v}_{\text{part}}^{n} + \boldsymbol{v}_{\text{grid}}^{n+1} - \boldsymbol{v}_{\text{grid}}^{n}) \tag{42}$$

where $\boldsymbol{v}_{\text{part}}^{n}$ is the speed of the particle at time-step n, $\boldsymbol{v}_{\text{grid}}^{n}$ is the speed interpolated from the grid at time-step n, and α is the ratio of FLIP over PIC ($\alpha = 1$ means full FLIP, $\alpha = 0$ is full PIC).

2.6.3 Creation and deletion of particles

If, after having been advected, a particle is either out of the domain, or in a solid cell, it is deleted.

The introduction of new particles is complex. Two situations are to be considered. The first consists in an inlet flow, through which new particles arrive. The second is quite less physical in nature, as it is about ensuring that enough particles are present in each liquid cell.

Inlet flows are handled by first determining the number of particles to be introduced, based on the speed of the flow. This number is computed as

$$c^{\dagger} = \frac{U\,\rho_p}{\Delta x}, \tag{43}$$

where U is the inlet velocity, ρ_p is the target particle density per cell, Δx is the cell size, and $c^{\dagger}$ is the resulting particle creation rate per inlet cell. Thus, the target number of particles per cell at the inlet is $c^{\dagger}\Delta t$, with Δt the duration of a time step.

Once this target number has been computed, the two situations described above converge to the same logic, with the exception that inside of the domain, the target particle number is obtained from a user input.

When creating a particle, two values are random: its initial position in the cell, and its creation time during the time step. After determining its position, its speed is obtained by interpolating the velocity from the grid. From this, the particle is advected for the rest of the time step. Its temperature is also interpolated.

It is possible to disable the refilling of cells inside of the domain, which is necessary for free-surface flows.

2.6.4 Handling of free surfaces

Free surfaces are handled in a naive way. If a particle is present in a cell after being advected, the cell is considered liquid, and thus a pressure is computed inside it. Accordingly, if a cell becomes empty, it is switched to air, leading to its pressure being zero. Solid cells never change.

2.6.5 Main loop

Each time loop happens as follow:

1. gravity is applied to the particles,
2. the speed of the particles is brought back to the grid,
3. if applicable, the temperature of particles is also brought back,
4. the divergence of the speed is computed,
5. the pressure field is computed,
6. if applicable, Eq. 5 is solved,
7. from the new pressure, the velocity is computed,
8. the velocity is interpolated at the particles,
9. results are saved,
10. the particles are advected,
11. particles are created or deleted.

2.7 APIC

The Affine Particle-In-Cell (APIC) method is an extension of the PIC approach designed to reduce numerical dissipation while maintaining stability. Unlike PIC and FLIP, each particle carries not only a velocity but also a local affine approximation of the velocity field. This allows preserving subgrid velocity structure during particle–grid transfers.

2.7.1 Grid-particles interactions

The general idea follows the PIC/FLIP framework, in which the contribution of each particle is distributed to nearby grid nodes using a bilinear kernel with compact support over one cell.

The key difference of the APIC method lies in the introduction of an affine velocity representation carried by each particle, through the matrix $\mathbf{C}_p$. This matrix allows the particle to locally approximate variations of the velocity field, rather than only storing a constant velocity.

Thus, for each particle located at position $\boldsymbol{x}_p$ with velocity $\boldsymbol{v}_p$, the contribution to the velocity at a grid node located at $\boldsymbol{x}_i$ is given by,

$$\boldsymbol{v}_i \mathrel{+}= m_p \, w_{pi} \, (\boldsymbol{v}_p + \mathbf{C}_p(\boldsymbol{x}_i - \boldsymbol{x}_p)) \tag{44}$$

where m_p is the particle mass and w_{pi} is the interpolation kernel.

In this formulation, the affine term $\mathbf{C}_p(\boldsymbol{x}_i - \boldsymbol{x}_p)$ enables the transfer of local velocity gradients from particles to the grid, thereby reducing the loss of information associated with standard PIC transfers.

Once all particle contributions have been accumulated, the grid velocities are normalized by the total mass at each grid node to recover the final velocity field. On the other end, the transfer from grid to particles reconstructs both the particle velocity and its affine coefficients. The new particle velocity is computed by interpolating the grid velocity,

$$\boldsymbol{v}_p^{n+1} = \sum_i w_{pi} \boldsymbol{v}_i \tag{45}$$

Additionally, the affine coefficients are reconstructed by computing weighted moments,

$$\boldsymbol{b}_p = \sum_i w_{pi} \boldsymbol{v}_i (\boldsymbol{x}_i - \boldsymbol{x}_p) \tag{46}$$

$$\mathbf{D}_p = \sum_i w_{pi} (\boldsymbol{x}_i - \boldsymbol{x}_p)(\boldsymbol{x}_i - \boldsymbol{x}_p)^T \tag{47}$$

The affine matrix is then obtained by,

$$\mathbf{C}_p = \boldsymbol{b}_p \mathbf{D}_p^{-1}. \tag{48}$$

In practice, the inversion of $\mathbf{D}_p$ is performed explicitly. When the matrix is singular or nearly singular, the affine contribution is discarded, effectively reverting to a PIC-like update. A similar fallback is applied near Dirichlet boundary cells to avoid instability.

2.7.2 Particle-grid interaction loop

Each time step follows the sequence:

1. gravity is applied to the particles if enabled,
2. particle velocities are transferred to the grid using the APIC formulation,
3. divergence of the velocity field is computed,
4. the pressure Poisson equation is solved,
5. velocity is projected to enforce incompressibility,
6. grid velocities are transferred back to the particles, reconstructing both velocity and affine coefficients,
7. particles are advected using the updated velocity field,
8. particles are created or removed if needed,
9. diagnostic quantities are computed.

2.7.3 Handling of degeneracies

Two situations require special treatment:

- If the moment matrix $\mathbf{D}_p$ is singular, the affine term is discarded and the update reduces to PIC behavior.
- Near solid or Dirichlet boundary cells, the affine reconstruction is damped to prevent numerical instability.

3 Results

In this section, each method implemented in Section 2 is validated through a set of test cases, including inlet flow problems and free-surface flows (when applicable). These cases are selected based on their increasing level of complexity and the availability of reference solutions, either analytical or numerical results from the literature, allowing for meaningful quantitative and qualitative comparisons.

Inlet flow problems will focus on an uniform flow between parallel plates, Taylor-Green vortex and a cylindrical in a flow. Free-surface flows discussed include dam break test cases, as well as sloshing tests. Finally, the thermal coupling is considered, via Rayleigh-Bénard convection, and an idealized heat exchanger.

3.1 Taylor–Green vortex

The Taylor–Green vortex is a classical unsteady benchmark describing the decay of a two-dimensional vortex flow. It admits an exact analytical solution of the incompressible Navier–Stokes equations in Cartesian coordinates, for both inviscid and viscous cases. Owing to its simplicity and well-defined analytical behaviour, it is widely used to assess the accuracy and stability of numerical methods, as explicitely done in [3].

The flow is initialised at $t = 0$ s over the periodic domain $0 \leq x, y \leq \pi$ with the following velocity field,

$$u(x, y, 0) = \sin(x)\cos(y) \tag{49}$$
$$v(x, y, 0) = -\cos(x)\sin(y) \tag{50}$$

with these initial and boundary conditions, the analytical solution is given by

$$u(x, y, t) = \sin(x)\cos(y)F(t) \tag{51}$$
$$v(x, y, t) = -\cos(x)\sin(y)F(t) \tag{52}$$

where $F(t) = e^{-2\nu t}$ is the viscous decay term, with ν the kinematic viscosity. These analytical results will be compared with the numerical solution throughout the simulation. In addition, the normalised kinetic energy $\overline{E}(t)$ is monitored to further evaluate the behaviour of the different numerical schemes. It is defined as

$$\overline{E}(t) = \frac{\int_A u(t)v(t)\mathrm{d}A}{\overline{E}(0)} \tag{53}$$

with

$$\overline{E}(0) = \int_A u(0)v(0), \mathrm{d}A. \tag{54}$$

in order to get a more global parameter to analyse. Let's note that if the decay term for each velocity component is $e^{-2\nu t}$, the decay term for the normalized energy will be $e^{-4\nu t}$. Another definition of the velocity decay term is computed in Gong and Al. [4] and scales as $e^{-2t/Re}$, with x, y being the non-dimension coordinates meaning that $L = 1\,\mathrm{m}$ and using $U_0 = 1\,\mathrm{m/s}$ as it was done here, one has

$$Re = \frac{U_0 L}{\nu} = \frac{1}{\nu} \tag{55}$$

so, the definition corresponds.

As discussed in Subsection 2.5 for the semi-Lagrangian advection, the finite differences used in the different schemes introduce a viscosity-like term in the advection equation, with a coefficient scaling as $u\Delta x$ when the CFL condition is satisfied. In addition, Bridson [1] proposed a theoretical relationship linking the effective numerical viscosity to the spatial and temporal discretisations, scaling as $(\Delta x)^2/\Delta t$. Consequently, even in the absence of physical viscosity, dissipative effects are expected to appear through numerical diffusion, leading to an exponential decay of the flow quantities that can be associated to numerical viscosity.

For consistency, all simulations were performed on the same physical domain, with $L_x = L_y = \pi$ m, and over the same total simulation time, $T = 20$ s. The velocity components were extracted at the point $[x; y] = [\pi/4; \pi/4]$ throughout the simulation. The only varying parameters were the numerical method, the spatial resolution, and the temporal resolution.

First, a convergence study was carried out by varying the spatial discretisation while maintaining a constant CFL number equal to 0.5. For each spatial resolution N_x, the corresponding time step was adjusted to preserve the CFL condition. The numerical parameters employed are summarised in Table 1.

N_x	Δx	N_t	Δt
50	6.2832×10^{-2}	636	3.1416×10^{-2}
100	3.1416×10^{-2}	1273	1.5708×10^{-2}
150	2.0944×10^{-2}	1909	1.0472×10^{-2}
200	1.5708×10^{-2}	2546	7.8540×10^{-3}
250	1.2566×10^{-2}	3183	6.2832×10^{-3}

Table 1: Spatial and temporal discretisation parameters employed in the convergence study.

As a first step, the evolution of the normalised energy $\overline{E}(t)$ was analysed over time.

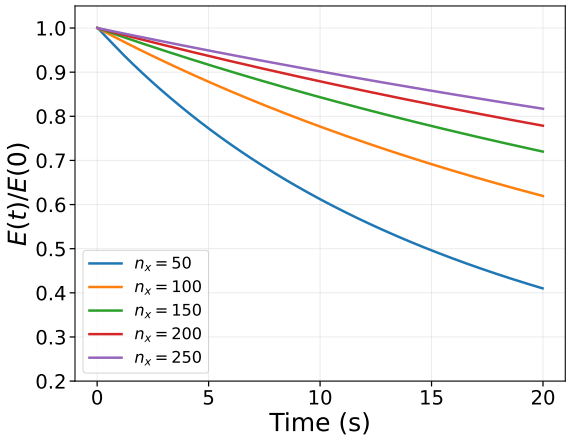


Figure 2: Normalized energy as a function of time for each mesh resolution using the semi-Lagrangian method

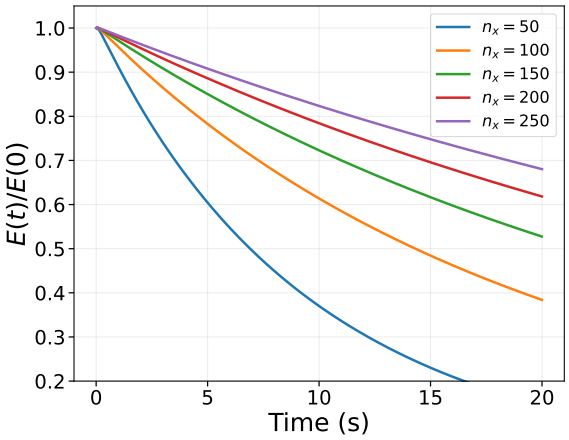


Figure 3: Normalized energy as a function of time for each mesh resolution using the PIC method

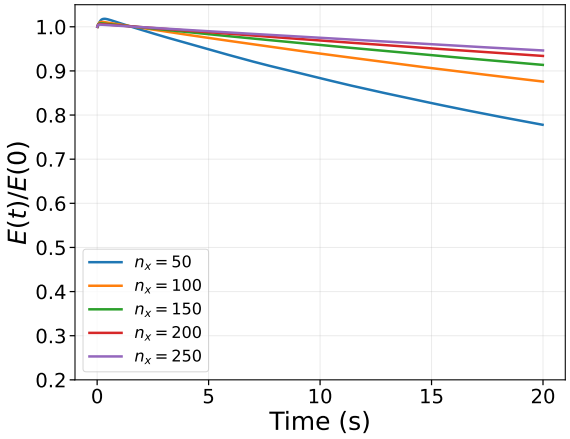


Figure 4: Normalized energy as a function of time for each mesh resolution using the FLIP method

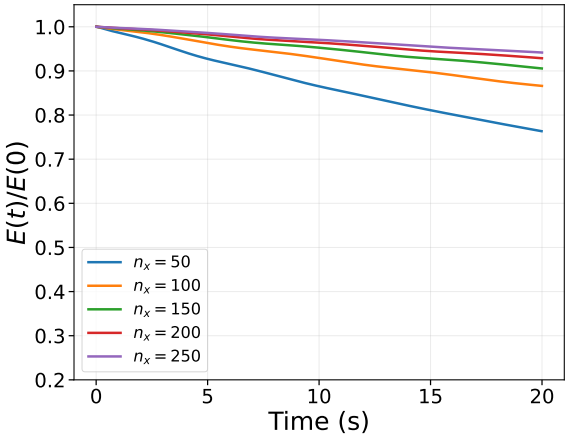


Figure 5: Normalized energy as a function of time for each mesh resolution using the APIC method

Each method dissipates energy as time evolves, although not at the same rate. The semi-Lagrangian and PIC methods introduce significant numerical diffusion as it can be see in Fig. 2 and Fig. 3, leading to a strong damping of the numerical solution and a rapid decrease of the normalized total energy. This behaviour mainly originates from the interpolation and backtracing steps involved in these approaches.

FLIP and APIC were implemented to reduce these dissipative effects, Fig. 4 and Fig. 5 confirm that expected behaviour as both methods perform significantly better and really similarly in terms of energy conservation. Nevertheless, a non-negligible amount of diffusion can still be observed in the results. In all cases, mesh refinement improves energy conservation, as the numerical errors associated with interpolation and discretization are reduced.

It is also worth noting that the total kinetic energy slightly increases during the first few timesteps of the FLIP simulations, particularly on coarse meshes. This behaviour can be explained by the fact that the initial timestep effectively projects the velocity field into a consistent incompressible state. Due to discretization errors—especially pronounced on coarse grids—this projection can introduce a small amount of non-physical energy into the system. Moreover, the FLIP scheme is known to preserve particle kinetic energy with very low numerical dissipation. As a consequence, any small inconsistency or spurious energy introduced during this initial correction is not immediately damped out, but instead carried by the particles for a short period of time. Only after a few timesteps do the usual dissipative mechanisms dominates once again, leading to the expected decay of total energy.

An investigation of the velocity will now be realised to assess the local behavior of these methods. Let's note that the two velocity components were evaluated at the center of the cell containing the point $[x; y] = [\pi/4; \pi/4]$. Depending on the discretisation, the initial numerical value may slightly differ from the exact inviscid solution due to the spatial positioning of the sampling point within the grid.

