POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria matematica

Tesi di Laurea

Filter stabilization for the compressible Navier Stokes equations with applications to atmosphere dynamics

Advisors
prof. Claudio Canuto
prof. Gianluigi Rozza
Co-advisors
Dr. Michele Girfoglio
Prof. Annalisa Quaini
firma dei relatori

. . . . . .
. . . . . .

Candidate
Nicola Clinco

firma del candidato

. . . . . .

Academic Year 2022-2023

A mio Padre, a mia
Madre Mimma,alla mia
Famiglia ed ai miei Amici

Abstract

This thesis discusses a stabilization technique for the compressible Euler equations with application in the atmospheric flow dynamics. Despite the computational resources available, a Direct Numerical Simulation (DNS) for the atmospherical flow is still far away from our possibilities. This is not only due to the mesh size requirement in terms of storage and computational power, but also to the restriction of the time step in order to fulfill the CFL condition. One of the remedies to these issues is the LES (Large Eddy Simulation) technique, which consists in simulating the large structures, while the smallest are modeled by a "sub-grid model". Inspired by the LES methodology, another approach to stabilize the oscillations in the domain is the Evolve-Filer-Relax algorithm (EFR) that is well investigated in the literature only for the incompressible flows.
The main objective of this thesis is to extend the methodology to the compressible framework. In particular to test the method in the context of atmospheric flow dynamics when the spatial scales are on the order of tens of kilometers (mesoscale flows).
The Finite Volume Method (FVM) is employed for the space discretization while a segregated pressure-based solver is used for solving the equations.
The characteristics of three different filter types will be analyzed: Linear, Smagorinski-like and Deconvolution-type filters. The three filters are tested on two different benchmarks: the non-linear density current and the rising thermal bubble.
Numerical results confirm the selectivity of the deconvolution-based filter while the linear filter is the most dissipative. All filters give perfectly comparable results to those in the literature obtained with high-order methods.

Acknowledgements

At the final stage of my Master's, i would like to thank prof. Claudio Canuto of Politecnico di Torino not only for giving me the opportunity to have a fantastic experience in SISSA with the MathLab group, but also for making me appreciate the theory of the partial differential equations during my whole Master's course.
A huge thank you also to Gianluigi Rozza of SISSA for his support, thank you for believing in my abilities and for the enthusiasm.
A special thank you to Michele Girfoglio, thank you for the special support you gave me and thank you for teaching me that it is necessary to proceed with caution during the projects.
A special thanks also to Annalisa Quaini from Houston for her patience and listening to my doubts at every moment.
A special thank you also to Dario, Isabella, Gaia and Andrea for sharing with me many ideas during my internship at SISSA MathLab.
I would also like to thank my family and friends for their continued encouragement during the past years.

Introduction: A brief history of numerical weather prediction

The first attempts at predicting the behavior of the atmosphere occurred nearly a century and a half ago by Robert FitzRoy in the 1860s. Using only telegraph systems to relay local weather information between base stations across Europe, he produced the first synoptic charts of England and coined the term “weather forecast.
The first real attempt to formulate the atmosphere as a mathematical system was done by the Norwegian meteorologist Vilhelm Bjerknes, which introduced the first equation set of the history to describe motion of the atmosphere. It was composed of:

- The equation of mass
- Three equations for the components of the velocity
- The equation of state
- The equation of the conservation of energy
- A conservation for the water mass

Although the work of Bjerknes was considered the symbol in the study of atmospheric motion, his equations were too complicated to investigate. In 1922, Fry Richardson was the first person who has tried to simulate the processes in the atmosphere. He modelled Europe with a finite difference grid but this attempt was revealed as a failure since spurious oscillations appeared in the solution. Courant, Friederich and Lewy investigated the numerical instabilities found by Richardson formalizing the theory for the finite-difference equations. However, it was not until the advent of computer simulation in the mid of the 20^{th}-century that numerical weather predictions produced some consistent results.
Nowadays, it is still a challenge the numerical modelling in this context, since the scales involved range from thousand of kilometers (synoptic scale) to tens of kilometers (mesoscale modelling). The first strategy to avoid solving the complete Navier Stokes equations is to trust "simplified models": without loss of generality, after an evaluation of the terms in the equations, only the terms that matters are taken into account, disregarding the "smaller terms". Despite its simplicity, this approach leads to some discrepancies in the simulations. The second strategy is to use the complete equations and then apply some stabilization strategies (e.g. RANS, LES), since the computational power required to simulate all the scales involved in turbulence (Direct numerical simulation) is still too high and several techniques are developed to tackle this problem. The most used technique (nowadays) in numerical weather prediction is to filter the small scales while retaining the biggest and is known in literature as LES (Large Eddy simulation). However, we are still far away from creating the perfect model for weather forecasting and the research is running still. Thus, the motivation of this thesis is to develop a novel methodology that is a suitable stabilization technique for the Euler equations.

Contents

List of Tables

List of Figures

Chapter 1

Mathematical Framework

1.1 General notions about the Compressible Navier-Stokes equations

In this chapter we will present the equation set that we will use in the following simulations.

We introduce the Euler equations that describe the motion of a compressible, inviscid fluid: in particular the conservation of mass, momentum and energy is given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \text{in } \Omega \tag{1.1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p + \rho g \widehat{\mathbf{k}} = 0 \qquad \text{in } \Omega \tag{1.2}$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) + \nabla \cdot (p \mathbf{u}) = 0 \qquad \text{in } \Omega \tag{1.3}$$

where $\mathbf{k}$ is the unitary vector aligned along the vertical direction and e is the total energy per unit volume ($e = c_p T + U^2/2$) which is the sum of the enthalpy and the kinetic-energy. Since the unknowns for this problem are 4 $(\rho, \mathbf{u}, p, e)$ we need a further equation to close the system: By assuming the gas ideal, we relate the pressure to the temperature T and density ρ by the following equation:

$$p = \rho R T \tag{1.4}$$

Note that the equations (1.1)-(1.4) are non linear (due to the presence of the convective terms) and also they are fully coupled. Several methods were developed in the literature to deal with this problem, and the best way to solve it is still an open problem.

As long as the quantity of interest in the atmospheric modelling framework is the potential energy θ it is popular to use the following equation set:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \text{in } \Omega \tag{1.5}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p + \rho g \widehat{\mathbf{k}} = 0 \qquad \text{in } \Omega \tag{1.6}$$

$$\frac{\partial (\rho \theta)}{\partial t} + \nabla \cdot (\rho \mathbf{u} \theta) = 0 \qquad \text{in } \Omega \tag{1.7}$$

In our case, we use only the first equation set since numerical experiments suggest that if we decouple the equations following the same splitting procedure as done for the Euler equations (1.2.4), we don't obtain satisfying results (spurious vertical waves due to the fact that the scheme is not well-balanced are originated in the domain).
In order to decouple the equations, we make use of the so-called "splitting methods", which basically are strategies to solve the fully coupled problem with the minor computational time as possible.

1.2 General schemes for time discretization

There are a lot of ways to discretize the Euler system in space and time.
In this section we will give an overview of the most widely used time discretization strategies.

1.2.1 Coupled Methods/Monolitic Methods

In these strategies the velocity and the pressure are still coupled at each time step.
The coupled methods are divided into fully implicit schemes and semi-implicit schemes:

- **Fully Implicit Schemes:**

  All the terms of the equations are implicit, thus, one must solve a fully coupled non linear system with a Newton-Krylov method (with a preconditioner). However, one has to pay the price of a huge computational cost that makes this method not convenient when we have many degrees of freedom. This method could be suitable to approximate a steady-state solution by using a time-dependent problem since we don't have a restriction on the time step.

- **Semi-Implicit Schemes:**

  In this case it is convenient to reserve the implicit treatment for some of the terms while there are some terms treated explicitly (eg. the incompressibility constraint in the incompressible Navier-Stokes equations). The explicit terms introduce a restriction on the time step needed for the stability.

1.2.2 Splitting Methods/Fractional-step methods

These methods produce a cascade of subproblems involving the velocity field and the pressure every time step.

Indeed, even though the quantities of interest are fully coupled in the problem (1.1)-(1.3) , the most well-known strategy to handle this issue is to "decouple" the equations in a certain manner (with appropriate boundary conditions).
Before presenting the complete methodology for the full compressible system, we will present the basic idea of the decoupling strategy by looking at a simple scalar ODE equation:
Suppose that we want to find Φ such that:

$$\frac{\partial \Phi}{\partial t} = A\Phi + B\Phi \tag{1.8}$$

$$\Phi(0) = \Phi_0 \tag{1.9}$$

where A and B $\in \mathbb{R}$: Evidently, the solution is given by:

$$\Phi(t) = e^{A\Delta t} e^{B\Delta t}\Phi_0 = e^{A\Delta t}(e^{B\Delta t}\Phi_0) \tag{1.10}$$

Solving the full problem is equivalent to "split" the problem into two small subproblems:

1. **Solve the problem with A=0**:
   Find Φ_s such that:

$$\frac{\partial \Phi}{\partial t} = B\Phi \tag{1.11}$$

$$\Phi(0) = \Phi_0 \tag{1.12}$$

2. **Solve the problem with B=0**:
   Starting from Φ_s , find Φ such that:

$$\frac{\partial \Phi}{\partial t} = A\Phi \tag{1.13}$$

$$\Phi(0) = \Phi_s \tag{1.14}$$

It is possible to demonstrate that this result is also valid if A and B are square matrixes that have the commutativity property:

$$AB = BA \rightarrow e^{A\Delta t} e^{B\Delta t} = e^{B\Delta t} e^{A\Delta t} \tag{1.15}$$

Nevertheless, if A and B don't commutate, there is an error due to the splitting that is proportional Δt.
The splitting procedure allows us to manage with the less computational time the fully coupled problem (1.1)-(1.4) and for this reason the splitting methods are largely employed in mostly commercial cfd-codes. The methodology allows us to build a class of method as well known as "Predictor-corrector" methods. To explain the basic principle of the latter, suppose we want to solve the ODE system (1.16) by using the implicit Euler as time-stepping scheme.

$$\frac{\partial \Phi}{\partial t} = \mathbf{L}(\Phi) + \mathbf{S}(\Phi) + \mathbf{g} \tag{1.16}$$

$$\Phi(0) = \Phi_0 \tag{1.17}$$

Where **L** and **S** are linear operators depending by Φ and **g** is a source term that could depend on the time.
The problem is divided into two parts:

1. **Solve for Φ_s**:
$$\frac{\Phi_s - \Phi^n}{\Delta t} = \mathbf{L}(\Phi_s) + \mathbf{g}^{n+1} \tag{1.18}$$

2. **Solve for Φ^{n+1}**:
$$\frac{\Phi^{n+1} - \Phi_s}{\Delta t} = \mathbf{S}(\Phi^{n+1}) \tag{1.19}$$

In particular, the first step is named "predictor", because we are not solving the real problem, but we are solving a "simpler" equation, and the second step is named "corrector" since we "hope" to correct the solution of the predictor step thanks to a sort of "refinement" by applying the operator **S**.
In order to show how big the error introduced by the splitting is, we can substitute $\Phi_s = \Phi^{n+1} - \Delta t \mathbf{S}(\Phi^{n+1})$ in the first equation, and rearranging the terms we obtain:

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} = \mathbf{S}(\Phi^{n+1}) + \mathbf{L}(\Phi^{n+1}) + \mathbf{g}^{n+1} - \Delta t \mathbf{L}\mathbf{S}(\Phi^{n+1}) \tag{1.20}$$

Note that in the eq.(1.20) there is a term proportional to Δt that tells us that the equation obtained by the splitting procedure is different from the exact equation.
This methodology will be extended for the complete scheme (1.1)-(1.4).
However,the decision of which term treat implicitly or explicitly and what terms retain in the predictor and in the corrector step is a real question mark: indeed, it is an activity of research build new temporal schemes in order to apply the greatest time step as possible while maintaining the error low. Fortunately, for the general problem (1.1)-(1.4) several guidelines (in terms of "what operator" should be treated implicitly or explicitly) are developed in the course of history : in this framework it is used the term "stiff" identify an operator that should be treated implicitly (eg. the pressure gradient in our case).

1.2.3 The Chorin Splitting-scheme

Nowadays, all the most common splitting schemes adopted in fluid-dynamics (at least for the pressure-based solvers) are all based on a fundamental scheme: the well-known "Chorin-Themam scheme" [16]. This scheme was originally developed for the incompressible flows and was used as a starting point for the development of many other schemes such as the PISO [15] (Pressure Implicit with Splitting of Operators).
Indeed, the compressible solver used in this thesis is a generalization of the PISO obtained by adding the energy equation and the perfect gas law, thus, for completeness we report the original Chorin scheme in the present thesis.
We are interested in solving the incompressible Navier-Stokes equations with a constant source term **g**

$$\begin{aligned} &\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{g} && in\ \Omega \\ &\nabla\cdot\mathbf{u} = 0 && in\ \Omega \\ &\mathbf{u} = 0 && on\ \partial\Omega \end{aligned} \tag{1.21}$$

The problem is solved in two parts: In the first part we obtain a solution by applying only the operators due to the viscosity, the convection term and the source term:

$$\begin{aligned} &\frac{\mathbf{u}_s - \mathbf{u}^k}{\Delta t} - \nu\nabla^2\mathbf{u}_s = \mathbf{g} - \mathbf{u}^k\cdot\nabla\mathbf{u}^k && in\ \Omega \\ &\mathbf{u}_s = \mathbf{0} && on\ \partial\Omega \end{aligned} \tag{1.22}$$

In the second step, we apply the pressure-gradient that enforce the velocity to be divergence free.

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}_s}{\Delta t} + \frac{1}{\rho}\nabla p^{k+1} = 0 \qquad in\ \Omega \tag{1.23}$$

$$\nabla\cdot\mathbf{u}^{k+1} = 0 \qquad in\ \Omega \tag{1.24}$$

$$\mathbf{u}^{k+1}\cdot\mathbf{n} = 0 \qquad on\ \partial\Omega \tag{1.25}$$

Exploiting the divergence-constraint of the velocity field we end up with a Poisson equation for the pressure:

$$\nabla^2 p^{k+1} = \frac{1}{\Delta t}\left(\nabla\cdot\mathbf{u}_s\right) \qquad in\ \Omega \tag{1.26}$$

By multiplying by $\mathbf{n}$ the equation (1.23) we obtain a suitable boundary condition for the pressure:

$$\frac{\partial p^{k+1}}{\partial n} = 0 \tag{1.27}$$

At the end we update the velocity field with the pressure.

$$\mathbf{u}^{k+1} = \mathbf{u}_s - \frac{1}{\rho}\nabla p^{k+1} \tag{1.28}$$

Note that the latter equation (1.28) can be regarded as a correction of the intermediate velocity $\mathbf{u}_s$ in order to obtain a divergence-free field. Although its simplicity, this method suffers from the low accuracy in time: indeed, in the case of no-slip boundary conditions we commit an error of magnitude $\mathrm{O}(\nu\sqrt{\Delta t})$ due to the error in the tangential component as reported in ([28]).
In substance, in order to decouple the system, Chorin created a "fictitious" boundary condition for the velocity (1.25),but in turn we have found a suitable condition for the pressure (1.27).
The second step of the scheme is named "Projection-step", since the velocity field $\mathbf{u}^{k+1}$ is the L^2-projection of $\mathbf{u}_s$ into the subspace of the divergence-free functions with a normal component to the boundary equals to zero (see [17] for further information).

1.2.4 The Compressible-PISO splitting scheme

The most famous splitting scheme that is nowadays adopted in the commercial codes of computational fluid dynamics is the "PISO" scheme (Pressure implicit with Splitting of Operators). Developed in [15], it was thought at the beginning only for the incompressible flows, and then it was extended for the compressible flows by adding the energy equation and the ideal gas-law. In this thesis, we will use a generalization of this type of scheme to handle the problem.
The algorithm is the following:

- **Step 1**: Find an intermediate density ρ^* from the mass equation:

$$\frac{\rho^*}{\Delta t} + \nabla \cdot (\rho^n \mathbf{v}^n) = b_\rho^n \quad b_\rho^n = \frac{\rho^n}{\Delta t} \tag{1.29}$$

- **Step 2**: Solve the energy equation for enthalpy Λ^{n+1}:

$$\frac{\rho^* \Lambda^{n+1} - \rho^n h^n}{\Delta t} + \frac{\rho^* - \rho^n}{\Delta t} k^n + \nabla \cdot (\rho^n \mathbf{v}^n (\Lambda^{n+1} + k^n)) = \rho^* (\mathbf{v}^n \cdot g\widehat{\mathbf{k}}) \tag{1.30}$$

- **Step 3**: Calculate the temperature T, the compressibility ψ^{n+1}, and a new density $\widetilde{\rho}^{n+1}$:

$$T^{n+1} = \frac{\Lambda^{n+1}}{c_p} \quad \psi^{n+1} = \frac{1}{RT^{n+1}} \quad \widetilde{\rho}^{n+1} = \frac{p^n}{RT^{n+1}} \tag{1.31}$$

- **Step 4**: Find the end step velocity $\mathbf{v}^{n+1}$ and pressure p^{n+1}

$$\frac{\widetilde{\rho}^{n+1}\mathbf{v}^{n+1}}{\Delta t} + \nabla \cdot (\rho^n \mathbf{v}^n \otimes \mathbf{v}^{n+1}) + \nabla p'^{n+1} + gz\nabla\widetilde{\rho}^{n+1} = \mathbf{b}_v^{n+1} \quad \mathbf{b}_v^{n+1} = \frac{\rho^n \mathbf{v}^n}{\Delta t} \tag{1.32}$$

$$p^{n+1} = p'^{n+1} + \widetilde{\rho}^{n+1} gz \tag{1.33}$$

Details about the procedure that allows us to decouple the pressure and the velocity will be given later, after a quick explanation of the discretization procedure because, as will be clear later, the p'-U decoupling procedure is dependent on the discretization.

Chapter 2

Stabilization techniques for the N.S. Equation

In this chapter we will present the concept of "spatial filtering" starting from the incompressible Navier Stokes equation. We will see the connection between the spatial filtering and the approach that will be adopted in the numerical experiments.

2.1 Motivation

Nowadays, it is computationally prohibitive solve the Navier-Stokes equation with a DNS[1], since it requires a too fine mesh for the convective-dominant flows. If we refer to the Kolmogoroff theory, the turbulent kinetic energy is extracted from the larger eddies from the mean flow, and when they break down, they transfer energy to the smaller scale eddies. At a scale comparable to the Kolmogoroff scale $\eta = Re^{-3/4}L$ the energy is dissipated. This process is called the "Energy-cascade" process. Thus, in order to simulate all the scales, a DNS necessitates a mesh grid comparable with the Kolmogoroff scale. In general, we are interested in simulating large scales flows $L \approx 10$km. Since the order of magnitude of the speed is $\simeq 10m/s$ for our case the smallest eddy has a dimension of about:

$$\eta \approx 1.7e-4\text{m} \tag{2.1}$$

From here, we understand that doing a mesh grid big as $L = 10$km but dense as $h = \eta$ is computationally prohibitive. For this reason, the Navier Stokes equations are averaged in time (bringing to the URANS) or in space (bringing to the LES). In this thesis, we adopt a generalization of the second approach, as will be discussed later.

[1]DNS which stands for Direct Numerical Simulation

2.1.1 Large Eddy Simulation

One of the most popular techniques to simulate turbulent flows is the Large Eddy Simulation (LES). Following the Kolmogoroff theory, large eddies are geometry-dependent, while the smallest one are "universal". This observation allows us to treat the calculations only for the large ones, while the smallest are treated implicitly thanks to a filter operation. Despite its originality, after the filtering operation many non linear-terms are created and usually they are neglected or related to the filtered quantity for mathematical convenience. The LES technique is attractive for the atmospheric community since the large scale unsteadiness is the focus (especially at the scale of tens of kilometers). Indeed, much of the pioneering work on the LES was motivated by meteorological applications [18],[21]
In summary, the errors introduced by the LES approach could be due to the following issues

1. Errors due to the disregarding of non linear terms
2. Modelling errors in the Reynolds stresses
3. Errors due to space and time discretization

Note that the LES methodologies are developed not only on the basis of the flow we are interested in simulating, but also on the numerical method used. There are several variants of the LES, depending on the applications:

- **Large eddy simulation with near-wall resolution:** This is the case in which the grid is sufficiently fine everywhere in order to resolve the 80% of the energy
- **Large eddy simulation with near-wall modelling:** This is the case in which the grid is very fine remote from the walls, but not in the rear wall region
- **Very Large Eddy Simulation (VLES):** This is the case in which the grid is too coarse to resolve the 80% of energy

In LES, the filtered velocity field accounts for a fraction of the turbulent kinetic energy (say 80%) in the flow field, while in the VLES a substantial fraction of the kinetic energy resides in the residual motions, thus, the simulation is strongly dependent on the modelling of the residual motion itself.

Outline of the method:
In DNS, the velocity field $\mathbf{U}(\mathbf{x}, t)$ has to be resolved on lenghtscales smaller than the kolmogoroff scale η. In LES, a low pass filtering operation is performed so that the resulting filtered velocity field $\overline{\mathbf{U}}(\mathbf{x}, t)$ can be resolved on a coarser grid with respect to the DNS. Specifically, the grid spacing h is related to a "filter width" Δ.
We end up, after the filtering operation, with a decomposition:

$$\mathbf{U}(\mathbf{x}, t) = \overline{\mathbf{U}}(\mathbf{x}, t) + \mathbf{u}'(\mathbf{x}, t)$$

Where $\mathbf{u}'(\mathbf{x}, t)$ is a field that has a filtered quantity which is not zero in general:

$$\overline{\mathbf{u}}'(\mathbf{x}, t) \neq 0$$

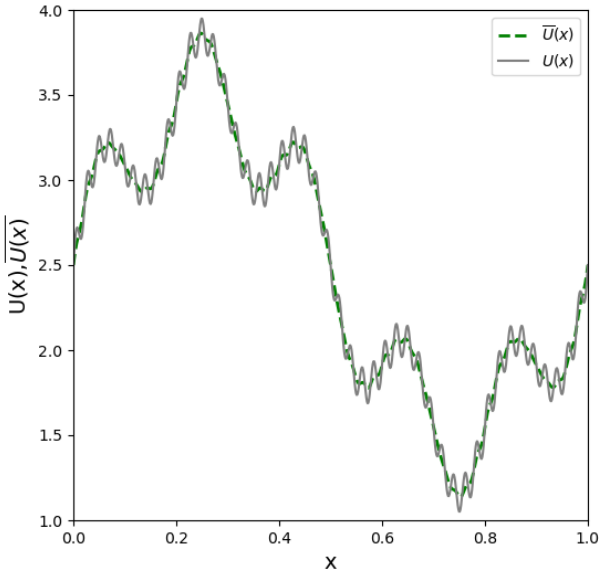

Figure 2.1: Effect of the filtering operation: we have denoted with $\overline{U}$ the filtered quantity

As one can see from fig. 2.1, the effect of the filter, in general, is to "damp" the high oscillations, thus, it acts as a low pass filter.
Next, we will give a general overview of the filtering operation from the mathematical point of view. We will start to analyze the incompressible framework and then we will extend the theory to the compressible one.

