

RESEARCH ARTICLE | MAY 22 2023

Validation of an OpenFOAM®-based solver for the Euler equations with benchmarks for mesoscale atmospheric modeling

Michele Girfoglio ; Annalisa Quaini; Gianluigi Rozza 

AIP Advances 13, 055024 (2023)

<https://doi.org/10.1063/5.0147457>

Articles You May Be Interested In

Pressurized turbulent premixed CH₄/H₂/air flame validation using OpenFOAM

AIP Advances (July 2022)

Evaluation of transition-sensitive eddy-viscosity turbulence models for separated flow in OpenFOAM

AIP Conf. Proc. (July 2017)

An OpenFOAM solver for multiphase and turbulent flow

Physics of Fluids (April 2020)



AIP Advances
Special Topics Now Online

[Learn More](#)

 AIP
Publishing

Validation of an OpenFOAM®-based solver for the Euler equations with benchmarks for mesoscale atmospheric modeling

Cite as: AIP Advances 13, 055024 (2023); doi: 10.1063/5.0147457

Submitted: 22 February 2023 • Accepted: 12 April 2023 •

Published Online: 22 May 2023



Michele Girfoglio,^{1,a)} Annalisa Quaini,² and Gianluigi Rozza^{1,a)} 

AFFILIATIONS

¹ mathLab, Mathematics Area, SISSA, via Bonomea 265, I-34136 Trieste, Italy

² Department of Mathematics, University of Houston, Houston, Texas 77204, USA

^{a)} Authors to whom correspondence should be addressed: mgirfogl@sissa.it and grozza@sissa.it

ABSTRACT

Within OpenFOAM, we develop a pressure-based solver for the Euler equations written in conservative form using density, momentum, and total energy as variables. Under simplifying assumptions, these equations are used to describe non-hydrostatic atmospheric flow. For the stabilization of the Euler equations and to capture sub-grid processes, we consider two Large Eddy Simulation models: the classical Smagorinsky model and the one equation eddy-viscosity model. To achieve high computational efficiency, our solver uses a splitting scheme that decouples the computation of each variable. The numerical results obtained with our solver are validated against numerical data available in the literature for two classical benchmarks: the rising thermal bubble and the density current. Through qualitative and quantitative comparisons, we show that our approach is accurate. This paper is meant to lay the foundation for a new open-source package specifically created for the quick assessment of new computational approaches for the simulation of atmospheric flows at the mesoscale level.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0147457>

I. INTRODUCTION

Forecasting the rapid changes in the Earth's climate is one of the biggest challenges of our times. Since the complexity of the problem requires inputs and close collaboration from scientists across various disciplines, ranging from physics of the atmosphere to computer science, open-source software has emerged as the evident choice. In fact, the availability of open-source software for use, modification, and distribution makes it ideal for collaborative development. Some examples of open-source software for climate simulations are the Community Earth System Model,¹ the Energy Exascale Earth System Model,² the MIT General Circulation Model,³ and the Climate Machine.⁴ All of these examples have large communities of active users and developers that have contributed over several years. Since climate simulations are an extension of weather forecasting, we note that open-source software has been a tool of choice for weather prediction as well. Examples include WRF⁵ and Atlas.⁶

Fast and accurate weather/climate forecasts need state-of-the-art numerical and computational methodologies. Although the

above-mentioned software packages are great tools for realistic simulations, testing and assessing new numerical approaches within them are non-trivial. In fact, the very complexity that allows for realistic simulations requires a considerable amount of time to familiarize with, making such software impractical as a testbed. This paper lays the foundation for a new open-source package, called Geophysical and Environmental Applications (GEA),⁷ specifically created for the assessment of new computational approaches for the simulation of mesoscale atmospheric flows and ocean flows.^{8,9} This idea is that if a new approach fails to meet desired accuracy or efficiency standards in our simplified software, it would not be considered for implementation in more advanced software. To maximize the reach and impact, we choose to build our software package on OpenFOAM®,¹⁰ an open-source, freely available C++ finite volume library that has become widely used in Computational Fluid Dynamics (CFD).

