Computers and Fluids 266 (2023) 106057

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Filter stabilization for the mildly compressible Euler equations with application to atmosphere dynamics simulations

Nicola Clinco [a], Michele Girfoglio [a], Annalisa Quaini [b], Gianluigi Rozza [a,*]

[a] mathLab, Mathematics Area, SISSA, via Bonomea, 265, Trieste, I-34136, Italy
[b] Department of Mathematics, University of Houston, Houston TX 77204, USA

ARTICLE INFO

Keywords:
Filter stabilization
Large Eddy Simulation
Non-hydrostatic atmospheric flows
Finite volume approximation
Evolve-Filter-Relax algorithm

ABSTRACT

We present a filter stabilization technique for the mildly compressible Euler equations that relies on a linear or nonlinear indicator function to identify the regions of the domain where artificial viscosity is needed and determine its amount. For the realization of this technique, we adopt a three step algorithm called Evolve-Filter-Relax (EFR), which at every time step evolves the solution (i.e., solves the Euler equations on a coarse mesh), then filters the computed solution, and finally performs a relaxation step to combine the filtered and non-filtered solutions. We show that the EFR algorithm is equivalent to an eddy-viscosity model in Large Eddy Simulation. Three indicator functions are considered: a constant function (leading to a linear filter), a function proportional to the norm of the velocity gradient (recovering a Smagorinsky-like model), and a function based on approximate deconvolution operators. Through well-known benchmarks for atmospheric flow, we show that the deconvolution-based filter yields stable solutions that are much less dissipative than the linear filter and the Smagorinsky-like model and we highlight the efficiency of the EFR algorithm.

1. Introduction

The Direct Numerical Simulation (DNS) is a simulation in Computational Fluid Dynamics (CFD) that solves the equations governing the fluid motion by resolving the entire range of relevant spatial and temporal scales. In many practical CFD applications, the smallest spatial scales can be several orders of magnitude smaller than the largest scales in the flow. An example is atmospheric flow, whose smallest spatial scales are typically of the order of 10^{-4} m while the typical domain size is of the order of $10^4 - 10^5$ m. For these applications, a DNS is beyond reach for nowadays computing machines and it will be for the foreseeable future.

One way to keep the computational cost affordable without sacrificing accuracy is to solve for the flow using a coarser mesh and model the effects of the small scales that are not directly solved through a so-called subgrid-scale (SGS) model. This is the principal idea behind Large Eddy Simulation (LES). Traditionally, SGS models introduce the effects of the unresolved scales with momentum fluxes that are linearly dependent upon the rate of strain of the large scales. This is known eddy-viscosity closure. The most famous eddy-viscosity model is the Smagorinsky model [1]. Its success is due to several factors: (i) it is relatively simple and easy to implement, (ii) it is computationally inexpensive compared to other SGS models, and (iii) it features parameters that can be tuned for the particular application at hand so that the results are realistic. The main limitation of the Smagorinsky model is the assumption of local balance between the subgrid scale energy production and dissipation. Since such equilibrium conditions do not hold in many practical applications, the Smagorinsky model often results into over-diffusive simulations. A large body of research has been motivated by improving upon the Smagorinsky model.

Some alternative methods introduce artificial diffusion that can be solution-dependent (see, e.g., [2–6]) or residual-based (see, e.g., [7–11]). These methods are driven by the intent to have an artificial viscosity that vanishes where the solution is smooth and/or decreases as the grid is refined. Other methods add a set of equations to the discrete governing equations formulated on a coarse mesh (coarse is meant with respect to the resolution required by DNS). This extra-problem can be devised in different ways, for example by a functional splitting of the solved and unresolved scales as in variational multiscale methods (see, e.g., [12–15]). In this paper, we propose an extra problem that acts as a differential (linear or nonlinear) low-pass filter added sequentially to the mildly compressible Euler equations for stratified flows. This sequential algorithm is called Evolve-Filter-Relax (EFR) since, at every time step, one first evolves the solution, i.e., solves the Euler equations on a coarse mesh, then filters the computed solution, and finally performs a relaxation step to combine the filtered and non-filtered solutions. This techniques is also known with the name of filter

* Corresponding author.
E-mail address: gianluigi.rozza@sissa.it (G. Rozza).

https://doi.org/10.1016/j.compfluid.2023.106057
Received 20 May 2023; Received in revised form 31 August 2023; Accepted 14 September 2023
Available online 22 September 2023

stabilization because it reduces or eliminates unphysical fluctuations in the computed solution. We will show that the EFR algorithm is an eddy-viscosity model.

Introduced in [16], EFR algorithms have been widely applied to the incompressible Navier–Stokes equations [16–23]. It was shown in [16,18–20,22] that numerical results obtained with nonlinear differential filters are more precise in localizing where eddy viscosity is needed and are overall more accurate than results obtained with plain Smagorinsky-type models. Despite these promising results for the incompressible Navier–Stokes equations, the application of filter stabilization to the Euler equations has received much less attention [24–27]. A possible alternative to nonlinear filters is given by variational multiscale methods, which approximate (i.e., not model or filter) the unresolved scale. See, e.g., [28–30].

In this paper, we consider the EFR algorithm for the Euler equations with both linear and nonlinear filters. Developed in [31–33], stabilization based on linear filters has been widely studied (see, e.g., [34–36]). However, it was noted in [37] that a linear stabilization can, at most, give a solution converging to a weak solution that is not the entropy solution, hence the need to investigate nonlinear filter stabilization techniques. We will show that the EFR algorithm with a deconvolution-based filter yields stable solutions that are much less dissipative than the Smagorinsky model. We recall that the use of deconvolution operators in SGS models to increase accuracy is well established and mathematically grounded [38–40].

The main advantages of the EFR algorithm are: (i) modularity, i.e., its implementation does not require any major modification of a legacy solver, and (ii) flexibility in the choice of the filter. In addition, if one chooses the deconvolution-based filter, the viscosity introduced by the EFR algorithm vanishes where the solution is smooth and decreases as the mesh is refined. We will show that we obtain numerical results that agree very well with data published in the literature for well-known 2D benchmark problems involving stratified and gravity driven atmospheres. We will also show that the computational cost to solve the additional filter problem is a fraction of the computational cost required by the Euler solver.

All the simulations in this paper have been carried out with GEA (Geophysical and Environmental Applications) [41], a new open-source atmosphere and ocean modeling framework within the finite volume C++ library OpenFOAM® [42]. For more details on GEA, see [43,44]. Although we demonstrate numerically the accuracy and efficiency of EFR algorithm using a finite volume method for space discretization, the algorithm itself can be used with any space discretization method.

The outline of the paper is as follows. Section 2 describes the compressible Euler equations for low Mach stratified flows and introduces the filter stabilization for this model. In Section 3, we discuss space discretization and the perturbation terms introduced by the filter stabilization to the Euler equations. Numerical results are presented in Section 4 and conclusions are drawn in Section 5.

2. Problem definition

2.1. The compressible Euler equations

We consider mildly compressible Euler equations to describe the motion of the dry atmosphere, i.e., a compressible inviscid fluid, assumed to behave like an ideal gas. Let Ω be a spatial domain of interest and $(0, t_f]$ a time interval of interest. Let ρ, $\mathbf{u} = (u, v, w)$, and p be the fluid density, velocity, and pressure. Moreover, let $e = c_v T + |\mathbf{u}|^2/2 + gz$ be the total energy density, where c_v is the specific heat capacity at constant volume, T is the absolute temperature, g is the gravitational constant, and z is the vertical coordinate. The conservation of mass, momentum, and total energy can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \text{in } \Omega \times (0, t_f], \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p + \rho g \hat{\mathbf{k}} = \mathbf{0} \qquad \text{in } \Omega \times (0, t_f], \tag{2}$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) + \nabla \cdot (p \mathbf{u}) = 0 \qquad \text{in } \Omega \times (0, t_f], \tag{3}$$

where $\mathbf{k}$ is the unit vector aligned with the vertical axis z. We close system (1)–(3) using the following thermodynamics equation of state for p:

$$p = \rho R T, \tag{4}$$

where R is the specific gas constant of dry air.

Let us to write the pressure as the sum of a fluctuation p' with respect to a hydrostatic term:

$$p = p' + \rho g z. \tag{5}$$

By plugging (5) into (2), we obtain:

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p' + g z \nabla \rho = 0 \quad \text{in } \Omega \times (0, t_f]. \tag{6}$$

Let c_p be the specific heat capacity at constant pressure for dry air and let

$$K = |\mathbf{u}|^2/2, \quad h = c_v T + p/\rho = c_p T, \tag{7}$$

be the kinetic energy density and the specific enthalpy, respectively. The total energy density can be written as $e = h - p/\rho + K + gz$. Then, Eq. (3) can be rewritten as:

$$\frac{\partial (\rho h)}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) + \frac{\partial (\rho K)}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) - \frac{\partial p}{\partial t} + \rho g \mathbf{u} \cdot \hat{\mathbf{k}} = 0, \tag{8}$$

where we have used Eq. (1) for further simplification.

This paper focuses on formulation (1),(4)–(8) of the Euler equations.

A quantity of interest for atmospheric problems is the potential temperature

$$\theta = \frac{T}{\pi}, \quad \pi = \left(\frac{p}{p_0} \right)^{\frac{R}{c_p}}, \tag{9}$$

where $p_0 = 10^5$ Pa, which is the atmospheric pressure at the ground. Additionally, we define the potential temperature fluctuation θ' as the difference between θ and its mean hydrostatic value θ_0:

$$\theta'(x, y, z, t) = \theta(x, y, z, t) - \theta_0(z). \tag{10}$$

See, e.g., [45] for more details.

2.2. Filter stabilization as an eddy viscosity model

A numerical solution of system (1),(4)–(8) computed with a mesh coarser than necessary for a DNS will be affected by non-physical oscillations that will eventually lead to a simulation breakdown. In order to avoid incurring into non-physical solutions, we adapt to the Euler equations an algorithm that has been shown to be accurate, efficient, and robust for the incompressible Navier–Stokes equations [16–23]. This algorithm consists of three steps: in the first step (called Evolve) one approximates the solution to the Euler equations with a coarse mesh, in the second step (called Filter) the numerical oscillations are smoothed out with a differential filter to obtain a filtered solution, and in the third step (called Relax) one combines the filtered and non-filtered solutions. This Evolve-Filter-Relax (EFR) algorithm, which is a computationally efficient realization of filter stabilization, is described next.

Let $\Delta t \in \mathbb{R}$, $t^n = n\Delta t$, with $n = 0, \dots, N_f$ and $t_f = N_f \Delta t$. Moreover, we denote by y^n the approximation of a generic quantity y at the time t^n. We adopt a Backward Differentiation Formula of order 1 (BDF1) for the discretization of the Eulerian time derivatives in (1),(6),(8). Other time discretization schemes are possible (see, e.g., [16,17,21,22]). The EFR algorithm reads as follows: given ρ^0, $\mathbf{u}^0$, h^0, p^0, and T^0, set $K^0 = |\mathbf{u}^0|^2/2$ and for $n \geq 0$ perform the following steps:

- *Step 1 - Evolve*: find density ρ^{n+1} and intermediate variables $\mathbf{v}^{n+1}, l^{n+1}, K_{\mathbf{v}}^{n+1}, q^{n+1}, q'^{,n+1}, T_l^{n+1}$ such that:

$$\frac{\rho^{n+1}-\rho^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1})=0, \tag{11}$$

$$\frac{\rho^{n+1}\mathbf{v}^{n+1}-\rho^n\mathbf{u}^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1}\otimes\mathbf{v}^{n+1})+\nabla q'^{,n+1}+gz\nabla\rho^{n+1}=\mathbf{0}, \tag{12}$$

$$\frac{\rho^{n+1}l^{n+1}-\rho^n h^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1}l^{n+1})+\frac{\rho^{n+1}K_{\mathbf{v}}^{n+1}-\rho^n K^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1}K_{\mathbf{v}}^{n+1})-\frac{q^{n+1}-p^n}{\Delta t}+\rho^{n+1}g\mathbf{v}^{n+1}\cdot\hat{\mathbf{k}}=0, \tag{13}$$

$$q^{n+1}=q'^{,n+1}+\rho^{n+1}gz, \tag{14}$$

$$q^{n+1}=\rho^{n+1}RT_l^{n+1}, \tag{15}$$

$$l^{n+1}-l^n=c_p(T_l^{n+1}-T_l^n), \tag{16}$$

$$K_{\mathbf{v}}^{n+1}=\frac{|\mathbf{v}^{n+1}|^2}{2}. \tag{17}$$

Notice that in (16) we have chosen to update the value of the intermediate specific enthalpy in an incremental fashion.

- *Step 2 - Filter*: find filtered variables $\bar{\mathbf{v}}^{n+1}, \bar{l}^{n+1}$ such that

$$\bar{\mathbf{v}}^{n+1}=F\mathbf{v}^{n+1}, \tag{18}$$

$$\bar{l}^{n+1}=Fl^{n+1}, \tag{19}$$

where F is a generic filter that could be linear or nonlinear. We will present possible choices for F in Section 2.3.

