31st International Conference Nuclear Energy for New Europe
NENE 2022 September 12-15 PORTOROŽ SLOVENIA

Inviscid Fluid Simulation Through Incompressible Schrödinger Flow Method: A Finite Element Approach

Stefano Riva
Politecnico di Milano, Dept. of Energy,
CeSNEF-Nuclear Engineering Division, Nuclear Reactors Group
Via La Masa, 34
20156 Milano, Italy
stefano.riva@polimi.it
Antonio Cammi, Carolina Introini
Politecnico di Milano, Dept. of Energy,
CeSNEF-Nuclear Engineering Division, Nuclear Reactors Group
Via La Masa, 34
20156 Milano, Italy
antonio.cammi@polimi.it, carolina.introini@polimi.it

ABSTRACT

Computational fluid dynamics is the standard approach to simulate the behaviour of fluids governed by the Navier-Stokes equations. This problem always involves a suitable treatment of the non-linearity of the advection term in the equations, which is the main bottleneck in performing fast simulations. Moreover, as the Reynolds number increases, its importance becomes larger and larger; for this reason, the Navier-Stokes equations are rarely directly numerically solved, preferring a solution with RANS or LES approaches. These methods model the behaviour of the small scales (totally or partially, respectively), and only the larger scales are directly solved.

In the limit of Reynolds number going to infinite (i.e., viscosity goes to 0), the flow obeys the Euler equations. These equations are still strongly non-linear, which limits the usable time step for stability (a common issue in hyperbolic PDEs). These fluids are referred to as ideal fluids, neglecting the dissipation given by the viscosity. In 1926, Madelung proposed a hydrodynamical form of quantum mechanics, showing a link between the linear Schrödinger equation and the non-linear Euler ones. In particular, he derived the latter from the former, linking the two different physics. Thus, a novel approach to solving complex non-linear PDEs has been proposed, substituting the non-linear Euler equations with the linear one derived by Madelung. The fluid state is now a vector of two complex wave functions which satisfy the Schrödinger equation with an incompressibility constraint. This method is called Incompressible Schrödinger Flow; literature studies solve this problem using FFT, showing impressive results in predicting vortex dynamics, an example of this is provided in this work, starting from the MATLAB code available in literature.

This work aims at implementing this novel approach in a Finite Element framework so that it is easier to extend it to complex geometries and showing some preliminary results of different simulations compared with a classical, state-of-the-art CFD approach. In the future, it would be interesting to investigate the possibility of linking this approach with the temperature equation to include buoyancy effects.

1 INTRODUCTION

The behaviour of inviscid fluids, characterised by null viscosity, is described by the Euler equations. These are a set of quasi-linear hyperbolic Partial Differential Equations (PDEs), typically expensive to solve due to time step and mesh size requirements [1]. The solution of these equations is of primary interest in the aeronautical field when supersonic and hypersonic flows or shock wave propagation are studied [2]. Different methods exist in the literature, both considering Computational Fluid Dynamics (CFD) approaches [3] or other paths, such as the Lattice Boltzmann Methods [4]. Specifically, the former employs different discretisation techniques in time and space to solve ″directly″ the Euler equations, however, the mathematical shape of the PDEs results in a heavy numerical problem.

This work aims at looking at this problem with a different perspective for predicting the behaviour of inviscid flows: the starting point is the connection between hydrodynamics and quantum mechanics proposed by Madelung in 1926 [5]. In particular, he derived the compressible Euler equations (continuity and momentum) from the Schrödinger equation. This idea has been explored by the pioneering work of Chern [6, 7], which developed a novel method under the name of *Incompressible Schrödinger Flow* (ISF). This technique can successfully simulate vortex dynamics; however, its implementation using Fast Fourier Transform (FFT) [8] is a bit limited as it cannot be directly adopted in the industrial engineering context. Therefore, this work aims at extending the range of applicability of this technique by implementing it using the Finite Element (FE) library FEniCSx [9, 10, 11] for Python.