2.1.2 Filtering Techniques for fluid-flow

Before presenting the final methodology that will be used for the numerical experiments, we provide some fundamental notions in order to understand why our approach works. The concept of "stabilization" could be associated with the dumping of the high frequency content contained in the solution. One of the methodologies for doing so is to solve the following "Helmholtz" problem for $\alpha \in (0,1)$:

$$-\nabla^2\overline{u} + \alpha\overline{u} = \alpha u \ in \ \Omega \tag{2.2}$$

$$\overline{u} = u \ on \ \ \partial\Omega \tag{2.3}$$

For simplicity, suppose that we are in 1D (the same apply in more dimensions), we can express u via a sine expansions $u(x) = \sum_k \widehat{u_k} sin(k\pi x)$, resulting in:

$$\overline{u(x)} = \sum_k \frac{\alpha}{\alpha + \pi^2 k^2} \widehat{u}_k sin(k\pi x) \tag{2.4}$$

Eq.(2.4) tells us that the filtered solution $\overline{u}$ has a coefficient in the sine expansion $\overline{\widehat{u}_k} = \frac{\alpha}{\alpha+\pi^2k^2}\widehat{u}_k < \widehat{u}_k$. This justify "intuitively" why the operator $I - \nabla^2$ acts as a low-pass filter:

indeed, the bigger the spatial frequency k, the more the coefficient is dumped respect the original coefficient. For clarity, we show in fig. 2.2 the coefficient expansion for the function $U(x) = 0.25sin(2x)+0.4sin(4x)+0.5sin(8x)+0.45sin(10x)+0.4sin(20x)$ before and after the filtering operation. Note the quadratic dependence of the coefficient on the frequency: this tells us that only the modes with the lowest frequency are retained, while the modes with high frequencies are easily dumped, making the solution more smooth.

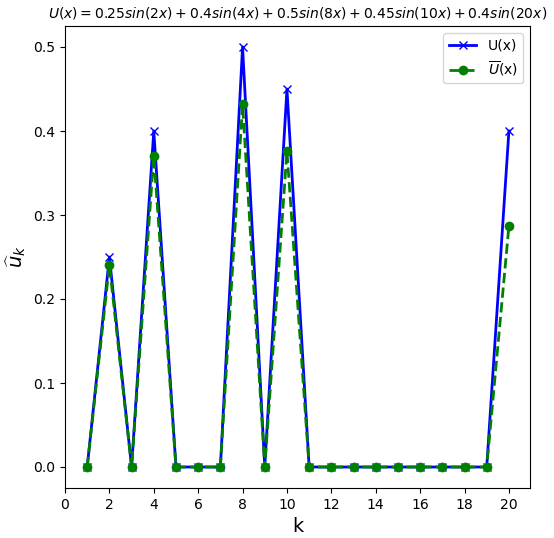

Figure 2.2: Coefficients of the function $U(x) = 0.25sin(2x) + 0.4sin(4x) + 0.5sin(8x) + 0.45sin(10x) + 0.4sin(20x)$ and their coefficients after the filtering operation (2.3)

2.1.3 Differential Filters for the Large Eddy Simulation

In the Large Eddy Simulation the variables are filtered in space such that:

$$\overline{f}(\mathbf{x}, t; \delta) = \int \mathbf{G}(\mathbf{x} - \mathbf{x}'; \delta) f(\mathbf{x}', t) d^3\mathbf{x}' \tag{2.5}$$

Where **G** is the Kernel of the filter. Intuitively, the sense of this integral is to do a "spatial average" with a "weight" spatial distribution given by G the quantity f. Assuming to adopt uniform spatial filters (in this manner the derivatives commute with the filtering operation), the filtered incompressible Navier-Stokes equations become:

$$\frac{\partial \overline{u_k}}{\partial x_k} = 0 \tag{2.6}$$

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial \overline{u_i u_k}}{\partial x_k} = -\frac{\partial \overline{p}}{\rho \partial x_i} + \nu \nabla^2 \overline{u_i} \tag{2.7}$$

These equations differs from the Navier–Stokes equations because the term $\overline{u_i u_k}$ is different than the product of the filtered velocities $\overline{u_i}\ \overline{u_k}$. The difference is called "residual stress tensor" $\tau^R = \overline{u_i u_k} - \overline{u_i}\ \overline{u_k}$

By decomposing the residual stress tensor in an anisotropic part and in an isotropic part:

$$\tau_{ij}^A = \tau_{ij}^R - \frac{2}{3}k_r\delta_{ij}$$

Where $k_r = \frac{1}{2}\tau_{ii}^R$ is the residual kinetic energy, we can include the isotropic part into the pressure:

$$\overline{p} = \overline{p} + \frac{2}{3}\rho k_r$$

We obtain the filtered momentum equation (2.8):

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u}\cdot\nabla\overline{u_i} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu\nabla^2\overline{u_i} - \nabla\cdot\boldsymbol{\tau}^r \tag{2.8}$$

Like the Reyndolds averaged Navier Stokes equations, eq.(2.8) and (2.6) are unclosed. Closure is achieved by modelling the "subgrid scale stress" $\boldsymbol{\tau}^r$.
One of the earliest model to do so was proposed by [18] in the context of meteorology. The anisotropic part of the residual stress tensor is linked to the filtered rate of strain:

$$\tau_{ij}^r = -2\nu_r\overline{S_{ij}}$$

Where $\nu_r(\mathbf{x},t) = (C_s\Delta)^2\overline{S}$ is the eddy viscosity related to the characteristic filtered rate of strain $\overline{S} = (2\overline{S_{ij}}\ \overline{S_{ij}})^{1/2}$. With this closure, we can solve directly the equations (2.8)-(2.6) and obtain the filtered quantities without the explicit specification of the filter. At this point a question arises: How do we calculate the specific filter from the filtered quantities $\overline{\phi}$? From homogeneous and isotropic turbulence there exists an unique filter, the "*Smagorinsky filter*" that resembles the Gaussian filter, see [5],chapter 13 for further references.
The most well known filters are the *Gaussian*, the *Box* filter and the *Sharp spectral* filter depending on the function $\mathbf{G}$.
A suitable integral filter is the Gaussian one, with the Kernel given by:

$$\mathbf{G}_1(\mathbf{x}-\mathbf{x}';\delta) = \left(\frac{6}{\pi\delta^2}\right)^{3/2} exp\left(-\frac{6(\mathbf{x}-\mathbf{x}')^2}{\delta^2}\right) \tag{2.9}$$

Observation 1. *The Gaussian filter gives a filtered function $\overline{f}$ such that:*

$$\frac{\partial \overline{f}}{\partial \epsilon} = \nabla^2\overline{f} \quad \epsilon = \delta^2/24 \tag{2.10}$$

Indeed, if we look closely at the function $\mathbf{G}$ (2.9), we observe that it is the Green function associated by the heat equation with $\epsilon = \delta^2/24$.
In this manner, if we assume that:

$$\frac{\partial \overline{f}}{\partial \epsilon} \approx \frac{\overline{f} - f}{\epsilon} \tag{2.11}$$

We obtain the equation(2.2) with $\epsilon = 1/\alpha$. In summary, the application of the filter function $\mathbf{G}$ is almost equivalent (for small *epsilon*) to the problem (2.3). As demonstrated

above, the filtered solution contains frequencies that are dumped, for this reason the strategy in discussion is a suitable stabilization technique.
Obviusly, the results obtained can be generalized in various forms, including many differential filters: for example, the kernel function $\mathbf{G}$ such that:

$$\mathbf{G}_2(\mathbf{x}-\mathbf{x}';\delta) = \frac{1}{2\delta}\exp\left(-\frac{|x-x'|}{\delta}\right) \tag{2.12}$$

is a solution of the differential equation:

$$f = \overline{f} - \delta^2 \frac{\partial \overline{f}}{\partial x^2} \tag{2.13}$$

Note also that in this case $\overline{f}$ is exactly the solution of an Helmholtz problem with $\alpha = 1/\delta^2$, in contrast than before, where with the Gaussian Filter the filtered solution was only an "approximation" of the problem. The two Kernel functions (2.9),(2.12) are shown in fig. 2.3.

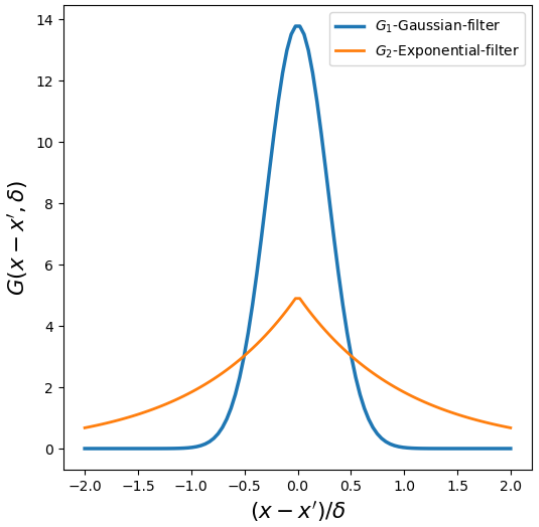

Figure 2.3: Kernel functions for the Gaussian filter (2.9) and for the Exponential filter (2.12)

Having given an overview of the idea behind the LES for incompressible flows, we will present the LES approach in the compressible framework.
For compressible flows, Favre[9] introduced the following filtering operation:

$$\widetilde{\phi} = \frac{\overline{\rho\phi}}{\overline{\rho}} \tag{2.14}$$

Which in general does not commute with the derivative operators ∂_t, ∂_i.
Applying the spatial-filtering operation to the original Navier-Stokes equations we obtain:

$$\begin{aligned} \partial_t \overline{\rho} + \partial_j(\overline{\rho}\widetilde{u}_j) &= 0 \\ \partial_t(\overline{\rho}\widetilde{u}_i) + \partial_j(\overline{\rho}\widetilde{u}_i\widetilde{u}_j) &= -\partial_i\overline{p} + [\partial_j\widetilde{\tau}_{ij} - \partial_j(\overline{\rho}a_{ij})] \\ &\quad +[\partial_j(\overline{\tau}_{ij} - \widetilde{\tau}_{ij})] \end{aligned} \tag{2.15}$$

Where the quantity a_{ij} and $\widehat{\tau}_{ij}$ are the quantities defined as:

$$\begin{aligned} a_{ij} &= \widetilde{u_i u_j} - \widetilde{u}_i\widetilde{u}_j \\ \widehat{\tau}_{ij} &= \frac{\mu}{Re}(\partial_j\widetilde{u}_i + \partial_i\widetilde{u}_j - \frac{2}{3}\delta_{ij}\partial_k\widetilde{u_k}) \end{aligned} \tag{2.16}$$

In the filtered energy equation the total energy is written as:

$$\widehat{e} = \frac{\overline{p}}{\gamma - 1} + \frac{1}{2}\overline{\rho}\widetilde{u}_k u_k = \overline{e} + \frac{1}{2}\overline{\rho}a_{kk} \tag{2.17}$$

When applying the same filtering operation to the energy equation we obtain many non-linear terms:

$$\begin{aligned} \partial_t\widehat{e} = &-(\partial_j((\widehat{e} + \overline{p})\widetilde{u}_j)) \\ &+ \left[\partial_j(\widehat{\tau_{ij}}\widetilde{u}_i) - \partial_j\widehat{q}_j - \frac{1}{\gamma - 1}\partial_j(\overline{pu_j} - \overline{p}\widetilde{u_j}) - (\overline{p\partial_k u_k} - \overline{p}\partial_k\widetilde{u_k}) - \partial_j(\overline{\rho}a_{kj}\widetilde{u_k})\right] \\ &+ \left[\overline{\rho}a_{kj}\partial_j\widetilde{u_k} + \overline{\tau_{kj}\partial_j u_k} - \overline{\tau_{kj}}\partial_j\widetilde{u_k} + \partial_j(\overline{\tau_{ij}}\widetilde{u}_i - \widehat{\tau_{ij}}\widetilde{u}_i) - \partial_j(\overline{q_j} - \widehat{q}_j)\right] \end{aligned} \tag{2.18}$$

It is evident that the additional terms that appear in this equation are unrelated to the filtered quantities $(\overline{\rho}, \widetilde{u}, \widehat{e})$ and furthermore their treatment is extremely computational expensive. To derive the solvable equations, we must approximate all the terms that involve correlations between fluctuating quantities and most of the time they are approximated. Indeed, the non linear terms are often neglected and only the terms a_{ij} and $\partial_j(\overline{pu_j} - \overline{p}\widetilde{u_j})$ are modelled. Thus, dealing with this approach is quite complicated and most often we don't have a complete picture of this methodology.

2.2 Filter Based Stabilization for evolution equations

In this chapter we will generalize the concept of "stabilization" for general evolutions equations. In particular, we will build the basic building blocks that will be used for the Navier Stokes case.
Let us consider the explicit method for a non linear evolution equation:

$$\frac{\partial u}{\partial t} + F(u) = 0$$

Indicating with the overbar the action of a local spatial filter (eg. the differential filter presented in section 2.1.2), the so-called methodology "Evolve then Filter" provides the following uncoupled steps:

Algorithm 1 Evolve then Filter for Evolution Equations

1: **Step 1: Evolve** $\frac{w^{n+1}-u^n}{\Delta t} + F(u^n) = 0$
2: **Step 2: Filter** $u^{n+1} = \overline{w^{n+1}}$

Note that both steps can be done by black box modules.
Eliminating the step 2 gives :

$$\frac{u^{n+1} - u^n}{\Delta t} + F(u^n) + \frac{1}{\Delta t}(w^{n+1} - \overline{w^{n+1}}) = 0$$

which is a time relaxation discretization with time relaxation coefficient $1/\Delta t$. Using the differential filter of eq.(2.2) we obtain the following problem:

$$\frac{u^{n+1} - u^n}{\Delta t} + F(u^n) - \frac{\delta^2}{\Delta t}\nabla^2 u^{n+1} = 0$$

Which is a time discretization of the Artificial Viscosity-Method. As pointed out in [7], the artificial viscosity coefficient is $\delta^2/\Delta t$ which can give a large amount of numerical diffusion depending on Δt and δ.

2.2.1 Filter based stabilization of implicit methods

The algorithm 1, if not properly done, introduces a too large amount of numerical diffusion: indeed, one approach to overcome this issue is to apply a deconvolution and then "relax" after the filter step.
Let $V \hookrightarrow L \hookrightarrow V'$ be hilbert spaces with duality pairing an extension of the L-inner product, denoted by $\langle ., . \rangle_L$ and $\| \cdot \|_L$ its associated norm.
Consider the following problem:
Find $u : [0, T] \to V$ satisfying $u(0) = u^0 \in V$ such that:

$$\frac{du}{dt} + F(u) = 0 \quad \forall\, t > 0 \tag{2.19}$$

Let us introduce two linear, self adjoint, bounded and positive operators $G : V' \to V$ (the Filter operator), and $D : V \to V$(the Deconvolution operator).
In order to have an exactly energy-conserving method, suppose that $F : V \to V'$ satisfy:

$$\langle F(v), v \rangle_L = 0 \quad \forall\, v \in V$$

The algorithm 2 is a modified version of the algorithm 1 that aims to reduce the numerical dissipation by the application of a "Deconvolution operator" and the application of a "relax step".

Algorithm 2 Evolve-filter-Deconvolve-Relax algorithm

1: **Step 1: Evolve** $\frac{w^{n+1}-u^n}{\Delta t} + F(\frac{w^{n+1}+u^n}{2}) = 0$
2: **Step 2: Filter** $\overline{w^{n+1}} = G(w^{n+1})$
3: **Step 3: Deconvolve** $\widetilde{w} = D(\overline{w^{n+1}})$
4: **Step 4: Relax** $u^{n+1} = (1-\chi)w^{n+1} + \chi\widetilde{w}$

The relaxation has the goal of reducing the numerical diffusion introduced by the filter step by doing a linear combination between the Evolve and the Deconvolution step solutions, while the Deconvolution has the role of increasing the accuracy.
As demonstrated in [7] (theorem 1.2), the following estimate holds:

$$\|w^{n+1}\|_L^2 - \|u^{n+1}\|_L^2 = \chi(2-\chi)\langle(I - D\,G)w, w\rangle_L + \chi^2\langle(I - D\,G)w, D\,Gw\rangle_L$$

Furthermore, we have an estimate of the numerical viscosity injected every time step:

$$\frac{\chi(2-\chi)}{\Delta t}\langle(I - D\,G)w, w\rangle_L + \frac{\chi^2}{\Delta t}\langle(I - D\,G)w, D\,Gw\rangle_L \geq 0 \tag{2.20}$$

Since $(I - D\,G)$ is semidefinite positive and $D\,G$ commutes with $(I - D\,G)$ the two terms in eq.(2.20) are both positive.
Adopting $\chi \simeq \Delta t$ the second term becomes an high order term, thus, the numerical dissipation is dominated by the first term. However, the analytical determination of the optimal parameters χ and δ is an open problem since fluid flow problems are complex in general.

Remark. *(Interpretation of the numerical dissipation terms)*
Suppose that the problem is periodic $L^2(0,2\pi)$, G a differential filter ($G = (-\delta^2\Delta + 1)^{-1}$ under periodic boundary conditions), D a van-Cittert deconvolution operator 2.3.7. The action of the numerical dissipation can be calculated by the wavenumber of the fourier series associated to the solution. Let $w(x) = \sum_k \widehat{w}(k)e^{-ikx}$, the first two van-Cittert deconvolution operators are $D_0 = I$ and $D_1 = 2I - G$.
Trasforming the first term of the numerical dissipation in eq. (2.20) give us:

$$((I - D\,G)w, w) = \sum_k \widehat{(I - D\,G)}(k)|\widehat{w}(k)|^2$$

Indeed, for D_0 the Fourier coefficient $\widehat{(I - D_0\,G)}$ can be calculated by applying the Fourier-transform to the original equation $(-\delta^2\nabla^2 + 1)\overline{u} = u$, allowing us to obtain $\widehat{\overline{u}}(k) = \frac{1}{1+\delta^2k^2}\widehat{u}(k)$, thus, $\widehat{G} = \frac{1}{1+\delta^2k^2}$. Using the linearity property of the Fourier transform we obtain: $\widehat{I - D_0\,G} = 1 - \frac{1}{1+\delta^2k^2} = \frac{\delta^2k^2}{1+\delta^2k^2}$. Repeating the procedure for $\widehat{(I - D_1\,G)}$ we obtain: $\widehat{I - D_1\,G} = \left(\frac{\delta^2k^2}{1+\delta^2k^2}\right)^2$.
Analogous to the first term, the Fourier transform for the second term in eq.(2.20) give us (for N=0): $D_0\,\widehat{G(I - D_0\,G)}(k) = \frac{\delta^2k^2}{(1+\delta^2k^2)^2}$.
Fig. 2.4 shows the coefficients we stated before (for fixed $\delta = 1.0$). Note that:

- *The numerical diffusion (amplification factor) increases as the wave number increases*

- *Increasing the order of the deconvolution decreases the diffusion.*
- *The green dashed line suggest that the second term is small respect the first term.*

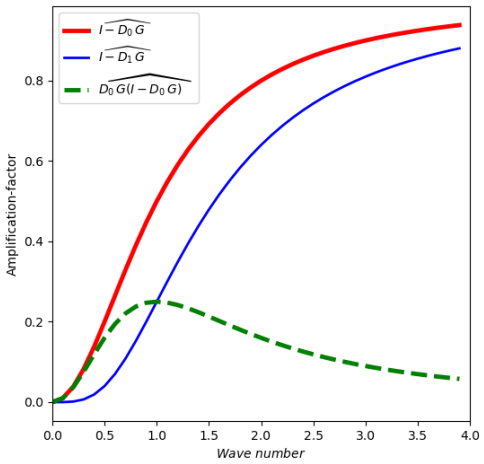

Figure 2.4: Fourier coefficients for $(\widehat{I - D_0 G})$, $(\widehat{I - D_1 G})$, $(\widehat{D_0 G(I - D_0 G)})$

The theoretical analysis of the Evolve then Filter for the incompressible Navier-Stokes equations is performed in [7] in the context of the finite elements. However, despite its simplicity and its property of being well-posed, the method is well-known to over-regularize solutions and even remove critical flow structures.
The first attempt to solve this problem was made in [34] by modifying the filter of the model and adaptively choose the filtering radius. The basic idea is to modify the filter-step by introducing a sort of function that tells us "where" we must regularize the flow:

$$-\nabla^2 (a(u)\overline{u}) + \alpha\overline{u} = \alpha u$$

Hereinafter, we will refer to the Evolve-Filter-Relax algorithm by considering this modified version for the filter step.

2.3 The Evolve-Filter-Relax approach for the incompressible Navier Stokes problem

Having given all the elements necessary for the full methodology, in this section we will describe how to couple the filtering procedure with the complete problem 1.3.
In this work, we consider a generalization of the Evolve-Filter-Relax approach proposed originally in [34] for the incompressible case by using the finite element method as discretization technique.