- *Step 3 - Relax*: find end of step $\mathbf{u}^{n+1}, h^{n+1}, K^{n+1}, p^{n+1}, p'^{,n+1}, T^{n+1}$ such that

$$\mathbf{u}^{n+1}=(1-\chi)\mathbf{v}^{n+1}+\chi\bar{\mathbf{v}}^{n+1}, \tag{20}$$

$$h^{n+1}=(1-\xi)l^{n+1}+\xi\bar{l}^{n+1}, \tag{21}$$

$$p^{n+1}=p'^{,n+1}+\rho^{n+1}gz, \tag{22}$$

$$p^{n+1}=\rho^{n+1}RT^{n+1}, \tag{23}$$

$$h^{n+1}-h^n=c_p(T^{n+1}-T^n), \tag{24}$$

$$K^{n+1}=\frac{|\mathbf{u}^{n+1}|^2}{2}, \tag{25}$$

where $\chi, \xi\in[0,1]$ are relaxation parameters.

The connection between the EFR algorithm and LES modeling is easily seen by shifting the index $n+1$ to n in (18)–(21) and plugging them into (12)–(13) to obtain:

$$\frac{\rho^{n+1}\mathbf{v}^{n+1}-\rho^n\mathbf{v}^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1}\otimes\mathbf{v}^{n+1})+\nabla q'^{,n+1}+gz\nabla\rho^{n+1}+\frac{\chi}{\Delta t}G\mathbf{v}^n=\mathbf{0}, \tag{26}$$

$$\frac{\rho^{n+1}l^{n+1}-\rho^n l^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1}l^{n+1})+\frac{\rho^{n+1}K_{\mathbf{v}}^{n+1}-\rho^n K^n}{\Delta t}+\nabla\cdot(\rho^{n+1}\mathbf{v}^{n+1}K_{\mathbf{v}}^{n+1})-\frac{q^{n+1}-p^n}{\Delta t}+\rho^{n+1}g\mathbf{v}^{n+1}\cdot\hat{\mathbf{k}}+\frac{\xi}{\Delta t}Gl^n=0, \tag{27}$$

with $G=I-F$, I being the identity operator. System (11), (26), (27), (14)–(17) gives us an implicit discretization of problem (1),(4)–(8) with BDF1 and an additional explicitly treated (linear or nonlinear) dissipation term.

Let us assume that $\chi=\chi_0\Delta t$ and $\xi=\xi_0\Delta t$, where χ_0 and ξ_0 are time-independent constants. Then, system (11), (26), (27), (14)–(17) can be seen as a time-stepping scheme for problem:

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=0, \tag{28}$$

$$\frac{\partial(\rho\mathbf{v})}{\partial t}+\nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v})+\nabla p'+gz\nabla\rho+\chi_0G\mathbf{v}=\mathbf{0}, \tag{29}$$

$$\frac{\partial(\rho l)}{\partial t}+\nabla\cdot(\rho\mathbf{u}l)+\frac{\partial(\rho K)}{\partial t}+\nabla\cdot(\rho\mathbf{u}K)-\frac{\partial p}{\partial t}+\rho g\mathbf{u}\cdot\hat{\mathbf{k}}+\xi_0Gl=0, \tag{30}$$

$$p=p'+\rho gz, \tag{31}$$

$$p=\rho RT, \tag{32}$$

$$h=c_pT, \tag{33}$$

$$K=|\mathbf{v}|^2/2. \tag{34}$$

Thus, filter stabilization algorithm (11)–(25) can be interpreted as a splitting scheme for problem (28)–(34).

Notice that model (28)–(34) can be considered as a LES model of the eddy-viscosity type with closure:

$$\nabla\cdot(\rho\overline{\mathbf{v}\otimes\mathbf{v}}-\rho\bar{\mathbf{v}}\otimes\bar{\mathbf{v}})\approx\chi_0G\mathbf{v}, \tag{35}$$

$$\nabla\cdot(\rho\overline{\mathbf{v}l}-\rho\bar{\mathbf{v}}\bar{l})\approx\xi_0Gl. \tag{36}$$

This shows the connection between algorithm (11)–(25) and LES modeling.

2.3. A possible choice for the filter

We will consider the following filter for step 2 (18)–(19):

$$F=(I+L)^{-1},\quad L=-\nabla\cdot(\delta\nabla) \tag{37}$$

where $\delta>0$ is a linear or nonlinear artificial "viscosity". Such a filter applied to $\mathbf{v}^{n+1}$ as in (18) amounts to solving the following problem: find $\bar{\mathbf{v}}^{n+1}$ such that

$$-\nabla\cdot(\delta\nabla(\bar{\mathbf{v}}^{n+1}))+\bar{\mathbf{v}}^{n+1}=\mathbf{v}^{n+1},\quad \delta=\alpha^2a(\mathbf{v}^{n+1}), \tag{38}$$

where α can be interpreted as the filtering radius and $a(\cdot)\in(0,1]$ is the so-called indicator function. Note that δ is not properly a viscosity since it has the dimension of a length squared. However, if we multiply (38) by $\rho^{n+1}/\Delta t$, we obtain Stokes problem:

$$\frac{\rho^{n+1}}{\Delta t}(\bar{\mathbf{v}}^{n+1}-\mathbf{v}^{n+1})-\nabla\cdot(\overline{\mu}\nabla\bar{\mathbf{v}}^{n+1})=\mathbf{0},\quad \overline{\mu}=\rho^{n+1}\frac{\alpha^2}{\Delta t}a(\mathbf{v}^{n+1}), \tag{39}$$

where $\overline{\mu}$ is dimensionally a dynamic viscosity.

The same filter applied to l^{n+1} as in (19) yields:

$$\frac{\rho^{n+1}}{\Delta t}(\bar{l}^{n+1}-l^{n+1})-\nabla\cdot(\overline{\mu}\nabla\bar{l}^{n+1})=0. \tag{40}$$

In summary, the EFR algorithm we will use in this paper entails performing the following steps:

- *Step 1 - Evolve*: find density $\rho^{n+1}, \mathbf{v}^{n+1}, l^{n+1}, K_{\mathbf{v}}^{n+1}, q^{n+1}, q'^{,n+1}, T_l^{n+1}$ such that (11)–(17) hold.
- *Step 2 - Filter*: find filtered variables $\bar{\mathbf{v}}^{n+1}, \bar{l}^{n+1}$ such that (39)–(40) hold.
- *Step 3 - Relax*: set (20)–(25).

2.4. Possible choices for the indicator function

The success of the EFR algorithm in the simulation of atmospheric flows ultimately depends on the reliability of the indicator function. The indicator function has to be such that it takes values close to zero where its argument (i.e., the Euler velocity or specific enthalpy) does not need regularization, while it takes values close to 1 where the argument does need to be regularized. Different choices for indicator function $a(\cdot)$ have been proposed in the literature for the incompressible Navier–Stokes equations [16,18,46,47]. Some indicator functions [16,18] are based on physical quantities that are known to vanish for coherent flow structures. The drawback for these indicator functions is that they do not allow for a rigorous convergence theory to verify the robustness of the associated filtering method. Hence, mathematics-based (instead of physics-based) indicator functions were proposed [16,19]. In this paper, we will consider and compare three mathematics-based choices.

The first and easiest choice corresponds to a linear filter, i.e., we take

$$a(\mathbf{v}) = a_L(\mathbf{v}) = 1, \tag{41}$$

in (39). Besides linearity, another advantage of this choice is that it makes the operator in the filter equations constant in time. However, its efficacy is rather limited, since it introduces the same amount of regularization everywhere in the domain. This is likely to introduce overdiffusion as we will show in Section 4

A second mathematically convenient indicator function is

$$a(\mathbf{v}) = a_S(\mathbf{v}) = \frac{|\nabla \mathbf{v}|}{\|\nabla \mathbf{v}\|_\infty}, \tag{42}$$

which has strong monotonicity properties. With $a_S(\cdot)$ as indicator function for the EFR algorithm, we recover a Smagorinsky-like model, which is an improvement over the linear filter obtained with $a_L(\cdot)$.

Finally, we consider a class of deconvolution-based indicator functions, which were shown to be particularly accurate for realistic incompressible flow problems [17,21]. Such functions are defined as:

$$a(\mathbf{v}) = a_D(\mathbf{v}) = |\mathbf{v} - D(F(\mathbf{v}))|, \tag{43}$$

where F is the linear Helmholtz filter (i.e., (37) with δ constant in space and time) and D is the Van Cittert deconvolution:

$$D = \sum_{n=0}^{N} (I - F)^n. \tag{44}$$

We remark that D is a bounded regularized approximation of F^{-1}. Typically, N in (44) is set to 0, 1 [17,21]. In this paper, we consider $N = 0$, which means $D = I$. For this choice of N, indicator function (43) becomes

$$a_D(\mathbf{v}) = |\mathbf{v} - F(\mathbf{v})|. \tag{45}$$

3. Space discretization of the steps in the EFR algorithm

For space discretization, we adopt a finite volume method. The Evolve step is the most computationally intensive step in the EFR algorithm and to contain its computational cost we use a splitting scheme thoroughly described in [44]. This section focuses on the space discretization of the Filter and Relax steps.

Let us consider a partition of the computational domain Ω into cells or control volumes Ω_i, with $i = 1, \dots, N_c$, 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 $j = 1, \dots, M$. We will start with the space discretization of the Filter problem (39)–(40) and then write the space discrete version of the Relax Step (20)–(25)

The integral form of the Eq. (39) for each volume Ω_i is given by:

$$\frac{1}{\Delta t}\int_{\Omega_i} \rho^{n+1}\overline{\mathbf{v}}^{n+1}\, d\Omega - \int_{\Omega_i} \nabla\cdot(\overline{\mu}\nabla\overline{\mathbf{v}}^{n+1})\, d\Omega = \frac{1}{\Delta t}\int_{\Omega_i} \rho^{n+1}\mathbf{v}^{n+1}\, d\Omega.$$

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

$$\frac{1}{\Delta t}\int_{\Omega_i} \rho^{n+1}\overline{\mathbf{v}}^{n+1}\, d\Omega - \int_{\Omega_i} (\overline{\mu}\nabla\overline{\mathbf{v}}^{n+1})\cdot d\mathbf{A} = \frac{1}{\Delta t}\int_{\Omega_i} \rho^{n+1}\mathbf{v}^{n+1} d\Omega. \tag{46}$$

Let us denote with $(\overline{\mu}\nabla\overline{\mathbf{v}}^{n+1})_i$ and $\overline{\mathbf{v}}_i^{n+1}$ the average stress tensor and filtered velocity in control volume Ω_i, respectively. Similarly, we denote with ρ_i^{n+1} and $\mathbf{v}_i^{n+1}$ the average density and intermediate velocity in Ω_i. Then, Eq. (46) is approximated as follows:

$$\frac{1}{\Delta t}\rho_i^{n+1}\overline{\mathbf{v}}_i^{n+1} - \sum_j (\overline{\mu}\nabla\overline{\mathbf{v}}^{n+1})_{i,j}\cdot\mathbf{A}_j = \frac{1}{\Delta t}\rho_i^{n+1}\mathbf{v}_i^{n+1}. \tag{47}$$

We choose to approximate the gradient of $\overline{\mathbf{v}}_i^{n+1}$ at face j with second order accuracy. See [48] for more details.

Following a similar procedure for (40), we obtain:

$$\frac{1}{\Delta t}\rho_i^{n+1}\overline{l}_i^{n+1} - \sum_j (\overline{\mu}\nabla\overline{l}^{n+1})_{i,j}\cdot\mathbf{A}_j = \frac{1}{\Delta t}\rho_i^{n+1}l_i^{n+1}, \tag{48}$$

where $\overline{l}_i^{n+1}$ and l_i^{n+1} are the average filtered and intermediate specific enthalpy in Ω_i. For the approximation of the gradient of $\overline{l}_i^{n+1}$ at face j, we use the same formula used for the components of $\overline{\mathbf{v}}_i^{n+1}$.

Now, let us turn to the Relax step. The discretized form of each equation in the Relax step is simply given by taking the average of each variable in Ω_i:

$$\mathbf{u}_i^{n+1} = (1-\chi)\mathbf{v}_i^{n+1} + \chi\overline{\mathbf{v}}_i^{n+1}, \tag{49}$$

$$h_i^{n+1} = (1-\xi)l_i^{n+1} + \xi\overline{l}_i^{n+1}, \tag{50}$$

$$p_i^{n+1} = p_i'^{,n+1} + \rho_i^{n+1} g z_i, \tag{51}$$

$$p_i^{n+1} = \rho^{n+1} R T_i^{n+1}, \tag{52}$$

$$h_i^{n+1} - h_i^n = c_p(T_i^{n+1} - T_i^n), \tag{53}$$

$$K_i^{n+1} = \frac{|\mathbf{u}_i^{n+1}|^2}{2}, \tag{54}$$

where z_i is the vertical coordinate of the centroid of cell Ω_i.