The structure of the paper is as follows: Section 2 is devoted to a brief description of the ISF method set in a FE framework; Section 3 presents and discusses some numerical results, employing the original code developed by [6, 7] and the novel implementation using FEniCSx; finally, in Section 4 the main conclusions are drawn.

2 SCHRÖDINGER'S SMOKE

The *Incompressible Schrödinger Flow* is a novel numerical technique, developed in [6, 7], able to describe inviscid fluids behaviour using the Schrödinger equation. This section gives some basic concepts on this analogy between hydrodynamics and quantum mechanics [5], along with the description of the ISF solution algorithm.

In quantum mechanics, a wave function represents the physical state of a system and its amplitude represents the probability of finding a particle inside a portion of the spatial and temporal domain. In the context of ISF, the flow field cannot be described by a single wave function wave $\psi \in \mathbb{C}$, because the associated velocity field is always characterised by a null vorticity, which is not generally true [7]. Therefore, the state of the system is described by a 2 components wave function, i.e. $\Psi = [\psi_1, \psi_2]^T \in \mathbb{C}^2$, which is the solution of the Schrödinger equation, i.e.

$$i\hbar\frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2}\Delta\Psi + p\Psi \longleftrightarrow i\hbar\frac{\partial}{\partial t}\begin{bmatrix}\psi_1\\ \psi_2\end{bmatrix} = -\frac{\hbar^2}{2}\Delta\begin{bmatrix}\psi_1\\ \psi_2\end{bmatrix} + p\begin{bmatrix}\psi_1\\ \psi_2\end{bmatrix}; \tag{1}$$

by writing the components of the wave function in polar coordinates and by taking the real and imaginary part of Eq. (1), the compressible Euler equations are obtained (the proof is found in [5]). This work will consider only incompressible flows with constant density, so that the

transformed equations are

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \dfrac{\partial \mathbf{u}}{\partial t} = -\left(\mathbf{u} \cdot \nabla\right)\mathbf{u} - \nabla p \end{cases} \qquad \text{given} \qquad \mathbf{u} = \hbar \mathrm{Re}\left\{-\left(\Psi^T\right)^* i \, \nabla \Psi\right\}. \tag{2}$$

The spatial operators on the wave function Ψ in Eq. (1), i.e. $-\frac{\hbar^2}{2}\Delta$ and p, have a physical interpretation in the Euler equations (2) on $\mathbf{u}$, i.e. $\left(\mathbf{u} \cdot \nabla\right)\mathbf{u}$ and $-\nabla p$. One of the most important results of this transformation, referred to as Madelung transformation, is the change of a non-linear term $\left(\mathbf{u} \cdot \nabla\right)\mathbf{u}$ into a linear one $-\frac{\hbar^2}{2}\Delta\Psi$: this is very important from the numerical point of view, as, in principle, it can lead to a reduction of the solution time.

In the end, the incompressibility constraint ($\nabla \cdot \mathbf{u} = 0$) can be equivalently imposed to the wave function as

$$\mathrm{Re}\left\{\left(\Psi^T\right)^* i \, \Delta \Psi\right\} = 0. \tag{3}$$

Using the linear Schrodinger equations avoids dealing with the convective term in the Euler equations, which makes the problem transport-dominated, which usually imposes limitations in terms of time step size, mesh size and stabilisation requirements [1, 12]. The following section will present how the ISF works in a Finite Element framework.

2.1 Solution Algorithm

Let Ω be the spatial domain, $\partial\Omega$ be its boundary, usually composed by $\Gamma_{in} \cup \Gamma_w \cup \Gamma_o$, and let $[0, T]$ be the time interval considered. Given an initial condition Ψ^0, the Schrödinger equation (1), with the constraint (3), can be solved with a first order[1] time splitting method[2]. At every time step $t_n \rightarrow t_{n+1}$, the algorithm must solve two differential problems:

1. The prediction step consists of a free-particle Schrödinger equation with the associated boundary conditions (a plane wave at the inlet and homogeneous Neumann boundaries for the rest), i.e.