2.3.1 The EFR in the incompressible framework

Thanks to the non linear convective term of the Navier Stokes equation, the different scales present in the fluid are not break down uniformly, moreover, intermittence, locality and backscatter are typical features in the energy cascade of high Reynolds number flow problems. Indeed, linear terms break down eddies into smaller ones until they are small enough to be dissipated by molecular viscosity or by numerical dissipation. The first attempt to develop a methodology to stabilize non-linearly the velocity in the incompressible framework was done in [34] in the context of the finite element discretization. To give a brief introduction of the algorithm, consider the Navier-Stokes equations (NSE) under no slip boundary conditions in domain Ω:
Let be $(\mathbf{u}, p)$ the velocity and the pressure respectively.
We are interested in finding $(\mathbf{u}, p)$ such that:

$$\mathbf{u}_t + u \cdot \nabla \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p = f(x,t) \qquad \text{in } \Omega \,\mathrm{x}\, (0,T] \tag{2.21}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \partial\Omega \,\mathrm{x}\, (0,T] \tag{2.22}$$

$$\mathbf{u}(x,0) = \mathbf{u}^0(x) \qquad \text{in } \Omega \tag{2.23}$$

Given a method for the NSE (after the discretization in time we obtain a generalized stokes problem), we consider a method for adapting the generalized stokes problem to high Reynolds number flows that is modular and consist of adding viscosity "locally", based on the behaviour of the flow.
Suppressing the spatial discretization, consider the algorithm 3.

Algorithm 3 Evolve-Filter-Relax for the incompressible N.S.E.

1: **Step 1: Evolve** $\frac{w^{n+1}-w^n}{\Delta t} + w^{n+1} \cdot \nabla w^{n+1} - \nu \nabla^2 w^{n+1} + \nabla p^{n+1} = f^{n+1}$
and $\nabla \cdot w^{n+1} = 0$
2: **Step 2: Nonlinear Filter** $w^{n+1} \to \overline{w^{n+1}}$
3: **Step 3: Relax** $u^{n+1} = (1-\chi) w^{n+1} + \chi \overline{w^{n+1}}$

To specify this nonlinear filter, we select an indicator function $a = a(u, \nabla u, ...)$ (denoted by a(u)) with the properties:

$$0 \le a(u) \le 1 \text{ for any fluid velocity } u(x,t) \tag{2.24}$$

$$a(u) \simeq 0 \text{ selects regions requiring no local filtering} \tag{2.25}$$

$$a(u) \simeq 1 \text{ selects regions requiring local filtering} \tag{2.26}$$

We define the non-linear filtered average $\overline{w}$ as the solution, under appropriate boundary conditions, of:

$$-\nabla \cdot (\delta^2 a(w) \nabla \overline{w}) + \nabla \lambda + \overline{w} = w \quad \text{and} \quad \nabla \cdot \overline{w} = 0 \text{ in } \Omega \tag{2.27}$$

Where λ is the Lagrange multiplier that enforces the incompressibility constraint for $\overline{w}$

To motivate step 2 and 3, let $\chi = 1$ and rearrange steps 1 and 2, this gives :

$$\frac{u^{n+1}-u^n}{\Delta t} + w^{n+1}\cdot\nabla w^{n+1} - \nu\nabla^2 w^{n+1} + \nabla p^{n+1} + \frac{1}{\Delta t}(w^{n+1} - \overline{w^{n+1}}) = f^{n+1} \tag{2.28}$$

By using the filtering step 2.27 :

$$\frac{u^{n+1}-u^n}{\Delta t} + w^{n+1}\cdot\nabla w^{n+1} - \nu\nabla^2 w^{n+1} + \nabla(p^{n+1}+\lambda^{n+1}) - \frac{\delta^2}{\Delta t}\nabla\cdot(a(w^{n+1})\nabla u^{n+1}) = f^{n+1} \tag{2.29}$$

Remark. *(Non linear filters)*
For simplicity, we have adopted an explicit method, however, the result can be extended immediately to the so-called IMEX method (with $w^{n+1}\cdot\nabla w^{n+1}$ replaced by $u^n\cdot\nabla u^{n+1}$ or used with the Crank-Nicholson scheme).

Note that, as for the original Evolve-Filter-Relax (2.2), we have obtained a term that injects artificial viscosity.

2.3.2 The EFR in the compressible framework

Let us discretize in time problem 1.3, let $\Delta t = t_0 + \Delta t$, with $n = 0,,,N_T$ and $T = t_0 + N_T\Delta t$. Denote by y^n the approximation of a quantity y at time n and denote by Ω the domain of the equations. For time discretization of system (1.1)-(1.4) we adopt a Backward Differentiation Formula of order 1 (BDF1). At our knowledge it is the first time that this method is applied to the compressible framework, while it is well known that this methodology was applied for the incompressible case in [22, 24] in a finite volume settings.
Denoting by $\Lambda = c_pT$ the specific enthalpy, let us consider the following algorithm for the equation (1.1)-(1.4):

- **Step 1: Evolve:** Find ρ^{n+1}, $\mathbf{v}^{n+1}$, Λ^{n+1}, p'^{n+1} such that:

$$\frac{\rho^{n+1}-\rho^n}{\Delta t} + \nabla\cdot(\rho\mathbf{v})^{n+1} = 0 \tag{2.30}$$

$$\frac{(\rho\mathbf{v})^{n+1}-(\rho\mathbf{u}^n)}{\Delta t} + \nabla\cdot((\rho\mathbf{v})^{n+1}\otimes\mathbf{v}^{n+1}) + \nabla p'^{n+1} + gz\nabla\rho^{n+1} = 0 \tag{2.31}$$

$$\frac{\rho^{n+1}(\Lambda^{n+1}+(\mathbf{v}^{n+1})^2/2) - \rho^n(\Lambda^n+(\mathbf{u}^n)^2/2)}{\Delta t} + \nabla\cdot\left((\rho\mathbf{v})^{n+1}\left(\Lambda^{n+1}+\frac{(\mathbf{v}^{n+1})^2}{2}\right)\right) - \frac{p^{n+1}-p^n}{\Delta t} = \rho^{n+1}\mathbf{v}^{n+1}\cdot g\widehat{\mathbf{k}} \tag{2.32}$$

$$p'^{n+1} = \rho^{n+1}RT^{n+1} - \rho^{n+1}gz \tag{2.33}$$

- **Step 2: Filter:** Find $\overline{\mathbf{u}}^{n+1}$, $\overline{\Lambda}^{n+1}$ such that:

$$\overline{\mathbf{u}}^{n+1} - \nabla\cdot(\nu_{art}(\mathbf{v})\nabla\overline{\mathbf{u}}^{n+1}) = \mathbf{v}^{n+1} \tag{2.34}$$

$$\overline{\Lambda}^{n+1} - \nabla\cdot(\nu_{art}(\mathbf{v})\ \nabla\overline{\Lambda}^{n+1}) = \Lambda^{n+1} \tag{2.35}$$

In this step we "inject" a quantity of viscosity dependent from ν_{art} which is a function of the velocity at the end of the evolve step:

$$\nu_{art}(\mathbf{v}) = \delta^2 a(\mathbf{v}) \tag{2.36}$$

Where δ is the filtering radius, which is the radius of the neighborhood from which the filter extracts information. The scalar function $a(\mathbf{v})$ is the so-called "indicator function". Its behavior is the following:

$$a(\mathbf{v}) \approx 0 \quad \text{where we don't regularize the flow}$$

$$a(\mathbf{v}) \approx 1 \quad \text{where we regularize the flow}$$

The indicator function is extremely important for the success of the model. We will discuss the different choices of the indicator function later.

- **Step 3: Relax:** Find the final step entalphy h and velocity $\mathbf{u}$ such that:

$$h^{n+1} = (1-\chi)\Lambda^{n+1} + \chi\overline{\Lambda}^{n+1} \tag{2.37}$$
$$\mathbf{u}^{n+1} = (1-\xi)\mathbf{v}^{n+1} + \xi\overline{\mathbf{u}}^{n+1} \tag{2.38}$$

Where $\xi, \chi \in [0,1]$ are two relaxation parameters that are problem dependent

We will refer to this stabilization strategy as EFR (EVOLVE-FILTER-RELAX) algorithm. The advantages of the EFR algorithm are summarized as follows:

1. Modularity: Its implementation does not require any modification of a legacy solver
2. Flexibility in the choice of filters in step 2 and also the possibility to adjust the solution at the final step 3.

Note that the Filter step in our case is applied also to the enthalpy, while in the incompressible framework it was applied only to velocity
Moreover, in our case there is no need to enforce the incompressibility constraint as in the original algorithm (1) since the velocity is not exactly divergence free.
Let us divide the equations (2.34)-(2.35) by Δt:

$$\frac{\overline{\mathbf{u}}^{n+1} - \mathbf{v}^{n+1}}{\Delta t} - \nabla \cdot \left(\frac{\nu_{art}(\mathbf{v})}{\Delta t} \nabla \overline{\mathbf{u}}^{n+1} \right) = 0 \tag{2.39}$$
$$\frac{\overline{\Lambda}^{n+1} - \Lambda^{n+1}}{\Delta t} - \nabla \cdot \left(\frac{\nu_{art}(\mathbf{v})}{\Delta t} \nabla \overline{\Lambda}^{n+1} \right) = 0 \tag{2.40}$$

Equations (2.39)-(2.40) resemble a generalized Stokes problem in which the pressure contribution and the convective term is neglected.

2.3.3 The EFR as an inexact splitting

To take the computational cost low, the system (2.30)-(2.33) is linearized: the convective terms are linearized, the kinetic energy term and the gravity term are treated explicitly.

$$\frac{\rho^{n+1}-\rho^n}{\Delta t}+\nabla\cdot(\rho\mathbf{v})^n=0 \tag{2.41}$$

$$\frac{(\rho\mathbf{v})^{n+1}-(\rho\mathbf{u}^n)}{\Delta t}+\nabla\cdot((\rho\mathbf{u})^n\otimes\mathbf{v}^{n+1})+\nabla p'^{n+1}+gz\nabla\rho^n=0 \tag{2.42}$$

$$\frac{\rho^{n+1}(\Lambda^{n+1}+(\mathbf{v}^n)^2/2)-\rho^n(h^n+(\mathbf{u}^n)^2/2)}{\Delta t}+\nabla\cdot\left((\rho\mathbf{u})^n(\Lambda^{n+1}+\frac{(\mathbf{v}^n)^2}{2})\right)-\frac{p^n-p^{n-1}}{\Delta t}=\rho^n\mathbf{v}^n\cdot g\widehat{\mathbf{k}} \tag{2.43}$$

$$p'^{n+1}=\rho^{n+1}RT^{n+1}-\rho^{n+1}gz \tag{2.44}$$

Let us define the linear operators associated with the convective terms for the momentum and the energy equations (2.42)-(2.43):

$$\mathcal{L}_c[(\rho\mathbf{u})^n]\mathbf{v}^{n+1}=\nabla\cdot((\rho\mathbf{u})^n\otimes\mathbf{v}^{n+1}) \tag{2.45}$$

$$\mathcal{L}_h[(\rho\mathbf{u})^n]\Lambda^{n+1}=\nabla\cdot((\rho\mathbf{u})^n\Lambda^{n+1}) \tag{2.46}$$

$$\mathcal{L}_k[(\rho\mathbf{u})^n]K^n=\nabla\cdot((\rho\mathbf{u})^nK^n) \tag{2.47}$$

Here, the notations $A[\mathbf{x}]\mathbf{y}$ means that the operator A is computed at $\mathbf{x}$ and then applied to the function $\mathbf{y}$.
Now, we will show that the EFR algorithm brings at a perturbation of the original problem (1.1)-(1.4). For simplicity, we will demonstrate this fact referring to the equation for the energy (2.32) with the assumption of treating explicitly the source term due to the gravity,the terms related to the kinetic energy and the time derivative of the pressure for simplify the notation.
Let us multiply the equation of the filtering step by $\rho^{n+1}\chi$ and sum to the equation for the evolve-step:

$$\frac{\rho^{n+1}\Lambda^{n+1}-\rho^nh^n}{\Delta t}+\frac{\rho^{n+1}(\mathbf{v}^{n+1})^2/2-\rho^n(\mathbf{v}^n)^2/2}{\Delta t}+\mathcal{L}_h[(\rho\mathbf{u})^n]\Lambda^{n+1}+\mathcal{L}_h[(\rho\mathbf{u})^n]K^n+$$
$$+\rho^{n+1}\frac{\chi}{\Delta t}(\overline{\Lambda}^{n+1}-\Lambda^{n+1})-\rho^{n+1}\chi\nabla\cdot(\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla\overline{\Lambda}^{n+1})=\frac{p^n-p^{n-1}}{\Delta t}+\rho^n\mathbf{v}^n\cdot\widehat{\mathbf{k}} \tag{2.48}$$

Rearranging conveniently the terms and exploiting the relaxation relation (2.38), we obtain:

$$\frac{\rho^{n+1}h^{n+1}-\rho^nh^n}{\Delta t}+\mathcal{L}_h[(\rho\mathbf{u})^n]h^{n+1}+\frac{\rho^{n+1}(\mathbf{v}^{n+1})^2/2-\rho^n(\mathbf{v}^n)^2/2}{\Delta t}+\mathcal{L}_k[(\rho\mathbf{u})^n]K^n-$$
$$\chi\rho^{n+1}\nabla\cdot(\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla\overline{\Lambda}^{n+1})+\chi\mathcal{L}_h[(\rho\mathbf{u})^n](\Lambda^{n+1}-\overline{\Lambda}^{n+1})=\frac{p^n-p^{n-1}}{\Delta t}+\rho^n\mathbf{v}^n\cdot\widehat{\mathbf{k}} \tag{2.49}$$

As we can see, the equation (2.49) represent the original equation in (1.3) with a perturbation term, that vanish when χ goes to zero. Hence, by choosing $\chi = O(\Delta t)$ we obtain an error of the order $\approx \Delta t$ which is coherent with the error of the discretization scheme (BDF1) and is coherent with the choice to treat explicitly the terms associated with the gravity and the convective terms.
Note also that the presence of the terms related to the kinetic energy could affect also the accuracy of the method since we don't use the velocity calculated at the end of the relaxation step.
Similar to the energy equation, we can substitute the filtering step onto the momentum equation and find the following equation:

$$\frac{\rho^{n+1}\mathbf{u}^{n+1} - \rho^n\mathbf{u}^n}{\Delta t} + \mathcal{L}_c[(\rho\mathbf{u})^n]\mathbf{u}^{n+1} + \xi\mathcal{L}_c[(\rho\mathbf{u})^n](\mathbf{v}^{n+1} - \overline{\mathbf{v}}^{n+1}) + \nabla p'^{n+1} + g z \nabla\rho^n - \xi\rho^{n+1}\nabla\cdot\left(\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla\overline{\mathbf{v}}^{n+1}\right) = 0 \tag{2.50}$$

Also for the momentum equation we obtain a perturbation for the convective term that vanishes as ξ goes to zero, while we obtain an artificial viscosity term that has the role to stabilize the solution.
Note that if we use eq. $(2.38)_1$ in the diffusion term we obtain:

$$\frac{\rho^{n+1}\mathbf{u}^{n+1} - \rho^n\mathbf{u}^n}{\Delta t} + \mathcal{L}_c[(\rho\mathbf{u})^n]\mathbf{u}^{n+1} + \xi\mathcal{L}_c[(\rho\mathbf{u})^n](\mathbf{v}^{n+1} - \overline{\mathbf{v}}^{n+1}) + \nabla p'^{n+1} + g z \nabla\rho^n - \rho^{n+1}\xi\nabla\cdot\left(\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla\mathbf{u}^{n+1}\right) - \rho^{n+1}\xi(1-\xi)\nabla\cdot\left(\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla(\overline{\mathbf{v}}^{n+1} - \mathbf{v}^{n+1})\right) = 0 \tag{2.51}$$

The treatment for the energy equation is analogous.
Equation (2.51) tells us that the E.F.R. algorithm introduces two inexact splitting errors that both depend on the relaxation parameter ξ

- A "convective" perturbation term
- Two extra "diffusion" perturbation terms

2.3.4 An estimate of the Relaxation parameter

In this subsection we will provide an estimate for the relaxation parameter.
Suppose we are able to solve the linearized momentum equation with a DNS:

$$\frac{(\rho\mathbf{u})^{n+1} - (\rho\mathbf{u})^n}{\Delta t} + \nabla\cdot((\rho\mathbf{u})^{n+1}\otimes\mathbf{u}^{n+1}) - \nabla\cdot{\boldsymbol{\sigma_{DNS}}}^{n+1} + g z\nabla\rho^n = 0 \tag{2.52}$$

With

$${\boldsymbol{\sigma_{DNS}}}^{n+1} = \nu\left[\left(\nabla\mathbf{u}^{n+1} + (\nabla\mathbf{u}^{n+1})^T\right) - \frac{2}{3}\left(\nabla\cdot\mathbf{u}^{n+1}\right)\mathbf{I}\right] - p'^{n+1}\mathbf{I} \tag{2.53}$$

By defining the following operator:

$$\mathcal{L}_E(\mathbf{v}^{n+1}, p'^{n+1}) = \nabla \cdot \left(\mathbf{u}^n \otimes \mathbf{v}^{n+1}\right) + \nabla p'^{n+1} + gz\frac{\nabla \rho^n}{\rho^n}$$

Assuming a constant density, we can rewrite the evolve step 2.31 as follows:

$$\frac{\mathbf{v}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathcal{L}_E(\mathbf{v}^{n+1}, p'^{n+1}) = 0$$

Multiplying the filter step (2.39) by ξ and adding to the equation 2.3.4 we obtain:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathcal{L}_E(\mathbf{v}^{n+1}, p'^{n+1}) - \nabla \cdot \boldsymbol{\sigma}_{EFR}^{n+1} = 0 \tag{2.54}$$

$$\sigma_{EFR}^{n+1} = \xi\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla \overline{\mathbf{u}}^{n+1} \tag{2.55}$$

Let $\nabla_\epsilon()$ denote the discrete gradient with a generic mesh size ϵ.
We require that the viscous contribution from the equivalent stress tensor (2.55) with a mesh size h matches perfectly the stress tensor of the DNS (2.53):

$$\xi\frac{\nu_{art}(\mathbf{v})}{\Delta t}\nabla_h \overline{\mathbf{u}}^{n+1} \simeq \nu\left[\left(\nabla_\eta \mathbf{u}^{n+1} + (\nabla_\eta \mathbf{u}^{n+1})^T\right) - \frac{2}{3}\left(\nabla_\eta \cdot \mathbf{u}^{n+1}\right)\mathbf{I}\right] \tag{2.56}$$

Assuming all the velocities of the same order of magnitude, we obtain:

$$\frac{\xi\nu_{art}}{h\Delta t} = \frac{4\nu}{3\eta} \qquad \nu_{art} = \delta^2 a(\mathbf{v})$$

Since we know that $a \leq \|a\|_\infty \leq 1$, we obtain a lower bound for the relaxation parameter:

$$\xi \simeq \frac{4}{3}\frac{\nu}{\nu_{art}}\frac{h}{\eta}\Delta t \geq \frac{4}{3}\frac{\nu}{\delta^2\|a\|_\infty}\frac{h}{\eta}\Delta t \geq \frac{4}{3}\frac{\nu}{\delta^2}\frac{h}{\eta}\Delta t \tag{2.57}$$

As will be clear from the results, while we have a practical estimate for the filtering radius for both the non-linear models, the estimate given by eq.(2.57) does not provide sufficient dissipation to stabilize the solution. For this reason, we will present most of the results by setting $\xi = \chi = 1$. With these values of the relaxation parameters, one of the two diffusion perturbation terms will vanish, while the inexact convection term is small as the numerical experiments will reveal.

2.3.5 Indicator Function for nonlinear filters

The breaking down of eddies into smaller ones until they are dissipated is a highly nonlinear process, even tough many models use linear filters to select the eddies to be dumped[2]. The indicator function dictates how the nonlinear filter manage the process of dissipation of the eddies. The most mathematically conveniently indicator function is related to the gradient of the velocity, thus, $a(\mathbf{u}) = \nabla\mathbf{u}$ (suitably normalized) could be a good choice because of its strong monotonicity properties. With this choice, we recover a Smagorinsky-like model [18], which is not sufficiently selective (it selects laminar shear flow as a regions of the domain with large turbulent fluctuations). However, this type of indicator function not only has strong monotonicity properties, but it is also meaningful from the physical point of view: e.g. the flow is regularized where there is a large shear flow.
Indicator function based on detecting coherent vortex-structures are based on the Q-criterion [34] and the model proposed by Vreman proposed in [33].
The Q-criterion is based on the quantity:

$$Q(\mathbf{u},\mathbf{u}) = \frac{1}{2}(\nabla^{ss}\mathbf{u} : \nabla^{ss}\mathbf{u} - \nabla^{s}\mathbf{u} : \nabla^{s}\mathbf{u}) > 0$$

Where $\nabla^{ss}\mathbf{u}$ and $\nabla^{s}\mathbf{u}$ are the antisymmetric part and the symmetric part of the gradient of velocity. An indicator function is then given:

$$a_Q(\mathbf{u}) = \frac{1}{2} - \frac{1}{\pi}\arctan\left(\frac{Q(\mathbf{u},\mathbf{u})}{\delta(|Q(\mathbf{u},\mathbf{u})| + \delta^2)}\right)$$

The model of Vreman, instead, vanishes for 320 coherent vortex structures and is based on a complex combination of the component of the velocity gradient tensor:

$$a_V(\mathbf{u}) = \sqrt{\frac{B(\mathbf{u})}{|\nabla\mathbf{u}|_F^4}}$$

$$B = \beta_{11}\beta_{22} + \beta_{22}\beta_{33} + \beta_{11}\beta_{33} - \beta_{12}^2 - \beta_{13}^2 - \beta_{23}^2 \quad \beta_{ij}(\mathbf{u}) = \sum_{m=1,2,3}\frac{\partial u_i}{\partial x_m}\frac{\partial u_j}{\partial x_m}$$

This filter was tested successfully in [1] with the Finite Element method.
Finally, the last type of filter related to the physics of the problem use the local helicity, defined as:

$$H = \mathbf{u}\cdot\boldsymbol{\omega} \quad \boldsymbol{\omega} = \nabla\times\mathbf{u}$$

From the Navier-Stokes equations in rotational form, it is possible to see that local high helicity suppresses local turbulent dissipation caused by breakdown of eddies into smaller ones, for this reason, an indicator function is then given by:

[2]This could be accomplished by solving the standard Helmholtz problem 2.3

$$a_H(\mathbf{u}) = 1 - \left| \frac{\mathbf{u} \cdot \omega}{|\mathbf{u}||\boldsymbol{\omega}| + \delta^2} \right|$$

Note that the filtering radius acts a normalizing factors for this type of filter.