3.1. The EFR algorithm as a solver for the Euler equations with perturbations

In this section, we will show that by combining the equations at the Evolve, Filter, and Relax steps we obtain the Euler equations perturbed by some extra terms and we discuss such terms.

Let us use a subindex h to denote the space-discrete solution, where h refers to the mesh size. We rewrite (12)

$$\frac{\rho_h^{n+1}\mathbf{v}_h^{n+1} - \rho_h^n\mathbf{u}_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes\mathbf{v}_h^{n+1}) + H(q_h'^{,n+1}, \rho_h^{n+1}) = \mathbf{0}, \tag{55}$$

where $\mathbf{v}_h^*$ is a suitable extrapolation of $\mathbf{v}_h^{n+1}$ to linearize the convective term and $H(q_h'^{,n+1}, \rho_h^{n+1}) = \nabla q_h'^{,n+1} + gz\nabla\rho_h^{n+1}$. Let us also write the space-discrete version of Eq. (20)

$$\mathbf{u}_h^{n+1} = (1-\chi)\mathbf{v}_h^{n+1} + \chi\overline{\mathbf{v}}_h^{n+1}, \tag{56}$$

and Eq. (39)

$$\frac{\rho_h^{n+1}}{\Delta t}(\overline{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) - \nabla\cdot(\overline{\mu}_h\nabla\overline{\mathbf{v}}_h^{n+1}) = \mathbf{0}, \tag{57}$$

where

$$\overline{\mu}_h = \rho_h^{n+1}\frac{\alpha^2}{\Delta t}a(\mathbf{v}_h^{n+1}). \tag{58}$$

We multiply (57) by χ, add it to (55), and make use of (56) to obtain:

$$\frac{\rho_h^{n+1}\mathbf{u}_h^{n+1} - \rho_h^n\mathbf{u}_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes\mathbf{v}_h^{n+1}) + H(q_h'^{,n+1}, \rho_h^{n+1}) - \chi\nabla\cdot(\overline{\mu}_h\nabla\overline{\mathbf{v}}_h^{n+1}) = \mathbf{0}.$$

Using (56) once more, we get:

$$\frac{\rho_h^{n+1}\mathbf{u}_h^{n+1} - \rho_h^n\mathbf{u}_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes\mathbf{u}_h^{n+1}) + H(q_h'^{,n+1}, \rho_h^{n+1}) + \chi\nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes(\mathbf{v}_h^{n+1} - \overline{\mathbf{v}}_h^{n+1})) - \chi\nabla\cdot(\overline{\mu}_h\nabla\overline{\mathbf{v}}_h^{n+1}) = \mathbf{0}.$$

which can be rewritten as

$$\frac{\rho_h^{n+1}\mathbf{u}_h^{n+1} - \rho_h^n\mathbf{u}_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes\mathbf{u}_h^{n+1}) + H(q_h'^{,n+1}, \rho_h^{n+1}) - \chi\nabla\cdot(\overline{\mu}_h\nabla\mathbf{u}_h^{n+1}) + \chi\nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes(\mathbf{v}_h^{n+1} - \overline{\mathbf{v}}_h^{n+1})) - \chi\nabla\cdot(\overline{\mu}_h\nabla(\overline{\mathbf{v}}_h^{n+1} - \mathbf{u}_h^{n+1})) = \mathbf{0}. \tag{59}$$

In (59), we have explicitly written a diffusive term involving only the end-of-step velocity $\mathbf{u}_h^{n+1}$, i.e., the last term in the first line. The last term at the left-hand side in (59) can be rewritten using (56) to get:

$$\frac{\rho_h^{n+1}\mathbf{u}_h^{n+1} - \rho_h^n\mathbf{u}_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes\mathbf{u}_h^{n+1}) + H(q_h'^{,n+1}, \rho_h^{n+1}) - \chi\nabla\cdot(\overline{\mu}_h\nabla\mathbf{u}_h^{n+1}) + \chi\nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes(\mathbf{v}_h^{n+1} - \overline{\mathbf{v}}_h^{n+1})) - \chi(1-\chi)\nabla\cdot(\overline{\mu}_h\nabla(\overline{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1})) = \mathbf{0}. \tag{60}$$

Eq. (60) shows that the end-of-step velocity $\mathbf{u}_h^{n+1}$ provided by the EFR algorithm solves a perturbed discrete momentum balance equation. The perturbation consists of three terms, all multiplied by χ: an extra convection term and two extra diffusion terms. As mentioned in Section 2.2, χ should be a multiple of Δt. Thus, as Δt tends to zero, the perturbation terms vanish and we recover the discrete momentum balance equation. In addition, we notice that when $\alpha^2/\Delta t$ tends to zero, the artificial viscosity $\overline{\mu}_h$ (58) vanishes and $\overline{\mathbf{v}}_h^{n+1}$ tends to $\mathbf{v}_h^{n+1}$. If α is a multiple of h (as it is typically the case), then Eq. (60) is consistent with Eq. (2) so long as the mesh size and time step go to zero at the same rate.

Combining in a similar fashion the space discrete version of Eq. (13):

$$\frac{\rho_h^{n+1} l_h^{n+1} - \rho_h^n h_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^* l_h^{n+1}) + G\left(\rho_h^{n+1}, K_{\mathbf{v},h}^{n+1}, q_h^{n+1}, \mathbf{v}_h^{n+1}\right) = 0,$$

$$G\left(\rho_h^{n+1}, K_{\mathbf{v},h}^{n+1}, q_h^{n+1}, \mathbf{v}_h^{n+1}\right) = \frac{\rho_h^{n+1} K_{\mathbf{v},h}^{n+1} - \rho_h^n K_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^* K_{\mathbf{v},h}^{n+1}) - \frac{q_h^{n+1} - p_h^n}{\Delta t} + \rho_h^{n+1} g \mathbf{v}_h^{n+1}\cdot\hat{\mathbf{k}},$$

with the discrete versions of Eq. (40):

$$\frac{\rho_h^{n+1}}{\Delta t}(\bar{l}_h^{n+1} - l_h^{n+1}) - \nabla\cdot(\overline{\mu}_h\nabla\bar{l}_h^{n+1}) = 0,$$

and Eq. (21):

$$h_h^{n+1} = (1-\xi) l_h^{n+1} + \xi \bar{l}_h^{n+1},$$

we obtain

$$\frac{\rho_h^{n+1} h_h^{n+1} - \rho_h^n h_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^* h_h^{n+1}) + G\left(\rho_h^{n+1}, K_{\mathbf{v},h}^{n+1}, q_h^{n+1}, \mathbf{v}_h^{n+1}\right) - \xi\nabla\cdot(\overline{\mu}_h\nabla h_h^{n+1}) + \xi\nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*(l_h^{n+1} - \bar{l}_h^{n+1})) - \xi(1-\xi)\nabla\cdot(\overline{\mu}_h\nabla(\bar{l}_h^{n+1} - l_h^{n+1})) = 0. \quad (61)$$

Eq. (61) is a perturbed discrete enthalpy balance equation, with the perturbation coming from an extra convection term and two extra diffusion terms. Like in the case of Eq. (60), the perturbation terms are multiplied by the relaxation parameter.

Remark 1. For $\chi = 1$, which corresponds to accepting the filtered velocity as the end-of-step velocity, Eq. (60) becomes:

$$\frac{\rho_h^{n+1}\mathbf{u}_h^{n+1} - \rho_h^n\mathbf{u}_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes\mathbf{u}_h^{n+1}) + H(q_h'^{,n+1}, \rho_h^{n+1}) - \nabla\cdot(\overline{\mu}_h\nabla\mathbf{u}_h^{n+1}) + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*\otimes(\mathbf{v}_h^{n+1} - \overline{\mathbf{v}}_h^{n+1})) = \mathbf{0},$$

where we clearly see that the artificial diffusion introduced by the EFR algorithm is $\overline{\mu}_h$ (58). Similarly, by setting $\xi = 1$ (i.e, we take the filtered enthalpy as the end-of-step enthalpy) Eq. (61) becomes:

$$\frac{\rho_h^{n+1} h_h^{n+1} - \rho_h^n h_h^n}{\Delta t} + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^* h_h^{n+1}) + G\left(\rho_h^{n+1}, K_{\mathbf{v},h}^{n+1}, q_h^{n+1}, \mathbf{v}_h^{n+1}\right) - \nabla\cdot(\overline{\mu}_h\nabla h_h^{n+1}) + \nabla\cdot(\rho_h^{n+1}\mathbf{v}_h^*(l_h^{n+1} - \bar{l}_h^{n+1})) = 0.$$

Remark 2. Eddy viscosity models are equivalent to introducing additional terms in Eq. (6) and (8) of the form

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) + \nabla p' + gz\nabla\rho - \nabla\cdot(2\mu_a\epsilon(\mathbf{u})) + \nabla\left(\frac{2}{3}\mu_a\nabla\cdot\mathbf{u}\right) = 0, \quad (62)$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla\cdot(\rho\mathbf{u}h) + \frac{\partial(\rho K)}{\partial t} + \nabla\cdot(\rho\mathbf{u}K) - \frac{\partial p}{\partial t} + \rho g\mathbf{u}\cdot\hat{\mathbf{k}} - \nabla\cdot\left(\frac{\mu_a}{Pr}\nabla h\right) = 0. \quad (63)$$

where μ_a is an artificial viscosity, $\epsilon(\mathbf{u}) = (\nabla\mathbf{u} + (\nabla\mathbf{u})^T)/2$ is the strain-rate tensor, and Pr is the Prandtl number, i.e., the dimensionless number defined as the ratio of momentum diffusivity to thermal diffusivity. The artificial viscosity μ_a is defined differently for the different LES models. Comparing (62)–(63) with (60)–(61) suggests choosing $\xi = \chi/Pr$.

Similar derivations to those presented in Remarks 1 and 2 apply for eddy viscosity models derived from variational multiscale methods. See, e.g., [49].

Remark 3. The Smagorinsky model sets μ_a in (62)–(63) as

$$\mu_a = \rho(C_s\delta)^2\sqrt{2\epsilon:\epsilon}, \quad C_s^2 = C_k\sqrt{\frac{C_k}{C_\epsilon}} \quad (64)$$

where δ is the filter width (typically comparable with the mesh size), and C_k and C_ϵ are model parameters. In order to obtain the same amount of artificial viscosity with the EFR algorithm and a_S (42), one needs $\alpha \approx C_s\delta\sqrt{\Delta t}\|\nabla\mathbf{v}_h^{n+1}\|_\infty$ at time t^{n+1}. In practice, one can easily calculate $C_s\delta\sqrt{\Delta t}$ while $\|\nabla\mathbf{v}_h^{n+1}\|_\infty$ would have to be guessed to tune α.

4. Numerical results

We consider two well-known benchmarks: the rising thermal bubble as presented in [50,51] and the density current [52,53]. Both test cases involve a perturbation of a neutrally stratified atmosphere with uniform background potential temperature over a flat terrain and the boundaries are treated as if the problem were inviscid (i.e., free-slip boundary conditions are imposed). So, these are not boundary layer flow problems. We presents our results for the rising thermal bubble and the density current in Sections 4.1 and 4.2, respectively, and compare them with other numerical data available in the literature since there is no exact solution for these benchmarks.

4.1. Rising thermal bubble

In computational domain $\Omega = [0, 5000]\times[0, 10\,000]$ m^2, a neutrally stratified atmosphere with uniform background potential temperature θ_0=300 K is perturbed by a circular bubble of warmer air. The initial temperature field is

$$\theta^0 = 300 + 2\left[1 - \frac{r}{r_0}\right] \text{ if } r \le r_0 = 2000 \text{ m}, \quad \theta^0 = 300 \text{ otherwise}, \quad (65)$$

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 [50,54]. The initial density is given by

$$\rho^0 = \frac{p_g}{R\theta_0}\left(\frac{p}{p_g}\right)^{c_v/c_p}, \quad p = p_g\left(1 - \frac{gz}{c_p\theta^0}\right)^{c_p/R}, \quad (66)$$

with $c_p = R + c_v$, c_v = 715.5 J/(Kg K), R = 287 J/(Kg K). The initial velocity field is zero everywhere. Finally, the initial specific enthalpy is given by:

$$h^0 = c_p\theta^0\left(\frac{p}{p_g}\right)^{\frac{R}{c_p}}. \quad (67)$$

We let the bubble evolve in the time interval of interest is $(0, 1020]$ s. Impenetrable, free-slip boundary conditions are imposed on all walls.

We consider five different meshes with uniform resolution $h = \Delta x = \Delta z = 125, 62.5, 31.25, 15.625$ m. The time step is set to $\Delta t = 0.1$ s for all the simulations. In all the cases, we set $\chi = \xi = 1$ so that the artificial diffusion introduced by the EFR algorithm can easily be calculated (see Remark 1).