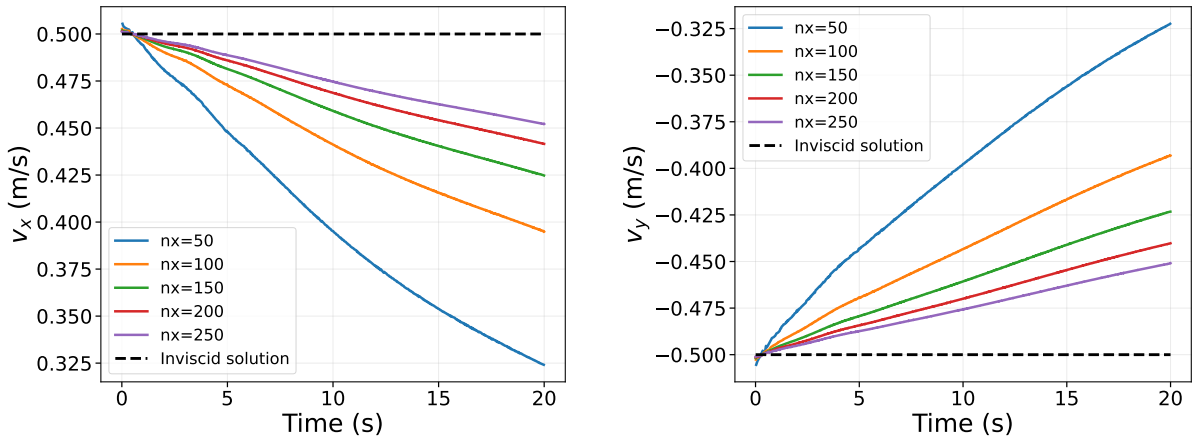


Figure 6: Values of v_x and v_y at $(i; j) = (Nx * dx/4; Nx * dx/4)$ for each mesh resolution, using semi-Lagrangian method

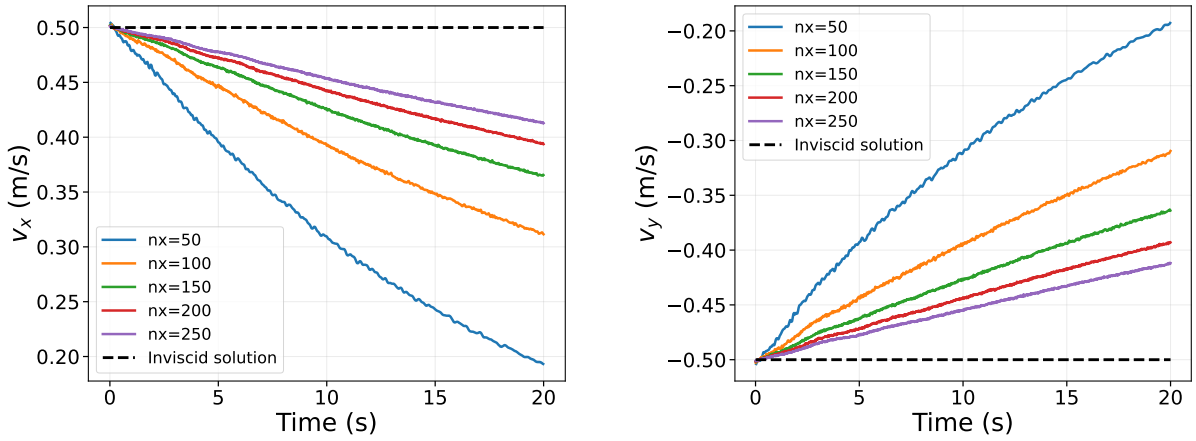


Figure 7: Values of v_x and v_y at $(x; y) = (\pi/4; \pi/4)$ for each mesh resolution, using PIC method

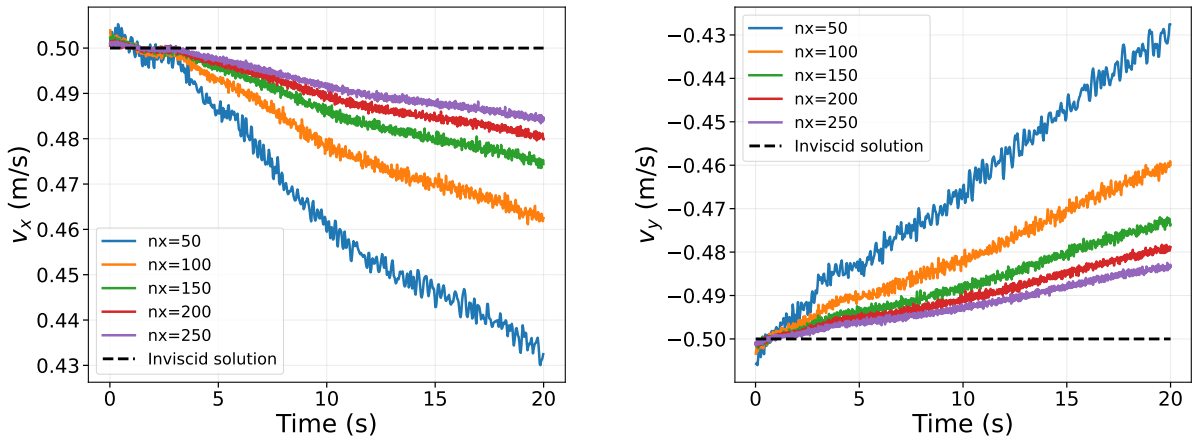


Figure 8: Values of v_x and v_y at $(x; y) = (\pi/4; \pi/4)$ for each mesh resolution, using FLIP method

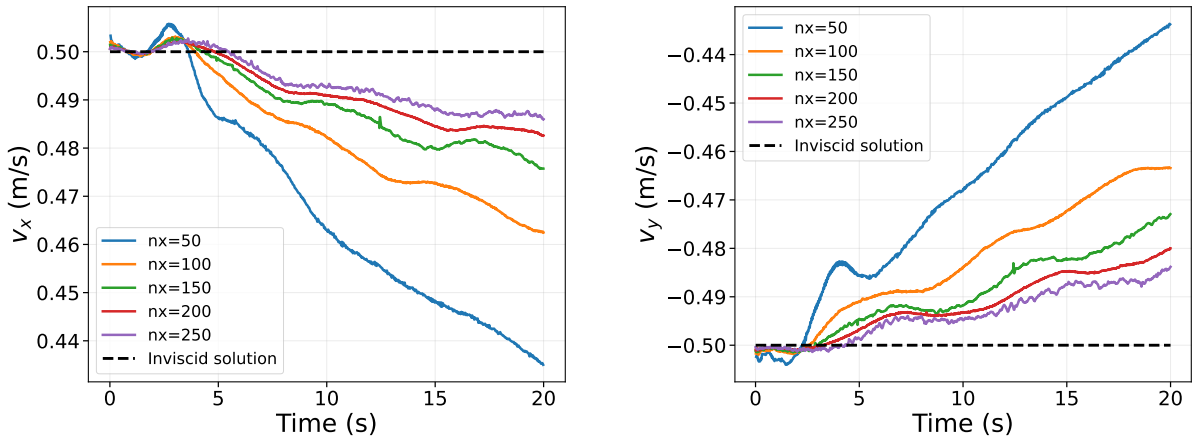


Figure 9: Values of v_x and v_y at $(x; y) = (\pi/4; \pi/4)$ for each mesh resolution, using APIC method

For all methods, mesh refinement consistently leads to solutions that are closer to the analytical (inviscid) reference. However, this improvement comes at the cost of increased computational time. In this subsection, the CFL number was kept constant; therefore, the increase in runtime is influenced by coupled changes in both spatial and temporal discretization parameters (notably N_x and N_t). A more detailed analysis is required to decouple the respective effects of the time step and space step. This would allow a more precise assessment of the computational cost and the actual benefit gained from mesh refinement.

Another clear feature is the overall symmetric behaviour of both velocity components. This symmetry is expected just by looking at the Eqs 52 and the point where the speed is computed, it still constitutes an interesting qualitative observation supporting the validity of the implementations. This symmetry could also indicates that the numerical diffusion introduced by each scheme is essentially isotropic, meaning that dissipation acts uniformly in all directions and does not privilege any particular component of the velocity field. Such isotropy is a desirable property, as anisotropic numerical viscosity would artificially break the physical symmetry of the problem.

Clearly, as it was seen in the normalized energy analysis, the semi-Lagrangian and PIC methods tend to produce overly smooth solutions by damping sharp gradients through interpolation, as illustrated in Figs. 6 and 7. This damping has two main effects over the velocity components : it reduces high-frequency variations in the solution and introduces non-negligible numerical diffusion. The general shape of the curves seems to follow an exponential decay, matching the expected results of the analytical solution.

As a consequence, the numerical results no longer accurately represent an inviscid flow. Instead, the solution amplitude decreases (in absolute value) toward zero in an apparently exponential manner, reminiscent of the exponential decay observed in the viscous analytical solution. As discussed in Subsection 2.5, the numerical diffusion may be interpreted as a form of numerical viscosity. A more detailed investigation of this numerical viscosity is done later on, in order to quantify its magnitude and study its dependence on the discretization parameters.

The same conclusion over the dissipative effect of FLIP and APIC methods can be seen in Figs. 8 and 9. However, their behaviour differs in various points.

For FLIP, the numerical solution exhibits a "plateau" where the flow is almost exactly conserved for around 3 seconds before the dissipative effects starts dominating. Fig 8

also shows an increase in high-frequency oscillations, leading to growing numerical noise. The initial "plateau" seems to be more constant and holds longer with mesh refinement whereas numerical noise is decreased as the mesh is refined.

In contrast, APIC exhibits a less noisy behaviour: the numerical solution remains smoother, with significantly reduced spurious oscillations, and the velocity components evolve in a more regular manner. The same type of "plateau" (but less noisy) can be seen in Fig 9 but another interesting behaviour is the appearance of what looks like solution modes (local maximum and minimum of the velocity components) of different amplitude and at different time. In order to asses this possibility, a spectral analysis of this method would be needed but will not be done here.

Overall, these results highlight the classical trade-off between dissipation and stability. Semi-Lagrangian and PIC methods provide robust and stable solutions, but at the cost of excessive numerical diffusion. In contrast, FLIP preserves fine-scale flow structures more effectively thanks to its low numerical dissipation, although this comes with increased numerical noise and a gradual long-term drift away from the analytical solution. APIC, however, achieves the most satisfactory compromise between these two behaviors. It retains the low-dissipation characteristics of FLIP while significantly improving stability and reducing spurious oscillations.

As discussed in the previously in this section, the decay of the numerical solution in each methods can be investigated by finding the exponential matching the decay.

N_x	$\nu_{\text{num}}(v_x)$	$\nu_{\text{num}}(v_y)$	$\nu_{\text{num}}(\overline{E})$
50	1.1601×10^{-2}	1.1102×10^{-2}	1.1087×10^{-2}
100	6.2509×10^{-3}	6.0437×10^{-3}	5.9482×10^{-3}
150	4.3145×10^{-3}	4.1953×10^{-3}	4.0847×10^{-3}
200	3.2783×10^{-3}	3.1960×10^{-3}	3.1136×10^{-3}
250	2.6548×10^{-3}	2.5935×10^{-3}	2.5164×10^{-3}

Table 2: Estimated numerical viscosity coefficients based on velocity components and normalized kinetic energy decay for the semi-Lagrangian method as a function of mesh resolution.

N_x	$\nu_{\text{num}}(v_x)$	$\nu_{\text{num}}(v_y)$	$\nu_{\text{num}}(\overline{E})$
50	2.4437×10^{-2}	2.4048×10^{-2}	2.4357×10^{-2}
100	1.2173×10^{-2}	1.1960×10^{-2}	1.2013×10^{-2}
150	8.1551×10^{-3}	8.0333×10^{-3}	8.0276×10^{-3}
200	6.1389×10^{-3}	6.0572×10^{-3}	6.0342×10^{-3}
250	4.9246×10^{-3}	4.8669×10^{-3}	4.8375×10^{-3}

Table 3: Estimated numerical viscosity coefficients based on velocity components and normalized kinetic energy decay for the PIC method as a function of mesh resolution.

N_x	$\nu_{\text{num}}(v_x)$	$\nu_{\text{num}}(v_y)$	$\nu_{\text{num}}(E)$
50	4.0910×10^{-3}	3.8834×10^{-3}	3.3993×10^{-3}
100	2.2264×10^{-3}	2.1152×10^{-3}	1.8067×10^{-3}
150	1.4921×10^{-3}	1.4250×10^{-3}	1.2332×10^{-3}
200	1.1419×10^{-3}	1.0913×10^{-3}	9.3633×10^{-4}
250	9.1319×10^{-4}	8.7244×10^{-4}	7.5512×10^{-4}

Table 4: Estimated numerical viscosity coefficients based on velocity components and kinetic energy decay for the FLIP method as a function of mesh resolution.

N_x	$\nu_{\text{num}}(v_x)$	$\nu_{\text{num}}(v_y)$	$\nu_{\text{num}}(E)$
50	4.0578×10^{-3}	3.8824×10^{-3}	3.3713×10^{-3}
100	2.2386×10^{-3}	2.1826×10^{-3}	1.8248×10^{-3}
150	1.4345×10^{-3}	1.4400×10^{-3}	1.2476×10^{-3}
200	1.1181×10^{-3}	1.0934×10^{-3}	9.4426×10^{-4}
250	9.1752×10^{-4}	8.9666×10^{-4}	7.7368×10^{-4}

Table 5: Estimated numerical viscosity coefficients based on velocity components and kinetic energy decay for the APIC method as a function of mesh resolution.

The numerical viscosity coefficients reported in Tables 2 to 5 confirm the differences in dissipation behaviour across the four methods and the consistent values between the values computed from velocity (local) parameters and the normalized kinetic energy (global) parameter. For all schemes, ν_{num} decreases monotonically with increasing mesh resolution, as expected, and the values obtained for v_x and v_y remain consistently close to one another, confirming the suggestion about the isotropic numerical dissipation. The Semi-Lagrangian method exhibits moderate viscosity, ranging from 1.16×10^{-2} at $N_x = 50$ down to 2.6×10^{-3} at $N_x = 250$, while PIC displays the highest dissipation overall, with coefficients roughly twice those of Semi-Lagrangian at every resolution, confirming the diffusive character of the PIC interpolation. In contrast, FLIP and APIC yield substantially lower numerical viscosity values, remaining below 4.1×10^{-3} even at the coarsest resolution and converging towards 9×10^{-4} at $N_x = 250$. FLIP and APIC produce nearly identical coefficients across all tested resolutions, indicating that the affine correction introduced in APIC does not significantly alter the dissipation rate compared to standard FLIP, it simply reduces its noisy behaviour.

The results from Tables 2 and 3 are plotted in Fig 10 and validate the evolution viscosity term coefficient as linear with respect to the space step at constant CFL with characteristic velocity $U_0 = 1\,\text{m/s}$.

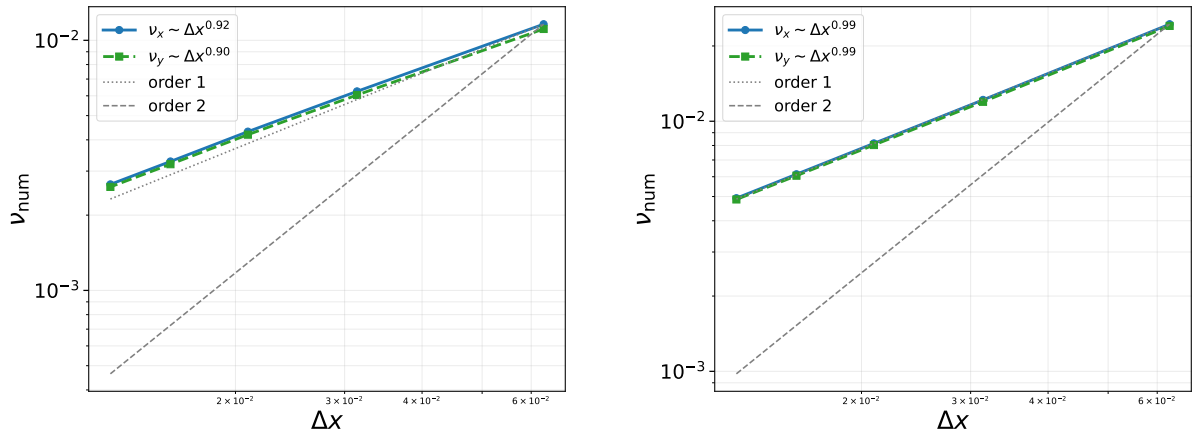


Figure 10: Scaling of the numerical viscosity ν_{num} with respect to the grid spacing Δx for the semi-Lagrangian (left) and PIC (right) methods on a logarithmic scale, for a constant CFL.