Another class of indicator functions are the Deconvolution-Based types. As will be clear from the following subsection, these types of functions are meaningful from a mathematical perspective rather than a physical point of view. Despite that, the results will show that they are valid candidates to describe the process of the Kolmogoroff scale breaking.

2.3.6 Deconvolution-based indicator functions

Let V to be an Hilbert space. Let $\mathcal{F}$ be a linear, invertible, self-adjoint, compact operator $\mathcal{F} : V \to V$. The spectral theorem says that:

$$\mathcal{F}x = \sum_{i=0}^{\infty} \lambda_i \langle x, e_i \rangle e_i \quad \mathcal{F}^{-1}y = \sum_{i=0}^{\infty} \frac{1}{\lambda_i} \langle y, e_i \rangle e_i \tag{2.58}$$

Where λ_i are the eigenvalues of $\mathcal{F}$, and e_i are the corresponding eigenfunctions, which form an orthonormal basis for V. Let $\mathcal{D}$ be a bounded regularized approximation of $\mathcal{F}^{-1}$, whose action on y is given by:

$$\mathcal{D}y = \sum_{i=0}^{\infty} \psi\left(\frac{1}{\lambda_i}\right) \langle y, e_i \rangle e_i$$

with

$$\psi\left(\frac{1}{\lambda_i}\right) \approx \begin{cases} \frac{1}{\lambda_i} & if\ i\ is\ small \\ 0 & if\ i\ is\ large \end{cases}$$

Then:

$$\|x - \mathcal{D}\mathcal{F}x\|\ \ is\ \ \begin{cases} small\ \ if\ x\ is\ \ smooth \\ large\ \ if\ x\ is\ not\ \ smooth \end{cases}$$

For example, if we consider the problem 2.3, the approximation $\mathcal{D}$ is determined by truncating the series 2.4 for low values for the spatial frequency k. The composition of the two operator $\mathcal{D}\mathcal{F}$ acts like a low-pass filter, where the more is the spatial frequency, the more the coefficients are truncated. Following these guidelines, we choose an indicator function such that:

$$a_D(\mathbf{u}) = |\mathbf{u} - \mathcal{D}(\mathcal{F}(\mathbf{u}))| \tag{2.59}$$

At this point, is evident that we need a suitable approximation of $\mathcal{D}$.

2.3.7 The Van Cittert Deconvolution operator

A common choice for $\mathcal{D}$ is the Van Cittert deconvolution operator $\mathcal{D}_N$ defined as:

$$\mathcal{D}_N = \sum_{i=0}^{N} (I - \mathcal{F})^i \tag{2.60}$$

To evaluate the indicator function (2.59) we apply the filter $\mathcal{F}$ a total of N+1 times. In order to save computational time and have a trade-off between accuracy and filtering N is typically small (in this thesis we will test the model for N=0).
In order to normalize the indicator function, we divide (2.59) by a normalizing factor:

$$a_D(\mathbf{u}) = \frac{|\mathbf{u} - \mathcal{D}(\mathcal{F}(\mathbf{u}))|}{max(1, \|\mathbf{u} - \mathcal{D}(\mathcal{F}(\mathbf{u}))\|_\infty)} \tag{2.61}$$

2.3.8 The Van Cittert-Helmholtz operator

Analyzing the problem (2.3), we can select $\mathcal{F}$ as the linear Helmholtz filter operator $\mathcal{F}_H$ defined as:

$$\mathcal{F} = \mathcal{F}_H = (I - \frac{1}{\alpha}\nabla^2)^{-1} = (I - \delta^2\nabla^2)^{-1} \qquad \delta = \frac{1}{\sqrt{\alpha}} \tag{2.62}$$

Note that in eq.(2.3) α (or δ) has the task to decrease the coefficients of the series (the less is α, or the more is δ, the more the denominator will increase). Note also from eq. (2.62) that δ has a units of measurement in length. The operator $\mathcal{F}_H$ has the prominent role to regularize the solution and δ dictates how big is the distance affected by the regularizing effect of the filter.
From a practical point of view, the workflow for N=0 is structured as follows:

1. Firstly, we solve the Helmholtz problem 2.3 obtaining $\widetilde{\mathbf{u}} = \mathcal{F}(\mathbf{u})$:

$$-\delta^2\nabla^2\widetilde{\mathbf{u}} + \widetilde{\mathbf{u}} = \mathbf{u} \; in \; \Omega \tag{2.63}$$

2. Secondly, we evaluate the indicator function with (2.61)

3. Thirdly, we filter the solution (the velocity and the enthalpy) with equations (2.35) by using the indicator function evaluated at point 2.

2.4 Summary of the indicator functions

In the present thesis we will test three different types of indicator functions:

The linear indicator function: a_L

$$a(\mathbf{u}) = 1 \tag{2.64}$$

This function is extremely diffusive since it assumes that the flow need regularization everywhere. However, a proper tuning of the filtering radius δ gives us the possibility to tune the numerical dissipation.

The Non-linear Gradient-type indicator function: a_S

$$a(\mathbf{u}) = \frac{\nabla \mathbf{u}}{max(1, \|\nabla \mathbf{u}\|_\infty)} \tag{2.65}$$

As explained above, this type of indicator function regularize where there is a shear flow. The connection with the subgrid artificial viscosity given by the standard Smagorinsky model [18] gives us the possibility to find an estimate for the filtering radius.
Indeed, for the standard Smagorinsky:

$$\mu_{art}^{SM} = \rho (C_s h)^2 \sqrt{2\epsilon : \epsilon} \quad \epsilon = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla^T \mathbf{u}\right)$$

Where h is the mesh size, ϵ is the symmetric part of gradient velocity tensor and C_s is related to the constant of turbulent dissipation and production.
The connection with the Gradient indicator function is easily seen by equaling the two viscosities:

$$\mu_{art}^{GT} = \rho \frac{\delta^2}{\Delta t} a(\mathbf{u})$$

By approximating $\sqrt{2\epsilon : \epsilon} \simeq \sqrt{2}\nabla \mathbf{u}$, we obtain:

$$\delta = 2^{1/4} C_s \, h \, \sqrt{\Delta t} \tag{2.66}$$

Despite the clear connection with the standard Smagorinsky model, in our case the "global" numerical viscosity injected is not only determined by the filtering radius, but also by the splitting error that we commit between the various steps of the algorithm. However, the equation (2.66) gives us to obtain an estimate on the order of magnitude of the filtering radius. Note that the filtering radius is related to a modelling part thanks to C_s and to the mesh size and the time step, that are selected by the numerical methods.

The non-Linear Deconvolution-based indicator function: a_D

This type of function is extremely selective and the evaluation of the indicator function requires a solution of an Helmholtz problem as explained in sec.2.3.8:

1. Solve the Helmholtz problem, after having selected a radius δ
2. Evaluate the indicator function
3. Filter the equations

Note that from here, it is possible to have a general picture of the computational cost required by the three indicator functions.

- The Linear indicator function is obviously the cheaper method, since the filter step requires only the evaluation of the laplacian term.
- The Gradient-type is relatively expensive, since the indicator function requires the evaluation of the velocity gradient, that generally speaking requires a loop trough the cells of the computational mesh.
- The Deconvolution-based requires the evaluation of the indicator function 2.61 and the resolution of an Helmholtz problem, thus, it requires only one more step from the Gradient-type.

Chapter 3

Space Discretization: the Finite Volume Method

In this chapter we will explain in details the space discretization adopted in our work.

3.1 Colocated Finite Volume Method

Nowadays, one of the most widely used method for solving PDE'S in fluid dynamics is the finite volume method that consists in discretizing the domain in elementary volumes and treating the cell averages as unknowns:

$$\phi = \frac{1}{V}\int_V \widetilde{\phi}(\mathbf{x},t)dV \tag{3.1}$$

The algebraic equations are obtained by enforcing the conservation for each control volume (CV) or cell by using the Gauss theorem.
For sake of simplicity, we will explain how does it work by simply discretizing a general form of transport equation in 2D:

$$\frac{\partial \rho\Phi}{\partial t} + \nabla\cdot(\rho\mathbf{w}\Phi) = \nabla\cdot(\Gamma\nabla\Phi) + S \tag{3.2}$$

where Φ is a generic variable, Γ is the diffusion coefficient, S is a source term, $\mathbf{w}$ is the convection velocity.
By integrating over the control volume (or over the area in 2D), we obtain:

$$\int_V \left[\frac{\partial(\rho\Phi)}{\partial t} + \nabla\cdot(\rho\mathbf{w}\Phi) - \nabla\cdot(\Gamma\nabla\Phi) - S\right] dV = 0$$

By applying the Gauss Theorem, we obtain:

$$\int_V \frac{\partial(\rho\Phi)}{\partial t}dV + \int_{\partial V}\rho\Phi\mathbf{w}\cdot\mathbf{n}dA = \int_{\partial V}\Gamma\nabla\Phi\cdot\mathbf{n}dA + \int_V S$$

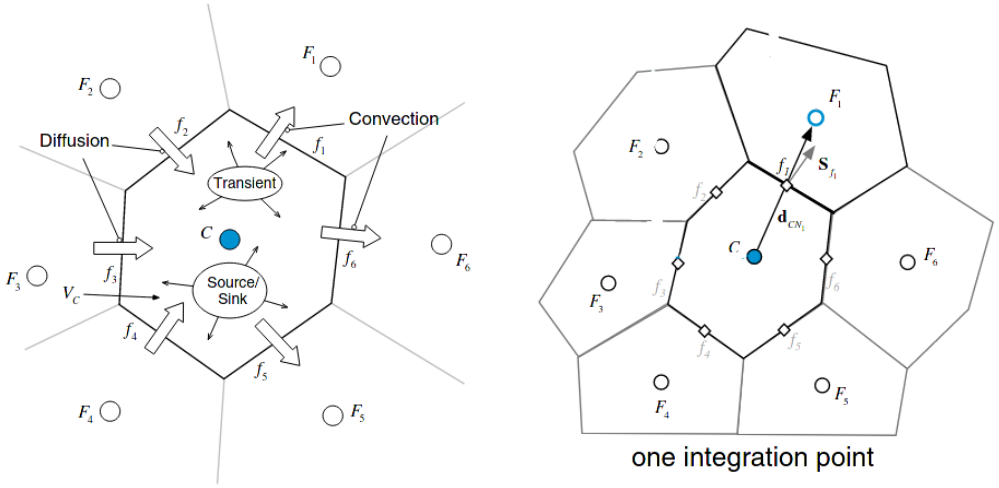

Figure 3.1: Finite volume discretization: in the left figure is shown where the terms of the equation (3.2) are evaluated, while in the right it is shown where the forces are evaluated.

Note that the convective term and the diffusion term are evaluated on the boundaries of the cell.
The surface integrals are replaced by a summation of the flux terms over the faces of the element, as reported in the right panel of fig. 3.1.

$$\int_{\partial V} \rho\Phi\mathbf{w}\cdot\mathbf{n}dA = \sum_{f\in faces}\int_f(\rho\Phi\mathbf{w}\cdot\mathbf{n})dA = \sum_{f\in faces}\Phi_f(\rho\mathbf{w})_f\cdot\mathbf{S}_f \tag{3.3}$$

$$\int_{\partial V} \Gamma\nabla\Phi\cdot\mathbf{n}dA = \sum_{f\in faces}\int_f\Gamma\nabla\Phi\cdot\mathbf{n}dA = \sum_{f\in faces}(\Gamma\nabla\Phi)_f\cdot\mathbf{S}_f \tag{3.4}$$

Note that at this point the problem of approximating the integrals along the faces arise. In the present thesis, we only do a simple mean value integration (known as trapezoidal rule), thus, as shown in fig. 3.1, we locate only one integration point at the centroid of the face: in this manner we obtain a trade-off between the computational cost and the discretization error, which is at the second order. This assumption is source of errors that are well known in the FVM framework:

- **Skewness:**

  This error is associated to the fact that the line that connect two adjacent cells does not intersect the center of the face.(see cap. 8 of the book of Moukalled et al. [8])

- **Non-orthogonality:**

  This aspect is associated with the angle between the normal of a face ($\mathbf{S}_f$) and the line that connect the two cells sharing the face. (see cap. 8 of the book of Moukalled et al. [8])

In our case, the errors due to the skewness and the non-orthogonality can be neglected because we will use perfect regular meshes.
We use a colocated grid arrangement approach, in which all the quantities are supposed to be in the geometrical centroid of a cell (e.g. the velocity and the pressure are stored in the cell's centroid). However, in the original formulation of the finite volume method, the quantities of interest are stored in a different location relative to the geometrical centroid. In order to reduce the computational cost low, we placed the quantities in the geometrical one. Indeed, for the most industrial application, the quantities are supposed to be located at the centroid of a cell and it is possible to show that the error is proportional to the square of the mesh size.

3.1.1 Details about the convection scheme

In the present thesis, we adopt an accurate third order centered-based numerical scheme for the convection term to improve accuracy.
In fig. 3.2 is shown the local stencil (with a normalized axis) for the convection scheme: the coefficients of a cubic polynomial are found by solving a linear system knowing the gradients and the values of the two cells that share the face. In the case of a perfectly uniform mesh $\lambda = \frac{1}{2}$, thus, the extrapolated value at the face is the following:

$$\Phi_f = \frac{1}{2}\Phi_P + \frac{1}{2}\Phi_N + \frac{1}{8}\Phi'_P - \frac{1}{8}\Phi'_N$$

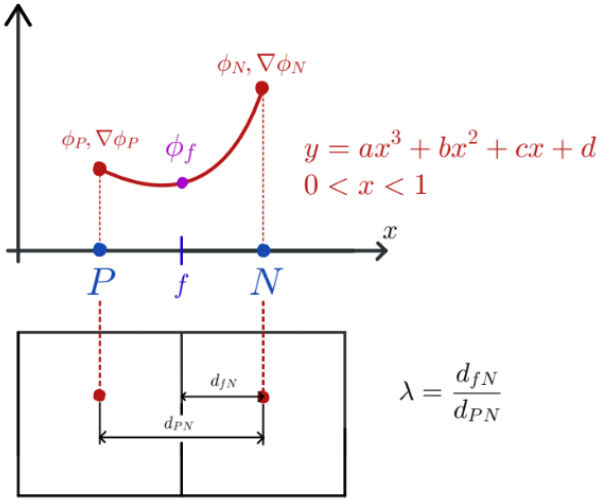

Figure 3.2: The "cubic" interpolation scheme: a third degree polynomial is fitted locally and it is extrapolated the value ϕ_f at the face

Despite being third order locally, since we are using the trapezoidal rule as quadrature formula for the integrals, our method is only second order accurate globally. However, we believe that this scheme is well suited for the LES simulations because it is a centered scheme, thus, it can give improvements in respect to an upwind biased stencil [4, 6, 29].

3.1.2 Space discretization for the Evolve-step

Having given an overview about the discretization scheme adopted in the present work, now we can explain in detail the discretization of the evolve step and in the following subsection the discretization of the filter-step.
We partition the computational domain Ω into cells or control volumes Ω_i with i=1,...N_c where N_c is the total number of cells in the mesh. Let $\mathbf{A}_j$ be the surface vector of each face of the control volume, with j=1,...,M the number of faces.
Let us start with the Evolve-step. After applying the Gauss-Divergence theorem, the integral form of eq. (1.29) for each volume Ω_i is given by:

$$\frac{1}{\Delta t}\int_{\Omega_i}\rho^* \, d\Omega + \int_{\partial\Omega_i}\rho^n\mathbf{v}^n \cdot d\mathbf{A} = \frac{1}{\Delta t}\int_{\Omega_i} b_\rho^n \, d\Omega \tag{3.5}$$

Let us denote with $(\rho^n\mathbf{v}^n)_i$ and ρ_i^* the average flux and intermediate density in control volume Ω_i, respectively. Eq.(3.5) is approximated as follows:

$$\frac{1}{\Delta t}\rho_i^* + \sum_j (\rho^n\mathbf{v}^n)_{i,j}\cdot\mathbf{A} = b_{\rho,i}^n \tag{3.6}$$

where $(\rho^n\mathbf{v}^n)_{i,j}$ denotes the flux associated to the centroid of face j divided by the volume Ω_i and $b_{\rho,i}^n$ is the average right hand side in Ω_i.
Secondly, we write the integral form of the eq.(1.30) in step 2:

$$\frac{1}{\Delta t}\int_{\Omega_i}\rho^*\Lambda^{n+1} \, d\Omega + \int_{\partial\Omega_i}\rho^n\mathbf{v}^n\Lambda^{n+1}\cdot \, d\mathbf{A} = b_h^n \tag{3.7}$$

with:

$$b_h^n = \int_{\Omega_i}\rho^*(\mathbf{v}^n\cdot g\widehat{\mathbf{k}})\, d\Omega - \frac{1}{\Delta t}\int_{\Omega_i}(\rho^*-\rho^n)k^n d\Omega + \frac{1}{\Delta t}\int_{\Omega_i}\rho^n h^n d\Omega - \int_{\partial\Omega_i}\rho^n\mathbf{v}^n k^n\cdot d\mathbf{A}$$

which figures as a source term. For sake of clarity we denote $\phi_{i,j}^n = (\rho^n\mathbf{v}^n)_{i,j}\cdot\mathbf{A}$ the flux associated to the centre of face j of the cell i.
The discretized form of eq.(3.7) is the following:

$$\frac{1}{\Delta t}\rho_i^*\Lambda_i^{n+1} + \sum_j \phi_{i,j}^n\Lambda_{i,j}^{n+1} = b_{h,i} \tag{3.8}$$

with

$$b_{h,i} = \rho_i^*(\mathbf{v}_i^n\cdot\widehat{\mathbf{k}}) - \frac{1}{\Delta t}(\rho_i^*-\rho_i^n)k_i^n + \frac{1}{\Delta t}\rho_i^n h_i^n - \sum_j \phi_{i,j}^n k_{i,j}^n$$

Denoting with the subscript i the average quantities associated with the cell "i" we can write the step 3 as follows:

$$T_i^{n+1} = \frac{\Lambda_i^{n+1}}{c_p} \quad \psi_i^{n+1} = \frac{1}{RT_i^{n+1}} \quad \widetilde{\rho_i}^{n+1} = \frac{p_i^n}{RT_i^{n+1}} \tag{3.9}$$

In the step 4, we choose to decouple the pressure field from the velocity field by using the conservation of mass to obtain a suitable approximation for the pressure fluctuation. Firstly, let us write the integral form for the momentum equation:

$$\frac{1}{\Delta t}\int_\Omega \widetilde{\rho}^{n+1}\mathbf{v}^{n+1}\,d\Omega+\int_{\partial\Omega}(\rho^n\mathbf{v}^n\otimes\mathbf{v}^{n+1})\cdot d\mathbf{A}+\int_\Omega \nabla p'^{n+1}\,d\Omega+\int_\Omega gz\nabla\widetilde{\rho}^{n+1}\,d\Omega=\int_\Omega \mathbf{b}_u^{n+1}\,d\Omega \tag{3.10}$$

Let us to write this equation in semi-discrete form, where the pressure contribution is in continuum form:

$$\frac{1}{\Delta t}\widetilde{\rho}_i^{n+1}\mathbf{v}_i^{n+1}+\sum_j \phi_{i,j}^n\mathbf{v}_{i,j}^{n+1}+\nabla p_i'^{n+1}+gz\nabla\widetilde{\rho}_i^{n+1}=\mathbf{b}_{v,i}^{n+1} \tag{3.11}$$

Following the approach in [15] we can write eq.(3.11) in semi-discretized form where the pressure term and the density term are in continuous form:

$$\mathbf{v}_i^{n+1}=\frac{\Delta t}{\widetilde{\rho}_i^{n+1}}(\mathbf{b}_{v,i}^{n+1}+\mathbf{H}(\mathbf{v}_i^n)-\nabla p_i'^{n+1}-gz\nabla\widetilde{\rho}_i^{n+1}) \tag{3.12}$$

$\mathbf{H}(\mathbf{v}_i^n)$ is the matrix obtained from the original matrix that multiply the velocity in eq. 3.12 that consists only of the off-diagonal terms applied to the velocity at the previous time step. This step is dependent from the discretization procedure and it is crucial for the resolution of the problem. Plugging this equation in the equation of mass (1.1) we obtain the equation for pressure fluctuations:

$$\frac{\widetilde{\rho}_i^{n+1}-\rho_i^*}{\Delta t}+\psi_i^{n+1}\frac{p_i'^{n+1}-p_i'^n}{\Delta t}+\nabla\cdot\left(\widetilde{\rho}_i^{n+1}\left(\frac{\Delta t}{\widetilde{\rho}_i^{n+1}}(\mathbf{b}_{v,i}^{n+1}+\mathbf{H}(\mathbf{v}_i^{n+1})-\nabla p_i'^{n+1}-gz\nabla\widetilde{\rho}_i^{n+1})\right)\right)=0 \tag{3.13}$$

Its discretization gives us the following equation:

$$\frac{\widetilde{\rho}_i^{n+1}-\rho_i^*}{\Delta t}+\psi_i^{n+1}\frac{p_i'^{n+1}-p_i'^n}{\Delta t}-\sum_j \widetilde{\rho_{i,j}}^{n+1}\frac{\Delta t}{\widetilde{\rho}_i^{n+1}}\nabla p_{i,j}'^{n+1}\cdot\mathbf{A}_{i,j}=\mathbf{b}_{p'}^{n+1} \tag{3.14}$$

where

$$\mathbf{b}_{p'}^{n+1}=\sum_j \widetilde{\rho_{i,j}}^{n+1}\frac{\Delta t}{\widetilde{\rho}_i^{n+1}}(\mathbf{b}_{v,i}^{n+1}-\mathbf{H}(\mathbf{v}_i^{n+1})+gz\nabla\widetilde{\rho_{i,j}}^{n+1})\cdot\mathbf{A}_{i,j}$$

The equation (3.14) is an elliptic equation for p' that is solved iteratively.
Then, we calculate the velocity field by eq. (3.12) and calculate the pressure with eq.(1.33) In order to check the consistency between the pressure equation and the continuity equation a check on the continuity error is performed because the density is linked with the pressure equation with the relation $\rho=\psi p$

- A density $\widetilde{\rho}$ is calculated with the quantity $\psi^{n+1}(p^{n+1}-p^n)$:
$$\widetilde{\rho}=\rho^{n+1}+\psi^{n+1}(p^{n+1}-p^n)$$
- The following check is done to see if there is a global continuity error in the whole domain:
$$err=\frac{\|\rho^{n+1}-\widetilde{\rho}\|_{L^1}}{\|\rho^{n+1}\|_{L^1}}$$

3.1.3 Space discretization for Filter-Step

In this subsection, we present the discretization of the filter problem (2.34)-(2.35). The integral form of the eq.(2.34) for each volume Ω_i is given by:

$$\int_{\Omega_i} \mathbf{u}^{n+1}\, d\Omega - \int_{\Omega_i} \nabla \cdot (\delta^2 a(\mathbf{v}^{n+1}) \nabla \mathbf{u}^{n+1})\, d\Omega = \int_{\Omega_i} \mathbf{v}^{n+1}\, d\Omega \tag{3.15}$$

By using the Gauss-divergence theorem, the above equation becomes:

$$\int_{\Omega_i} \mathbf{u}^{n+1}\, d\Omega - \int_{\Omega_i} (\delta^2 a(\mathbf{v}^{n+1}) \nabla \mathbf{u}^{n+1}) \cdot d\mathbf{A} = \int_{\Omega_i} \mathbf{v}^{n+1} d\Omega \tag{3.16}$$

With a similar discretization procedure of the evolve-step, we divide by the cell volume and obtain the following equation for the cell averages:

$$\mathbf{u}_i^{n+1} - \delta^2 \sum_j a(\mathbf{v}^{n+1})(\nabla \mathbf{u}_i^{n+1})_j \cdot \mathbf{A}_{i,j} = \mathbf{v}_i^{n+1} \tag{3.17}$$

Similarly, we obtain the discretization form for eq.(2.35):

$$h_i^{n+1} - \delta^2 \sum_j a(h^{n+1})(\nabla h_i^{n+1})_j \cdot \mathbf{A}_{i,j} = \Lambda_i^{n+1} \tag{3.18}$$

3.1.4 Hydrostatic Initialization for fluid flow

Before the discussion of the results, it is worth mentioning that the case we're dealing with is quite complicated numerically: we are simulating a flow regime which describes a combination of two factors:

- **The Mach number is low**

  This aspect has the implication that the standard density-based solvers created for high-mach numbers fails in this framework. The numerical diffusion introduced by the scheme itself is too high because most schemes are thought for supersonic-flow regimes.