We start with the linear filter, i.e., we take a_L (41) as indicator function, because it allows us to make a direct comparison with the results obtained by setting $\mu_a = 15$ and $Pr = 1$ in (62)–(63) [44,50]. We note that both of these are ad-hoc values chosen by the authors of [50] to stabilize the numerical simulations. It is not unusual in benchmarks to set $Pr = 1$ although the air Prandtl number is about 0.71 at 20 °C (see, e.g., [55]). Other authors have chosen other arbitrary values, like $Pr = 0.1$ in [56]. For a qualitative analysis of the results for the rising thermal bubble as Pr varies we refer the reader to [11]. To introduce the same amount of artificial viscosity with the EFR algorithm and a_L, we use (58) to get $\alpha \approx \sqrt{\mu_a\Delta t/\rho^{n+1}}$ at time t^{n+1}. For simplicity, we keep

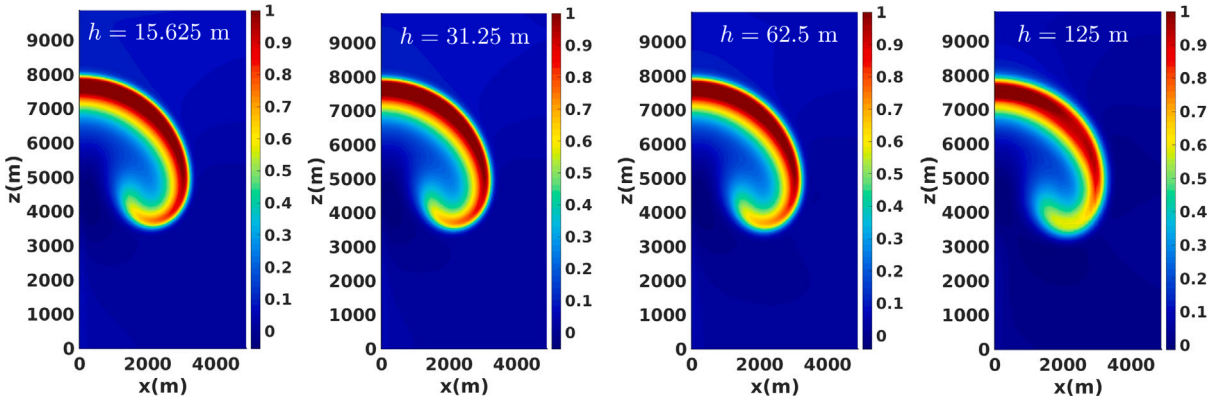

Fig. 1. Rising thermal bubble, a_L, $\alpha = 1.9$ m: perturbation of potential temperature at $t = 1020$ s computed with four different meshes. The mesh size, specified in each panel, is increasing from left to right.

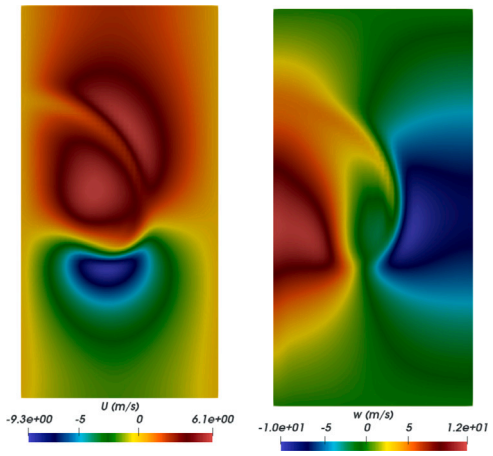

Fig. 2. Rising thermal bubble, a_L, $\alpha = 1.9$ m: contour plots of the horizontal velocity component u (left) and the vertical velocity component w (right) at $t = 1020$ s computed with mesh $h = 125$ m.

α constant in space and time and set it to 1.9, which is obtained by taking the minimum value of density. Fig. 1 shows the perturbation of potential temperature θ' at $t = 1020$ s computed by the EFR algorithm (with a_L and $\alpha = 1.9$) and all the meshes under consideration. From Fig. 1, we observe no visible change in the computed θ' when the mesh is refined past $h = 62.5$ m. In order to facilitate the comparison of the panels in Fig. 1 with data in the literature [11,44,50,54], we have forced the colorbar to range from 0 to 1. Overall, these results are in very good qualitative agreement with the corresponding figures in [11,44,50,54].

We obtain good qualitative agreement with data in the literature (e.g., Fig. 7 in [50]) also for Fig. 2, which displays velocity components u and w at $t = 1020$ s computed by the EFR algorithm (with a_L and $\alpha = 1.9$) with mesh $h = 125$ m.

Fig. 3 reports a more quantitative comparison. It compares the time evolution of the maximum perturbation of potential temperature θ'_{max} and maximum vertical component of the velocity w_{max} computed by the EFR algorithm (with a_L and $\alpha = 1.9$) against the corresponding results from [50]. We see that the evolution of θ'_{max} computed with meshes $h = 125$ m is affected by spurious oscillations. Oscillations of small amplitude affect also the θ'_{max} computed with mesh $h = 62.5$ m, but they disappear with finer meshes. Since θ'_{max} and w_{max} computed with meshes $h = 31.25$ m and $h = 15.625$ m are practically overlapped, we chose not to refine the mesh further. The "converged" w_{max} overlaps with the reference value till about $t = 800$ s, which is a remarkable improvement over our previous results in [44]. The "converged" θ'_{max} is also closer to the results from [50] than in [44], however there is still some distance between the two curves.

Table 1
Rising thermal bubble, a_L, $\alpha = 1.9$: minimum and maximum vertical velocity w and potential temperature θ' at $t = 1020$ s compared with the values extracted from the figures in [50].

Type	h (m)	w_{min} (m/s)	w_{max} (m/s)	θ'_{min} (K)	θ'_{max} (K)
Ref. [50]	125	−7.75	13.95	−0.013	1.4
$a_L, \alpha = 1.9$	125	−10.35	12.01	−0.012	1.23
$a_L, \alpha = 1.9$	62.5	−10.54	12.16	−0.041	1.24
$a_L, \alpha = 1.9$	31.25	−10.61	12.21	−0.050	1.22
$a_L, \alpha = 1.9$	15.625	−10.63	12.28	−0.052	1.22

Table 1 reports the extrema for the vertical velocity w and potential temperature perturbation θ' at $t = 1020$ s obtained with the EFR algorithm (with a_L and $\alpha = 1.9$), together with the values extracted from the figures in [50]. This tables confirms the findings from Fig. 3.

Next, we consider indicator functions a_S and a_D and focus on the two intermediate meshes ($h = 31.25$ m and $h = 62.5$ m). We set the value of α using $C_s = 0.094$ [44] and Remark 3, which suggests an order of magnitude for α rather than a strict value. We take $\alpha = 3$ m for mesh $h = 31.25$ m. Since Remark 3 suggests a linear dependence of α on the mesh size, we take $\alpha = 6$ m for mesh $h = 62.5$ m. We note that these values would apply only for a_S but we will use them for a_D too in order to show the differences in the solutions obtained with the two indicator functions. Fig. 4 shows the spatial distribution of θ' and the indicator function at $t = 1020$ s computed with the EFR algorithm and the nonlinear filters. Note that with the nonlinear filters we can capture a larger amount of vortical structures than with the linear filter (compare Fig. 4 with Fig. 1). The results computed with a_S and mesh $h = 31.25$ m (Fig. 4, first panel on the top row) agree very well with those obtained with the Smagorinsky model in [44] (Fig. 5, left panel). On a given mesh, the Rayleigh–Taylor instability at the edge of the bubble is more developed when using a_D instead of a_S, which indicates that a_D introduces less artificial viscosity than a_S. Recall that the artificial viscosity introduced by the EFR algorithm (58) is proportional to the indicator function. Indeed, the plots on the bottom row of Fig. 4 show that a_S at $t = 1020$ s has larger values over wider regions than a_D. This means that indicator function a_D is more selective in identifying the regions of the domain where diffusion is needed.

Table 2 reports the extrema for the vertical velocity w and potential temperature perturbation θ' at $t = 1020$ s obtained with the EFR algorithm, together with the values from [44] for the Smagorinsky model. The data in Table 2 confirm our observation from Fig. 4 about a_S vs a_D. Indeed, we see that larger extreme values are found with the EFR algorithm and a_D. In addition, we see that the Smagorinsky model

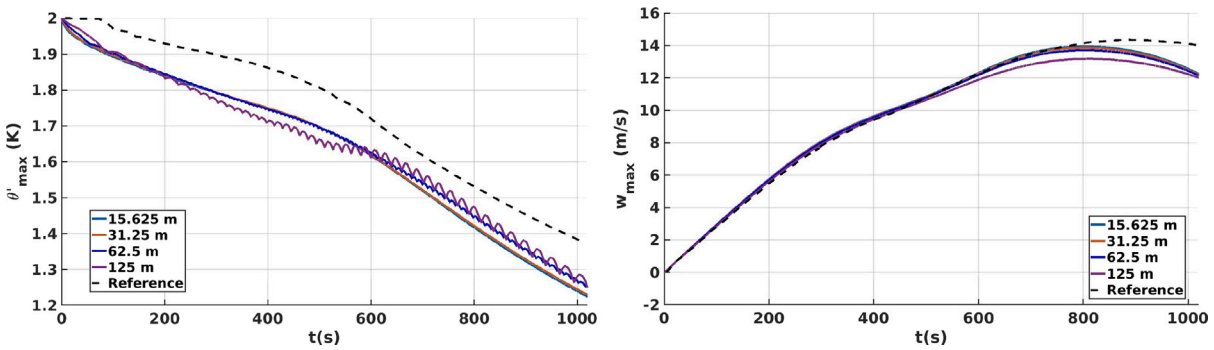

Fig. 3. Rising thermal bubble, a_L, $\alpha = 1.9$: time evolution of the maximum perturbation of potential temperature θ'_{max} (left) and the maximum vertical component of the velocity w_{max} (right) computed with all the meshes under consideration. The reference values are taken from [50] and refer to resolution 125 m.

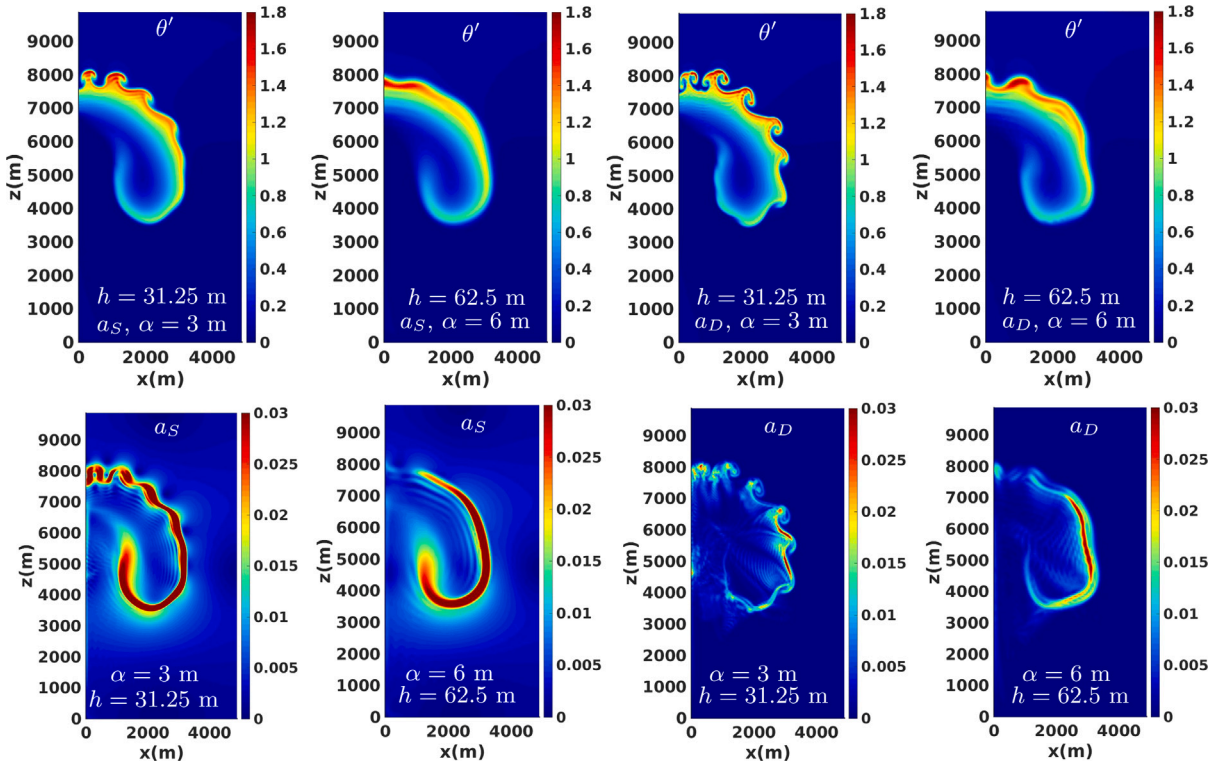

Fig. 4. Rising thermal bubble: perturbation of potential temperature (top row) and corresponding indicator function (bottom row) at $t = 1020$ s computed with the EFR and a_S (first two columns) and a_D (last two columns) for different mesh sizes.