3.1.1 Impacts of simulation parameters

One of the main advantages of these implementations is their ability to easily generate complex cases. An interesting investigation would then be the influence of the different governing parameters in order to identify a robust reference parameter set that remains effective across a wide range of cases and configurations. The analysis of the impact of each parameter separately is then needed.

Spatial resolution study Different spatial discretizations were considered while maintaining a constant time step of $\Delta t = 0.01\,\text{s}$ throughout all simulations.

N_x	Δx	N_t	Δt
50	6.2832×10^{-2}	2000	1.0×10^{-2}
100	3.1416×10^{-2}	2000	1.0×10^{-2}
150	2.0944×10^{-2}	2000	1.0×10^{-2}
200	1.5708×10^{-2}	2000	1.0×10^{-2}
250	1.2566×10^{-2}	2000	1.0×10^{-2}

Table 6: Spatial and temporal discretization parameters employed in the spatial convergence study with fixed time stepping.

This specific set of tests allows the spatial discretization errors to be isolated and enables an analysis of the numerical behavior with respect to the mesh resolution. It also provides additional insight and confirms several observations drawn in the previous subsection. The same analysis scheme will be followed.

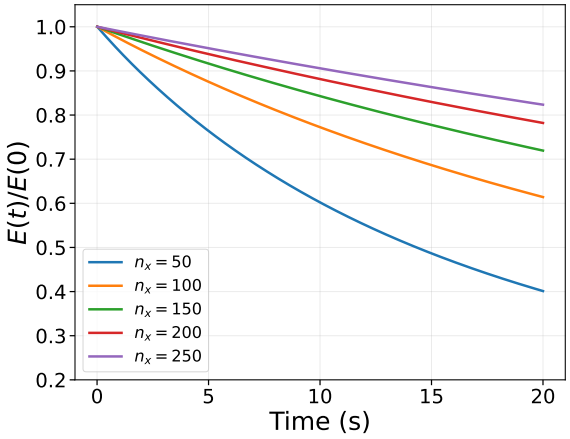


Figure 11: Normalized energy as a function of time for each mesh resolution using the semi-Lagrangian method

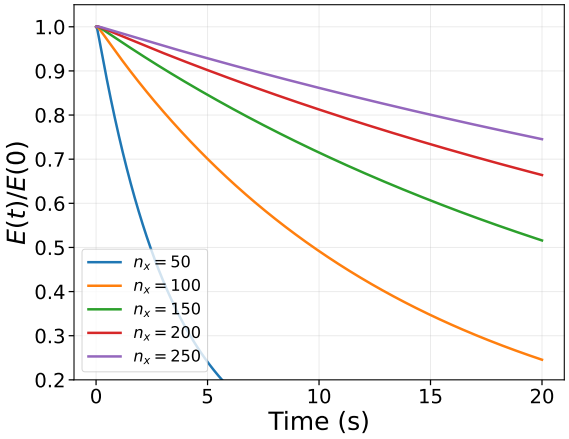


Figure 12: Normalized energy as a function of time for each mesh resolution using the PIC method

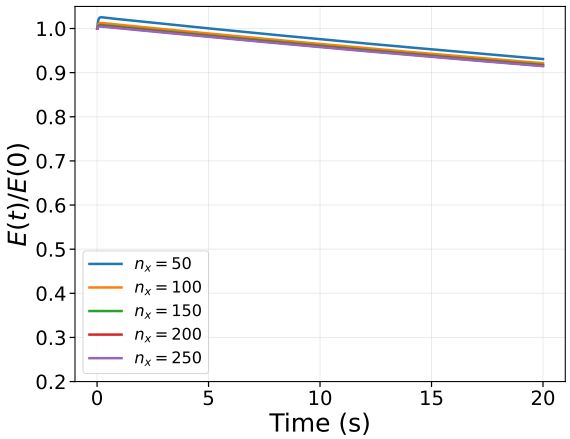


Figure 13: Normalized energy as a function of time for each mesh resolution using the FLIP method

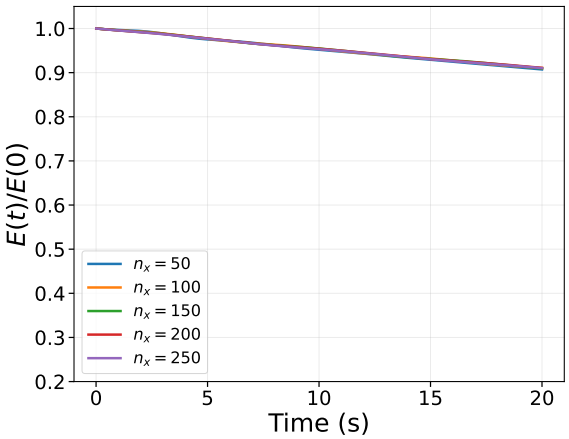


Figure 14: Normalized energy as a function of time for each mesh resolution using the APIC method

The normalized kinetic energy evolution confirms the independence of the numerical dissipation with respect to the spatial resolution for the semi-Lagrangian method and PIC, as Figs. 2 and 11 exhibit nearly identical curves for a given spatial discretization despite the use of different time steps. In contrast, the PIC method now appears to exhibit a dependence proportional to $(\Delta x)^2$, as confirmed by Fig 15, when the time step is fixed.

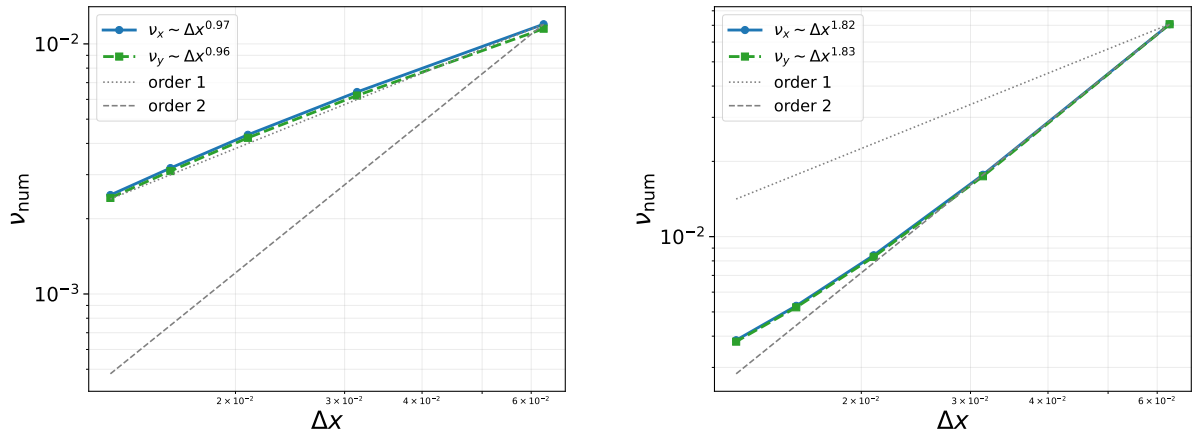


Figure 15: Scaling of the numerical viscosity ν_{num} with respect to the grid spacing Δx for the semi-Lagrangian (left) and PIC (right) methods, for a constant time step.

On the other hand, both the FLIP and APIC methods appear to be weakly dependent on the spatial resolution, since the normalized energy curves corresponding to the different mesh resolutions almost perfectly overlap in Figs. 13 and 14. Moreover, these results are fully consistent with those obtained previously in Figs. 4 and 5, in the simulation using a comparable time-step value.

The analysis of the velocity components provides additional insight into the dependency of the methods on the spatial resolution. The FLIP method appears to be nearly independent of the mesh resolution, provided that the CFL condition remains reasonably satisfied. Indeed, Fig. 16 shows that both the velocity amplitudes and the associated numerical noise remain very similar for all discretizations with $N_x > 50$. This suggests that the method preserves its dynamical behavior consistently across mesh refinements.

In contrast, the APIC method exhibits a stronger local initial sensitivity to the spatial discretization. As illustrated in Fig. 17, for coarse meshes, the velocity components experience a significant local amplification around $t = 3\,\text{s}$, despite the fact that the normalized energy evolution remains globally similar to the other resolutions. It could be due to the oscillatory modes being more uniformly distributed as the mesh is refined. Still, after this velocity jump, the general decay shape and value looks really similar independently of the spatial resolution used. It should however be emphasized that an appropriate balance between mesh refinement and CFL stability must be maintained in order to obtain reliable simulations. Indeed, additional numerical noise becomes visible for the finest discretization ($N_x = 250$), where the CFL condition is no longer strictly respected.

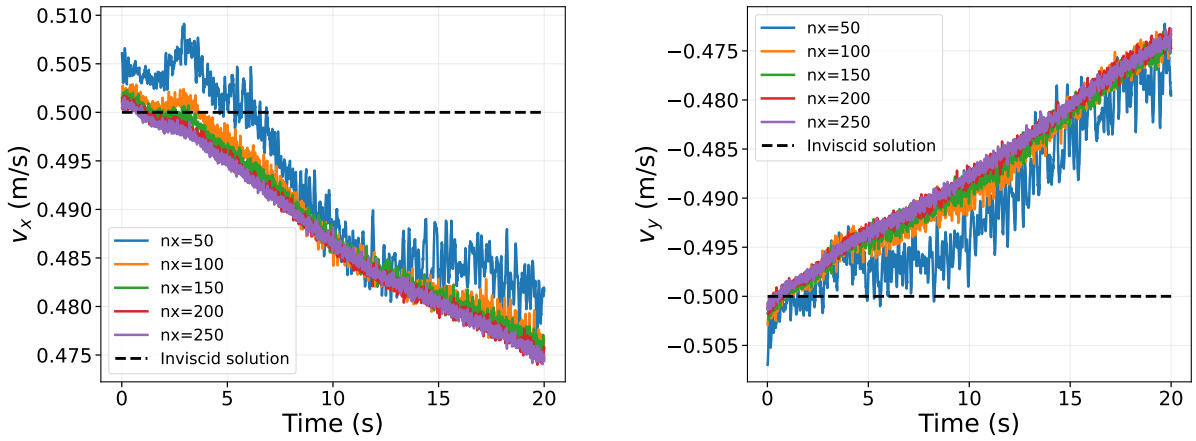


Figure 16: Values of v_x and v_y at $(x; y) = (\pi/4; \pi/4)$ for each mesh resolution, using FLIP method

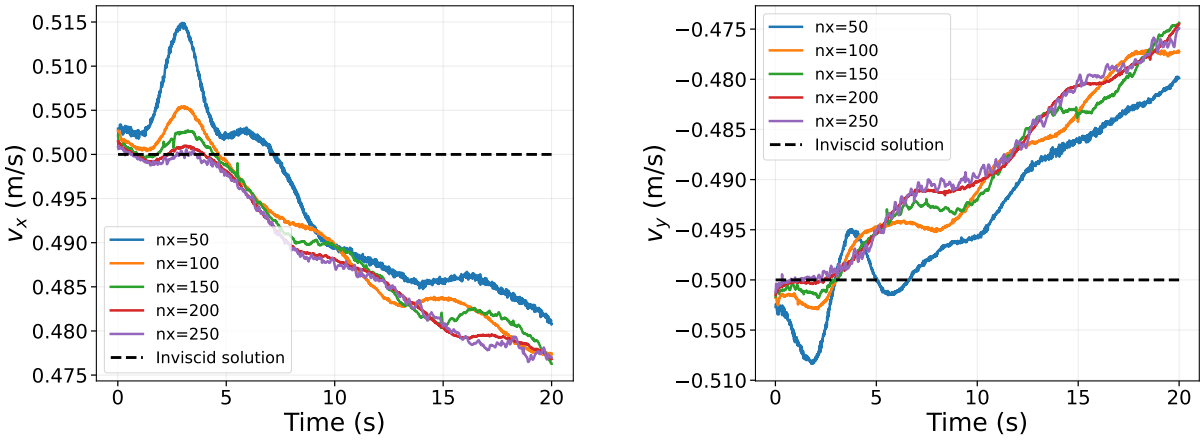


Figure 17: Values of v_x and v_y at $(x; y) = (\pi/4; \pi/4)$ for each mesh resolution, using APIC method

Time resolution The influence of temporal discretization on the simulations was investigated using several time-step sizes while keeping the spatial resolution fixed at $\Delta x = 3.1416 \times 10^{-2}\,\mathrm{m}$ for all simulations.

N_x	Δx	N_t	Δt
100	3.1416×10^{-2}	250	8.0×10^{-2}
100	3.1416×10^{-2}	500	4.0×10^{-2}
100	3.1416×10^{-2}	1000	2.0×10^{-2}
100	3.1416×10^{-2}	2000	1.0×10^{-2}
100	3.1416×10^{-2}	4000	5.0×10^{-3}

Table 7: Temporal discretization parameters used in the convergence study with fixed spatial resolution.

This test seeks to isolate temporal discretization errors and allows a systematic analysis of the numerical behavior as a function of the time-step size. The same analysis scheme will be performed here.

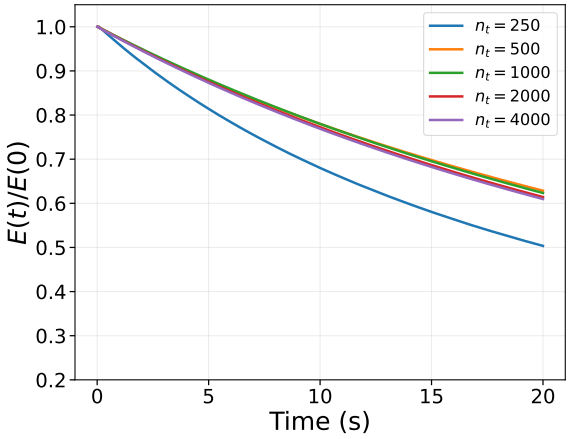


Figure 18: Normalized kinetic energy as a function of time for different time steps using the semi-Lagrangian method

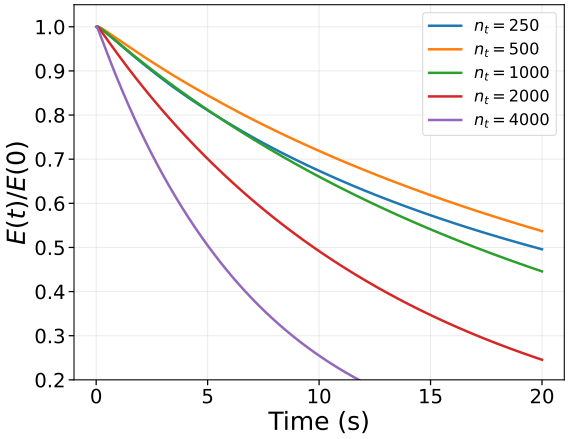


Figure 19: Normalized kinetic energy as a function of time for different time steps using the PIC method

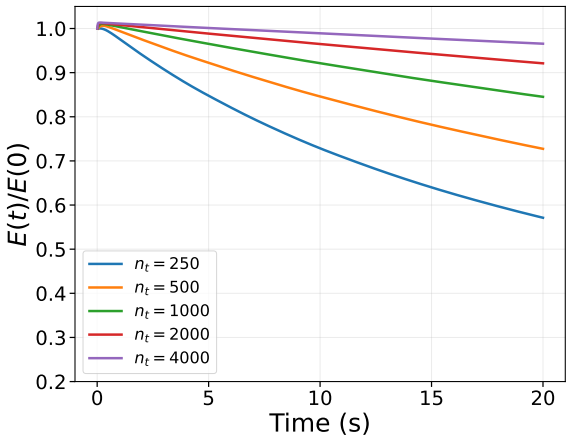


Figure 20: Normalized kinetic energy as a function of time for different time steps using the FLIP method

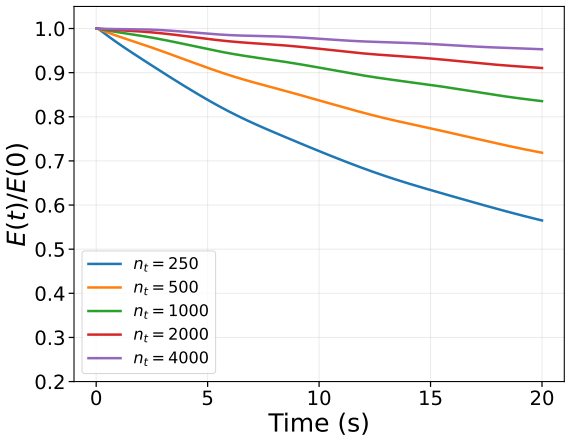


Figure 21: Normalized kinetic energy as a function of time for different time steps using the APIC method

No significant deviation is observed for the semi-Lagrangian and PIC methods: their numerical dissipation follows the previously identified scaling laws. The coarsest time step ($N_t = 250$) violates the CFL condition, which explains the observed discrepancy. Although seemingly counter-intuitive, the increase in numerical dissipation for PIC validates the $(\Delta t)^{-1}$ scaling law at fixed Δx identified earlier. Figure 22 confirms this by showing that, upon discarding the two time steps that violate the CFL condition ($N_t = 250$ and $N_t = 500$), the remaining data points recover the expected scaling behavior.