- **The presence of the gravity in a quasi static-flow regime**

  Since in our case we are dealing with a flow regime with very low velocities, it is necessary that the method is well-balanced: indeed, if the terms in the discretized vertical momentum equation are not consistent, a non-physical vertical acceleration arise. In the context of colocated finite volume settings, several strategies are developed to handle this problem, see [26],[2],[19] for further reference. Since we are using a pressure-based solver for the evolve step, we follow another strategy different to those mentioned previously.

Before doing the simulation, we must initialize the dynamic component of the pressure in such a way that the pressure gradient balance perfectly the gravity force at discrete level:

$$\nabla p = \rho \mathbf{g} \widehat{k} \tag{3.19}$$

And in terms of dynamic pressure:

$$\nabla p' = \nabla \rho (\mathbf{g} \widehat{k} \cdot \vec{x}) \tag{3.20}$$

With this trick, at the start of the time evolution the equation of momentum is perfectly balanced (numerically speaking) because we are binding the discretized pressure to the discretized gravity term that is affected by the mesh. Taking the divergence of the equation (3.20) we are able to obtain the equation for the hydrostatic initialization:

$$\nabla \cdot (\nabla p') = \nabla \cdot (\nabla \rho (\mathbf{g} \widehat{k} \cdot \vec{x})) \tag{3.21}$$

After solving eq.(3.21) we calculate the total pressure:

$$p = p' + \rho \mathbf{g} \widehat{k} \cdot \vec{x}$$

And after this step we calculate again the density by the equation of state:

$$\rho = \frac{p}{RT} \tag{3.22}$$

The procedure is repeated for several times (until the variation of the pressure between two successive steps is negligible)
Even tough this procedure could seem superfluous, it is extremely fundamental for the success of the simulation: this is justified by the fact that in the first time steps the left hand side of the vertical momentum equation must be balanced by the right hand side at a discrete level (if we are in a steady-state). Other techniques rely on a local reconstruction of the pressure and the density after having selected a background state in every cell of the mesh [26, 2, 19].

Chapter 4

Numerical Experiments and Results

In this chapter we will show the results of our methodology applied to the most well known benchmarks in the atmospherical community. We will assess the well-balanced property of the scheme with a flat terrain and then we will test the stabilization properties of the EFR algorithm on the rising thermal bubble and the density current benchmark.

4.1 Hydrostatic Atmosphere

The goal of this first test case is to verify that an initial resting atmosphere over a flat terrain remains still for a long time interval. This test will test the "well-balanced" property of the scheme because, as reported in section 3.1.4 if care is not taken during the discretization of the momentum equation, spurious vertical velocity could arise during the process.
We will use the same domain setting as in the Straka-test [31]. The computational domain x-z is $\Omega = [0,25600]\text{x}[0,6400]m^2$. Free slip boundary conditions are considered at all the boundaries. We consider a uniform mesh size of $\Delta x = \Delta z = h = 200m$ and we set a timestep of $\Delta t = 0.1s$. A uniform background potential temperature of $\theta = 300K$ is set in the whole domain, which implies a stable stratification. Figure 4.1 shows the time evolution of the maximum vertical velocity picked in the whole domain for three days of simulation: the method stabilizes at $1e-5$ m/s, which suggests that it is well suited for the evolution of perturbations, such as in the density current and in the rising thermal bubble test.

4.2 The Rising-thermal bubble benchmark

The rising thermal bubble test shows the evolution of a warm bubble in a constant potential temperature ($\theta_0 = 300K$) environment. Because the bubble is warmer than the ambient air, it rises while deforming as a consequence of the shearing motion caused by

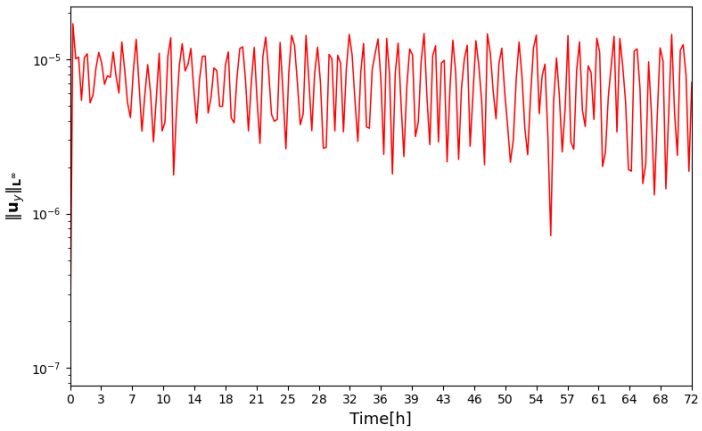

Figure 4.1: Hydrostatic atmosphere: time evolution of the maximum vertical velocity

the velocity field gradients until it forms a mushroom cloud. To drive the motion of the air, the following potential temperature perturbation is then added:

$$\theta' = 300 + 2(1 - \frac{r}{r_0}) \ \ if \ \ r \leq r_0 \ = 2000m \ \ \theta_0 \ = 300 \ otherwise \tag{4.1}$$

where r = $\sqrt{(x - x_c)^2 + (z - z_c)^2}$,$(x_c, z_c)$=(5000,2000)m is the radius of the circular perturbation.
In our case, in order to save precious computational time, we've decided to simulate only an half of the domain: this is justified since the resulting bubble is symmetric respect to the y axis (see fig. 4.2 for clarity). The interval time taken in consideration is (0,1020]s. In this case, as will be done in the density current test, it is necessary to add a small quantity of numerical viscosity in order to stabilize the solution. In the common literature a quantity of $\nu_{num} = 15m^2/s$ of kinematic viscosity is added in this test case, and the resulting model is named "AV-15" which stands for "Artificial-Viscosity-15", see [25],[27] for further references.

4.2.1 Linear-type Results

In order to compare our model with the standard AV15 model, we set the filtering radius to $\delta = 1.225m$, which gives an amount of diffusion of:

$$\nu_{num} = \frac{\delta^2 a(\mathbf{U})}{\Delta t} = \frac{1.225^2 m^2}{0.1s} = 15[\frac{m^2}{s}]$$

The mesh sizes selected for this benchmark are h = $\Delta x = \Delta z$ = [15.625,31.25,62.5,125]m, while the timestep is set to $\Delta t = 0.1s$ in order to respect the CFL condition. A Third-order accurate scheme for the convective-term and central difference scheme is employed for the laplacian term. The fig. 4.3 shows the temperature perturbation evolution for the finest mesh at t=340,680,1020s.

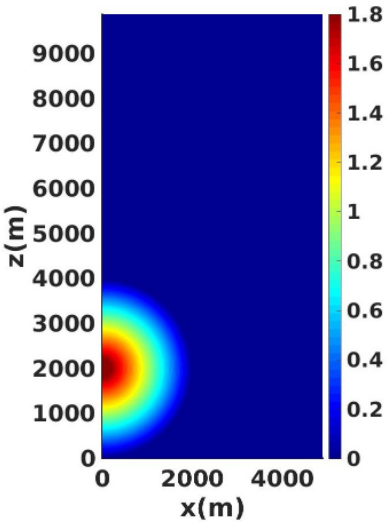

Figure 4.2: Initial setting for the thermal bubble: the temperature decreases linearly

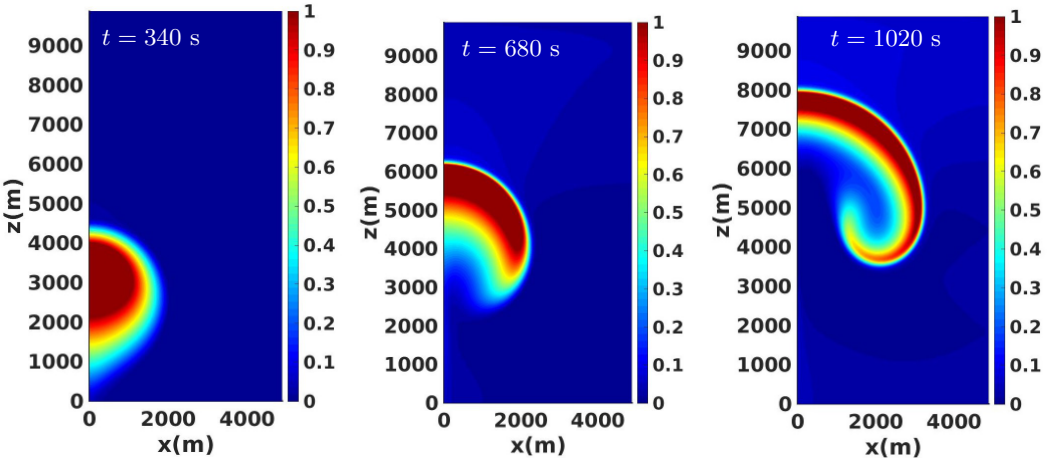

Figure 4.3: Rising thermal bubble, Linear model (a_L), $h = 31.25m$, $\delta = 1.225m$, temperature perturbation θ' computed at $t = 340,680,1020s$.

As the bubble is warmer than the surrounding environment, it rises and evolves into a mushroom-like shape due to its buoyancy.

Fig. 4.4 compares the results between the mesh sizes: qualitatively speaking, there are no major visible differences at the first appearance, even tough a careful analysis for the coarser mesh reveals some oscillation in the center of the bubble.

Despite the fact that, qualitatively, it seems that $\delta = 1.225m$ is sufficient to stabilize the solution, analyzing the maximum temperature perturbation and the maximum vertical velocity emerge the presence of large oscillations for the coarser mesh, furthermore, with this radius, the mesh size $h = 62.5m$ does not converge at the other curves for the other

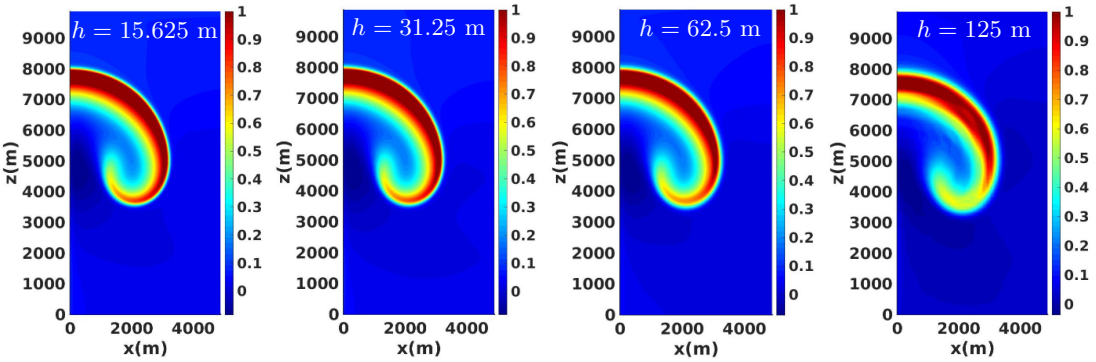

Figure 4.4: Rising thermal bubble, Linear model (a_L), $\delta = 1.225$ m: perturbation of potential temperature at $t = 1020$ s computed with four different meshes.

meshes.

Note also in the right panel of fig. 4.5 that for $t \simeq 840s$ there is an unphysical behaviour for the maximum vertical velocity. The post-process reveal that this numerical artefact is due to the small filtering radius, which is not sufficient to stabilize oscillations near the left boundary. This is the reason why we have augmented slightly the filtering radius up to $\delta = 1.9m$. With this value, no qualitative differences are observed in the solution: compare fig. 4.4 with fig. 4.6. Quantitatively speaking, this new value allows us to reach the convergence for the finest meshes, since the curves in the graph 4.7 are almost overlapped. Moreover, we have eliminated the numerical artefact in the maximum vertical velocity as revealed in the right panel of fig. 4.7.

As expected, the maximum temperature perturbation at t=1020s diminishes by increasing the filtering radius because we are putting more viscosity in the domain (compare fig. 4.5 and fig. 4.7).

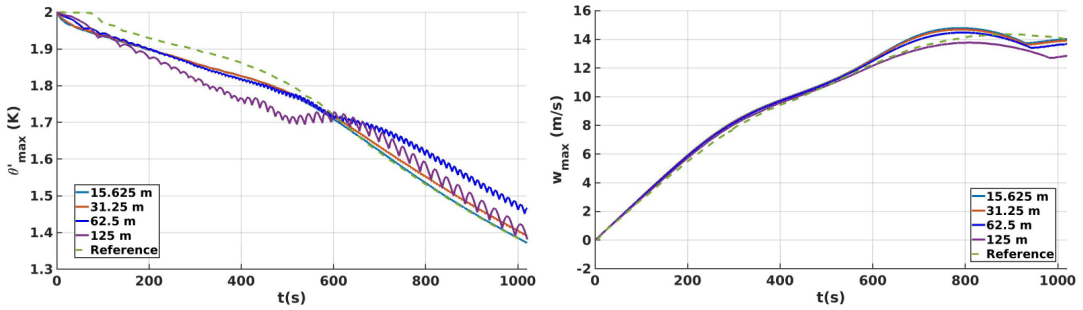

Figure 4.5: Rising thermal bubble, Linear model (a_L), $\delta = 1.225$: time evolution of the maximum perturbation of potential temperature θ'_{max} (left) and the maximum vertical component of the velocity w_{max} (right) computed with all the meshes under consideration. The reference values are taken from [25] and refer to a resolution h=125 m.

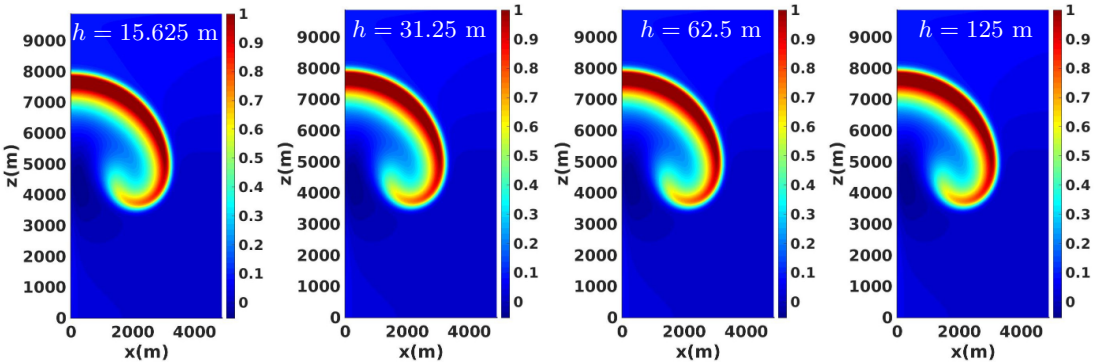

Figure 4.6: Rising thermal bubble, Linear model (a_L), $\delta = 1.9$ m: perturbation of potential temperature at $t = 1020$ s computed with four different meshes.

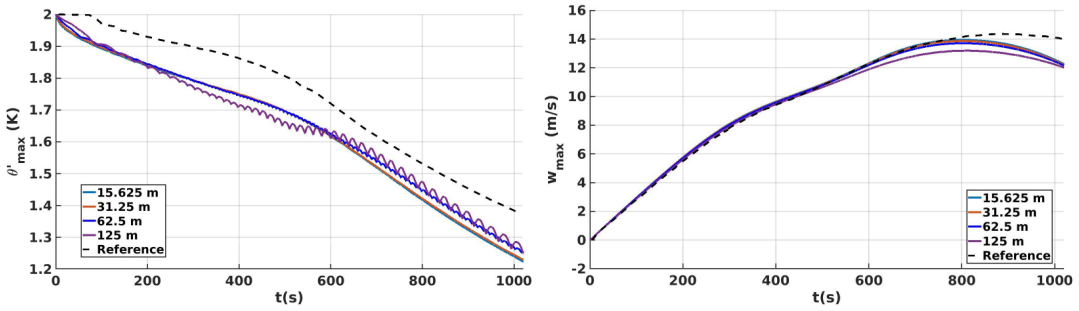

Figure 4.7: Rising thermal bubble, Linear model (a_L), $\delta = 1.9m$: time evolution of the maximum perturbation of potential temperature θ'_{max} (left) and the maximum vertical component of the velocity w_{max} (right) computed with all the meshes under consideration. The reference values are taken from [25] and refer to resolution 125 m.

4.2.2 Gradient-type results

Analyzing the results for the Gradient type, we obtain an order of magnitude for the filtering radius based on the Smagorinsky constant adopted in [23]: by taking $C_s \simeq 0.094$, we obtain $\delta \simeq 2^{1/4} 0.094 h\sqrt{\Delta t}$ as pointed out in the theory previously mentioned. In our case, we set $\delta = 2.5m$ for $h = 15.625m$, $\delta = 5m$ for $h = 31.625m$ and $\delta = 10m$ for $h = 62.5m$. Smaller filtering radius give rise to spurious oscillations in the domain.
Fig. 4.8 shows the results for all the meshes cited above. Note that $\delta = 10m$ gives a too high level of dissipation into the domain because the temperature perturbation is, on average, lower than in the other cases. The high level of dissipation is indeed caused by the fact that we decided to maintain the ratio between the filtering radius constant when increasing the mesh sizes instead of directly taking the filtering radius given by 2.66, which, in the case of $h = 62.5m$ gives $\delta \simeq 2$. As this estimate suggests, the filtering radius has to be of the order of some units, but the exact value of the radius is not known a priori. Observing carefully the top left pic of fig. 4.8, some oscillations rise below the bubble for the finest mesh: this artifact is probably due to the oscillations obtained for the indicator

Type	h (m)	w_{min} (m/s)	w_{max} (m/s)	θ'_{min} (K)	θ'_{max} (K)
[25]	125	-7.75	13.95	-0.013	1.4
$\delta = 1.9$	125	-10.35	12.01	-0.012	1.23
$\delta = 1.9$	62.5	-10.54	12.16	-0.041	1.24
$\delta = 1.9$	31.25	-10.61	12.21	-0.050	1.22
$\delta = 1.9$	15.625	-10.63	12.28	-0.052	1.22
$\delta = 1.225$	125	-10.56	12.70	-0.030	1.36
$\delta = 1.225$	62.5	-10.85	13.65	-0.070	1.43
$\delta = 1.225$	31.25	-10.97	13.90	-0.080	1.38
$\delta = 1.225$	15.625	-11.01	14.01	-0.085	1.37

Table 4.1: Rising thermal bubble, Linear model (a_L), minimum and maximum vertical velocity w and potential temperature θ' at $t = 1020$ s compared with the values extracted from the figures in [25].

function itself, as shown in the second row. Probably, the first order boundary conditions on the left boundary are responsible to create these oscillations during the first part of the time evolution. As the time proceeds, the small value of the filtering radius is not sufficiently high to delete these numerical artifacts that are convected in the domain.