$$\frac{\tilde{\Psi} - \Psi^n}{\Delta t} = \frac{i\hbar}{2}\Delta\tilde{\Psi} \quad \text{in } \Omega, \qquad \begin{cases} \tilde{\Psi} = e^{i(\mathbf{k}\cdot\mathbf{x} - \omega t)} \cdot \left[c_1, c_2\right]^T & \text{on } \Gamma_{in} \\ \nabla\tilde{\Psi} \cdot \mathbf{n} = 0 & \text{on } \Gamma_w \cup \Gamma_o \end{cases} \tag{4}$$

   The wave function needs also to be normalised to 1, since this property is not necessarily conserved in the time-discrete case, i.e.

$$\tilde{\Psi} \longleftarrow \tilde{\Psi} \cdot \left(||\tilde{\Psi}||\right)^{-1} = \tilde{\Psi} \cdot \left(\tilde{\psi}_1^* \tilde{\psi}_1 + \tilde{\psi}_2^* \tilde{\psi}_2\right)^{-\frac{1}{2}}. \tag{5}$$

2. The correction phase involves a Poisson problem to enforce the incompressibility constraint (3). The unknown is φ, which is referred to as *pressure*, even though its units of measure are $[m^2/s]$. This quantity has been introduced to make the velocity field divergence free following the idea developed in [7], i.e.

$$\Delta\varphi = \nabla \cdot \tilde{\mathbf{u}} \quad \text{in } \Omega, \qquad \begin{cases} \varphi = 0 & \text{on } \Gamma_{in} \\ \nabla\varphi \cdot \mathbf{n} = 0 & \text{on } \Gamma_w \\ \nabla\varphi \cdot \mathbf{n} = g & \text{on } \Gamma_o \end{cases} \tag{6}$$

   in which $\tilde{\mathbf{u}}$ is the velocity field computed using Eq. (2) with $\tilde{\Psi}$. This phase is also called *pressure projection*.

[1] Higher order time discretisation strategies may improve the convergence of the method itself, whilst having a larger computational cost.

[2] A similar solution strategy is also employed in CFD, a very important example is the work of [13].

The boundary conditions imposed on the wave function $\tilde{\Psi}$ and the pressure φ chosen are linked to the following ones on the velocity

$$\mathbf{u} = \hbar \cdot \mathbf{k} \quad \text{on } \Gamma_i, \qquad \mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_w, \qquad \mathbf{u} \cdot \mathbf{n} = -g \quad \text{on } \Gamma_o \tag{7}$$

The value of g can be determined as a consequence of a global mass balance

$$\int_{\Gamma_{in}} \mathbf{u}_{in} \cdot \mathbf{n} \, d\sigma = \int_{\Gamma_o} \mathbf{u} \cdot \mathbf{n} \, d\sigma = \int_{\Gamma_o} (\tilde{\mathbf{u}} - \nabla\varphi) \cdot \mathbf{n} \, d\sigma = \int_{\Gamma_o} g \cdot \mathbf{n} \, d\sigma \tag{8}$$

This is a compatibility condition for the problem. In the end, the wave function and the velocity are updated, i.e.

$$\Psi^{n+1} = e^{-i\frac{\varphi}{\hbar}} \tilde{\Psi}, \qquad \mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \nabla\varphi. \tag{9}$$

Algorithm 1 summarises the overall algorithm.

Algorithm 1: Incompressible Schrödinger Flow solution algorithm.

Input
$\Psi_0(\mathbf{x})$, $\hbar$, Δt, N_{max};
Output
$\{\Psi(\mathbf{x}, t_n), \mathbf{u}(\mathbf{x}, t_n)\}_{n=1}^{N_{max}}$;
for $n = 0 : N_{max}$ **do**
$\tilde{\Psi} \leftarrow$ Schrödinger$(\Psi_n, \Delta t, \hbar)$;
$\tilde{\Psi} \leftarrow \tilde{\Psi}/(\tilde{\psi}_1^* \tilde{\psi}_1 + \tilde{\psi}_2^* \tilde{\psi}_2)^{0.5}$;
$\varphi \leftarrow$ Poisson$(\tilde{\Psi})$;
$\Psi^{n+1} = e^{-i\frac{\varphi}{\hbar}} \tilde{\Psi}$;
$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \nabla\varphi$ or
$= \hbar \text{Re}\left\{ - \left([\Psi^{n+1}]^T \right)^* i \, \nabla \Psi^{n+1} \right\}$