The most notable effects appear for the FLIP and APIC methods. In these cases, reducing the time step significantly improves kinetic energy conservation and reduces numerical dissipation. This highlights their stronger sensitivity to temporal resolution compared to spatial resolution.

The analysis of the velocity components further supports these conclusions: decreasing the time step improves agreement with the expected inviscid behaviour and reduces numerical noise, while preserving the overall flow structure.

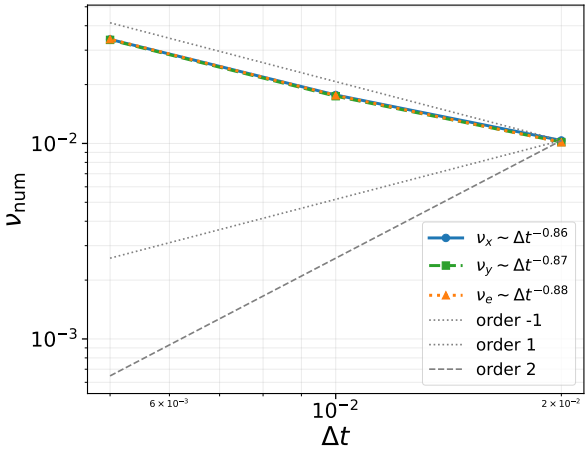


Figure 22: Scaling of the numerical viscosity ν_{num} with respect to the grid spacing Δx for PIC method on a logarithmic scale, for a constant space step.

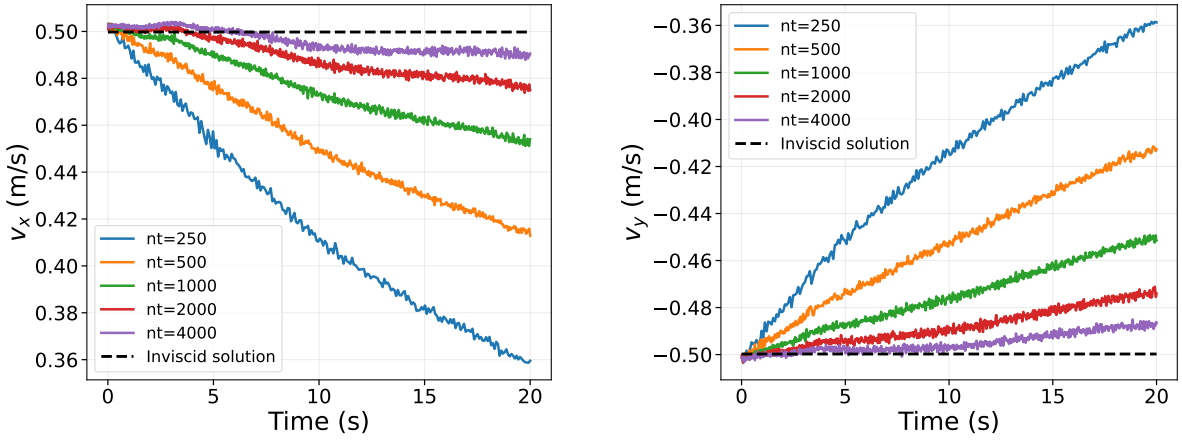


Figure 23: Velocity components v_x and v_y at $(x, y) = (\pi/4, \pi/4)$ for the FLIP method at different time steps

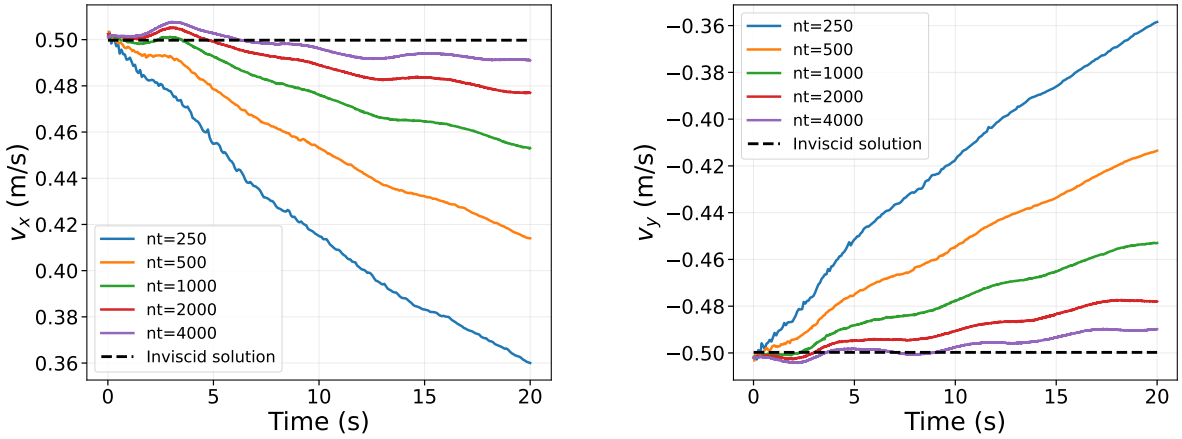


Figure 24: Velocity components v_x and v_y at $(x, y) = (\pi/4, \pi/4)$ for the APIC method at different time steps

3.2 Cylindrical in an uniform flow

The flow past a circular cylinder in a uniform stream is one of the classical benchmark problems in fluid mechanics and computational fluid dynamics. Although no closed-form analytical solution exists for the general viscous case governed by the Navier–Stokes equations, this configuration has been extensively investigated both experimentally and numerically. As a result, it provides a robust and widely accepted validation case for assessing the accuracy and physical fidelity of numerical methods.

Due to the absence of an analytical solution, the validation approach differs from the one adopted in the previous test case. Instead of comparing the solution to an exact reference, the validation is based on the periodicity of the pressure and velocity fields generated by the alternating vortex shedding behind the cylinder. In particular, a pressure-based coefficient evaluated on the cylinder surface, analogous to a lift coefficient, exhibits a periodic behavior from which the vortex shedding frequency can be extracted through a Fourier analysis.

The associated Strouhal number is defined as

$$St = \frac{fD}{U_\infty}, \tag{56}$$

where f is the vortex shedding frequency, D is the cylinder diameter, and U_∞ is the free-stream velocity. In this study, a computational domain of dimensions $L_x = 2\,\mathrm{m}$ and $L_y = 1\,\mathrm{m}$ is considered, containing a circular cylinder of diameter $D = 0.1\,\mathrm{m}$ centered at $[x, y] = [0.25,\, 0.5]$. The simulations are performed over a total physical time of $T = 7.5\,\mathrm{s}$.

In order to produce results comparable with those available in the literature, several geometric and numerical considerations must be satisfied. First, the cylinder diameter should not exceed approximately 10% of the channel opening. This ensures that blockage effects remain limited, so that the inlet velocity accurately represents the free-stream velocity experienced by the cylinder while allowing the shed vortices to evolve naturally without excessive confinement effects.

It is also important to ensure that the cylinder diameter is sufficiently resolved by the computational grid. A coarse spatial discretization would lead to a poor geometric representation of the obstacle and consequently affect the accuracy of the pressure and velocity fields around the cylinder. Finally, the downstream length of the domain should be at least ten times larger than the cylinder diameter in order to avoid any interaction between the wake and the outlet boundary condition.

Different spatial discretizations were considered while keeping the CFL number constant (CFL = 0.2), in order to study the evolution of the numerical results as the mesh is refined. The vertical spatial discretization N_y was prescribed and the corresponding numerical parameters were adjusted accordingly to preserve the same CFL condition throughout all simulations.

N_y	Δx	N_x	N_t	Δt
100	1.00×10^{-2}	150	3750	2.00×10^{-3}
150	6.67×10^{-3}	225	5624	1.33×10^{-3}
200	5.00×10^{-3}	300	7500	1.00×10^{-3}

Table 8: Spatial and temporal discretization parameters employed in the Strouhal number study.

The previous test case demonstrated that non-negligible numerical viscosity exists in each of the implemented methods. In the present test case, this manifests through an effective Reynolds number, defined as

$$Re_{\text{eff}} = \frac{U_0 D}{\nu_{\text{num}}}, \tag{57}$$

where U_0 is the free-stream velocity and D is the cylinder diameter. Depending on its value, Re_{eff} can be linked to a corresponding Strouhal number via an appropriate empirical correlation. The purpose of this test case is therefore to verify that each numerical scheme reproduces benchmark values, thereby providing a physically meaningful validation beyond purely mathematical convergence measures.

As established in the previous subsection, the numerical viscosity of the semi-Lagrangian scheme depends only on Δx. Taking the value reported for $N_x = 250$ in Table 2 and halving it to account for the spatial step being twice as large in the present configuration, the numerical viscosity is of order 10^{-3}, giving $Re_{\text{SL}} \approx 200$. For PIC, the law introduced in [1] and verified in Subsection 3.1 gives

$$\nu_{\text{PIC}} = \frac{(\Delta x)^2}{6\,\Delta t} \geq 4.16 \times 10^{-3}, \tag{58}$$

which yields $Re_{\text{PIC}} \leq 48$. For both FLIP and APIC, the numerical viscosity scales with the time step rather than the grid spacing; by extrapolating from Tables 4 and 5, it is of order 10^{-4}, so $Re_{\text{FLIP}} \approx Re_{\text{APIC}} \approx 2000$.

The values of the pressure based coefficient associated to each methods are represented in Figs 25 - 28

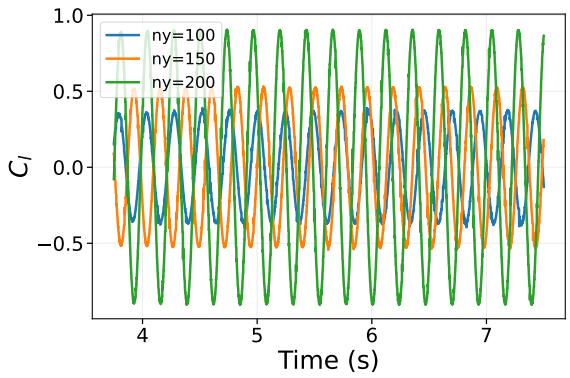


Figure 25: Pressure based lift coefficient for the spatial resolution, using the semi-Lagrangian method

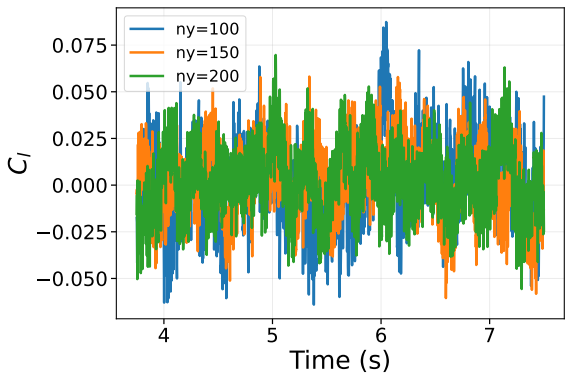


Figure 26: Pressure based lift coefficient for the spatial resolution, using the PIC method

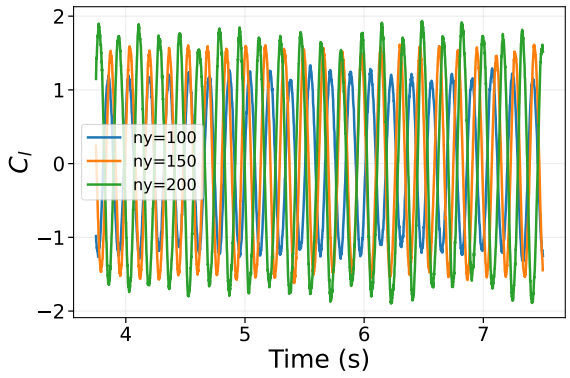


Figure 27: Pressure based lift coefficient for the spatial resolution, using the FLIP method

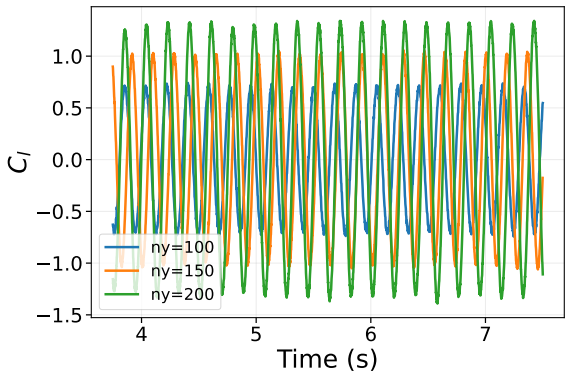


Figure 28: Pressure based lift coefficient for the spatial resolution, using the APIC method

As can be clearly seen in Fig. 26, the pressure-based lift coefficient remains essentially constant in time, indicating the absence of periodic vortex shedding. This behavior is consistent with the very low effective Reynolds number associated with this configuration, for which the flow remains steady and strongly damped.

Extracting the main frequency of these signals using a Fourier transform, one obtains the following results:

ny	f [Hz]	St
100	4.0000	0.2000
150	4.5341	0.2265
200	4.2667	0.2133

Table 9: Results obtained with the semi-Lagrangian method

ny	f [Hz]	St
100	5.8667	0.2933
150	6.1344	0.3067
200	5.8667	0.2933

Table 10: Results obtained with the FLIP method

ny	f [Hz]	St
100	5.3333	0.2665
150	5.6010	0.2800
200	5.3333	0.2665

Table 11: Results obtained with the APIC method

When comparing these results with those reported in [5], it can be observed that the semi-Lagrangian and APIC methods reproduce the expected trend reasonably well, with Strouhal numbers remaining in the correct order of magnitude and showing weak dependence on mesh refinement.