4.2.3 Deconvolution-based results

For the Deconvolution-based filter, we set the filtering radius as the same order of magnitude of the Gradient-type. We augment slightly the filtering radius for h=15.625m since smaller radius gives too oscillations in the domain. It is worth mentioning that we are allowed to change the ratio between the filtering radius for different mesh sizes, since, at the best of our knowledge, find a formula related to the standard LES models is still an open problem. The fig. 4.9 reveals that this type of filter is more selective respect to the others: compare the results for the two mesh finer meshes. With $h = 15.625m$, we observe more structures respect the Gradient-type: compare fig. 4.9 with fig. 4.8. With $h = 31.25m, \delta = 5m$, we observe that the Gradient-type is too diffusive, as confirmed by the indicator function (compare fig. 4.8, and 4.9, bottom row). This happen because the indicator function for $h = 15.625m$ is greater respect $h = 31.25m$ as shown by the plot of the indicator functions (the colorbar range is the same for the same mesh-size between the two non-linear models). As cited above, this behaviour is due to the fact that the spatial gradients increase as mesh size decreases. Table 4.2 shows the maximum temperature perturbation and the maximum vertical velocity for the Gradient-type and Deconvolution-based. As expected, the latter reaches higher values (in magnitude) for the maximum and minimum velocity. While the maximum temperature perturbation is, on average, higher for $h = 15.625m$ and lower for $h = 62.5m$ because, in the latter case, the discretization introduces more dissipation. However, an unexpected value for the maximum temperature perturbation is observed for $h = 31.25m$: θ'_{max} obtained with the Deconvolution-based is less respect to the Gradient-type. This aspect is probably a

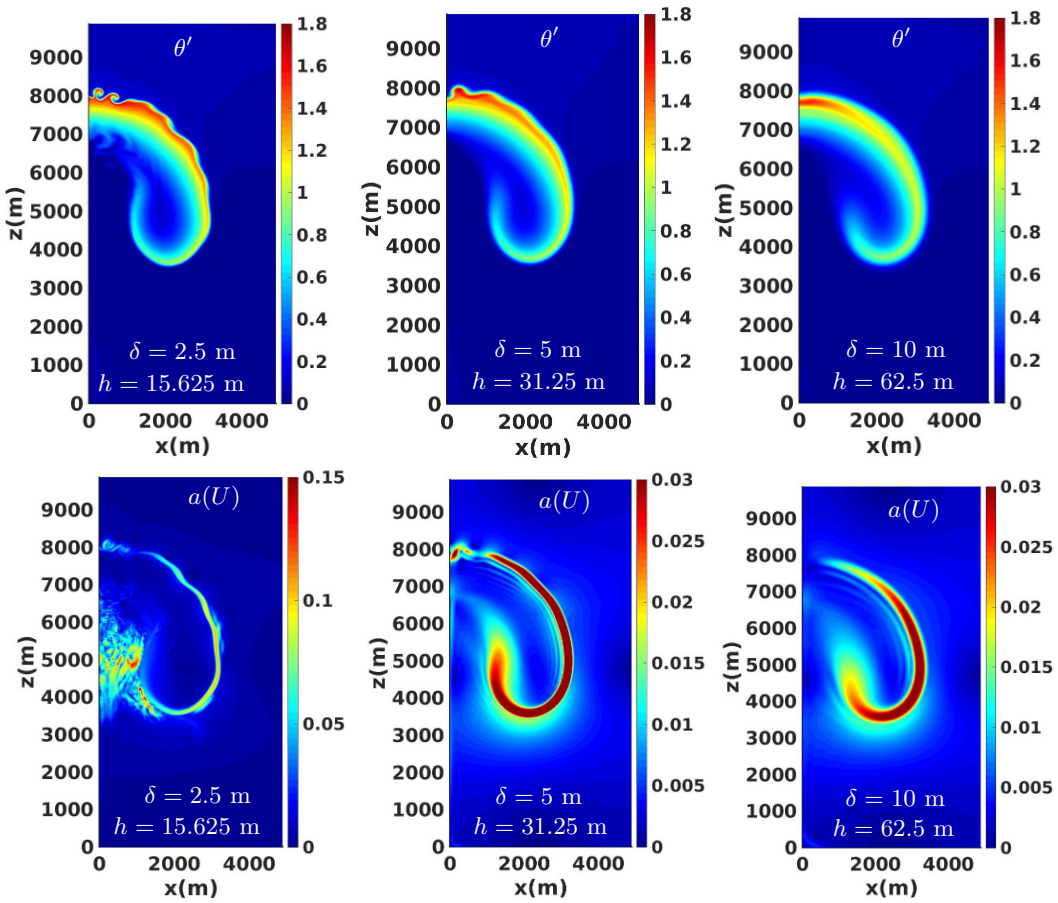

Figure 4.8: Rising thermal bubble, Gradient-type (a_S): perturbation of potential temperature (first row) computed for three different mesh size, bottom row: Indicator function at t=1020s

numerical artifact due to the extreme selectiveness of the filter.

Model	h (m)	δ (m)	w_{min} (m/s)	w_{max} (m/s)	θ'_{min} (K)	θ'_{max} (K)
EFR, a_S	15.625	2.5	-11.88	15.78	-0.39	1.77
EFR, a_D	15.625	3	-13.99	16.00	-0.36	1.82
EFR, a_D	31.25	5	-12.62	15.14	-0.141	1.72
EFR, a_S	31.25	5	-11.13	14.81	-0.22	1.78
EFR, a_S	62.5	10	-10.76	13.38	-0.45	1.55
EFR, a_D	62.5	10	-10.73	13.86	-0.15	1.62

Table 4.2: Rising thermal bubble: minimum and maximum vertical velocity w and potential temperature θ' at $t = 1020$ s computed with different mesh size for the two non linear indicator functions

We conclude the section by studying the effect of the filtering radius.
Fig. 4.10 shows the potential temperature perturbation at t=1020s computed by the Gradient type filter for different values of the radius δ. Despite the same order of magnitude, small variations in the radius imply large variations in the solution.

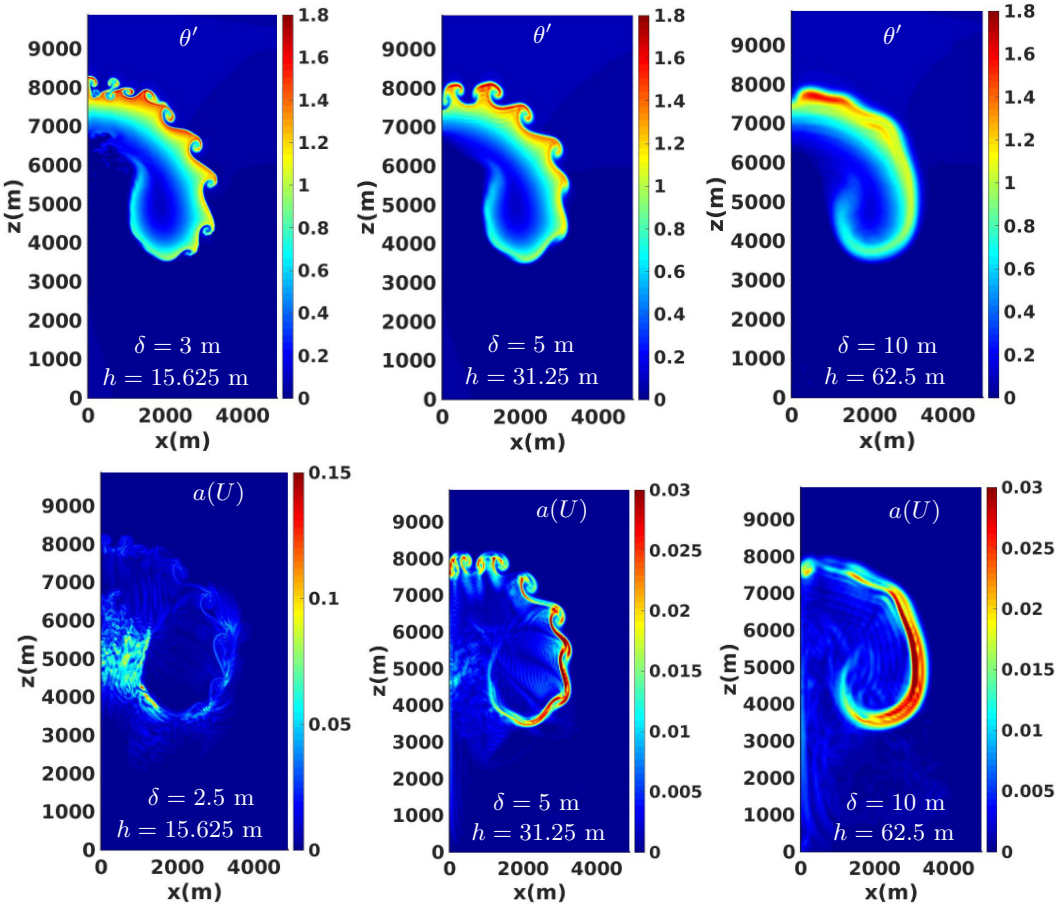

Figure 4.9: Rising thermal bubble, Decovolution-based (a_D): perturbation of potential temperature (first row) computed for three different mesh size , bottom row: Indicator function at t=1020s

4.3 The density current benchmark

The density current test was proposed by [31] and concerns the evolution of a cold bubble dropped in a neutrally stratified atmosphere. Because the bubble is cold, it sinks, eventually hitting the ground. At this point, the bubble begins to shear as it travels along the ground forming Kelvin–Helmholtz rotors. As discussed in [31], viscosity is required in order to obtain a grid-converged solution. The initial conditions for this case are quite similar to those of the rising thermal bubble, however, the differences are in the domain

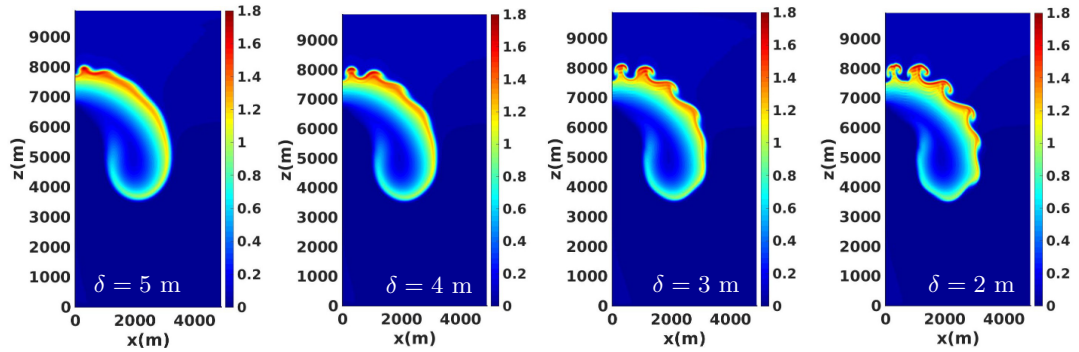

Figure 4.10: Rising thermal bubble, Gradient-type (a_S): perturbation of potential temperature at $t = 1020$ s computed by the EFR algorithm with mesh $h = 31.25$ m and (from left to right), δ = 5,4,3,2 m.

size, the shape of the cold cosine bubble (elliptic shape in this case). Moreover, the bubble is cold compared to the environment that surrounds it.
Straka noted that the rotors formed by the Kelvin-Helmoltz instability were created every 300s and for this reason we show the results for t=[300,600,900]s respectively. It is worth highlighting that in the same article, every numerical method was tested using a constant numerical viscosity of 75(Pa s) while in this work we inject a viscosity that depends on the behaviour of the flow. For this reason, as will be shown later, even more vortexes could be created during the evolution.
The initial temperature field is set by:

$$\theta^0 = 300 - \frac{15}{2}\left(1 + \cos(\pi r)\right) \;\; if \;\; r \leq 1 \tag{4.2}$$

where $r = \sqrt{(\frac{x-x_c}{x_r})^2 + (\frac{z-z_c}{z_r})^2}$, with $(x_r, z_r) = (4000,2000)$ and $(x_c, z_c) = (0,3000)$.

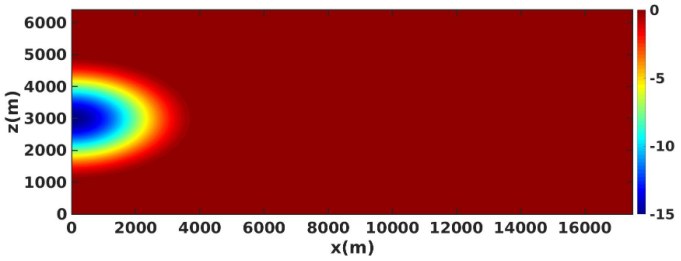

Figure 4.11: Initial temperature perturbation for the density current test

The mesh employed in this test case are all uniform, fully orthogonal in order to avoid non-orthogonality and skewness errors. The mesh sizes selected are respectively $h = \Delta x = \Delta z$ = [100,50,25] m. The time step is set to $\Delta t = 0.1$ s in order to respect the CFL condition. As in the rising thermal bubble, a third-order accurate scheme for the convective term and central difference scheme is employed for the laplacian term. If we don't add a sufficient

artificial viscosity, this test is unstable. This is confirmed by the fig. 4.12: the bubble is completely destroyed at the final time step.

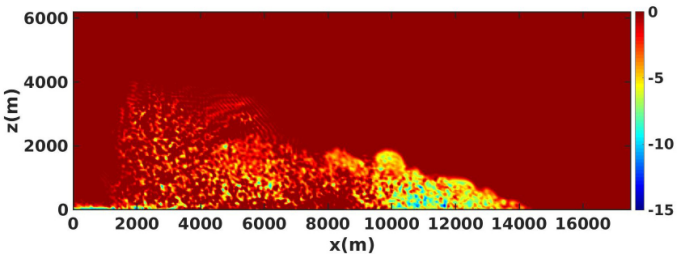

Figure 4.12: Final temperature perturbation at t=750s: the temperature perturbation is nonphysical

Firstly, we will present the results by setting $\xi = \chi = 1$, and then we will show the results obtained by setting the relaxation parameters by the estimate reported in section 2.3.4.

4.3.1 Linear-type results

Regarding the linear type, we set the filtering radius to $\delta = 2.7$, since with this radius we inject a numerical viscosity into the domain which is exactly the same used in [31] to stabilize the solution for the AV75 model.

$$\nu_{num} = \frac{\delta^2}{\Delta t} \simeq 75 m^2/\text{s} \tag{4.3}$$

Note that the classical theory of the Evolve-filter-relax approach suggests taking a filtering radius of the same order as the mesh size [34, 7], but at the same time, this assumption constraints us to set a value for the relaxation parameter which is not trivial to determine. Indeed, as revealed in fig. 4.13 this assumption provides a too high level of dissipation (with no relaxation), making the solution too smooth.

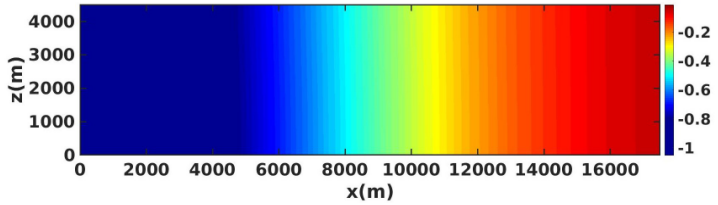

Figure 4.13: Final temperature perturbation at t=900s for the linear model with $\delta = h = 50m$: the solution is overdiffusive

A qualitative description of the flow evolution for the mesh size $h = 25m$ is given in fig. 4.14.
The negative buoyancy forces drag down the cold air bubble, then, when the ground is touched, the bubble begins to shear thanks to the fact that it is moving along the ground

generating Kelvin-Helmholtz rotors. Qualitatively speaking, we see a good agreement with the results in literature [31], [30], [27]. Note from fig. 4.15 that with this filtering radius, the coarser mesh does not provide a solution comparable with the other meshes. For a quantitative comparison of the results, we consider the position of the front location, which is defined as the position on the ground where the potential temperature perturbation is -1K. The table 4.3 shows that for all the meshes the location is close, indicating that $\delta = 2.74$ gives a good quantity of viscosity that allows us to reach the convergence.
For further quantitative comparison, we consider the profile of temperature perturbation at $z = 1200m$ in the left panel of fig. 4.16: note that both meshes gives a profile which is almost overlapped, indicating, as expected, convergence. However, in our case, the curves are translated slightly to the right, indicating the fact that the bubble convects faster respect to the reference, while the magnitude of the peaks and the global behaviour are almost the same.
In order to investigate how the filtering radius affects the curves, in the right panel of fig. 4.16 are shown the curves obtained by adopting a filtering radius $\delta = 3.5m$. The biggest negative peak, corresponding to the center of the biggest rotor shifts slightly to the right and also the amplitude of the other peaks diminish because of the dissipation in the domain. From the table 4.3 we note that with an high radius the front locations for both meshes are overlapped, confirming that this radius is sufficient to reach the convergence. These numerical experiments suggest that the "shifting" we have obtained in respect to the reference is not due to the dissipation, but probably to the pressure-velocity coupling and to the splitting error in the EFR algorithm.

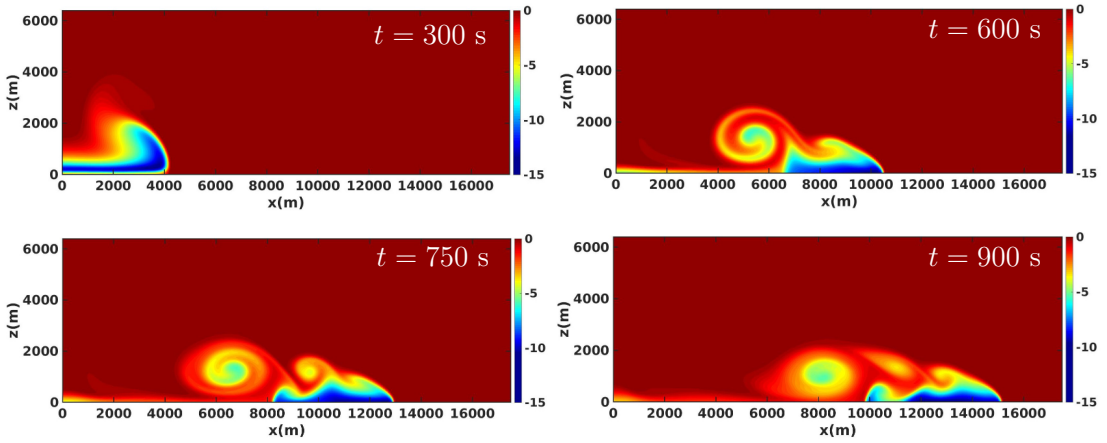

Figure 4.14: Density current, linear model: time evolution of potential temperature fluctuation θ' computed with mesh h=25m and filtering radius $\delta = 2.74m$

Figure 4.17 shows the horizontal and the vertical velocity obtained by the finest mesh with $\delta = 2.74m$. With a minimum of $U_x \simeq -14.55$ and a maximum of $U_x \simeq 37.34$ for the horizontal velocity and with a minimum for $U_y \simeq -15.25$ and a maximum of $U_y \simeq 12.16$ for the vertical velocity, the results matches quantitatively and qualitatively with [27].

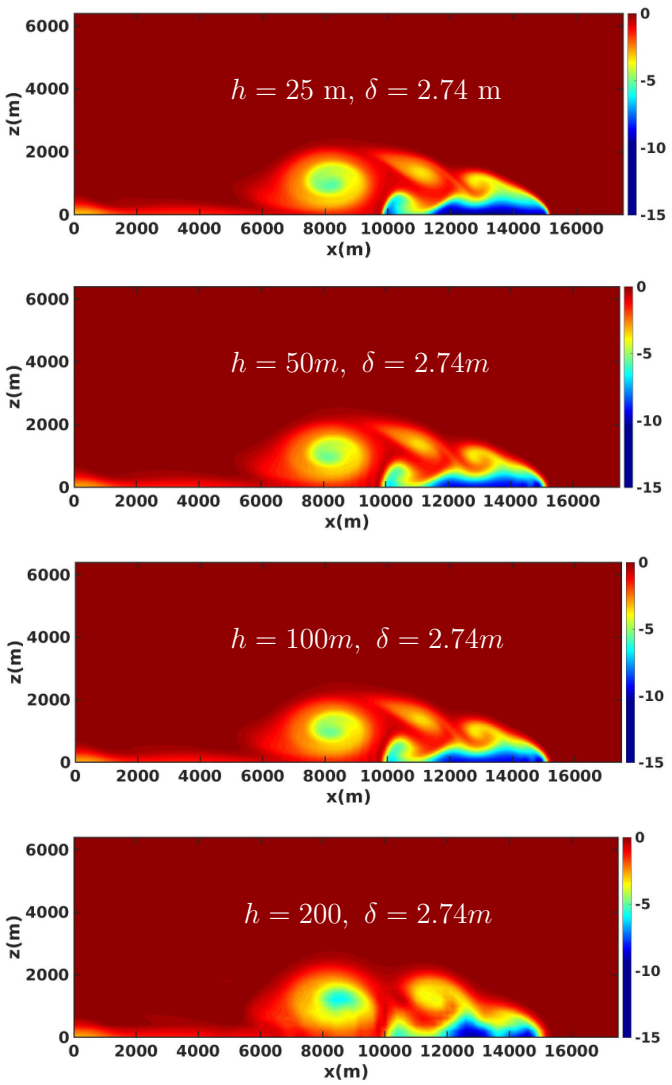

Figure 4.15: Density-current, Linear Model (a_L): Potential temperature fluctuation θ' at $t = 900$ s for meshes h= [25,50,100,200] m.

4.3.2 Gradient-type result

Switching to the second type of filter which resembles the Smagorinky model, we focus our attention on three different mesh sizes $h = 12.5{,}25{,}50m$ respectively. Adopting the Smagorinsky constant $C_s = 0.454$ used in [23], we obtain, as for the thermal-bubble test, an estimate for the filtering radius $\delta \simeq 2^{1/4}\, 0.454\, h\sqrt{\Delta t}$. Starting from the finest mesh $h = 12.5m$, we set $\delta = 4m$ because smaller values does not stabilize the solution, consequently, we set $\delta = 8{,}16$ for $h = 25{,}50m$.