In principle, several numerical discretisation techniques can solve the PDEs involved in this solution algorithm: the original work of [6, 7] employed FFT, which it is in general limited to structured grids, making it less flexible for complex geometry. For this reason, this work has set this method in a Finite Element (FE) framework to make it more suited for future developments on more complicated test cases. The FE approach requires the derivation of the weak formulations[3] [14] of the differential problems (not reported here for brevity). In the end, it is worth mentioning that it is still unclear if the choice of the functional spaces for Ψ and φ are connected, as in the classical CFD approach: in fact, in CFD the pressure and the velocity must have only Taylor-Hood compatible spaces [1], if this is not respected there is no guarantee that the FE solution converges to the true one.

3 NUMERICAL RESULTS

This section discusses some results on the potentiality of the ISF method: first, the MATLAB code by [6, 7] will obtain a Von Karman vortex street, to highlight the capability of ISF to predict vortex dynamics; then, the FEniCSx code will simulate an inviscid flow over a Backward Facing Step (BFS) [15], a typical CFD benchmark test case and its results will be compared with a potential solution using OpenFOAM [16].

[3]The differential problem in strong form, e.g. (1), is rewritten to look for a solution in a broader functional space. Weak formulations usually make use of the scalar product in L^2-sense and of the integration by parts.

3.1 Von Karman Vortex Street - MATLAB

The domain Ω is composed of a rectangle 20×8 m^2, whereas the cylinder obstacle with radius $r = 0.5\,m$ is enforced into the system as a constrained velocity region [6, 7], in which the wave function must assume a prescribed value related to the desired velocity field in that region. The original ISF code, developed in MATLAB, requires a uniform structured grid to apply FFT: this case uses 500×200 elements with a time step of size $\Delta t = 0.02\,s$. The CFL condition[4] [1] for this case is more or less equal to 10, which is greater than the ″classical″ limit ~ 1. Indeed, this is one of the main advantages of the ISF method compared to a CFD approach because having a non-hyperbolic linear PDE provides less restriction on the discretisation parameters.

Figure 1 shows the advection of particles by the Schrödinger flow at $T = 75\,s$. This result show how effective the ISF algorithm is in predicting vortex dynamics, which was the main focus of the original work of [6, 7]. Moreover, using a higher Δt gives an important advantage in terms of solution time.

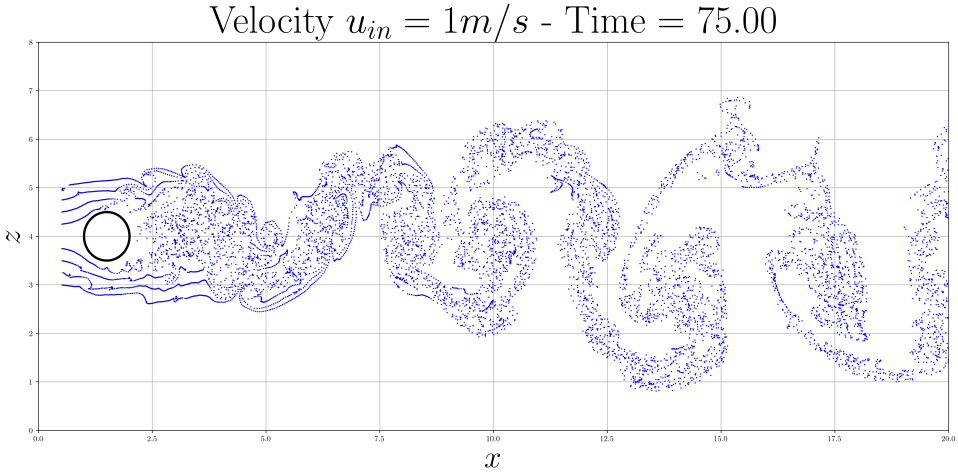

Figure 1: Particles advected by the velocity, computed using ISF (Matlab code, [6]).