Table 2
Rising thermal bubble: minimum and maximum vertical velocity w and potential temperature θ' at $t = 1020$ s computed by the EFR algorithm with different meshes, indicator functions, and values of α. For comparison, the table reports the values from [44] obtained with the Smagorinsky model.

Model	h (m)	α (m)	w_{min} (m/s)	w_{max} (m/s)	θ'_{min} (K)	θ'_{max} (K)
EFR, a_D	31.25	3	−13.39	15.59	−0.18	1.88
EFR, a_S	31.25	3	−11.60	15.29	−0.13	1.75
Smagorinsky [44]	31.25	–	−11.54	15.04	−0.072	1.89
EFR, a_D	62.5	6	−11.3	14.88	−0.12	1.78
EFR, a_S	62.5	6	−10.76	13.38	−0.14	1.72

from [44] gives smaller extreme values than EFR algorithm with a_S, which seems to be less diffusive.

We conclude this section by highlighting the important role played by the filtering radius. Fig. 5 shows θ' at $t = 1020$ s computed by the EFR algorithm with a_S and different values of α for mesh $h = 31.25$ m. Although the three values of α are all of the same order of magnitude, we see a big difference in the solution. This sensitivity to α can be mitigated by choosing $\chi \neq 1$ (see, e.g., [17]) and $\xi \neq 1$.

4.2. Density current

The computational domain in the xz-plane is $\Omega = [0, 25\,600] \times [0, 6400]$ m^2 and the time interval of interest is $(0, 900]$ s. Impenetrable, free-slip boundary conditions are imposed on all the walls. The initial density is given by (66) with initial potential temperature:

$$\theta^0 = 300 - \frac{15}{2}[1 + \cos(\pi r)] \text{ if } r \leq 1, \quad \theta^0 = 300 \text{ otherwise}, \tag{68}$$

where $r = \sqrt{\left(\frac{x-x_c}{x_r}\right)^2 + \left(\frac{z-z_c}{z_r}\right)^2}$, with $(x_r, z_r) = (4000, 2000)$ m and $(x_c, z_c) = (0, 3000)$ m. The initial bubble in this test is of cold air. The initial velocity field is zero everywhere and the initial specific enthalpy is given by (67).

We consider uniform, orthogonal meshes with mesh sizes $h = \Delta x = \Delta z = 200, 100, 50, 25$ m. The time step is set to $\Delta t = 0.1$ s. Just like in the case of the warm bubble, we set $\chi = \xi = 1$ in the EFR algorithm.

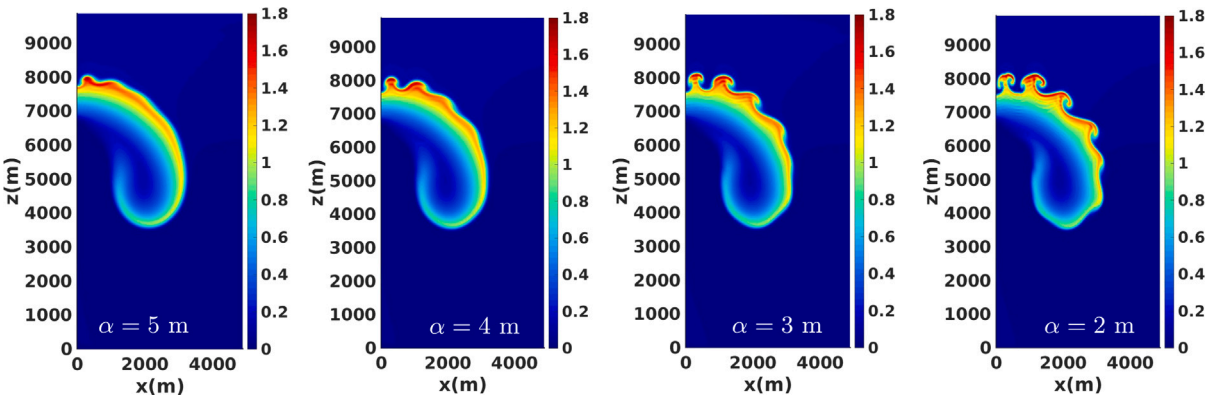

Fig. 5. Rising thermal bubble, a_S: perturbation of potential temperature at $t = 1020$ s computed by the EFR algorithm with mesh $h = 31.25$ m and (from left to right) $\alpha = 5, 4, 3, 2$ m.

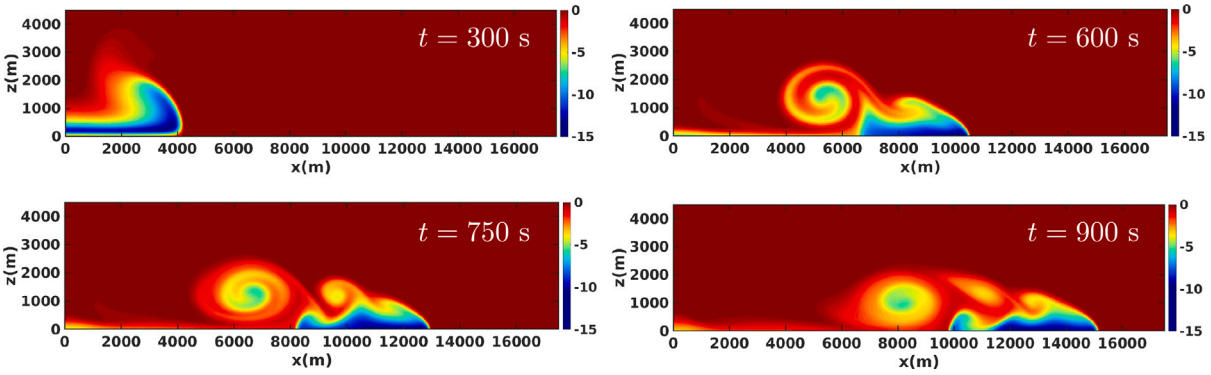

Fig. 6. Density current, a_L, $\alpha = 2.7$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 25$ m.

We start again with the linear filter, i.e., we take a_L (41) as indicator function, because it allows us to make a direct comparison with the results obtained by setting $\mu_a = 75$ and $Pr = 1$ in (62)–(63) [44,53]. To introduce the same amount of artificial viscosity with the EFR algorithm and a_L, we use (58) and get $\alpha = 2.7$ m by using the minimum density in the computational domain. Fig. 6 shows θ' computed with this value of α and mesh $h = 25$ m (i.e., the finest mesh among those considered) at $t = 300, 600, 750, 900$ s. We observe very good agreement with the results reported in Fig. 1 of [53], which were obtained with the same resolution. In order to understand the behavior of the liner filter as the mesh size is varied, we report θ' computed at $t = 900$ s with all the meshes mentioned above in Fig. 7. We observe that the EFR algorithm with a_L and $\alpha = 2.7$ m does not introduce sufficient artificial diffusion to stabilize the solution with the coarsest mesh we consider (i.e., $h = 200$ m). For all the other meshes though, we see the emergence of a clear three-rotor structure when the mesh is refined. Also the results in Fig. 7 are in very good agreement with those reported in the literature. See, e.g., [11,44,50,53,57,58].

For a quantitative comparison, we consider the potential temperature perturbation θ' at $t = 900$ s along the horizontal direction at height $z = 1200$ m. Fig. 8 displays a comparison between the results given by the EFR model with a_L and $\alpha = 2.7$ for meshes $h = 100, 50, 25$ m. We see that the curves associated to meshes $h = 50, 25$ m are practically superimposed. In Fig. 8, we report also the results from [57], which were obtained by setting a constant artificial viscosity (i.e., $\mu_a = 75$) and using a spectral element method. Such results are labeled as "Reference" and refer to resolution 25 m. We observe that our results are slightly out of phase with respect to the reference data. Each dip in θ' in Fig. 8 corresponds to a recirculation in Fig. 7, top three panels. So, from Fig. 8 we learn that in our simulations the front is faster than in the simulations from [57]. For this reason, Table 3 reports the front location (defined as the location on the ground where $\theta' = -1$ K) at

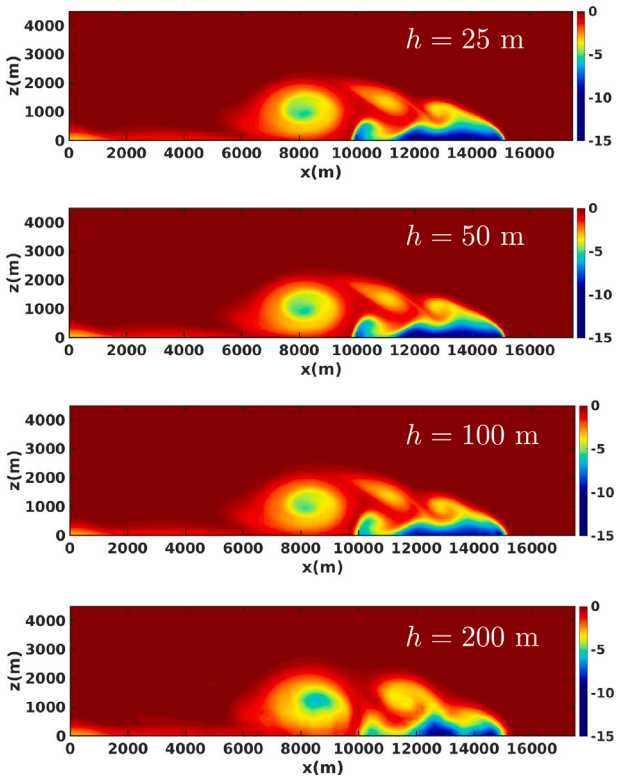

Fig. 7. Density current, a_L, $\alpha = 2.7$: potential temperature fluctuation θ' computed at $t = 900$ s with meshes $h = 25, 50, 100, 200, 400$ m. The mesh size is increasing from top to bottom.

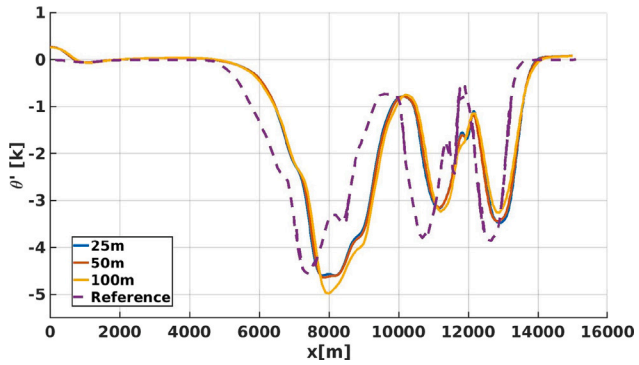

Fig. 8. Density current, a_L, $\alpha = 2.7$: potential temperature perturbation θ' at $t = 900$ s along the horizontal direction at a height of $z = 1200$ m for meshes $h = 100, 50, 25$ m compared against data from [57] (denoted as "Reference" and referred to resolution 25 m).

Table 3
Density current, a_L, $\alpha = 2.7$: our results for the front location at $t = 900$ s obtained with different meshes compared against results reported in [53]. For Ref. [53], we provide the range of mesh sizes and front location values obtained with different methods.

Method	h (m)	Front location (m)
EFR, a_L	25	15170
EFR, a_L	50	15190
EFR, a_L	100	15210
Ref. [53]	(25, 200)	(14533,17070)

$t = 900$ s obtained with EFR and a_L and compares it with the data in Table 4 of [53]. The data in [53] refer to the model with constant artificial viscosity (i.e., $\mu_a = 75$) and 14 different numerical approaches. We note that our results fall well within the values from [53]. Hence, we attribute the difference with the Reference in Fig. 8 to the use of different numerical methods.