$h(m)$	$\delta(m)$	front location (m)
25	2.74	15170
50	2.74	15190
100	2.74	15210
200	2.74	15220
25	3.5	14900
50	3.5	14910
100	3.5	14918

Table 4.3: Position of the front location for the Linear model (a_L) with $\delta = 2.74m$ and $\delta = 3.5m$

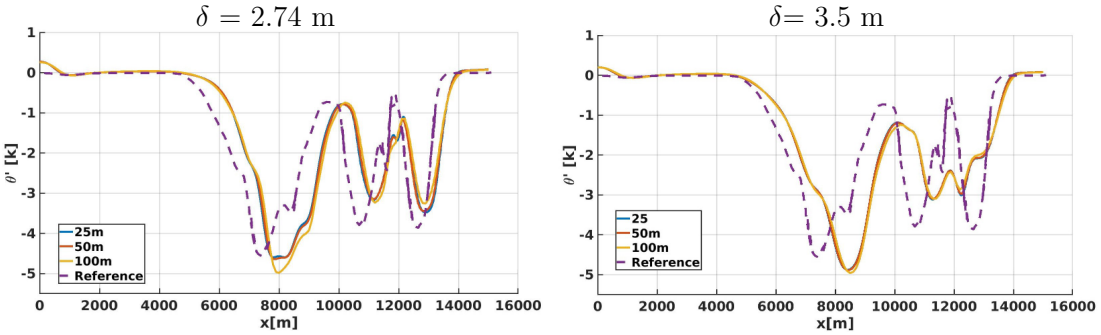

Figure 4.16: Comparison of temperature perturbation at z=1200 for two different values of the filtering radius with ref. [25]

Fig. 4.18 shows the evolution of temperature perturbation for the finest mesh. As expected, after a transition time of about $t \simeq 300s$, the bubble touches the ground and start to travel to the right. Meanwhile, the kelvin-helmholtz generates complex vortical structures. Analyzing the results for $h = 25m$ in fig. 4.19, we observe a different temperature distribution compared to the finest mesh. This is due to the different filtration radius and mesh size. Qualitatively speaking, the case $h = 25m$ matches pretty well with the results given by the standard Smagorinsky-model [30], [23], while the dynamics shown for h=12.5m is different. On the other hand, the figure 4.20 confirms that the adoption of a linear trend for the radius is not optimal: $\delta = 16m$ smoothen the solution in this case. To avoid overdiffusion, a proper tuning of the filtering must be accomplished by hand, allowing us to use a coarser mesh without compromising the accuracy, as will be seen. As shown in fig. 4.21, lowering the filtering radius allows us to recover the typical three-rotor structure of this test. The table 4.4 confirms that the front location obtained with this type of filter matches pretty well with the results obtained with high order methods [30], note also that increasing the filtering radius makes the front slightly slower (compare the fourth and the fifth row of the table).

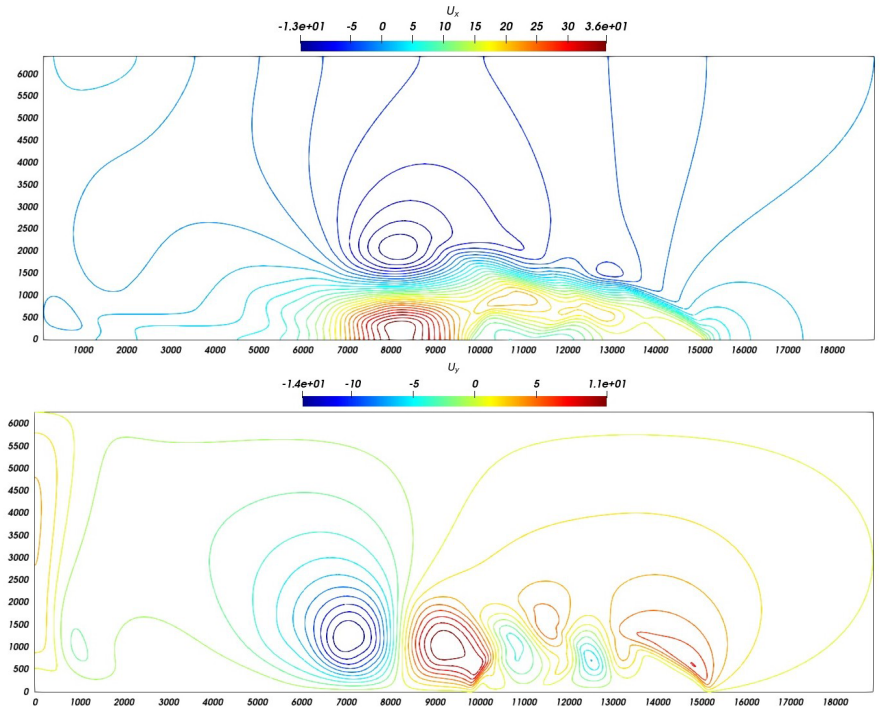

Figure 4.17: Horizontal velocity (top panel) and vertical velocity (bottom panel) at t=900s obtained with the Linear model (a_L), $\delta = 2.74m$

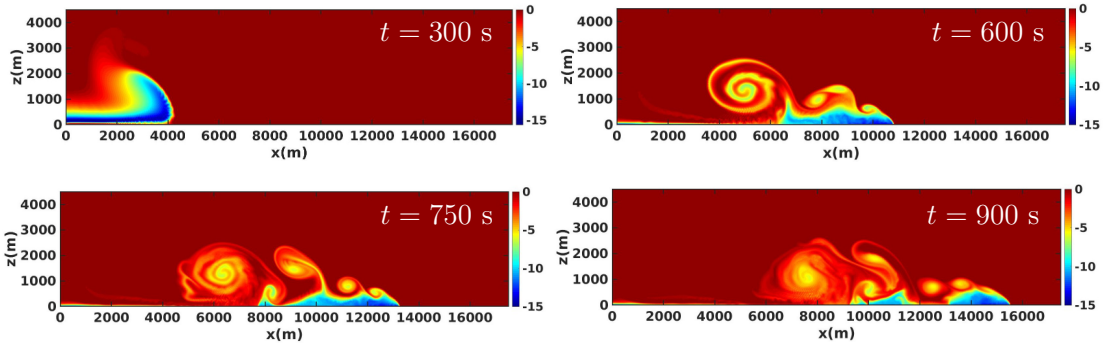

Figure 4.18: Density current, Gradient-type (a_S), $\delta = 4m$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 12.5$ m.

4.3.3 Deconvolution-based result

As shown in 4.2, the Deconvolution based is more selective compared to the Gradient-type, thus , we slightly increase the filtering radius δ for the gradient-type since at the moment we do not have practical criteria to set the filtering radius δ. We set $\delta = 5m$ for $h = 12.5m$, $\delta = 10m$ for $h = 25m$ and $\delta = 12m$ for $h = 50m$. Before showing the results obtained in this case, we compare the time evolution of the space-averaged artificial viscosity for the

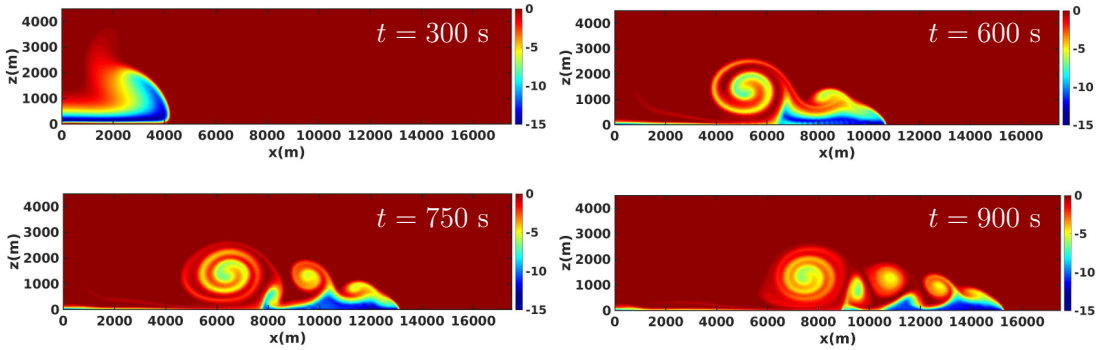

Figure 4.19: Density current, Gradient-type (a_S), $\delta = 8$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 25$ m.

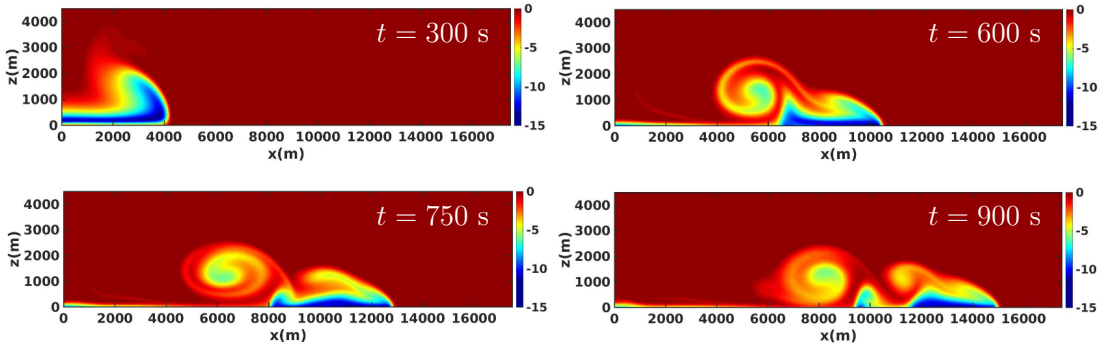

Figure 4.20: Density current, Gradient-type (a_S), $\delta = 16m$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m.

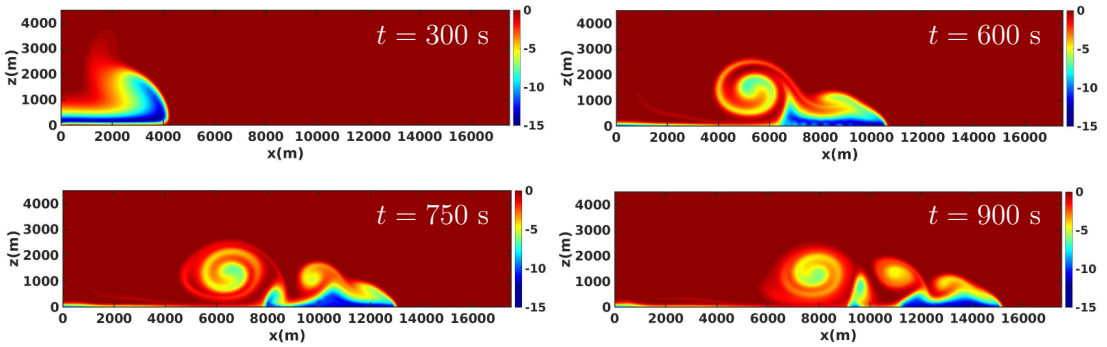

Figure 4.21: Density current, Gradient-type (a_S), $\delta = 11m$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m.

two non linear filters:

$$\mu_{av} = \frac{1}{\Omega}\int_{\Omega} \overline{\mu}_h d\Omega \tag{4.4}$$

Method	h (m)	δ (m)	Front Location (m)
Ref. [30]	12.5	-	15056
EFR, Gradient-type	12.5	4	15550
EFR, Gradient-type	25	8	15300
EFR, Gradient-type	50	11	15220
EFR, Gradient-type	50	16	15090
Ref. [31]	(25, 200)	-	(14533,17070)

Table 4.4: Density current, Gradient-type (a_S): front location at $t = 900$ s obtained with the EFR algorithm and different meshes. Our results are compared against results from [30, 31]. For reference [30], we report only the front location computed with the finest resolution. For reference [31], we provide the range of mesh sizes and front location values obtained with different methods.

The left panel of fig. 4.22 shows the trend for the Gradient filter: observe that, for all the meshes taken in consideration from $t \approx 300s$, the viscosity start to reach a "plateau". Physically speaking, from this time instant the flows begins to be convective-dominant and the relative high level of viscosity created by the filters is needed to stabilize the rotors generated during the time evolution. As expected, $\delta = 16m$ introduces too much artificial diffusion and for this reason we slightly decrease the value to $\delta \approx 11m$ which introduces a similar quantity of diffusion of $\delta \approx 8m$ with $h = 25m$.

Comparing the viscosity obtained by the Deconvolution-based, we see that quantitative speaking, the two filters matches pretty well. However, comparing carefully the time evolution for the finest mesh ($h = 12.5m$) we see that the Deconvolution-based starts to inject diffusion from $t \approx 100s$: this interesting result confirms the major selectiveness of this filter, since it start to stabilize only when the shear caused by the deformation of the bubble is sufficiently high.

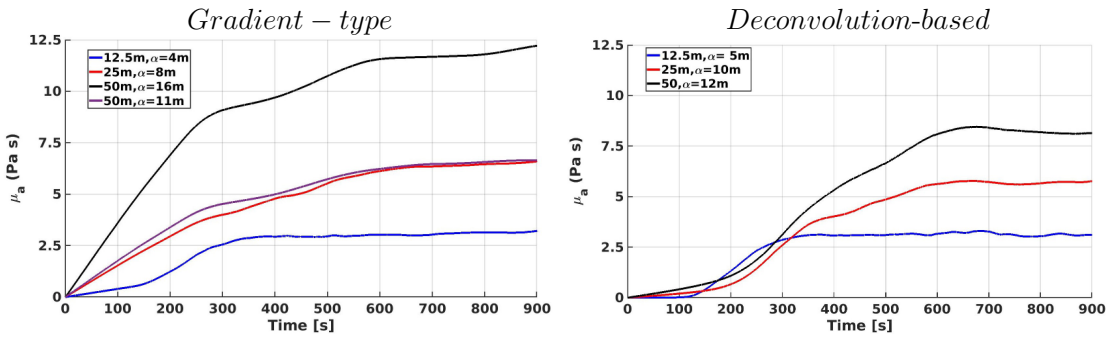

Figure 4.22: Density current: time evolution of the average eddy viscosity (4.4) for the Gradient-type (a_S) (left) and the Deconvolution-based (a_D) (right) with meshes $h = 12.5, 25, 50$ m. We have indicated with α the filtering radius δ

Figure 4.23 shows the results obtained by the finest mesh for the Deconvolution-based type. Even tough the solution is initially comparable with the other non-linear filter, the solution at $t = 900s$ is qualitative different: compare fig. 4.23 and fig. 4.18. The similar behaviour of the space averaged viscosity of fig. 4.22 for h=25m is reflected quantitatively in the dynamics of the potential temperature perturbation: the panels in fig. 4.19 and fig. 4.24 are almost similar. Surprisingly, despite the fig. 4.22 reveals that for h=50m the Deconvolution based with $\delta = 12m$ is (in average) more dissipative than the gradient type with $\delta = 11m$, the pictures in fig. 4.20 and 4.25 agrees qualitatively speaking.
Table 4.5 reports the position of the front location at t=900s obtained with the Deconvolution-based for the three meshes stated above. Also in this case, we see a good agreement with the results obtained by [30] and [31] since the values differs only by $\approx 400m$. Comparing table 4.5 with 4.4, we see clearly that not only the two filters shows similar values but also that the front becomes faster as the mesh is refined for both the two non-linear models. In contrast, for the linear model 4.1 with $\delta = 2.74m$, the trend is opposite because the more the mesh is refined, the more the front slow down.
Next, in order to have a comparison of the action of the two different non-linear models, in fig. 4.26 we compare the time evolution of the indicator functions for the simulations 4.18 and 4.23. It is clear that both the models present the largest values of the indicator function at the bottom of the largest rotor. However, a deep inspection of the panels reveals that the gradient-type has larger regions of intermediate values. Indeed, this is confirmed by comparing the results for h=50m in fig. 4.27 : the right column shows that the deconvolution type is able to stabilize "localized" regions while the gradient-type performs an "average". Notice that the finest mesh $h = 12.5m$ gives higher peaks respect $h = 50m$ for both non-linear models. This behaviour is found also by [3] in the incompressible framework.

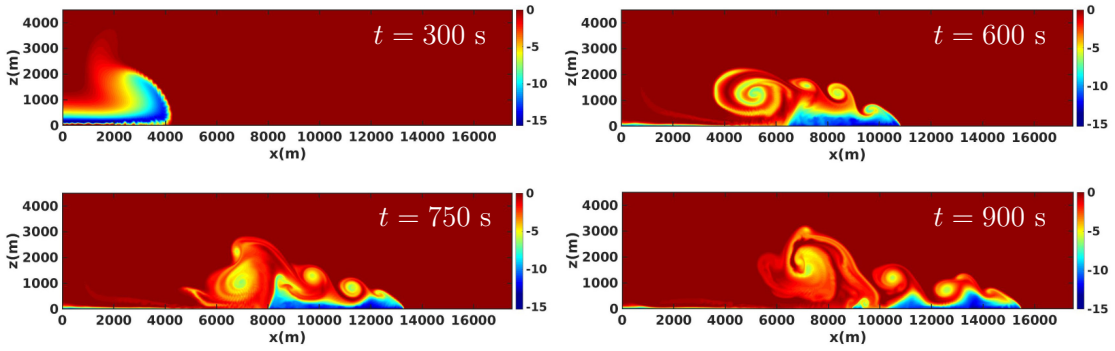

Figure 4.23: Density current, Deconvolution-based (a_D), $\delta = 5m$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 12.5$ m.

Figure 4.28 report the convective term error due to the inexact splitting that we commit when we perform the Evolve-Filter algorithm. For simplicity, the pics are refered to the mesh-size $h = 50m$. It is important to note that in both the two graphs the error is referred to the L^∞ norm. In all the simulations the average error (arithmetic average) has an order of magnitude $< 1e - 15$: this confirms that our method can be considered

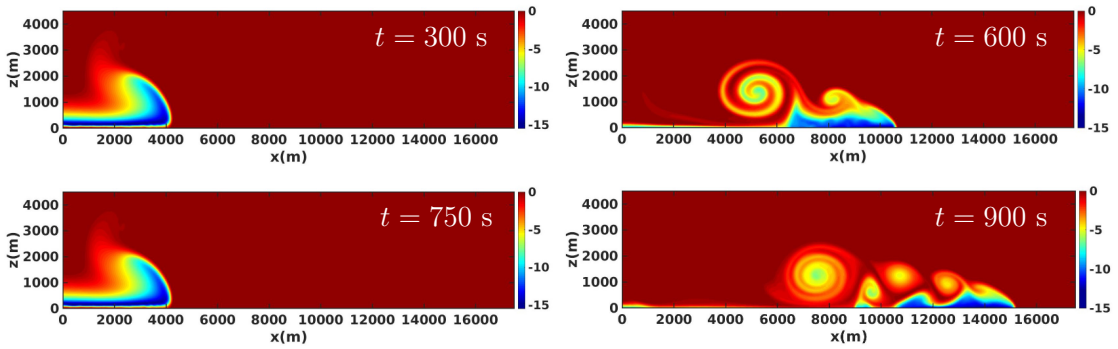

Figure 4.24: Density current, Deconvolution-based (a_D), $\delta = 10m$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 25$ m.

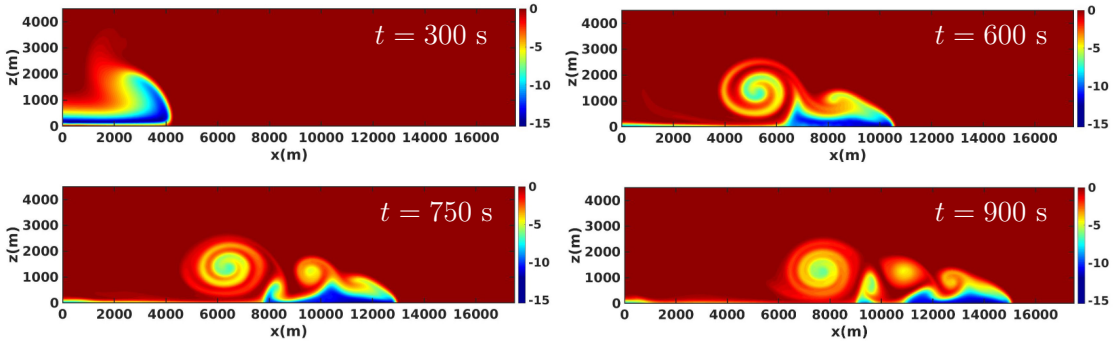

Figure 4.25: Density current, Deconvolution-based (a_D), $\delta = 12m$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m.