The main drawback of this MATLAB code is that it allows uniformly structured grids only and that obstacles are velocity-constrained regions: this is why the ISF must be extended to the FE framework to be more general and flexible for employing it in an engineering context.

3.2 Backward Facing Step

Validation of CFD codes widely uses the BFS benchmark, especially for viscous flows governed by the Stokes or Navier-Stokes equations. The results discussed are preliminary only, foreseeing an in-depth validation phase. The mesh has been created with GMSH [17] and it is composed by 16079 triangular elements, ($\Delta x \approx 10^{-3}\,m$); whereas, the time step size is equal to $0.02\,s$ and the final time is $T = 10\,s$. The code has been implemented on Google Colab using the FEM on Colab distribution by [18]; the case required almost 16 minutes to be simulated, which is reasonably little compared to a CFD approach: for instance, the time step for the standard approach with this mesh would be $\sim 10^{-4} \div 10^{-3}\ s$; hence much more steps on the time loop are needed.

[4] In the framework of hyperbolic equations, some numerical discretisation techniques restrict the time step for stability reasons with the Courant-Friedrichs-Lewy condition, i.e. CFL $= \max_{\mathcal{K} \in \mathcal{T}_h} \frac{\langle \mathbf{u} \rangle_{\mathcal{K}} \Delta t}{||\mathcal{K}||} \leq 1$, given $\mathcal{K}$ the element of the triangulation $\mathcal{T}_h$ of the domain Ω.

Figure 2 shows the two components of the wave function (real and imaginary part) at the final time T: the profiles are very similar to one another, apart from the order of magnitude between ψ_1 and ψ_2, thus the influence of ψ_1 on the velocity is greater than ψ_2. They represent a plane wave propagation inside the BFS domain, thus modelling the pure transport phenomenon; the effect on the step on the wave is visible and this enlargement of the wave function directly impacts the streamlines due to Eq. (2).

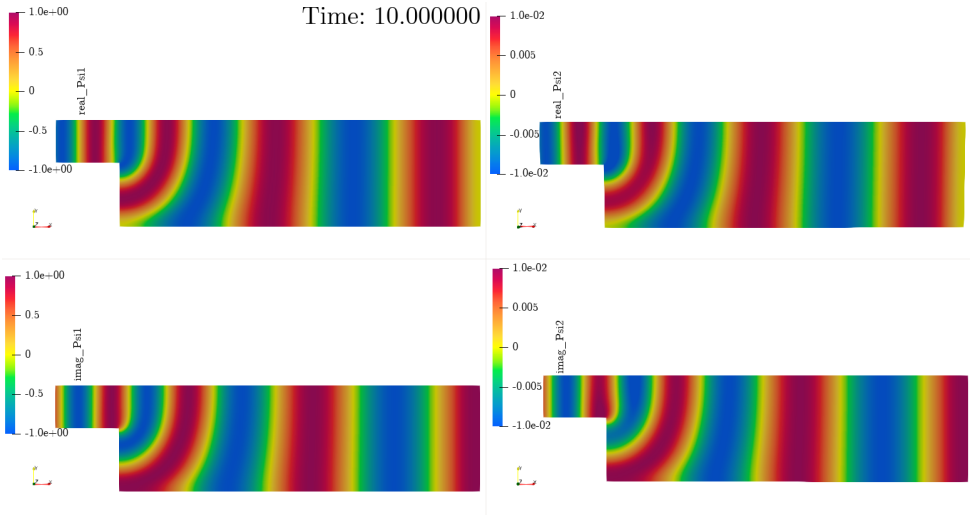

Figure 2: Real and imaginary part of wave function as contour plots (left: ψ_1, right: ψ_2) at final time $T = 10\,s$, using ISF on FEniCSx.