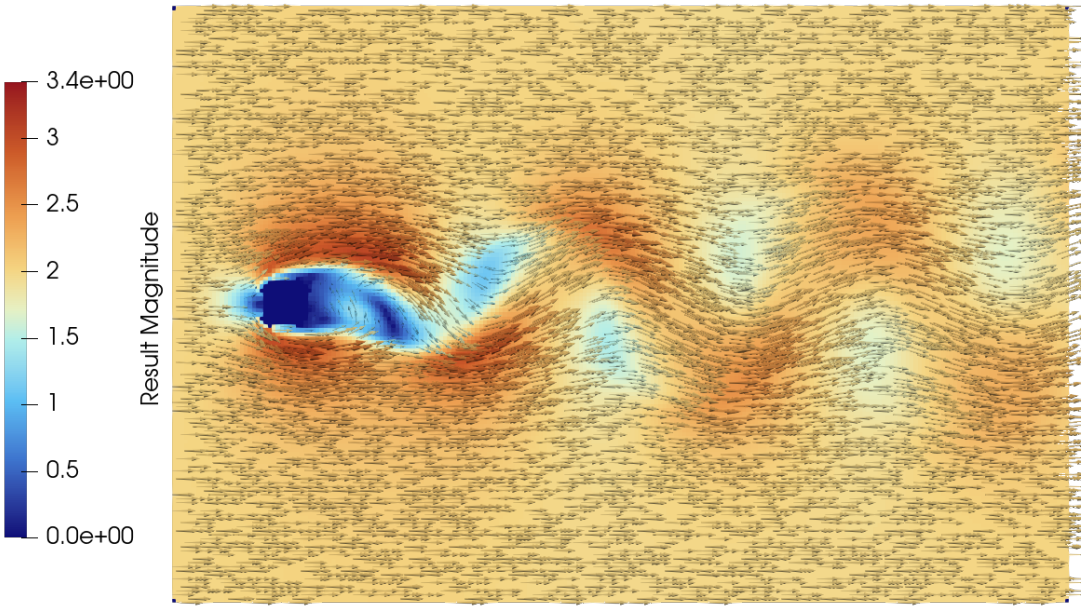


Figure 29: Velocity magnitude over the domain at t=5.824 s, for semi-Lagrange method with $N_y = 150$

On the contrary, the FLIP method yields systematically higher Strouhal numbers than expected. However, this result should not necessarily be interpreted as a failure of the method, since the effective numerical dissipation may have been underestimated in the present analysis. In fact, values of the Strouhal number around $St \approx 0.3$ for Reynolds numbers $Re \geq 2 \times 10^5$ can still be observed in the literature for certain flow regimes and numerical/experimental configurations, as discussed in [6].

To compare the methods more quantitatively, the velocity fields were computed for each scheme. As expected, PIC does not produce any vortices 30, while the semi-Lagrangian method dissipates them rapidly 29. FLIP 31 and APIC 32 yield almost identical results, although APIC exhibits a more periodic behaviour, consistent with theoretical expectations.

3.3 Dam break

A classic test for free-surface flows is the dam break. This problem consists in starting with a block of fluid at rest at one edge of the domain, and letting it collapse under the effect of gravity. In this project, the experimental setup of Lobovský et al. was reproduced numerically [7], which is illustrated in Fig. 33.

Two values were compared with the experimental data obtained by Lobovský et al., which are the mean speed of the wavefront and the impact pressure on the right 15 mm above the bottom of the container.

The parameters that were studied are the spatial step Δx and the time step Δt. Each parameter was varied independently of the others, and this study was realized for both the PIC/FLIP and APIC solvers. No tests were realized with the semi-Lagrangian solver, as it does not support free-surface flows. The default values for the tests are provided in Sec. B.3. It should be noted that all PIC/FLIP simulations used a ratio of 5%/95%, as it has the advantage of limiting the dissipation of the PIC method, while having less noise than the pure FLIP method. Such hybrid methods are used in the literature [8].

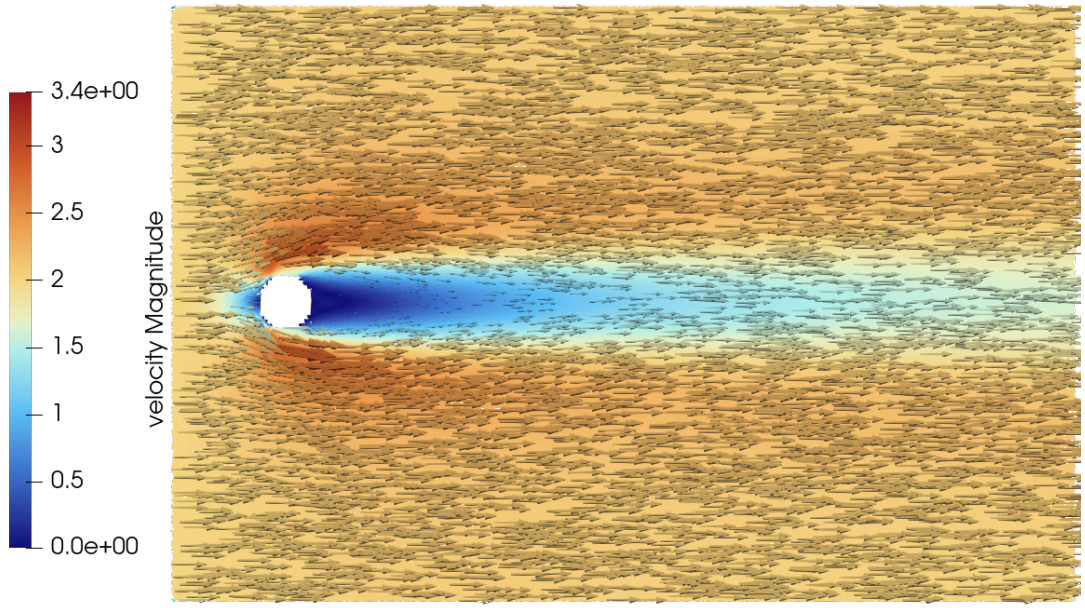


Figure 30: Velocity magnitude over the domain at t=5.824 s, for PIC method with $N_y = 150$

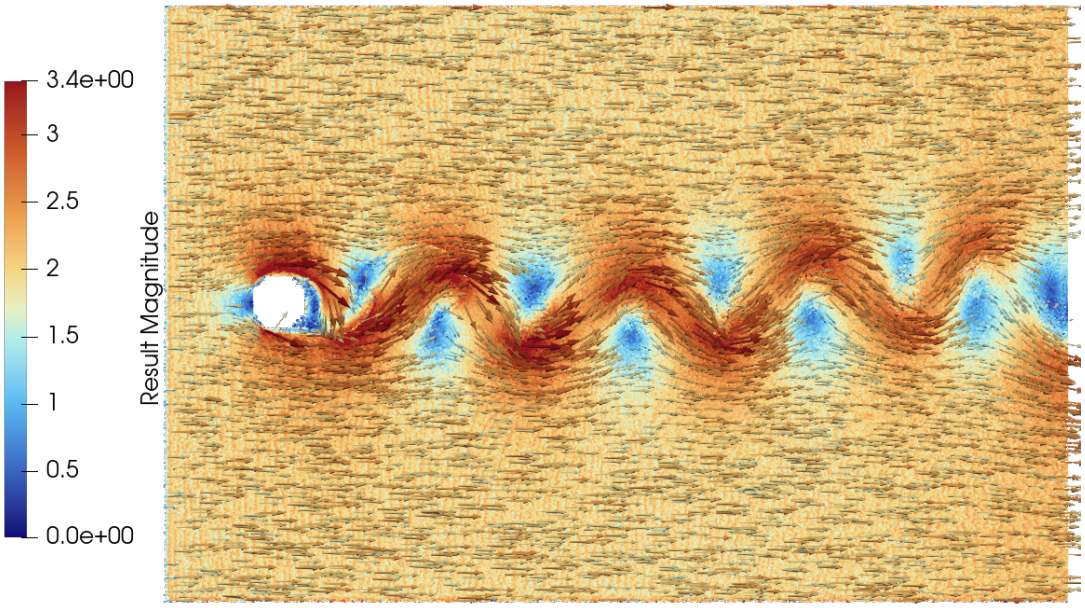


Figure 31: Velocity magnitude over the domain at t=5.824 s, for FLIP method with $N_y = 150$

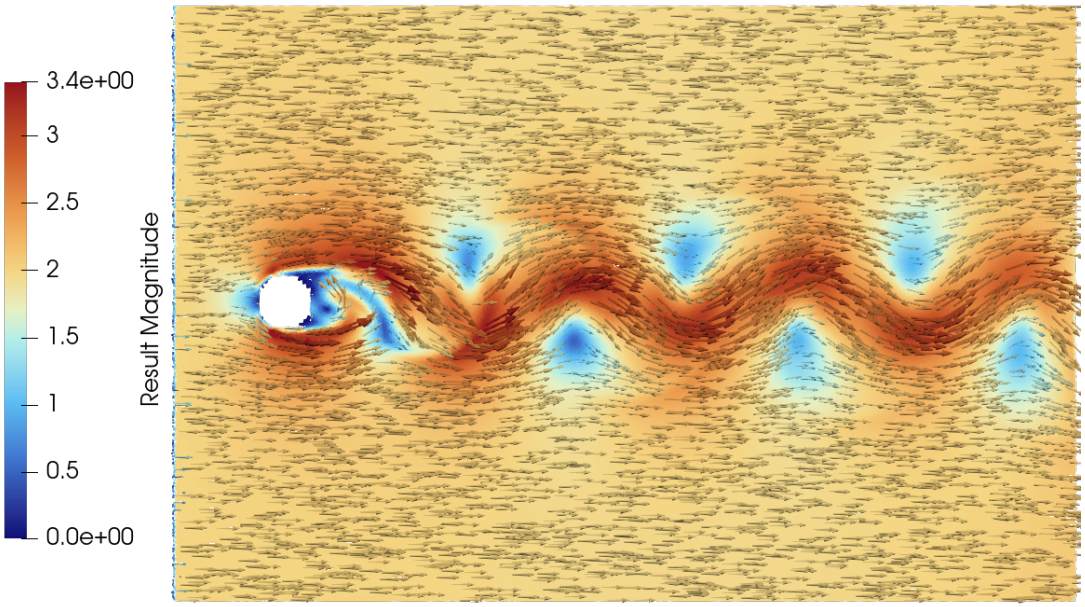


Figure 32: Velocity magnitude over the domain at t=5.824 s, for APIC method with $N_y = 150$

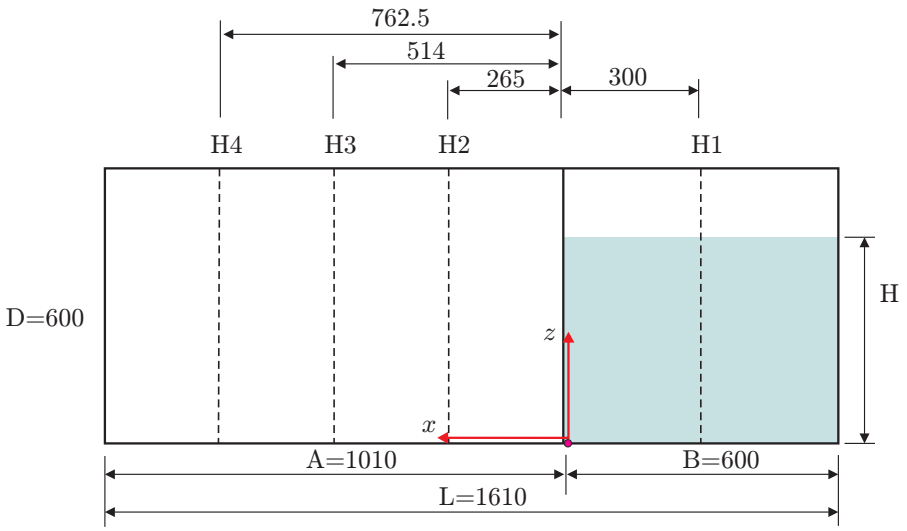


Figure 33: Geometry of the dam break test considered by Lobovský et al. [7]. The same dimensions were used for the tests here, with the water on the left, and $H = 300\,\text{mm}$. The height of the fluid was not considered, hence the positions of the height sensors `Hn` can be ignored. Units are in mm.

The non-dimensional average velocity of the wavefront has been studied in experiments, and reported values range from 1.14 to 1.74, with a mean of 1.47 and a standard deviation of 0.2 [7]. Values from simulations realized here are shown in Fig. 34. All are between 1.3 and 1.45, with the exception of a single test realized with very few cells ($n_x = 53$ along the length of the domain).

The influence of the spatial step seems to be minimal, except for extreme values such as the test $n_x = 53$. For simulations using the PIC/FLIP solver, a small slope in the wavefront speed can be seen near $n_x = 200$ (Fig. 34a). Such phenomenon is not present for those using the APIC solver, which thus hints at a faster convergence with respect to the size of the spatial step for this solver.

The situation is quite different when considering the influence of the time step. Simulations using the APIC solver converge towards a value similar to the ones obtained when studying the spatial step, with only a single lower value for the lowest number of time steps considered ($n_t = 1000$), as well as for $n_t = 17082$. However, results obtained with the PIC/FLIP solver exhibit a decreasing speed with respect to the number of time steps. This can be explained by the growing numerical viscosity when decreasing the duration of the time step for the PIC method, which is a result corroborated with the Taylor-Green results (Sec. 3.1).

The other quantity studied in this report was the impact pressure at $z = 15\,\text{mm}$. Lobovský et al. realized a hundred similar experiments, leading to a median of 5749 Pa, a 97.5 percentile of 6331 Pa, and a 2.5 percentile of 5326 Pa [7].

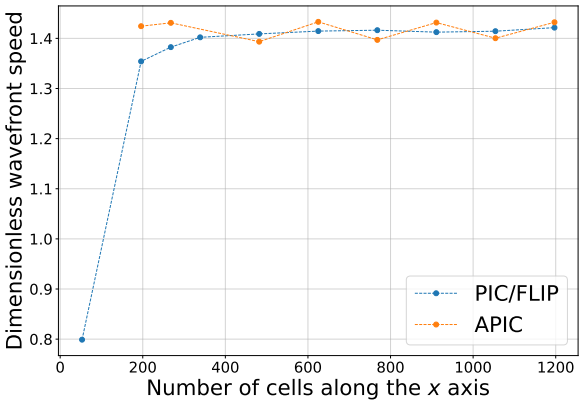
Results obtained numerically are shown in Fig. 35. When refining the grid, simulations using the PIC/FLIP solver converged towards values similar to the experimental ones, as, starting from $n_x = 339$, the impact pressure obtained was comprised between 5488 Pa and 5909 Pa, with a mean value of 5698 Pa.

Simulations using the APIC method exhibit much less reliable results. This is possibly due to the implementation here, more specifically to the inversion of the $\boldsymbol{D}$ matrix. In any case, values obtained when varying the spatial step follow a similar curve to the one outlined by the PIC/FLIP results, except for two outliers, which go up to 31 752 Pa.

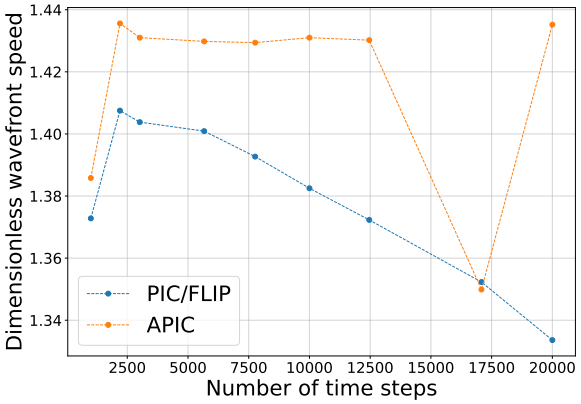
The impact of the duration of the time step lead to similar results as those obtained when considering the speed of the wavefront, for the PIC/FLIP solver. The mean impact pressure is 6395 Pa when the number of time steps is below or equal to 10000. This value is high compared to the experimental results, but that is due to the resolution of the grid being too low ($n_x = 268$). However, when the number of time steps becomes greater than 10000, the values of the impact pressure start growing quite fast, reaching 44 203 Pa for $n_t = 20000$.