Method	h (m)	δ (m)	Front Location (m)
Ref. [30]	12.5	-	15056
EFR, Deconvolution-based	12.5	5	15560
EFR, Deconvolution-based	25	10	15215
EFR, Deconvolution-based	50	12	15120
EFR, Deconvolution-based	50	20	14800
Ref. [31]	(25, 200)	-	(14533,17070)

Table 4.5: Density current, Deconvolution-based (a_D): front location at $t = 900$ s obtained with the EFR algorithm and different meshes. Our results are compared against results from [31, 30]. For reference [31], we provide the range of mesh sizes and front location values obtained with different methods. For reference [30], we report only the front location computed with the finest resolution.

effective in these types of problem. Note also that the error committed for the enthalpy does not create problems, indeed, the order of magnitude of the enthalpy is $\simeq 1e4$,

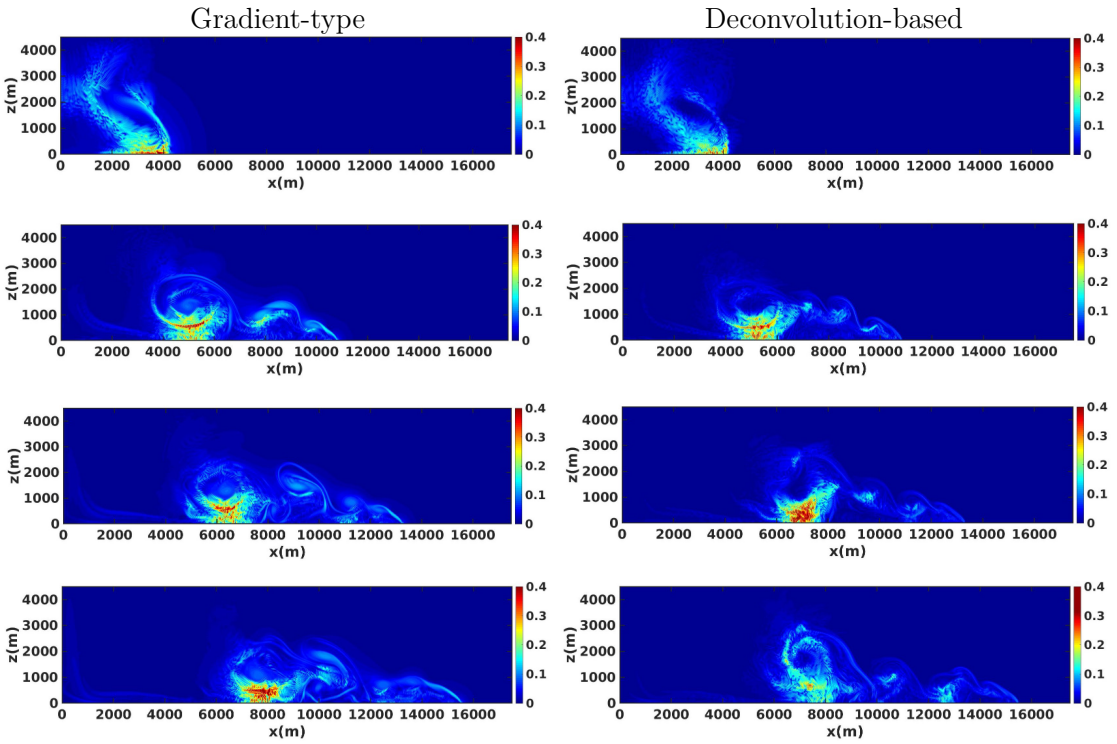

Figure 4.26: Density current, mesh $h = 12.5$ m: comparison between the Gradient-type (a_S), $\delta = 4m$ (left) and the Deconvolution-based (a_D), $\delta = 5m$ (right).

thus, the error is negligible. Interestingly, the more is the filter selective, the more is the maximum splitting error since as reveals fig. 4.28 the Deconvolution-based type has, in average, an error greater respect the others filter. This can be justified by the fact that this filter tends to regularize regions where the "peaks" are localized. Moreover, note that the maximum error is reached for $t \simeq 300s$ that is the moment when the cold bubble "stretches" after having hitting the bottom floor (see fig. 4.24, top left). Furthermore, as expected, the trends for the splitting terms matches the L^∞ norm of the difference between the filtered and unfiltered quantities as revealed in fig. 4.29.
We conclude reporting the computational cost. Table 4.6 reports the time taken by the evolve and the filter step separately and the total time of the entire simulation. The simulations are performed on a common laptop (AMD Ryzen 7 5700U, 16GB RAM). As expected, the extreme selectiveness of the Deconvolution-based must cope with the higher computational cost required by the filtering step. As explained in 2.3.5, the higher cost is due to the necessity of the evaluation of the indicator function which requires the resolution of a linear Helmholtz problem. Note also that the filter step for the Deconvolution based-requires half of the time employed by the evolve step: this is due to the solver choices. During the evolve step, the majority of the computational cost is spent to solve an elliptic equation for p' with the Diagonal incomplete Cholesky preconditioned conjugate gradient method, which is the same method used for the filter step. Since the equation of mass conservation is treated fully explicitly, it is very inexpensive to solve. The solver for the

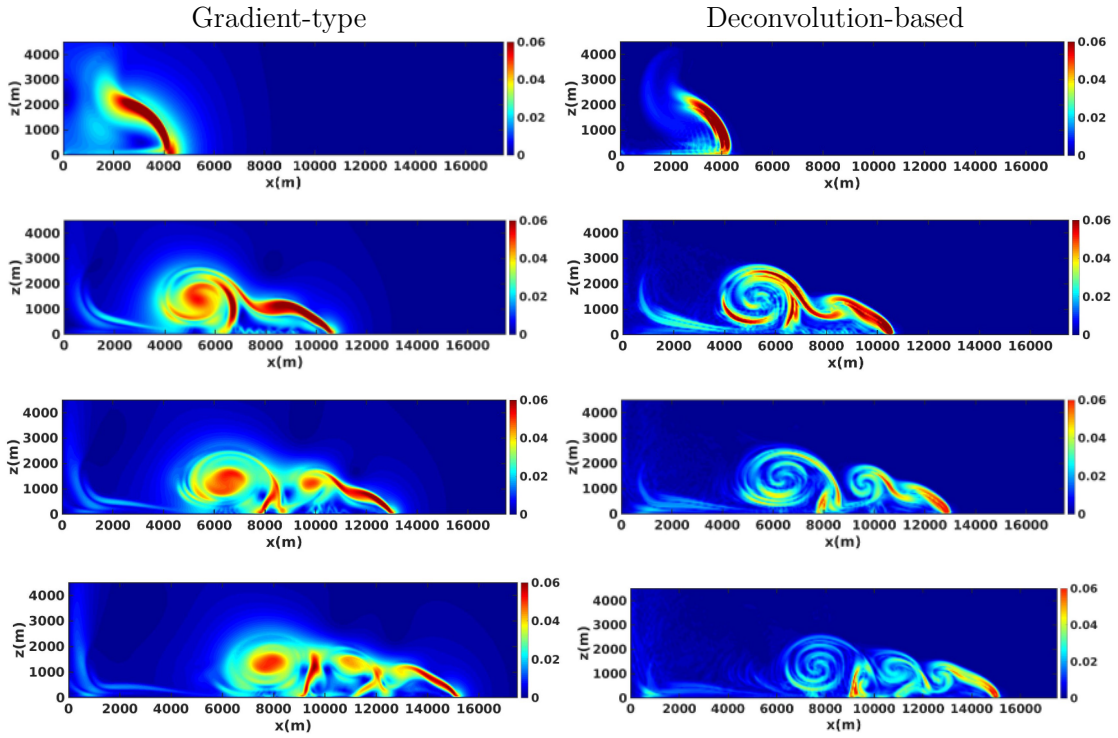

Figure 4.27: Density current, mesh $h = 50$ m: comparison between the Gradient-type (a_S), $\delta = 11m$ (left) and the Deconvolution-based (a_D), $\delta = 12m$ (right) indicator functions

conservation of energy equation uses the bi-conjugate gradient stabilized method with a diagonal-based incomplete LU preconditioner. The accuracy for the resolution of all the linear system is set to $1e-8$. The computational cost of the evolve step is also contained by not performing a momentum predictor step.

Model	h (m)	δ (m)	Evolve (s)	Filter (s)	Total (s)
Linear	25	2.7	0.3	0.070	3492
Gradient-type	25	8	0.3	0.108	3880
Deconvolution-based	25	10	0.3	0.159	4163
Linear	50	2.7	0.06	0.015	707
Gradient-type	50	11	0.06	0.023	772
Deconvolution-based	50	12	0.06	0.029	790

Table 4.6: Density current: computational time taken by the evolve step and filter step per time step and total simulation time for the EFR algorithm with the various indicator functions and the specified values of δ for meshes $h = 50$,25m.

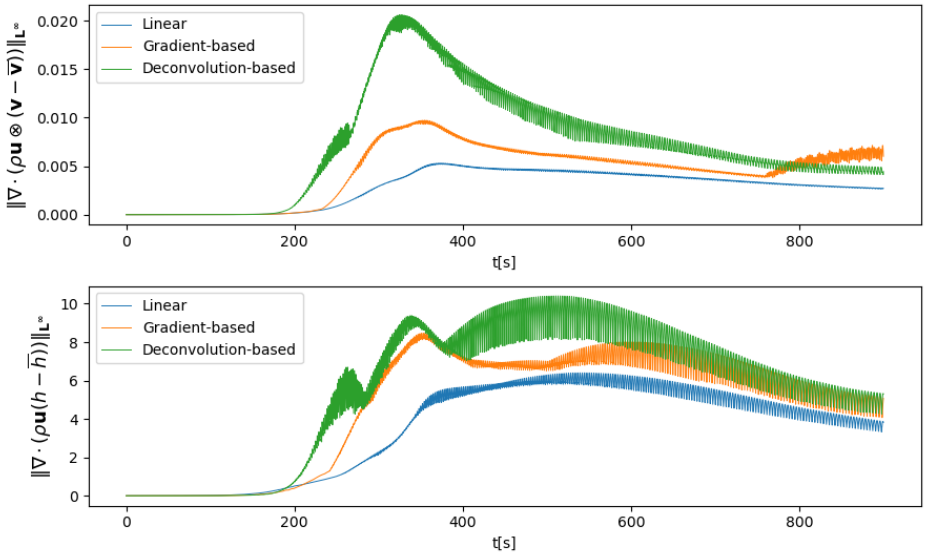

Figure 4.28: Density current: time evolution of the L^∞ norm for the inexact splitting term for the velocity (top panel) and for the energy (bottom panel)

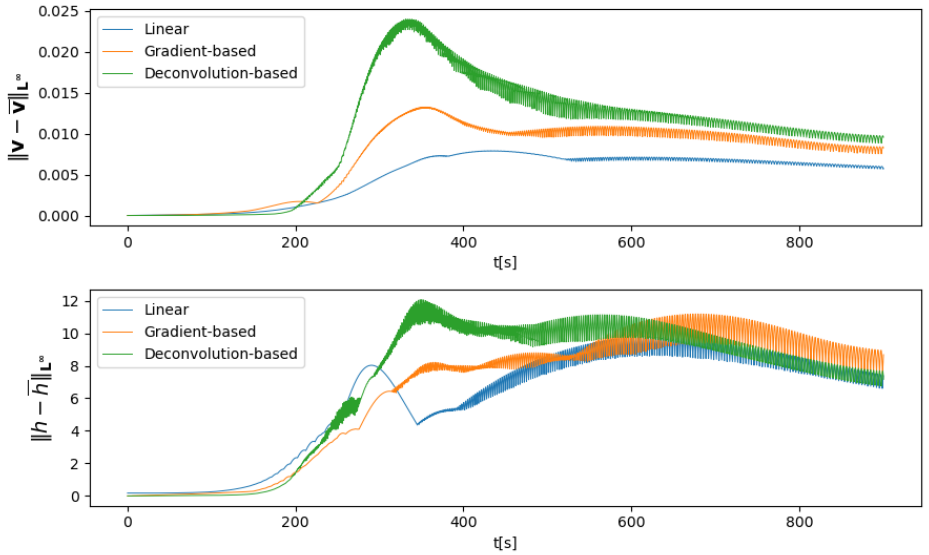

Figure 4.29: Density current: time evolution of the L^∞ norm for the difference between the filtered quantities and the quantities at the end of the evolve step

4.3.4 Results with Relaxation step

Based on the estimate of subsection 2.3.4 we calculate the relaxation parameter with $\Delta t = 0.1s, \eta \simeq 2e-4m, \nu \simeq 1.5e-5m^2/s, h = \delta = 50m$.

$$\xi \gtrsim \frac{4}{3}\frac{1.5e-5}{(\delta^2)\|a\|_\infty}\frac{h}{\eta}\Delta t = \xi_{min}$$

In figure 4.30 are shown the results for all the filters. This estimate of the relaxation parameter is not able to trigger the Kelvin-Helmholtz instability.
We remark that the correct value of the relaxation parameter is still an open problem. Moreover, in equation 4.3.4, the value ξ_{min} is only a lower bound for the correct value of the relaxation parameter, thus a sensitivity analysis for the radius must be accomplished to tune the relaxation parameter.

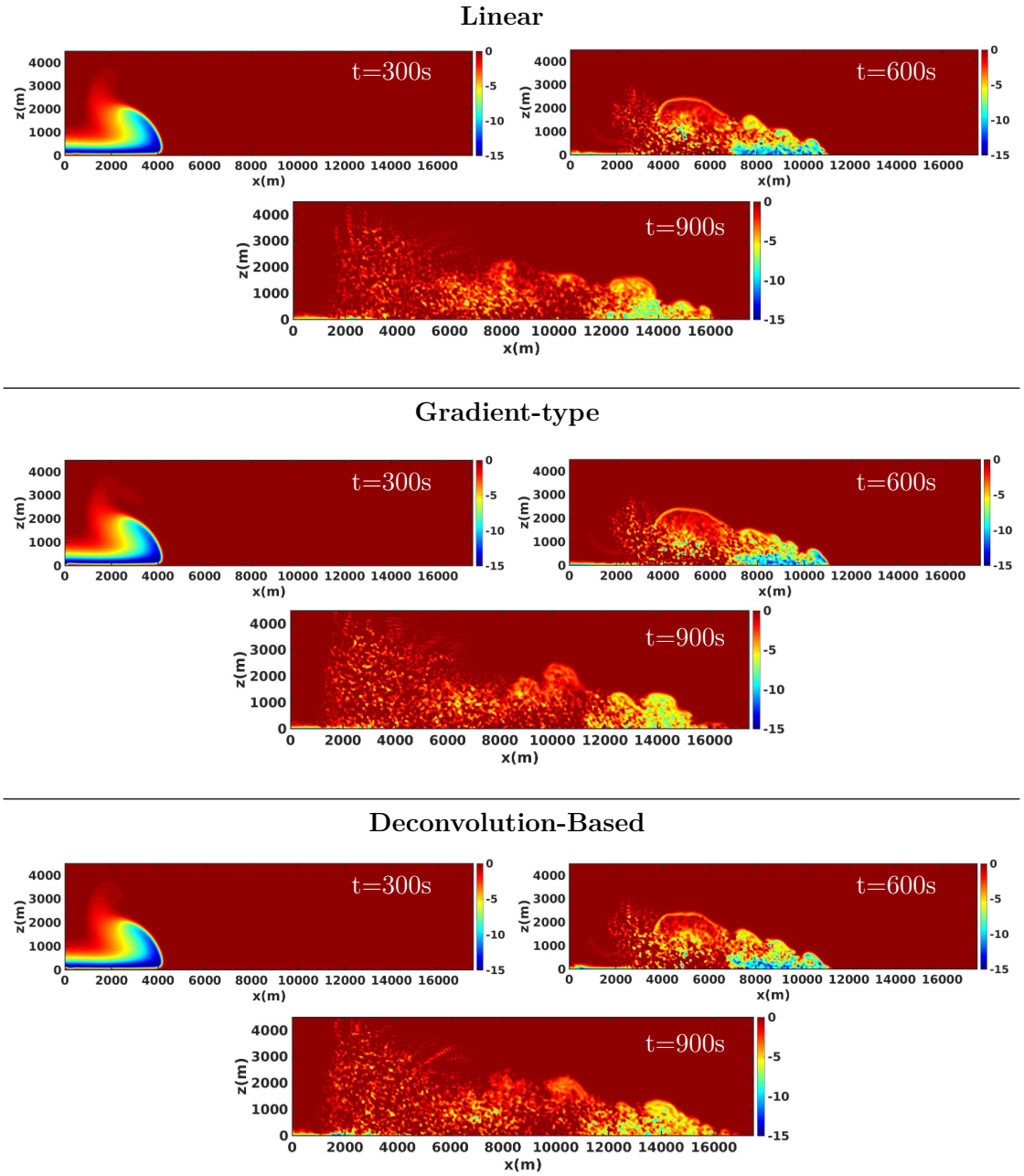

Figure 4.30: Density current, $h = \delta = 50m$, time evolution of the potential temperature perturbation calculated with the relaxation parameter ξ_{min}

4.4 Conclusion and future perspective

In this Master's Thesis we have investigated the Evolve-Filter-Relax algorithm in the context of the atmospherical flow dynamics simulation.
The inviscid compressible Navier Stokes equations are used as starting point for the work and the colocated finite volume method is used for the space discretization. A pressure-based solver was employed for the evolve step since the Mach Number for these applications is low.
Regarding the filter step, three type of filters are investigated in this work: a Linear, a Smagorinsky-like and a Deconvolution-based. The estimation of the parameters is based on the results present in literature for the well-known turbulence models (e.g. the classical Smagorinsky model). As expected, the simulations confirm the extreme selectiveness of the Deconvolution-based filter while the linear is the most diffusive.
However, the algorithm with the relax parameter based on a gross estimate is under diffusive and then it could be interesting to investigate a strategy to obtain the correct value of the relaxation step.
Further improvements for the methodology are possible: we could consider a different discretization scheme in space and in time and mostly important we can apply the EFR algorithm for the equation set with the potential temperature instead the of energy equation. More room of improvement could be achieved also by taking more steps in the Van-Cittert deconvolution operator.
An investigation of the correct value for the relaxation parameter should be beneficial for the complete algorithm.

Bibliography

[1] A.Takhirov A.L.Bowers, L.G.Rebholz and C.Trenchea. Improved accuracy in regularization models of incompressible flow via adaptive nonlinear filtering. *International Journal for Numerical Methods in Fluids*, 706:805–828, 2012.

[2] Tong Wu Alina Chertock, Alexander Kurganov and Jun Yan. Well-balanced numerical method for atmospheric flow equations with gravity. *Applied Mathematics and Computation*, 493:1–13, 2023.

[3] L. Bertagna, A. Quaini, and A. Veneziani. Deconvolution-based nonlinear filtering for incompressible flows at moderately large Reynolds numbers. *International Journal for Numerical Methods in Fluids*, 81(8):463–488, 2016.

[4] B.P.Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *computer methods in applied mechanics and engineering*, 1979.

[5] Stephen B.Pope. *Turbulent Flows*. Mathematics of Computations, 2000.

[6] B.Spalding. A novel finite difference formulation for differential expressions involving both first and second derivatives. *International journal for numerical methods in engineering*, 1972.

[7] Vincent J. Ervin, William J. Layton, and Monika Neda. Numerical analysis of filter-based stabilization for evolution equations. *SIAM Journal on Numerical Analysis*, 50(5):2307–2335, 2012.

[8] M. Darwish F. Moukalled, L. Mangani. *The Finite Volume Method in Computational Fluid Dynamics*. Springer, 2015.

[9] Alexandre Favre. *Physics of Fluids*, 26:2851–2863, 1983.

[10] Joel H Ferziger and Milovan Peric. *Computational methods for fluid mechanics*. Springer, 2002.

[11] GEA - Geophysical and Environmental Applications. https://github.com/GEA-Geophysical-and-Environmental-Apps/GEA.

[12] Vreman B. Geurts B., Kuerten H., and Theofilis V. Les modeling errors in free and wall bounded compressible shear layers. *Engineering turbulence modelling and experiments 2*, 2:325–334, 1993.

[13] Aimé Fournier Henry Weller, Hilary Weller. Voronoi, delaunay, and block-structured mesh refinement for solution of the shallow-water equations on the sphere. *Monthly Weather Review*, 2009.

[14] Hilary Weller Henry Weller. A high-order arbitrarily unstructured finite-volume model of the global atmosphere: tests solving the shallow-water equations. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 0, 2000.

[15] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1):40–65, 1986.

[16] J.A.Chorin. *Numerical Solution of the Navier Stokes Equations*. Mathematics of Computations, 1968.

[17] Jie Schen J.L. Guermond, P.Minev. An overview of projection methods for incompressible flows. *Computer Methods in applied mechanics and engineering*, 195(23):6011–6045, 2005.

[18] J.Smagorisnky. General circulation experiments with the primitive equations. *Mon. Wea. Rev.*, 91:99–164, 1963.

[19] Roger Käppeli. Well-balanced methods for computational astrophysics. *Living Reviews in Computational Astrophysics*, 8, 2022.

[20] A. Veneziani L. Bertagna, A. Quaini. Deconvolution-based nonlinear filtering for incompressible flows at moderately large reynolds numbers. *Numerical Methods in fluid*, 81(8):463–488, 2016.

[21] D. K Lilly. The representation of small-scale turbulence in numerical sim- ulation experiments. *IBM Scientific Computing Symp. on Environmental Sciences*, 8, 1967.

[22] G.Rozza M.Girfoglio, A. Quaini. A finite volume approximation of the navier stokes equations with nonlinear filtering stabilization. *Computer and Fluids*, 187(1):27–45, 2019.

[23] G.Rozza M.Girfoglio, A.Quaini. VALIDATION OF OPENFOAM ® SOLVERS FOR THE EULER EQUATIONS USING BENCHMARKS FOR MESOSCALE ATMOSPHERIC MODELING. 2022.

[24] Gianluigi Rozza Michele Girfoglio, Annalisa Quaini. A novel large eddy simulation model for the quasi-geostrophic equations in a finite volume setting. *Journal of Computational and Applied Mathematics*, 2023.

[25] J. Lindeman N. Ahmad. Euler solutions using flux-based wave decomposition. *Int. J. Numer. Meth. Fluids*, 54:47–72, 2007.

[26] S. Langenberg N. Botta, R. Klein and S. Lutzenkirchen. Well balanced finite volume methods for nearly hydro- static flows. *J. Comput. Phys.*, 196:539–565, 2004.

[27] N.N.Ahmad. High-resolution wave propagation method for stratified flows. Technical report, AIAA, 2018.

[28] Rolf Rannacher. On chorin's projection method for the incompressible navier-stokes equations. *Monthly Weather Review*, 2006.

[29] Richard M.Beam R.F.Warming. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA Journal*, 1976.

[30] S.Marras, Murtazo Nazarov, and Francis Giraldo. Stabilized high-order galerkin methods based on a parameter-free dynamic sgs model for les. *Journal of computational physics*, 301(2):77–101, 2015.

[31] J.M. Straka, R.B. Wilhelmson, L.J. Wicker, J.R. Anderson, and K.K. Droegemeier. Numerical solutions of a non-linear density current: a benchmark solution and comparisons. *International Journal for Numerical Methods in Fluids*, 17(2):1–22, 1997.

[32] Nathalie Tufenkji and Menachem Elimelech. Correlation equation for predicting single-collector efficiency in physicochemical filtration in saturated porous media. *Environmental science & technology*, 38(2):529–536, 2004.

[33] A.W. Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids.*, 16:3670—-3681, 2004.

[34] Catalin Trenchea W. layton, Leo. G. Rebholz. Modular nonlinear filter stabilization of methods for higher reynolds numbers flow. *Journal of Mathematical Fluid Mechanics*, 15:325–354, 2012.