Figure 3 shows the velocity defined from the previous wave function and the one computed using a potential flow[5] formulation [3]. There is no recirculation region behind the step since the flow is inviscid: in fact, there is no viscosity, and the flow follows the ″wall boundary″; moreover, the flow is almost steady. These results are reasonable from a physical point of view since there is no viscosity and no boundary layer: the flow follows the boundary keeping the non-penetration condition. The flow is very similar to the potential one, showing that the importance of the vorticity, for this particular case, is relatively low; however, in principle, the ISF is able to go beyond potential flows, due to the definition of the $2-$component wave function Ψ.

4 CONCLUSIONS

This work aims at studying the *Incompressible Schrödinger Flow*, an innovative technique for simulating inviscid incompressible fluids governed by the Euler equations. This method uses the analogy between hydrodynamics and quantum mechanics proposed by Madelung in 1926. Its main advantage is substituting a non-linear hyperbolic set of PDEs with a linear parabolic PDE, i.e. the Schrödinger equation, coupled with a Poisson problem to enforce the incompressibility constraint. The mathematical shape of the Schrödinger equation comes with an advantage in terms of simulation time since the time step size can be, in general, larger. Moreover, stabilisation techniques for transport-dominated problems in a FE framework [1] are seldom required, ensuring a simple formulation of the problem.

This work aims at extending ISF into a Finite Element framework, making it more suited for more general problems compared to the original formulation with FFT. In fact, the first re-

[5]This result has been obtained using the built-in solver *potentialFoam* in OpenFOAM.

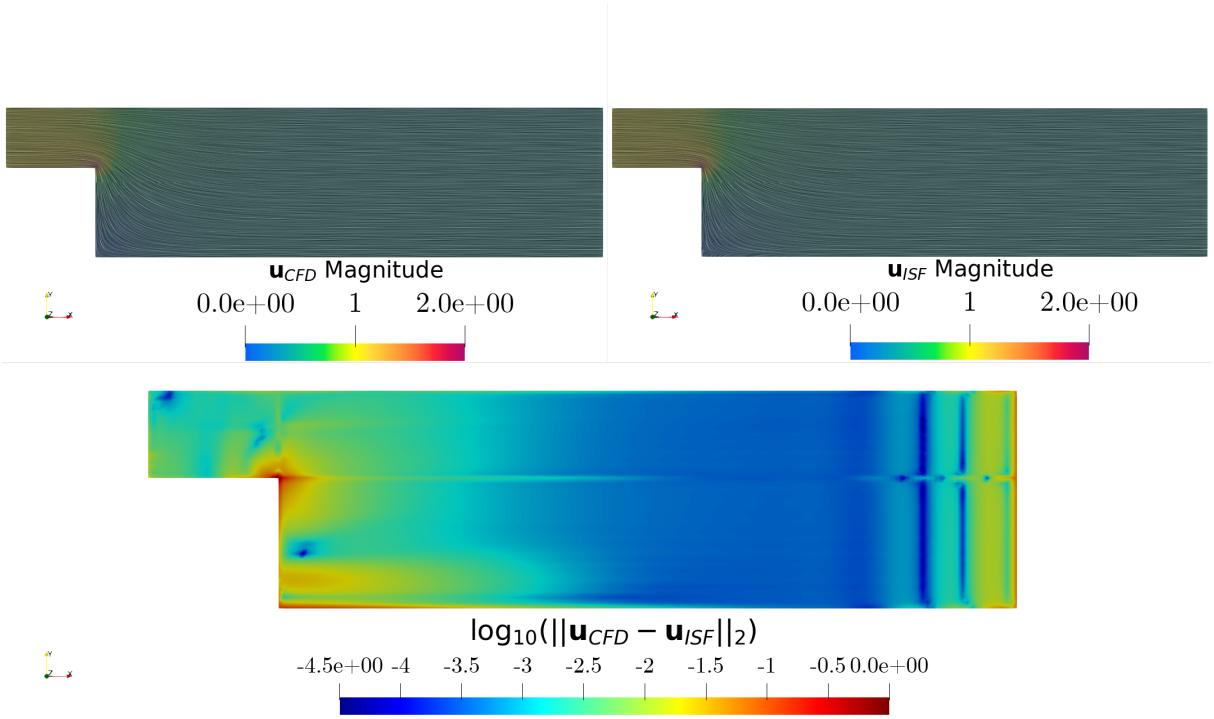

Figure 3: Comparison at $T = 10\,s$ between the ISF velocity streamlines and the potential one, using LIC (Line Integral Convolution). Below the norm of the difference in a logarithmic scale.