Next, we focus on the EFR algorithm with indicator function a_S. We restrict our attention to meshes $h = 25, 50$ m and, following [11], we further refine the mesh to get $h = 12.5$ m. Like in Section 4.1, we set the value of α using the rule of thumb in Remark 3 and $C_s = 0.454$ [44]. We take $\alpha = 4$ m for mesh $h = 12.5$ m because smaller values would lead to instabilities. Then, we use the linear dependence of α on the mesh size to set $\alpha = 8$ m for mesh $h = 25$ m and $\alpha = 16$ m for mesh $h = 50$ m. Fig. 9 (left), 9 (right), and 10 (left) show the time evolution of the potential temperature fluctuation computed with meshes $h = 12.5, 25, 50$ m, respectively. As expected, more vortical structures appear when we reduce the mesh size. The EFR algorithm with a_S produces very similar results to a standard implementation of the Smagorinsky model with mesh $h = 25$ m: compare Fig. 9 (right) with Fig. 10 in [44]. However, our method does a better job at stabilizing the larger eddies with mesh $h = 12.5$ m: Fig. 9 (left) with Fig. 9 in [44]. For mesh $h = 50$ m, Fig. 10 (left) indicates that the $\alpha = 16$ leads to overdiffusion. In fact, it provides a smoothed out solution even when compared to the linear filter (see Fig. 7, second panel from the top). A less dissipative solution can be found by lowering the values of α. Fig. 10 (right) shows θ' computed with mesh $h = 50$ m and $\alpha = 11$. We this new value of α the solution obtained with mesh $h = 50$ m looks similar to the solution given by mesh $h = 25$ m. This is confirmed by Table 4, which reports the front locations at $t = 900$ s obtained with EFR and a_S. The location computed with mesh $h = 25$ m and $\alpha = 8$ is very close to the location given by mesh $h = 50$ m and $\alpha = 11$. This is exactly what expected from the EFR algorithm: with a proper tuning of α, one can use coarser meshes without compromising accuracy. In [17,21], using heuristic arguments we provided guidelines to set α and χ for the EFR algorithm applied to the incompressible Navier–Stokes equations. Such arguments cannot be easily extended to the Euler equations, so further investigation is need to understand how to tune α and the relaxation parameters with no user intervention. This will be the object of a follow-up paper.

Table 4
Density current, a_S: front location at $t = 900$ s obtained with the EFR algorithm and different meshes. Our results are compared against results from [11,53]. For Ref. [11], we report only the front location computed with the finest resolution. For Ref. [53], we provide the range of mesh sizes and front location values obtained with different methods.

Method	h (m)	α (m)	Front location (m)
Ref. [11]	12.5	–	15056
EFR, a_S	12.5	4	15550
EFR, a_S	25	8	15300
EFR, a_S	50	11	15220
EFR, a_S	50	16	15090
Ref. [53]	(25, 200)	–	(14533,17070)

Now, let us turn our attention to a_D. In Section 4.1, we have shown that a_D is a more selective indicator function than a_S. Thus, we slightly increase the values of α used for a_S since at the moment we do not have a better criterion to set α for a_D. We take $\alpha = 5$ for mesh $h = 12.5$ m, $\alpha = 10$ for mesh $h = 25$ m, $\alpha = 12$ for mesh $h = 50$ m. Before showing the solutions obtained with a_D and these values of α, in Fig. 11 we compare the time evolution of the space-averaged artificial viscosity

$$\mu_{av} = \frac{1}{\Omega}\int_{\Omega} \overline{\mu}_h d\Omega \tag{69}$$

obtained with a_S and a_D for meshes $h = 12.5, 25, 50$ m. We recall that $\overline{\mu}_h$ is defined in (58). From Fig. 11, we see that a_S with $\alpha = 4$ and a_D with $\alpha = 5$ introduce roughly the same amount of space-averaged artificial viscosity for most of the time interval under consideration in the case of mesh $h = 12.5$ m. The main difference for this mesh is that a_D introduces almost no artificial diffusion till about 100 s and then ramps it up faster than a_S. A rather fast ramp is observed for a_D also in the case of meshes $h = 25, 50$ m: μ_{av} remains small while the cold perturbation falls due to negative buoyancy and it increases as the cold front propagates horizontally. For mesh $h = 25$ m, μ_{av} given by a_S with $\alpha = 8$ grows almost linearly till about 300 s and then around 600 s it flattens. As already evident from Fig. 10, a_S with $\alpha = 16$ introduces too much artificial viscosity for mesh $h = 50$ m. This can be fixed by decreasing the value to $\alpha = 11$, which introduces a similar amount of μ_{av} as $\alpha = 8$ does for mesh $h = 25$ m.

Fig. 12 (left), 12 (right), and 13 display the time evolution of the potential temperature fluctuation computed with a_D and the chosen values of α for meshes $h = 12.5, 25, 50$ m, respectively. When compared to the respective counterparts obtained with a_S (namely Fig. 9 (left), 9 (right), and 10 (right)), all the observations made about Fig. 11 are confirmed: the solutions obtained with the finer mesh are initially comparable early and then some differences are observed for the larger recirculations, the solutions for the intermediate mesh are remarkably similar. While Fig. 11 suggests that EFR with a_D and $\alpha = 12$ is more diffusive (in average) than with a_S and $\alpha = 11$, the respective solutions in Fig. 13 and Fig. 10 (right) are surprisingly similar.

Table 5 reports the front locations at $t = 900$ s obtained with EFR and a_D for the three meshes under consideration. The three locations are within about 400 m of each other, with the front becoming faster as the mesh is refined. This was the case also for a_S (see Table 4). The opposite trend is observed for a_L (see Table 3), i.e., the front slows down as the mesh is refined, although the locations are only roughly 40 m apart. In any case, our results fall well within the results from [53] and are close to the results from [11].

Next, in Fig. 14 we report with a comparison of indicator functions a_S and a_D for the simulations in Fig. 9 (left) and 12 (left). We see that, while both indicator functions have larger values (red to yellow shades) at the bottom of the largest recirculation, at a given time a_S

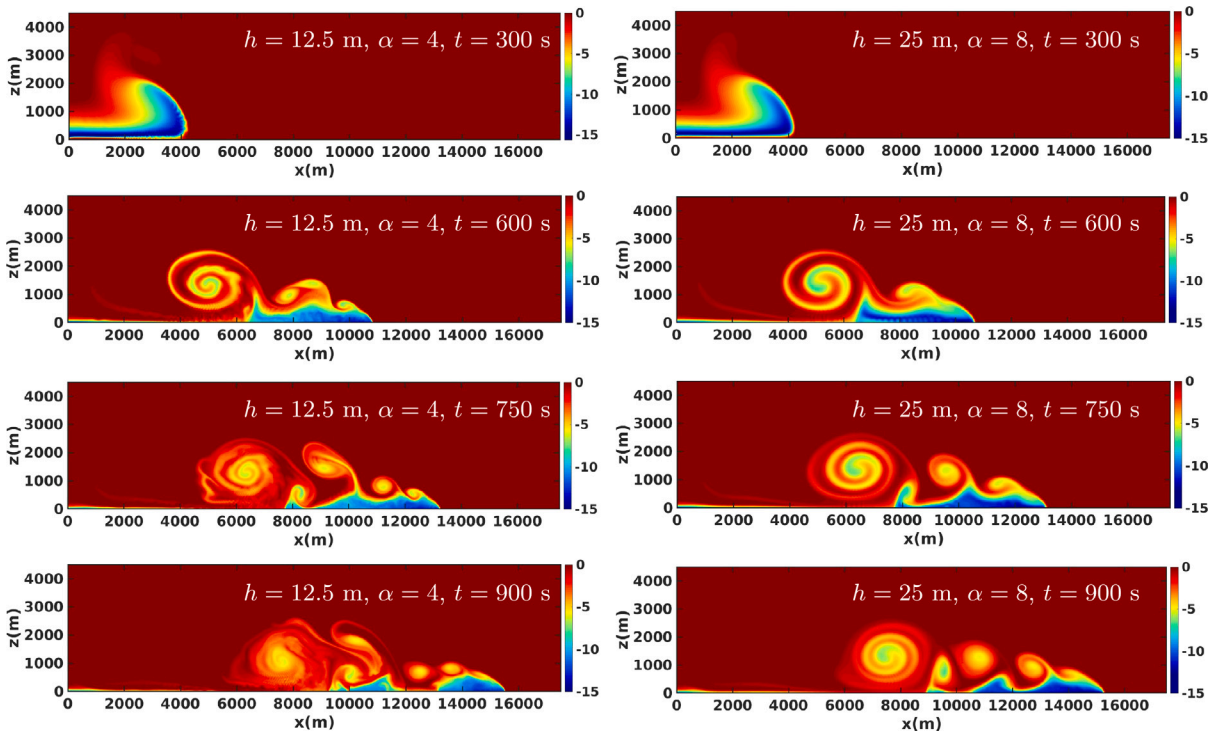

Fig. 9. Density current, a_S: time evolution of potential temperature fluctuation θ' computed with mesh $h = 12.5$ m and $\alpha = 4$ (left) and with mesh $h = 25$ m and $\alpha = 8$ (right).

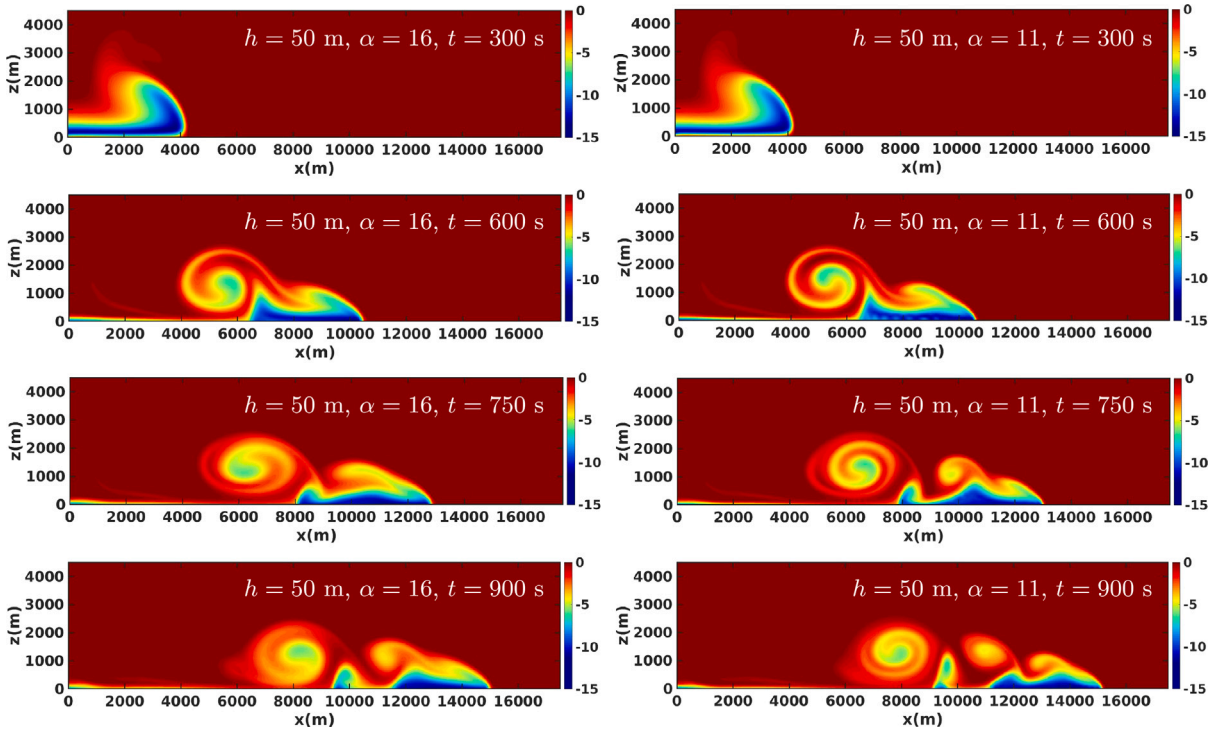

Fig. 10. Density current, a_S: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m and two values of α: $\alpha = 16$ (left) and $\alpha = 11$ (right).

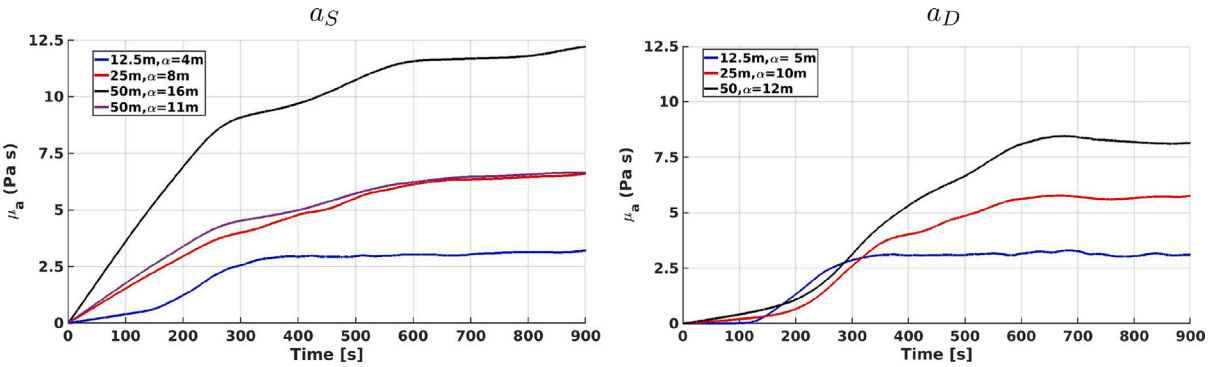

Fig. 11. Density current: time evolution of the average eddy viscosity (69) for a_S (left) and a_D (right) with meshes $h = 12.5, 25, 50$ m.