Simulations using the APIC solver give more stable results, even if an outlier is present at $n_x = 2199$. The obtained values are however higher than the ones from the PIC/FLIP simulations, when $n_t \leq 10000$. Afterwards, a small increase in the values from the APIC simulations can be seen, but it is much smaller than the one of the PIC/FLIP values.

Finally, the profile of the free surface can be compared qualitatively, as in Fig. 36. It can be observed that the profile obtained with a PIC/FLIP simulation using 911 cells along the length of the domain is very similar to the one obtained experimentally by Lobovský et al. [7]. This is not what is expected, as the simulations aim to reproduce the behaviour of perfect fluids, while experiments were realized with water. An analytical solution developed by Ritter exists for dam break tests with perfect fluids, which concludes that the curvature of the free surface profile should not change of sign [9]. This shows that there is a non-negligible numerical viscosity, as developed in Sec. 3.1.

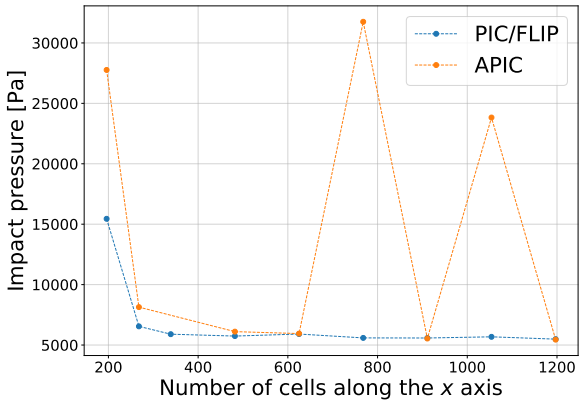


(a) Wavefront speed as a function of the number of cells along the x axis.

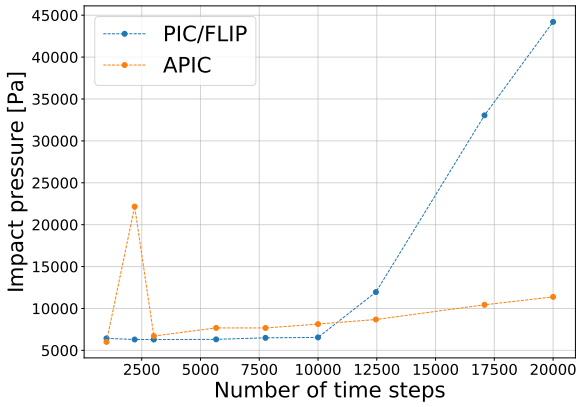


(b) Wavefront speed as a function of the number of time steps.

Figure 34: Dimensionless speed $v/\sqrt{gH}$ of the wavefront as a function of numerical parameters, for dam break tests.



(a) Impact pressure as a function of the number of cells along the x axis.



(b) Impact as a function of the number of time steps.

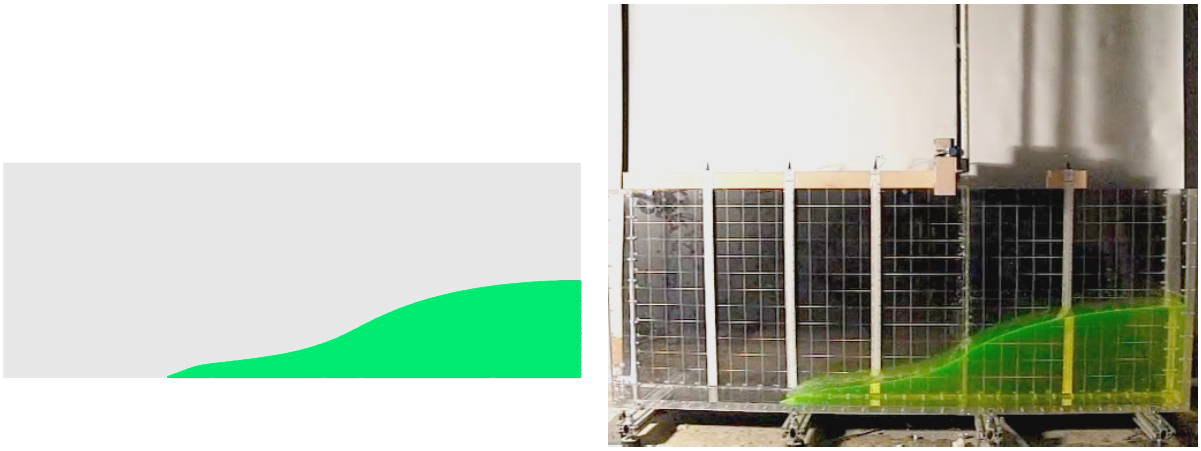
Figure 35: Impact pressure of the wavefront on the right wall of the domain, at $z = 15\,\text{mm}$, as a function of numerical parameters, for dam break tests.

Thus, it can be concluded that the implemented methods give physical results for the dam break problem, when the grid is not too coarse ($n_x \geq 200$), and with a time step duration staying above $2 \times 10^{-4}\,\text{s}$. It is worth noting that the APIC method yields more variable results, with several outliers present.

3.4 Sloshing

Sloshing is a phenomenon common in moving containers which are partially filled. Accordingly, this problem has been studied extensively in the literature [10]. Thus, even if the implemented solver is not able to reproduce oscillations of the container, studying the natural frequencies of the fluid when initialized with a sinusoidal free surface is still of interest.

The geometry studied by Chen and Xue [10], which is depicted in Fig. 37, was reproduced. The natural frequency of the oscillations was studied via the pressure at the bottom left of the tank, as well as via the height of the fluid on its left. The obtained



(a) Free surface profile for a PIC/FLIP simulation, with $n_x = 911$.

(b) Free surface profile from Lobovský et al. [7].

Figure 36: Free surface profile of a dam break test at $t = 0.276\,\mathrm{s}$.

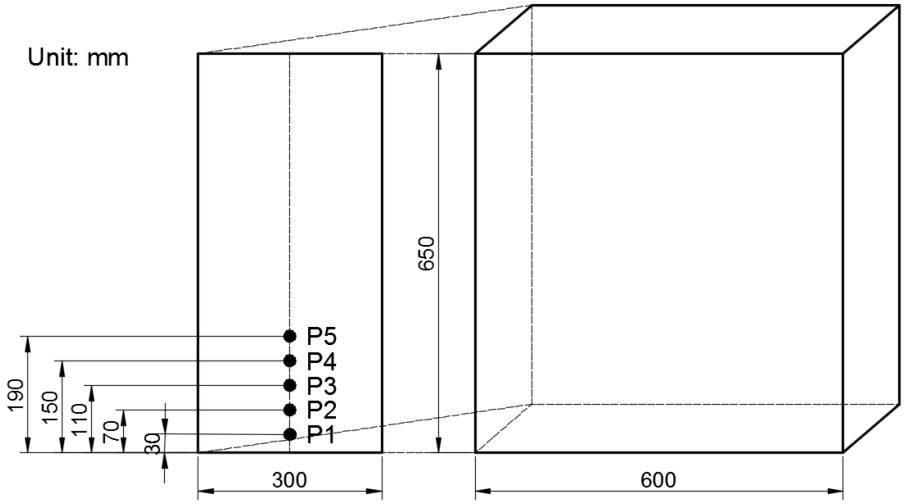


Figure 37: Geometry of the sloshing problem considered by Chen and Xue, which is reproduced here [10], with the container filled at 13.8 % of its capacity.

values of the frequency were compared with the analytical result obtained from potential flow theory [10], which are given by

$$\omega_n = \sqrt{g\frac{\pi n}{L}\tanh\left(\frac{\pi n}{L}h\right)}, \tag{59}$$

with n the mode number and h the filling level.

The parameters that will be varied in this study are the same as for the dam break test, which have been detailed in Sec. 3.3. The fluid was initialized with a sinusoidal free surface, with a single period and an amplitude of 45 mm. The tank was filled at 13.8 %, which corresponds to a mean height of 90 mm.

From Eq. 59, it is deduced that the first harmonic should be at 0.7559 Hz, the second at 1.3842 Hz, and the third at 1.8620 Hz. The frequencies were obtained by fitting damped sinuses of the form

$$s(t) = (At + B)\sin(2\pi f t + \phi) + C \tag{60}$$

on the data, with A, B, C, ϕ, f parameters to optimize, and f the frequency of interest.

It results that the PIC/FLIP and APIC methods were able to capture the fundamental harmonic quite well, as the obtained values are between 0.76 Hz and 0.79 Hz for the PIC/FLIP simulations, and between 0.77 Hz and 0.82 Hz for the APIC ones, except for some edge cases.

The impact of the number of cells along the x axis is low, as can be seen in Fig. 38. The fundamental frequency captured is too low when the number of cells is very low ($n_x = 37$). However, starting at $n_x = 275$, the value captured by the PIC/FLIP simulations is very stable, with variations only starting at the third decimal, the two first being 0.78 Hz. This is the case for both measurements using the pressure, and using the depth.

The results from APIC simulations are less stable. The obtained frequencies are higher than the ones resulting from the PIC/FLIP method for most of the parameter range. A decrease can nonetheless be observed for the second half of the range. This is possibly caused by the lack of numerical dissipation in the APIC method, which is more prominent in the PIC/FLIP method.

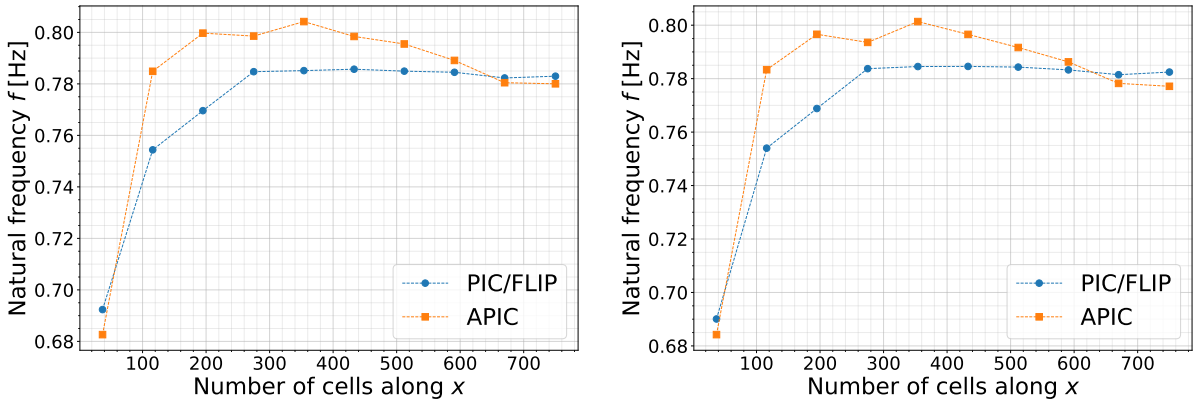
Interesting results appear when considering the influence of the number of time steps. While the frequencies are stable for most of the parameter range, larger values are obtained for smaller number of time steps, as illustrated in Fig. 39. As those are located in the neighbourhood of 1.32 Hz, it can be concluded that the second harmonic is prominent for those simulations. However, as this is still lower than the theoretical frequency of 1.38 Hz, it can be concluded that the fundamental harmonic is still not negligible in those tests. Moreover, the frequency obtained with the APIC method is generally higher than the one of the PIC/FLIP simulations, which is consistent with results from above.

Additionally, the fundamental frequencies obtained here are generally higher than the theoretical result of 0.7559 Hz. This hints at the presence of a second harmonic in all simulations, but weaker than the fundamental one. To confirm this, the pressure and depth measurement for a simulation using the default parameters outlined in Sec. B.4 were plotted in Fig. 40 and Fig. 41, with the fitted sinuses.

From these figures, it is clear that the depth measurement is more noisy and less precise than the pressure measurement. This has two causes. The first is that the depth measurement resolution is limited by the size of the grid, whereas the pressure can take a continuum of values. The second is due to the handling of free surfaces in this project. As only a single particle is needed to cause a cell to be considered liquid, and as projections to happen at the free surface, additional noise is present.

Moreover, these two signals are not regular sinuses. It appears that higher harmonics are indeed present. To further confirm this, the fast Fourier transforms (FFT) of the two signals were computed, and are shown in Fig. 42 and Fig. 43. As the frequencies of interest are quite small, the resolution in the chosen range is coarse. Nonetheless, a first peak is observed, which is near the theoretical fundamental frequency. A second peak can be seen, which is smaller than the first, and which is around the second harmonic. Finally, a third peak, smaller than the first two, is present in the FFT of the depth measurements. It might correspond to the fourth harmonic, which has a theoretical frequency of 2.229 Hz, but the small amplitude of this data, coupled with the low resolution of the FFT, makes this uncertain.

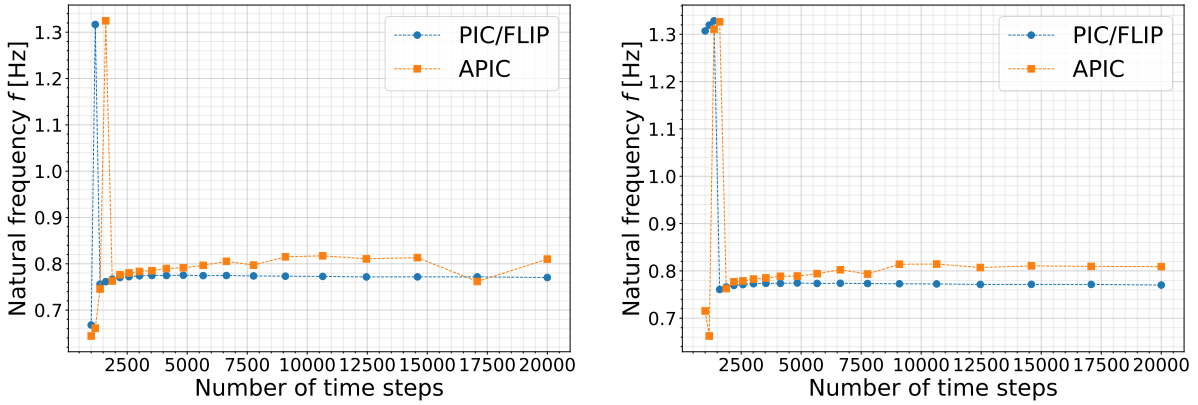
From these results, it can be concluded that the PIC/FLIP and APIC methods give reliable results for sloshing problems without excitation, with the former method performing better than the latter. It should be noted that the minimal number of cells along the x axis is larger for this problem ($n_x \geq 300$) than for the dam break one ($n_x \geq 200$).



(a) Measures using the pressure.

(b) Measures using the depth.

Figure 38: Frequency of the oscillations for a sloshing test as a function of the number of cells along the x axis, based on the pressure at the bottom left of the container, and on the depth of the liquid on the left of the domain.



(a) Measures using the pressure.

(b) Measures using the depth.

Figure 39: Frequency of the oscillations for a sloshing test as a function of the number of time steps, based on the pressure at the bottom left of the container, and on the depth of the liquid on the left of the domain.

However, there is no problem with smaller durations of time steps in this case.

3.5 Rayleigh-Bénard convection

Rayleigh-Bénard convection is a problem where the flow resulting from a hot plate at the bottom of the domain and a cold one at the top is studied. In this section, the number of convection cells n appearing when considering elongated domains is studied. Poel et al. have found that, when considering an aspect ratio Γ such that $1 \leq \Gamma \leq 12$, the ratio n/Γ approaches 0.77 for high Rayleigh numbers [11]. As in this study perfect fluids are considered, it can be assumed that the Rayleigh number obtained is very high, even if numerical dissipation is present.