sults by [6, 7] used structured grids with simple geometries. This work shows two main results: a preliminary test of Von Karman vortex street using the original MATLAB code and a first assessment of the capabilities of the new code on the Backward Facing Step. The former allows to fully understand the potentialities and the main critical issues of the method itself, giving promising results for vortex dynamics prediction and thus highlighting the necessity to study this technique. On the other hand, this work also tested the new implementation using FE for an inviscid flow over a Backward Facing Step. The most impressive outcome of this approach is the simulation time: this is much less than a classical CFD solution of Euler equations, mainly due to the possibility of considering a higher time step size. In this work, the ISF solution has been compared with a potential solution, for the sake of simplicity, since a direct solution of the incompressible Euler equations is not so easy to be implemented. Therefore, this technique is worth being studied for the simulation of inviscid incompressible flows.

Future works will involve: a further analysis of the variational formulations and the functional spaces of the differential problems; the extension of the original ISF to make it compute the actual pressure field; the investigation of the possibility of using these techniques in a predictor-corrector scheme for the simulation of viscous flows, governed by the Navier-Stokes equation; the inclusion of buoyancy effects coupled with the energy equation.

REFERENCES

[1] Alfio Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations*. Springer Berlin, Heidelberg, i edition, 1994.

[2] Denis Serre and À ma Mère. Shock Reflection in Gas Dynamics. In S. Friedlander and D. Serre, editors, *Handbook of Mathematical Fluid Dynamics*, volume 4, pages 39–122. North-Holland, 2007.

[3] Kaarle Versteeg, Henk and Weeratunge Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007.

[4] Jonas Latt, Christophe Coreixas, Joël Beny, and Andrea Parmigiani. Efficient supersonic flow simulations using lattice boltzmann methods based on numerical equilibria. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2175):20190559, 2020.

[5] Erwin Madelung. Eine anschauliche deutung der gleichung von schrödinger. *Naturwissenschaften*, 14:1004, 1926.

[6] Albert Chern, Felix Knöppel, Ulrich Pinkall, Peter Schröder, and Steffen Weißmann. Schrödinger's smoke. *ACM Trans. Graph.*, 35, 7 2016.

[7] Albert Chern. *Fluid Dynamics with Incompressible Schroedinger Flow*. PhD thesis, California Institute of Technology, 2017.

[8] M. Heideman, D. Johnson, and C. Burrus. Gauss and the history of the Fast Fourier Transform. *IEEE ASSP Magazine*, 1(4):14–21, 1984.

[9] M. S. Alnaesa, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40, 2014.

[10] M. W. Scroggs, I. A. Baratta, C. N. Richardson, and G. N. Wells. Basix: a runtime finite element basis evaluation library. *Journal of Open Source Software*, 7(73):3982, 2022.

[11] M. W. Scroggs, J. S. Dokken, C. N. Richardson, and G. N. Wells. Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes. *ACM Transactions on Mathematical Software*, 2022. To appear.

[12] Alfio Quarteroni. *Numerical Models for Differential Problems*. Springer, Cham, iii edition, 2016.

[13] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968.

[14] Sandro Salsa. *Partial Differential Equations in Action - From Modelling to Theory*. Springer Cham, ii edition, 2015.

[15] B. Armaly, F. Durst, Jose Pereira, and B. Schönung. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, 127:473 – 496, 01 1983.

[16] H G Weller, G Tabor, H Jasak, and C Fureby. A Tensorial Approach to Computational Continuum Mechanics using Object-Oriented Techniques. *Computers in Physics*, 12(6):620–631, 1998.

[17] Geuzaine, Christophe and Remacle, Jean-Francois. Gmsh, 2022. Version 4.10.4.

[18] Ballarin, Francesco. FEM on Colab, 2021. https://fem-on-colab.github.io.