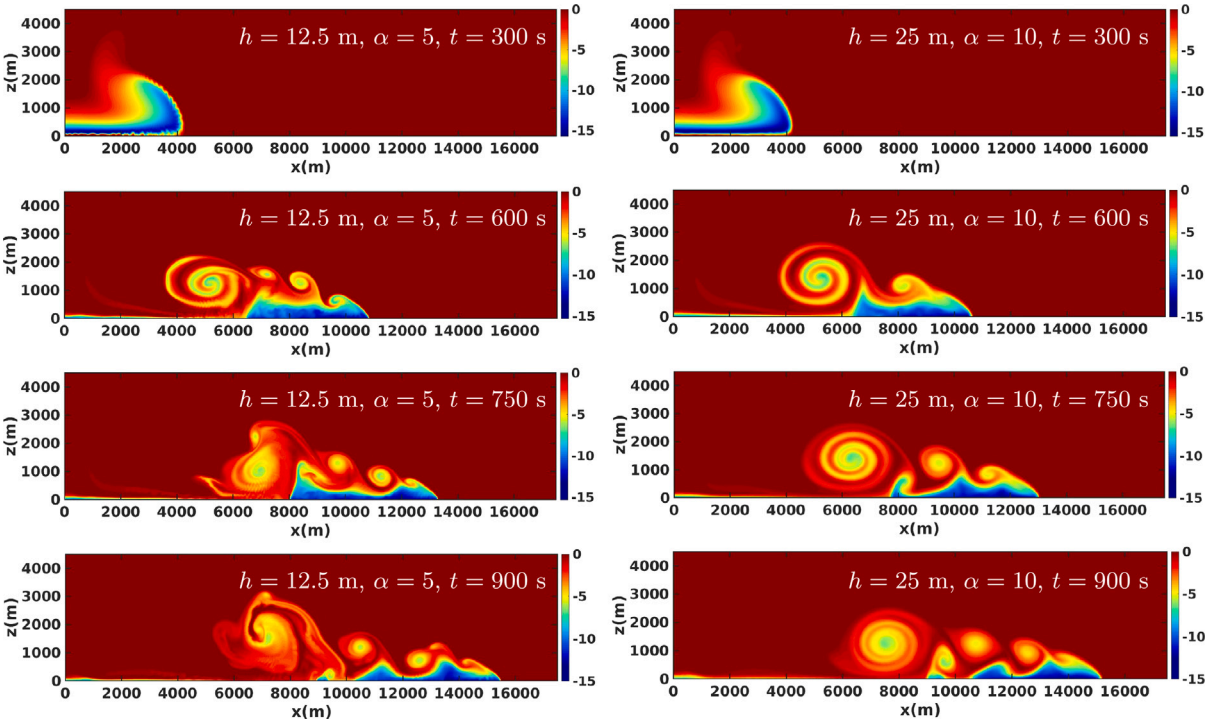

Fig. 12. Density current, a_D: time evolution of potential temperature fluctuation θ' computed with mesh $h = 12.5$ m and $\alpha = 5$ (left) and with mesh $h = 25$ m and $\alpha = 10$ (right).

Table 5
Density current, a_D: front location at $t = 900$ s obtained with the EFR algorithm and different meshes. Our results are compared against results from [11,53]. For Ref. [53], we provide the range of mesh sizes and front location values obtained with different methods. For Ref. [11], we report only the front location computed with the finest resolution.

Method	h (m)	α (m)	Front location (m)
Ref. [11]	12.5	–	15056
EFR, a_D	12.5	5	15560
EFR, a_D	25	10	15215
EFR, a_D	50	12	15120
Ref. [53]	(25, 200)	–	(14533,17070)

has larger regions of intermediate values (light blue shade) than a_D. This is due to the fact that a_S is a less selective indicator function, as mentioned earlier. This is more evident on mesh $h = 50$ m: see Fig. 15 for the plots of the indicator function for a_S and a_D for the simulations in Fig. 10 (left) and 13. The higher selectivity of a_D results in much smaller regions of high and intermediate values (red to green shades). Finally, we note that the maximum magnitude of the indicator function is higher for the finer mesh ($h = 25$ m), which gives rise to more localized and higher peaks than the coarser mesh ($h = 50$ m). This is in line with what already observed in [17].

We conclude with a comment on the computational cost. Table 6 reports the computational time taken by the evolve step and filter step per time step and total simulation time for the EFR algorithm with indicator functions a_L, a_S, and a_D and the specified values of α for meshes $h = 50, 25$ m. All the simulations were run on a common laptop (AMD Ryzen 7 5700U, 16 GB RAM). As expected, the total

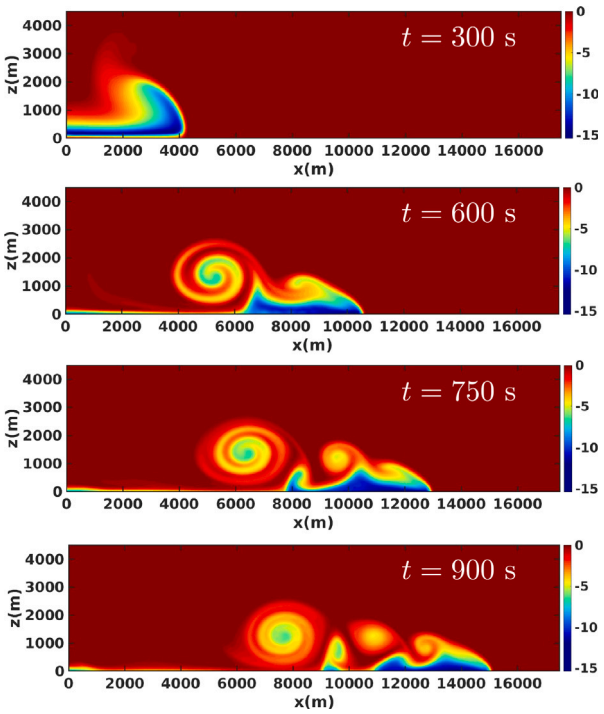

Fig. 13. Density current, a_D, $\alpha = 12$: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m.

computational cost increases when switching from the linear filter to nonlinear filters, with the deconvolution-based indicator function being the most expensive. In fact, while a_S requires a simple post-processing of the velocity field, a_D in (45) requires one application of the linear Helmholtz filter. Despite this increased cost, the simulation with mesh $h = 25$ m and a_D takes a little less than 1 h and 10 min, which means that the solver is rather efficient. We note that when using a_D, the filter step takes about half of the time needed for the evolve step. This might seem counter-intuitive given the relative complexity of the problems solved at the two steps. However, it can be explained with the solver choices. During the evolve step, the majority of the computational cost is spent to solve the equation for p' (recall we adopt a splitting scheme detailed in [44]) 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 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.

5. Concluding remarks

In this paper, we presented a filter stabilization technique for the mildly compressible Euler equations that is realized through a three step algorithm called Evolve-Filter-Relax (EFR). While filter stabilization and the EFR algorithm have been widely investigated for the incompressible Navier–Stokes equations, this work is one of the few papers that applies them to the Euler equations. We showed that the EFR algorithm is equivalent to an eddy viscosity model in LES and we considered three indicator functions to tune the amount and location of eddy viscosity: a constant indicator function, an indicator function proportional to the velocity gradient norm that recovers a Smagorinsky-like model, and indicator function based on approximate deconvolution operators. The first indicator function corresponds to a linear filter, which is known to be overdiffusive, while the other two lead to nonlinear filters.

Table 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 indicator functions a_L, a_S, and a_D and the specified values of α for meshes $h = 50, 25$ m.

Model	h (m)	α (m)	Evolve (s)	Filter (s)	Total (s)
EFR, a_L	25	2.7	0.3	0.070	3492
EFR, a_S	25	8	0.3	0.108	3880
EFR, a_D	25	10	0.3	0.159	4163
EFR, a_L	50	2.7	0.06	0.015	707
EFR, a_S	50	11	0.06	0.023	772
EFR, a_D	50	12	0.06	0.029	790

We tested our EFR approach with two well-known benchmarks for atmospheric flow: the rising thermal bubble and the density current. For both benchmarks, we showed that the linear filter provides results in excellent agreement with data in the literature obtained by setting an ad-hoc eddy viscosity. We showed that with the nonlinear filters we can capture a larger amount of vortical structures in the flows, which is expected and in line with other published data obtained with LES models. Of the three indicator functions under consideration, the deconvolution-based indicator function was shown to be more selective in identifying the regions of the domain where artificial diffusion is needed. Finally, we commented about the computational efficiency of our approach, highlighting that the filter step is computationally cheap with respect to the evolve step.

More work is needed to improve the EFR algorithm proposed in this paper. A parametric study of the relaxation parameters χ and ξ (set equal to 1 in this work) would inform us on their "optimal" value to improve the accuracy and would help to mitigate the sensitivity to the filtering radius α. In addition, the role of the order of the deconvolution N needs to be investigated.

CRediT authorship contribution statement

Nicola Clinco: Numerical test, Programming. **Michele Girfoglio:** Writing – original draft. **Annalisa Quaini:** Revision, Test planning. **Gianluigi Rozza:** Funding, Revision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We acknowledge the support provided by the European Research Council Executive Agency by the Consolidator Grant project AROMA-CFD "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" - GA 681447, H2020-ERC CoG 2015 AROMA-CFD, PI G. Rozza, and INdAM-GNCS 2019–2020 projects. This work was also partially supported by US National Science Foundation through grant DMS-1953535 (PI A. Quaini). A. Quaini acknowledges support from the Radcliffe Institute for Advanced Study at Harvard University, USA where she has been the 2021–2022 William and Flora Hewlett Foundation Fellow.

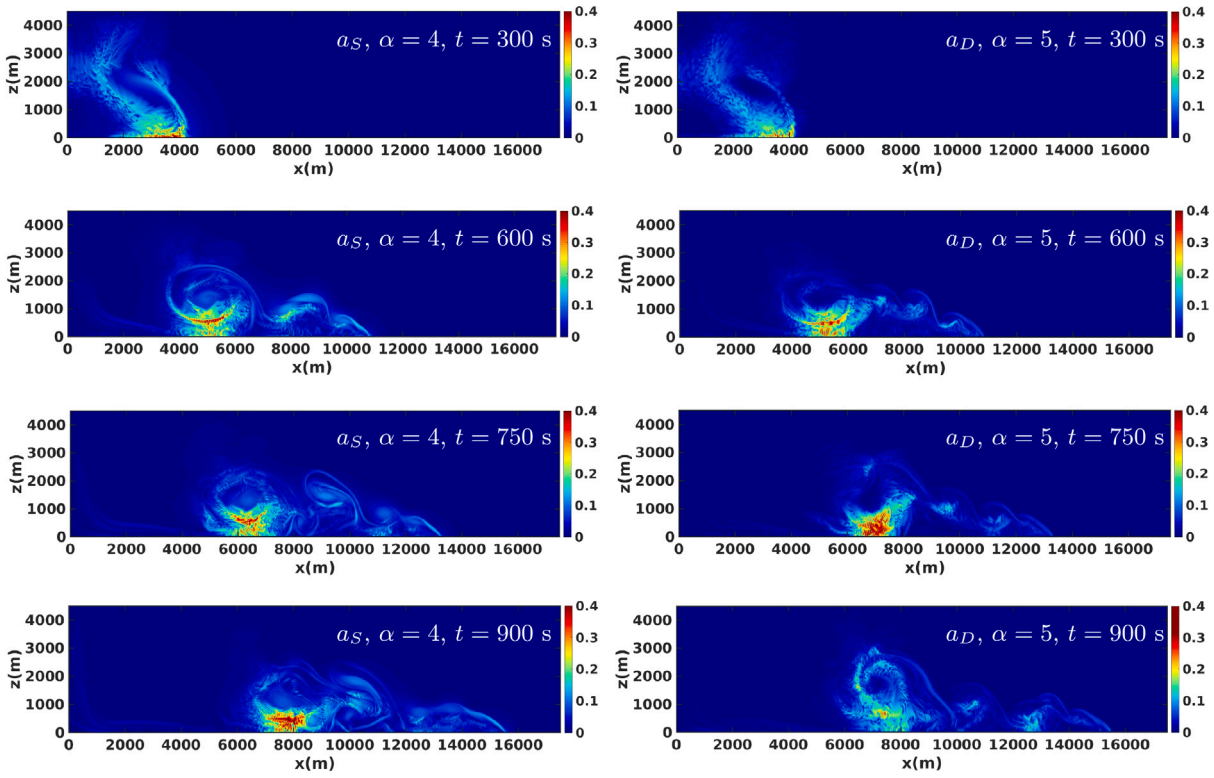

Fig. 14. Density current, mesh $h = 12.5$ m: time evolution of a_S with $\alpha = 4$ m (left) and a_D with $\alpha = 5$ m (right).