Physical and thermal properties used were those of water at 325 K [12]. The bottom plate was set to 350 K, while the top one was at 300 K. Left and right walls were isolated. The exact parameters used are given in Sec. B.5. Several aspect ratios were considered:

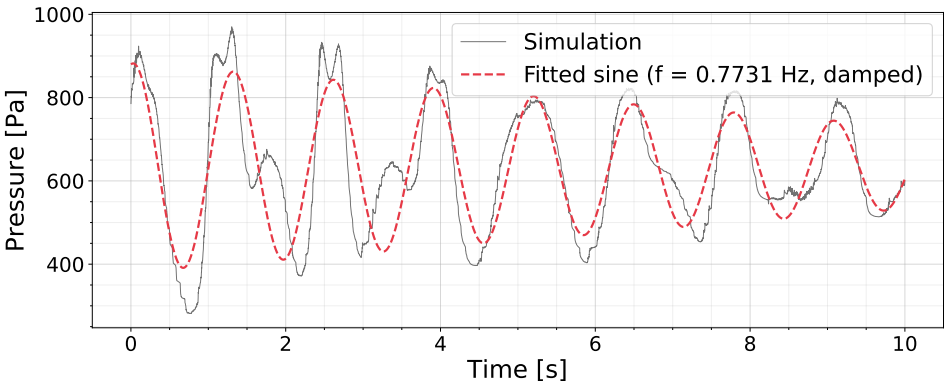


Figure 40: Pressure at the bottom of the container of a sloshing test as a function of time for a simulation with default parameters, as well as a fitted damped sine on this data.

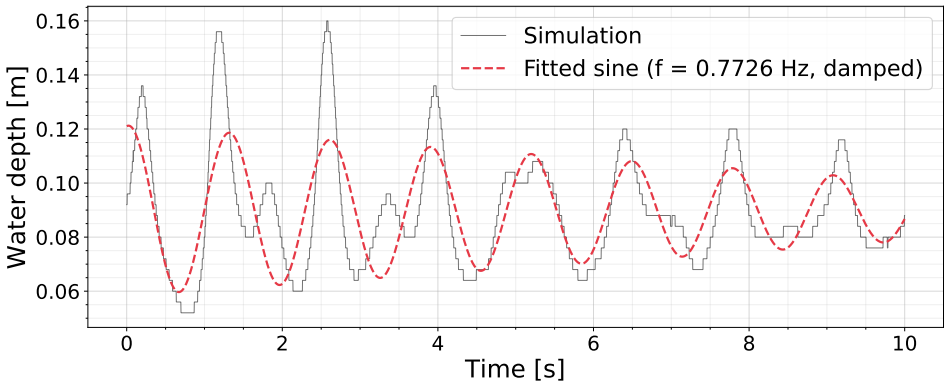


Figure 41: Depth of the fluid on the left of the domain of a sloshing test as a function of time for a simulation with default parameters, as well as a fitted damped sine on this data.

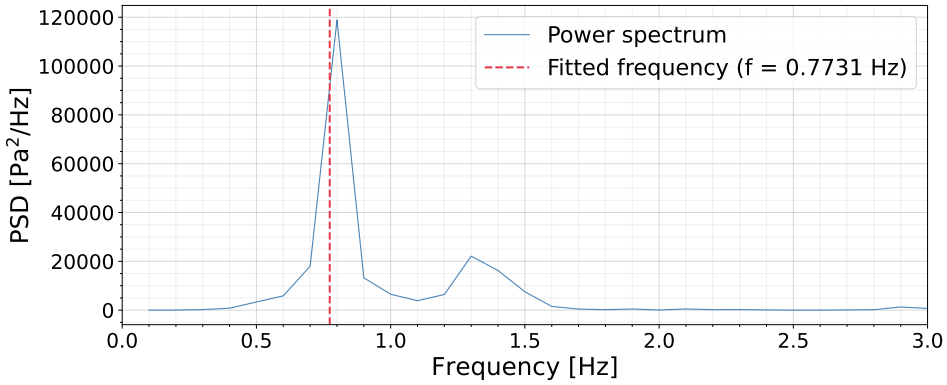


Figure 42: FFT of the pressure at the bottom of the container as a function of time for a sloshing simulation with default parameters.

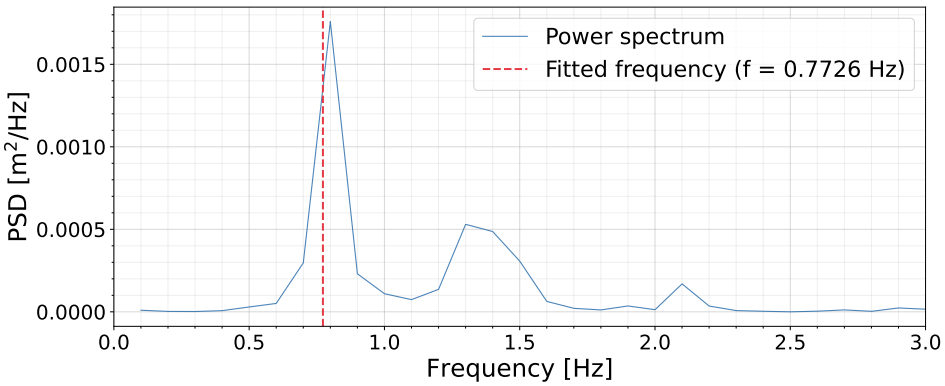


Figure 43: FFT of the depth of the fluid on the left of the domain as a function of time for a sloshing simulation with default parameters.

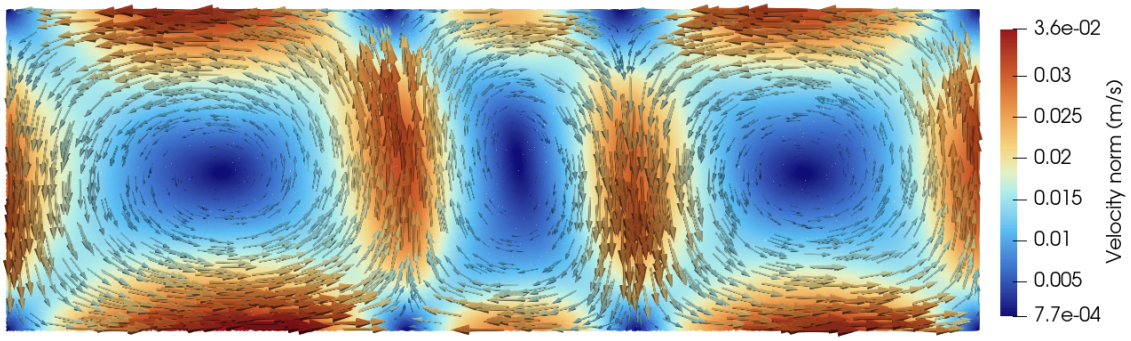


Figure 44: The velocity field at the end of the simulation ($t = 700\,\mathrm{s}$) for Rayleigh-Bénard convection with $\Gamma = 3$.

3, 5.5, and 10.5. Thus, the number of convection cells obtained should theoretically be 2.31, 4.235, and 8.085, which must be rounded, as those numbers must be integers. The first case is, according to Poel et al., at the limit between stabilizing at 2 or 3 convection cells. The two others should show 4 and 8 convection cells respectively.

It results that 3 convection cells can be seen at the end of the simulation for $\Gamma = 3$, as observed in Fig. 44. As expected, 4 cells are observed for $\Gamma = 5.5$, as illustrated in Fig. 45. However, only 5 cells are present when $\Gamma = 10.5$, as shown in Fig. 46. As those cells are quite elongated, this could indicate that the state observed is transient.

In all cases, convection happens spontaneously, and convection cells are created. It is however not possible to determine non-dimensional numbers to compare results more accurately, as those numbers include the viscosity, which is not possible to know precisely here.

3.6 Heat exchanger

As the advection is decoupled from the rest of the equations, it was possible to solve the equation for the temperature (Eq. 5) in solid cells present in the domain. Thus, the solver is able to simulate both flows and heat diffusion at the same time. A heat

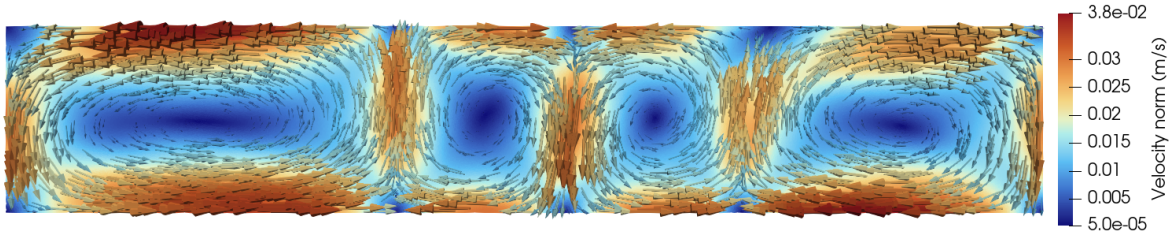


Figure 45: The velocity field at the end of the simulation ($t = 700\,\mathrm{s}$) for Rayleigh-Bénard convection with $\Gamma = 5.5$.

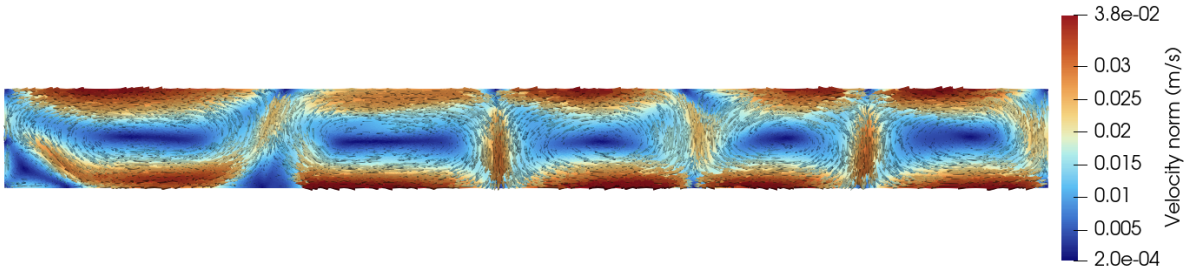


Figure 46: The velocity field at the end of the simulation ($t = 700\,\mathrm{s}$) for Rayleigh-Bénard convection with $\Gamma = 10.5$.



Figure 47: Temperature [K] in the heat exchanger at the end of the simulation ($t = 25\,\mathrm{s}$). The grey area corresponds to the solid boundary between the hot and cold flows.

exchanger was then reproduced. However, to ensure that the transfer of heat through the solid was visible, the specific heat capacity of the solid was set to $1\,\mathrm{J/kgK}$, and its thermal conductivity to $2000\,\mathrm{W/mK}$. Those parameters were also adapted for the fluid, with a heat capacity of $418\,\mathrm{J/kgK}$ and a thermal conductivity of $5\,\mathrm{W/mK}$. The hot entry flow was at $400\,\mathrm{K}$, while the cold one was at $300\,\mathrm{K}$. All the parameters of this test are described in Sec. B.6.

It results that the transfer of heat between the cold and heat flows is very visible, as in Fig. 47. The temperature at the end of the hot flow is comprised between $350\,\mathrm{K}$ and $373\,\mathrm{K}$, while the one at the end of the cold flow is between $350\,\mathrm{K}$ and $323\,\mathrm{K}$.

Thus, the solver is capable of simulating a heat exchanger, including the conduction of heat in the solid boundary.

4 Discussion and perspectives

All test cases considered in this project yielded physically meaningful results in good agreement with data from the literature, where such comparisons were possible. However, experimental measurements are performed with real fluids, whereas the implemented solver targets perfect fluids. The numerical dissipation introduced by the different schemes is therefore not negligible, as confirmed by the Taylor-Green vortex and the Von-Karman vortices analysis (Sec. 3.1, Sec. 3.2).

The numerical viscosity was shown to scale approximately as Δx (first-order behaviour) for the semi-Lagrangian scheme, as $(\Delta x)^2/\Delta t$ for PIC, while that of FLIP and APIC was found to depend almost exclusively on Δt. Overall, the semi-Lagrangian and PIC methods were found to be significantly more dissipative than FLIP and APIC. The scaling showed that even for potential really small resolutions, the numerical viscosity will remains non-negligible relative to the physical viscosity of common real fluids, which fundamentally limits the ability of these methods to represent truly inviscid dynamics without resorting to extremely fine grids.

However, this quantification was realized on a single test case. It is then necessary to quantify the viscosity in several other situations, and to determine whether it is geometry-dependent. Moreover, mathematical validation of the scaling order is needed.

Furthermore, the present validation of the methods results only from numerical experiments. The ranges of parameters that were given are consequently approximative. A mathematical proof of the convergence of the method is thus needed. It would also be of interest to compare results with other numerical methods, such as the finite volumes method.

This last remark is particularly true for the PIC/FLIP ratio. For free-surface flows, only a 5%/95% ratio was considered. As a result, the evolution of the results with respect to this parameter is unknown. The number of particles per cell was also chosen empirically. As such, it is not possible to conclude that these methods reproduce a physical behaviour in general. However, the results obtained here are promising, and encourage further research.

Some attention should be given to the APIC method. While for internal flows it yields results similar to the FLIP method, but with less noise, it is less conclusive for free-surface flows. Some simulations did not give any result for the dam break test case. Upon investigation, it was concluded that numerical instability appeared in the form of highly accelerated particles. Such phenomenon is possibly due to the inversion of the $\boldsymbol{D}$ matrix in the grid-particles interaction, as it can be singular. Further investigation is needed to solve those problems. This could also explain the outliers in the data.

An important point regarding free-surface flows is the naive handling of the free surface. As mentioned in Sec. 2.6, a cell is considered fluid if at least a particle is present in it, otherwise it is treated as air. Solid cells are set by the user in the input file, and do not change during the simulation. Such an elementary method can lead to noise. The implementation of level sets would improve this aspect of the simulations significantly.

Another capability of the solver, which is thermal coupling, was not studied very quantitatively. This is due to the necessity of a finite viscosity to compute non-dimensional numbers, such as the Rayleigh or Prandtl numbers. As it was not known in the considered test cases, it was not possible to compare results accurately with the literature. Moreover, while the term *viscosity* is used here, one must remember that it is fundamentally numerical dissipation and dispersion. Such phenomenon is different from real viscosity,

which is a diffusion of momentum. This limitation extends to many test cases, which study results based on the Reynolds number, among others.

Consequently, the study of the thermal coupling was mostly qualitative. While the number of convection cells was compared with the literature, the validation of the results was mostly visual. In the case of the heat exchanger, physical parameters were not used, as it would have resulted in an impractically long simulation, as well as a very big domain, increasing even more the computation costs. As merely observing diffusion across the solid boundary leading to a change of temperature in the fluid showed that the diffusion in the solid was happening, such difficulties were avoided by using a very low thermal resistance for the solid, and a high thermal conductivity for the fluid.

Such limitations could be lifted by extending the solver to include physical viscosity, which would allow to compute non-dimensional numbers. Another extension to consider would be adding a third spatial dimension, which is of interest, as the behaviour of fluids change between the 2D and 3D cases. Turbulence is particularly affected by this last point, as vortex stretching only happens in three dimensions.

From these observations, it can be concluded that the APIC method yields more physical results for internal flows, as its viscosity is lower in those situations. However, the PIC/FLIP method is more adapted to free-surface flows, given the reliability of its results for the dam break and sloshing tests, as well as the numerical instability exhibited by the APIC simulations.

5 Conclusion

In this study several methods to solve Euler's equations were implemented, which were the semi-Lagrangian method, the particle-in-cell (PIC) method, the fluid-implicit-particle (FLIP) method, and the affine-particle-in-cell (APIC) method. Moreover, thermal coupling was implemented for all methods, using the Boussinesq approximation.

Several test cases were studied. The analysis of the energy and the speed of the fluid in a Taylor-Green vortex case led to the quantification of the numerical viscosity for each method. Furthermore, the influence of the spatial step, as well as the duration of the time step were determined. Orders of convergence were obtained with respect to Δx, and showed that the behaviour was of first order for semi-Lagrangian method, of the second order for PIC with dependence in $(\Delta t)^{-1}$. Finally, the independence with respect to the spatial resolution was shown for FLIP and APIC.