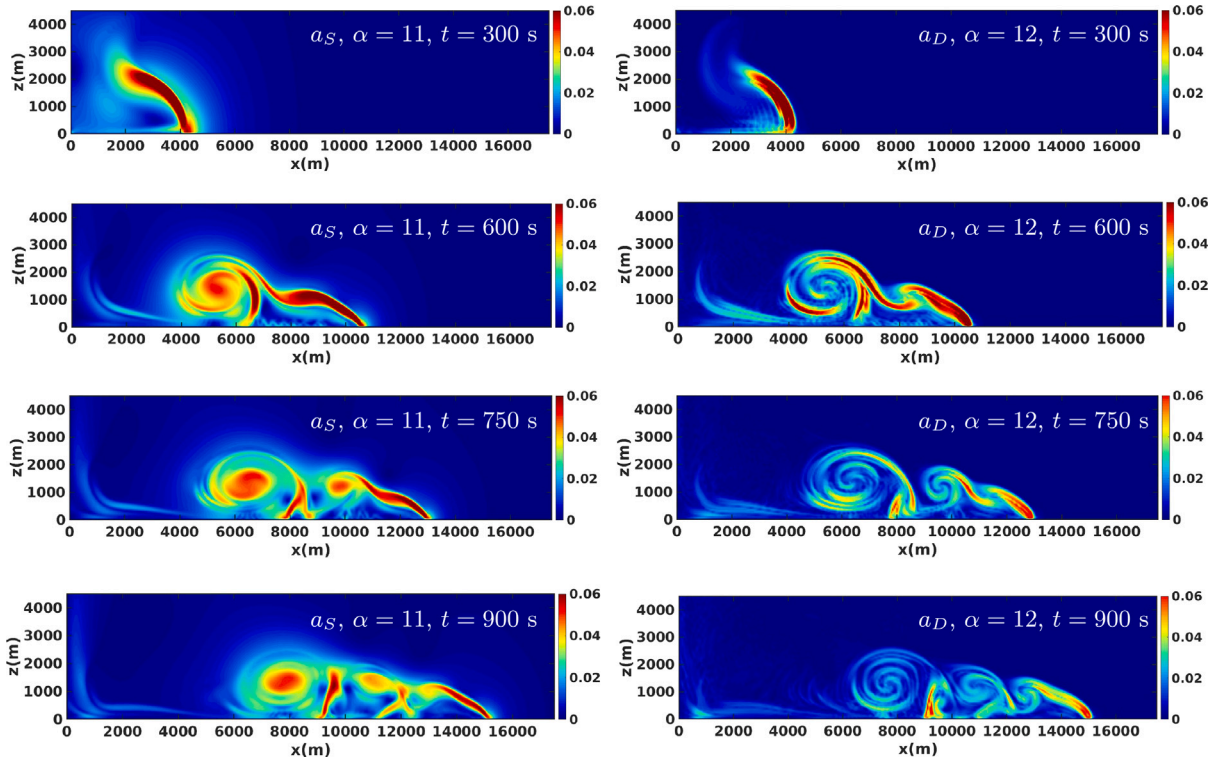

Fig. 15. Density current, mesh $h = 50$ m: time evolution of a_S with $\alpha = 11$ m (left) and a_D with $\alpha = 12$ m (right).

References

[1] Smagorinsky J. General circulation experiments with the primitive equations: I. The basic experiement. Mon Wea Rev 1963;91:99–164.
[2] Abgrall R. Toward the ultimate conservative scheme: Following the quest. J Comput Phys 2001;167(2):277–315.
[3] Hughes T. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Comput Methods Appl Mech Engrg 1995;127:387–401.
[4] Kloeckner A, Warburton T, Hesthaven JS. Viscous shock capturing in a time-explicit discontinuous Galerkin method. Math Model Nat Phenom 2011;6(3):57–83.
[5] Rispoli F, Saavedra R. A stabilized finite element method based on SGS models for compressible flows. Comp Meth Appl Mech Engrg 2006;196:652–64.
[6] Persson P-O, Peraire J. Sub-cell shock capturing for discontinuous Galerkin methods. In: Proc. of the 44th AIAA aerospace sciences meeting and exhibit, AIAA-2006-112. 2006.
[7] Guermond JL, Pasqueti R, Popov B. Entropy viscosity method for nonlinear conservation laws. J Comput Phys 2011;230(11):4248–67.
[8] Guermond JL, Pasquetti R. Entropy-based nonlinear viscosity for Fourier approximations of conservation laws. C R Acad Sci, Ser I 2008;346:801–6.
[9] Guermond JL, Popov B. Viscous regularization of the Euler equations and entropy principles. SIAM J Appl Math 2014;74(2):284–305.
[10] Kurganov A, Liu Y. New adaptive artificial viscosity method for hyperbolic systems of conservation laws. J Comput Phys 2012;231(24):8114–32.
[11] Marras S, Nazarov M, Giraldo FX. Stabilized high-order Galerkin methods based on a parameter-free dynamic SGS model for LES. J Comput Phys 2015;301:77–101.
[12] Bazilevs Y, Calo V, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput Methods Appl Mech Engrg 2007;197:173–201.
[13] Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. Comput Methods Appl Mech Engrg 2002;191:4295–321.
[14] Codina R, Badia S, Baiges J, Principe J. Variational multiscale methods in computational fluid dynamics. In: Encyclopedia of computational mechanics. 2nd ed.. John Wiley & Sons, Ltd; 2017, p. 1–28.
[15] Hughes TJR, Feijóo G, Mazzei L, Quincy J. The variational multiscale method – A paradigm for computational mechanics. Comput Methods Appl Mech Engrg 1998;166:3–24.
[16] Layton W, Rebholz L, Trenchea C. Modular nonlinear filter stabilization of methods for higher Reynolds numbers flow. J Math Fluid Mech 2012;14:325–54.
[17] Bertagna L, Quaini A, Veneziani A. Deconvolution-based nonlinear filtering for incompressible flows at moderately large Reynolds numbers. Internat J Numer Methods Fluids 2016;81(8):463–88.
[18] Bowers AL, Rebholz LG, Takhirov A, Trenchea C. Improved accuracy in regularization models of incompressible flow via adaptive nonlinear filtering. Internat J Numer Methods Fluids 2012;70(7):805–28.
[19] Bowers A, Rebholz L. Numerical study of a regularization model for incompressible flow with deconvolution-based adaptive nonlinear filtering. Comput Methods Appl Mech Engrg 2013;258:1–12.
[20] Ervin V, Layton W, Neda M. Numerical analysis of filter-based stabilization for evolution equations. SIAM J Numer Anal 2012;50(5):2307–35.
[21] Girfoglio M, Quaini A, Rozza G. A finite volume approximation of the Navier-Stokes equations with nonlinear filtering stabilization. Comput Fluids 2019;187:27–45.
[22] Layton W, Röhe L, Tran H. Explicitly uncoupled VMS stabilization of fluid flow. Comput Methods Appl Mech Engrg 2011;200(45):3183–99.
[23] Olshanskii M, Xiong X. A connection between filter stabilization and eddy viscosity models. Numer Methods Partial Differ Equ 2013;29(6):2061–80.
[24] Chehab J-P. Damping, stabilization and numerical filtering for the modeling and the simulation of time dependent PDEs. Discrete Contin Dyn Syst - S 2021;14(8):2693–728.
[25] Hesthaven JS, Warburton T. Nodal discontinuous galerkin methods: Algorithms, analysis, and applications. Springer Publishing Company, Incorporated; 2007.
[26] Holm DD. Averaged Lagrangians and the mean effects of fluctuations in ideal fluid dynamics. Phys D: Nonlinear Phenom 2002;170:253–86.
[27] Secchi P. An alpha model for compressible fluids. Discrete Contin Dyn Syst - S 2008;3:351–9.
[28] Gravemeier V, Gee MW, Kronbichler M, Wall WA. An algebraic variational multiscale-multigrid method for large eddy simulation of turbulent flow. Comput Methods Appl Mech Engrg 2010;199:853–64.
[29] Gravemeier V, Wall WA, Ramm E. Large eddy simulation of turbulent incompressible flows by a three-level finite element method. Internat J Numer Methods Fluids 2005;48:1067–99.
[30] Gravemeier V, Wall MW, Ramm E. A three-level finite element method for the stationary incompressible Navier-Stokes equations. Comput Methods Appl Mech Engrg 2004;193:1323–66.
[31] Mullen J, Fischer P. Filtering techniques for complex geometry fluid flows. Commun Numer Methods Eng 1999;15(1):9–18.
[32] Boyd JP. Two comments on filtering (artificial viscosity) for Chebyshev and Legendre spectral and spectral element methods: Preserving boundary conditions and interpretation of the filter as a diffusion. J Comput Phys 1998;143(1):283–8.
[33] Fischer P, Mullen J. Filter-based stabilization of spectral element methods. C R Acad Sci, Ser I 2001;332(3):265–70.
[34] Garnier E, Adams N, Sagaut P. Large eddy simulation for compressible flows. Springer, Berlin; 2009.
[35] Mathew J, Lechner R, Foysi H, Sesterhenn J, Friedrich R. An explicit filtering method for large eddy simulation of compressible flows. Phys Fluids 2003;15(8):2279–89.
[36] Visbal M, Rizzetta D. Large eddy simulation on curvilinear grids using compact differencing and filtering schemes. J Fluids Eng 2002;124:836–47.
[37] Ern A, Guermond J. Weighting the edge stabilization. SIAM J Numer Anal 2013;51(3):1655–77.
[38] Dunca A, Epshteyn Y. On the Stolz-Adams deconvolution model for the large-eddy simulation of turbulent flows. SIAM J Numer Anal 2005;37(6):1890–902.
[39] Stolz S, Adams N. An approximate deconvolution procedure for large-eddy simulation. Phys Fluids 1999;11(7):1699–701.
[40] Stolz S, Adams N, Kleiser L. An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows. Phys Fluids 2001;13(4):997–1015.
[41] GEA - Geophysical and Environmental Applications. https://github.com/{GEA}-Geophysical-and-Environmental-Apps/{GEA}.
[42] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620–31.
[43] Girfoglio M, Quaini A, Rozza G. GEA: A new finite volume-based open source code for the numerical simulation of atmospheric and ocean flows. In: Proceedings of the Finite Volumes for Complex Applications 10 conference. 2023, Accepted. https://arxiv.org/abs/2303.10499.
[44] Girfoglio M, Quaini A, Rozza G. Validation of an OpenFOAM®-based solver for the Euler equations with benchmarks for mesoscale atmospheric modeling. AIP Adv 2023;13(5):055024.
[45] Kelly JF, Giraldo FX. Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: Limited-area mode. J Comput Phys 2012;231:7988–8008.
[46] Borggaard J, Iliescu T, Roop J. A bounded artificial viscosity large eddy simulation model. SIAM J Numer Anal 2009;47:622–45.
[47] Hunt J, Wray A, Moin P. Eddies stream and convergence zones in turbulent flows. Technical report CTR-S88, CTR report; 1988.
[48] Jasak H. Error analysis and estimation for the finite volume method with applications to fluid flows [Ph.D. thesis], Imperial College, University of London; 1996.
[49] Oberai AA, Hughes TJR. A palette of fine-scale eddy viscosity and residual-based models for variational multiscale formulations of turbulence. Comput Mech 2016;57:629–35.
[50] Ahmad N, Lindeman J. Euler solutions using flux-based wave decomposition. Internat J Numer Methods Fluids 2007;54:47–72.
[51] Feng Y, Miranda-Fuentes J, Jacob J, Sagaut P. Hybrid lattice Boltzmann model for atmospheric flows under anelastic approximation. Phys Fluids 2021;33(3):036607.
[52] Carpenter R, Droegemeier K, Woodward P, Hane C. Application of the piecewise parabolic method (PPM) to meteorological modeling. Mon Wea Rev 1990;118:586–612.
[53] Straka J, Wilhelmson R, Wicker L, Anderson J, Droegemeier K. Numerical solution of a nonlinear density current: A benchmark solution and comparisons. Int J Num Meth Fluids 1993;17:1–22.
[54] Ahmad NN. High-resolution wave propagation method for stratified flows. In: AIAA aviation forum, Atlanta, GA. AIAA. 2018.
[55] Giraldo FX, Restelli M. A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling. SIAM J Sci Comput 2009;31:2231–57.
[56] Nazarov M, Hoffman J. Residual-based artificial viscosity for simulation of turbulent compressible flow using adaptive finite element methods. Internat J Numer Methods Fluids 2013;71:339–57.
[57] Giraldo FX, Restelli M. A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. J Comput Phys 2008;227:3849–77.
[58] Marras S, Moragues M, Vázquez M, Jorba O, Houzeaux G. A variational multiscale stabilized finite element method for the solution of the Euler equations of nonhydrostatic stratified flows. J Comput Phys 2013;236:380–407.