Afterwards, the shedding of von Karman vortices in the case of a cylinder in a uniform flow was considered and results were coherent with those from the Taylor-Green analysis.

Moreover, free-surface flows were studied. First, the influence of the grid resolution, as well as the number of time steps, was discussed for dam break test cases, for both PIC/FLIP and APIC simulations. Measurements used were the wavefront speed and the impact pressure. Results were compared with Lobovský et al. [7], and found in accord with experiments for a number of cells along the x axis $n_x \geq 200$, and for a number of time steps $n_t \leq 10\,000$.

Additionally, the natural frequencies in sloshing test cases were determined, and compared with results from potential flow theory [10]. Values from numerical simulations agreed well with the theory. Fast Fourier transforms were realized, and showed that the second harmonic was not negligible in the results of the simulations. It is worth noting that the APIC method performed worse than the PIC/FLIP method for these cases.

Consequently, it is deduced that the APIC method is more adapted to internal flows, as it exhibits lower numerical dissipation than other methods, as well as less noise than the FLIP method. However, the PIC/FLIP method is more suited for free-surface flows, as the APIC method then shows numerical instability. In any case, results obtained were in accordance with the literature, validating the methods.

Then, the thermal coupling was considered. Free convection was first discussed, via three Rayleigh-Bénard test cases. The number of convection cells was compared with the literature [11], and values found were as expected, except for the largest aspect ratio studied. Afterwards, it was shown that the solver was capable of reproducing forced convection, as well as conduction in solids, by simulating an idealized heat exchanger. This coupling was validated qualitatively, as the absence of well-defined viscosity prevents the use of most non-dimensional numbers, and thus preventing quantitative comparisons.

Finally, the limitations of the present study were discussed, and potential extensions were given. Notable blind spots were the lack of mathematical developments on the convergence of the solvers, the impossibility of using most non-dimensional numbers, and the naive handling of the free surfaces. Hence, potential extensions include the addition of viscosity to the model, and the use of level sets.

The implemented solver, and the results obtained from it, demonstrate that PIC methods are viable for fluid mechanics computations, as a wide range of phenomena can be reproduced, ranging from internal flows to free-surface problems and thermal convection. Additional research would further consolidate the validation of these methods, while implementing viscosity and three-dimensional effects would allow for more applications.

A Development of the temperature equation

The first principle of thermodynamics can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathcal{U}+\mathcal{K}\right)=W+P-Q \tag{61}$$

with

- $\mathcal{U}$ the internal energy of the system,
- $\mathcal{K}$ the kinetic energy of the system,
- W the power generated by the forces applied on the fluid,
- P the heat production inside of the system,
- and Q the total heat flux flowing out of the system.

The term $\mathcal{U}$ can be written as

$$\mathcal{U}=\int_{\mathbb{V}(t)}\rho e\,\mathrm{d}\mathbb{V}, \tag{62}$$

with e the internal energy density. This implies that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{U}=\int_{\mathbb{V}(t)}\rho\dot{e}\,\mathrm{d}\mathbb{V}, \tag{63}$$

per Reynold's Theorem. Assuming that $\dot{e}=c\dot{T}$ with c the specific heat capacity, in $[\mathrm{J\,kg^{-1}K^{-1}}]$, one then obtains

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{U}=\int_{\mathbb{V}(t)}\rho c\dot{T}\,\mathrm{d}\mathbb{V}. \tag{64}$$

The kinetic energy can be written as

$$\mathcal{K}=\int_{\mathbb{V}(t)}\frac{1}{2}\rho\|\boldsymbol{v}\|^2\,\mathrm{d}\mathbb{V}, \tag{65}$$

meaning that its time derivative is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{K}=\int_{\mathbb{V}(t)}\rho\dot{\boldsymbol{v}}\cdot\boldsymbol{v}\,\mathrm{d}\mathbb{V}. \tag{66}$$

The power generated by the forces W can be written as

$$W=\int_{\mathbb{V}(t)}\rho\boldsymbol{b}\cdot\boldsymbol{v}\,\mathrm{d}\mathbb{V}+\int_{\mathbb{S}(t)}\boldsymbol{t}\cdot\boldsymbol{v}\,\mathrm{d}\mathbb{S}, \tag{67}$$

with $\boldsymbol{b}$ the body forces and $\boldsymbol{t}$ the surface tractions. From the definition of the Cauchy stress tensor $\boldsymbol{\sigma}$, one has $\boldsymbol{t}=\boldsymbol{\sigma}\cdot\boldsymbol{n}$ with $\boldsymbol{n}$ the unit normal of the surface. This means that

$$W=\int_{\mathbb{V}(t)}\rho\boldsymbol{b}\cdot\boldsymbol{v}\,\mathrm{d}\mathbb{V}+\int_{\mathbb{S}(t)}\boldsymbol{\sigma}\cdot\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}\mathbb{S}. \tag{68}$$

Applying Gauss' theorem thus gives

$$W = \int_{\mathbb{V}(t)} \rho \boldsymbol{b} \cdot \boldsymbol{v} \, \mathrm{d}\mathbb{V} + \int_{\mathbb{V}(t)} \boldsymbol{\nabla} \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{v}) \, \mathrm{d}\mathbb{V} \tag{69}$$

$$= \int_{\mathbb{V}(t)} \rho \boldsymbol{b} \cdot \boldsymbol{v} \, \mathrm{d}\mathbb{V} + \int_{\mathbb{V}(t)} \left[(\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}) \cdot \boldsymbol{v} + \boldsymbol{\sigma} \cdot (\boldsymbol{\nabla} \cdot \boldsymbol{v}) \right] \mathrm{d}\mathbb{V}. \tag{70}$$

As it is assumed that the fluid is incompressible, the mass conservation equation gives $\boldsymbol{\nabla} \cdot \boldsymbol{v} = 0$. From the conservation of momentum equation,

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \rho \dot{\boldsymbol{v}} - \rho \boldsymbol{b}, \tag{71}$$

which then gives

$$W = \int_{\mathbb{V}(t)} \rho \boldsymbol{b} \cdot \boldsymbol{v} \, \mathrm{d}\mathbb{V} + \int_{\mathbb{V}(t)} \left[\rho \dot{\boldsymbol{v}} \cdot \boldsymbol{v} - \rho \boldsymbol{b} \cdot \boldsymbol{v} \right] \mathrm{d}\mathbb{V} \tag{72}$$

$$= \frac{\mathrm{d}}{\mathrm{d}t} \mathcal{K}. \tag{73}$$

The heat production inside the system can be described as

$$P = \int_{\mathbb{V}(t)} r \, \mathrm{d}\mathbb{V} \tag{74}$$

with r the heat production per unit volume.

The total heat flux can be written as

$$Q = \int_{\mathbb{S}(t)} q \, \mathrm{d}\mathbb{S}. \tag{75}$$

Assuming that $q = \boldsymbol{q} \cdot \boldsymbol{n}$ with $\boldsymbol{q}$ the heat flux vector, in [W/m^2] (Stokes' heat flux theorem), and applying Gauss' theorem, one finds

$$Q = \int_{\mathbb{V}(t)} \boldsymbol{\nabla} \cdot \boldsymbol{q} \, \mathrm{d}\mathbb{V}. \tag{76}$$

Furthermore, assuming that the material follows Fourier's law of heat conduction, one has

$$\boldsymbol{q} = -k \boldsymbol{\nabla} T, \tag{77}$$

thus

$$Q = \int_{\mathbb{V}(t)} -\boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) \, \mathrm{d}\mathbb{V}. \tag{78}$$

Introducing Eq. 64, Eq. 66, Eq. 73, Eq. 74 and Eq. 78 in Eq. 61, one obtains

$$\int_{\mathbb{V}(t)} \left[\rho c \dot{T} + \rho \dot{\boldsymbol{v}} \cdot \boldsymbol{v} \right] \mathrm{d}\mathbb{V} = \int_{\mathbb{V}(t)} \left[\rho \dot{\boldsymbol{v}} \cdot \boldsymbol{v} + r + \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) \right] \mathrm{d}\mathbb{V}. \tag{79}$$

Applying the localization theorem, one finds

$$\rho c \dot{T} = r + \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T). \tag{80}$$

Developing the material derivative,

$$\frac{\mathrm{d}}{\mathrm{d}t} T = \frac{\partial T}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} T, \tag{81}$$

the equation finally becomes

$$\frac{\partial T}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} T = \frac{r}{\rho c} + \frac{1}{\rho c} \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T). \tag{82}$$

B Simulation parameters

B.1 Taylor-Green vortex

The default values for the Taylor-Green tests were:

- a grid of 50×50 cells,
- a spatial step $\Delta x = 0.062\,832\,\mathrm{m}$,
- 636 time steps,
- a time step duration $\Delta t = 0.031\,416\,\mathrm{s}$,
- a tolerance for the pressure solver of 0.01,
- a density of $1000\,\mathrm{kg/m^3}$,
- 25 particles per cell,
- the Taylor-Green velocity field initialisation used $U_{inf} = 1\,\mathrm{m/s}$ and $k = 1$
- `SOLID` cells on each boundaries

B.2 Cylindrical in an uniform flow

The default values for the cylindrical in an uniform flow were:

- a grid of 200×100 cells,
- a spatial step $\Delta x = 0.01\,\mathrm{m}$,
- 3750 time steps,
- a time step duration $\Delta t = 0.02\,\mathrm{s}$,
- a tolerance for the pressure solver of 0.01,
- a density of $1000\,\mathrm{kg/m^3}$,
- 25 particles per cell,
- `SOLID` cells on the top and bottom of the domain
- `DIRICHLET` cells initialize at $U_0 = 2\,\mathrm{m/s}$ on the left hand side of the domain
- `AIR` cells on the right hand side of the domain

B.3 Dam break

The default values for the dam break tests were:

- a grid of 268×100 cells,
- a spatial step $\Delta x = 0.006\,\text{mm}$,
- 10000 time steps,
- a time step duration $\Delta t = 0.0002\,\text{s}$,
- a tolerance for the pressure solver of 0.003,
- a density of $997\,\text{kg/m}^3$,
- a gravitational acceleration of $9.81\,\text{m/s}^2$,
- 15 particles per cell,
- if ran with the PIC/FLIP solver, a PIC/FLIP ratio of 5%/95%.

B.4 Sloshing

The default values for the sloshign tests were:

- a grid of 150×150 cells, with a spatial step $\Delta x = 0.004\,\text{m}$,
- 10000 time steps, with a duration of $\Delta t = 1 \times 10^{-3}\,\text{s}$,
- a tolerance of 10^{-3} for the pressure solver,
- a density $\rho = 1000\,\text{kg/m}^3$,
- a gravity $g = 9.81\,\text{m/s}^2$,
- a mean fluid height of 23 cells,
- an initial amplitude of 11 cells for the sine shape of the free surface, with one period spanning the whole domain.

B.5 Rayleigh-Bénard convection

The parameters used were:

- a ratio PIC/FLIP of 5%/95%,
- a grid of $100\Gamma \times 100$, with Γ the aspect ratio considered,
- a spatial step $\Delta x = 5 \times 10^{-3}\,\text{m}$,
- 10000 time steps of duration $\Delta t = 0.07\,\text{s}$,
- a tolerance for the pressure solver of 7×10^{-5}, and for the temperature solver of 10^{-6},
- physical properties of water at 325 K [12], which where a density $\rho = 987.16\,\text{kg/m}^3$, a coefficient of thermal expansion $\beta = 4.7119 \times 10^{-4}\,\text{K}^{-1}$, a specific heat capacity $c = 4182\,\text{J/kgK}$, and a thermal conductivity $k = 0.645\,\text{W/mK}$.

B.6 Heat exchanger

The parameters used were:

- a ratio PIC/FLIP of 5%/95%,
- a grid of 800×50 cells, with a spatial step $\Delta x = 5 \times 10^{-3}\,\mathrm{m}$,
- 5000 time steps, with a time step duration of $5 \times 10^{-3}\,\mathrm{s}$,
- a tolerance for the pressure solver of 10^{-2}, and one of 10^{-6} for the thermal solver,
- a hot entry flow at $400\,\mathrm{K}$, and a cold one at $300\,\mathrm{K}$,
- a density $\rho = 987.16\,\mathrm{kg/m^3}$, a heat capacity $c = 418\,\mathrm{J/kgK}$, and a thermal conductivity $k = 5\,\mathrm{W/mK}$ for the fluid,
- a density $\rho_s = 8933\,\mathrm{kg/m^3}$, a heat capacity $c_s = 1\,\mathrm{J/kgK}$, and a thermal conductivity $k_s = 2000\,\mathrm{W/mK}$ for the solid,
- a solid separation going from cell 22 to cell 28 included, and spanning the whole length of the domain.

References

[1] R. Bridson, *Fluid Simulation for Computer Graphics*, 2nd. Boca Raton, FL: A K Peters/CRC Press, 2015.

[2] Y. Li, W. Wang, X. Su, G. Zhang, and H. Tang, "A comparative study on the acceleration techniques for solving finite difference discretization Poisson's equation in the PIC/MCC method," *Physica Scripta*, vol. 99, no. 5, p. 055 604, Apr. 2024. DOI: 10.1088/1402-4896/ad3697.

[3] Y. Dagan, "Analytical solutions for particle dispersion in Taylor–Green vortex flows," *Theoretical and Computational Fluid Dynamics*, vol. 39, no. 1, p. 14, Feb. 2025. DOI: 10.1007/s00162-024-00731-z.

[4] X. Gong, H. Suzuki, and T. Kouchi, "Multiscaled inviscid Taylor-Green vortex flow for examining energy conservation error in incompressible flows," *Journal of Physics: Conference Series*, vol. 2701, no. 1, p. 012 052, Feb. 2024. DOI: 10.1088/1742-6596/2701/1/012052.

[5] H. Jiang and L. Cheng, "Strouhal–Reynolds number relationship for flow past a circular cylinder," *Journal of Fluid Mechanics*, vol. 832, pp. 170–188, Dec. 2017. DOI: 10.1017/jfm.2017.685.

[6] I. Rodríguez, O. Lehmkuhl, J. Chiva, R. Borrell, and A. Oliva, "On the flow past a circular cylinder from critical to super-critical Reynolds numbers: Wake topology and vortex shedding," *International Journal of Heat and Fluid Flow*, vol. 55, pp. 91–103, Oct. 2015. DOI: 10.1016/j.ijheatfluidflow.2015.05.009.

[7] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias, "Experimental investigation of dynamic pressure loads during dam break," *Journal of Fluids and Structures*, vol. 48, pp. 407–434, Jul. 2014. DOI: 10.1016/j.jfluidstructs.2014.03.009.

[8] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The affine particle-in-cell method," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–10, 2015. DOI: 10.1145/2766996.

[9] O. Castro-Orgaz and H. Chanson, "Ritter's dry-bed dam-break flows: Positive and negative wave dynamics," *Environmental Fluid Mechanics*, vol. 17, no. 4, pp. 665–694, Aug. 2017. DOI: 10.1007/s10652-017-9512-5.

[10] Y. Chen and M.-A. Xue, "Numerical simulation of liquid sloshing with different filling levels using OpenFOAM and experimental validation," *Water*, vol. 10, no. 12, p. 1752, Nov. 2018. DOI: 10.3390/w10121752.

[11] E. P. van der Poel, R. J. A. M. Stevens, K. Sugiyama, and D. Lohse, "Flow states in two-dimensional Rayleigh-Bénard convection as a function of aspect-ratio and Rayleigh number," *Physics of Fluids*, vol. 24, no. 8, p. 085 104, Aug. 2012. DOI: 10.1063/1.4744988.

[12] T. L. Bergman and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 8th. Hoboken, NJ: John Wiley & Sons, 2017.