In recent years, OpenFOAM has been adopted to study a large variety of computational approaches for different CFD applications, including plasma cutting,¹¹ fire plumes,¹² and metal forming

processes.¹³ Thanks to parallel computing support, multiphase modeling capabilities, a wide range of existing turbulence models, and ease of implementation for new ones, OpenFOAM represents a great tool for the development and assessment of computational strategies for atmospheric modeling. In this context, researchers have chosen OpenFOAM to perform atmospheric boundary layer simulations (see, e.g., Refs. 14–16), multi-fluid atmospheric convection (see, e.g., Ref. 17), and mesh r -adaptivity for weather prediction over steep terrain (see, e.g., Ref. 18). The results in Refs. 17 and 18 have been obtained with AtmosFOAM,¹⁹ an OpenFOAM-based model of the global atmosphere using arbitrary-shaped cells. Our solver starts with a focus on local atmospheric problem. However, we do not exclude interfacing it with AtmosFOAM in the future.

As the core of our open-source package, we present an OpenFOAM-based solver for the Euler equations for stratified fluid flow, which are of importance in non-hydrostatic mesoscale atmospheric modeling, and assess it through two well-known test cases. We consider the Euler equations written in conservative form using density, momentum, and total energy as variables. It is shown in Ref. 20 that this set of equations yields less dissipative results than other forms of the Euler equations for relevant test problems. Hence, it was recommended to pursue this formulation for the development of mesoscale models, although other formulations (i.e., non-conservative forms and the conservative form using density, momentum, and potential temperature as variables) have received more attention for atmospheric studies. For the stabilization of the Euler equations and to capture sub-grid processes, we will assess and compare two Large Eddy Simulation (LES) models built in OpenFOAM: the classical Smagorinsky model²¹ and the one equation eddy-viscosity model.²² We will show that both LES models yield numerical results that agree very well with data published in the literature.

Although, currently, both pressure-based and density-based approaches for the Euler equations apply to a wide range of flows, from incompressible to highly compressible, pressure-based solvers have been conventionally used for mildly compressible flows (as is the case for atmospheric flows). Despite this, the majority of the software for weather prediction adopts density-based approaches, which were originally designed for high-speed compressible flows. In this paper, we opt for a pressure-based solver and show that it yields accurate results when compared to data in the literature. In the spirit of OpenFOAM, our solver uses a splitting scheme that decouples the computation of each variable in order to achieve high computational efficiency.

The rest of this paper is organized as follows: In Sec. II, we briefly describe the formulation of the Euler equations and the LES models under consideration. Section III presents our pressure-based approach and provides the details of space and time discretization. Numerical results for the two benchmark tests are discussed in Sec. IV. Conclusions are drawn in Sec. V.

II. PROBLEM DEFINITION

A. The compressible Euler equations

We consider the dynamics of dry atmosphere in a spatial domain of interest Ω by neglecting the effects of moisture, solar radiation, and heat flux from the ground. We assume that dry air behaves like an ideal gas.

Let ρ be the air density, $\mathbf{u} = (u, v, w)$ be the wind velocity, and e be the total energy density. Note that $e = c_v T + |\mathbf{u}|^2/2 + gz$, 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 equations stating the conservation of mass, momentum, and energy for the dry atmosphere written in terms of ρ , $\mathbf{u}$, and e over a time interval of interest $(0, t_f]$ read

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

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

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

where $\hat{\mathbf{k}}$ is the unit vector aligned with the vertical axis z and p is pressure. To close systems (1)–(3), we need a thermodynamics equation of state for p . Following the assumption that dry air behaves like an ideal gas, we have

$$p = \rho R T, \quad (4)$$

where R is the specific gas constant of dry air.

Let us write the pressure as the sum of a fluctuation p' with respect to a background state

$$p = p' + \rho g z. \quad (5)$$

By plugging Eqs. (5) into (2), we obtain

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

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

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

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

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

where we have used Eq. (1) for further simplification. We will devise a splitting approach for problems (1) and (4)–(8) because this formulation of the Euler equation facilitates the decoupling of all variables. In addition, it allows for an explicit treatment of the kinetic and potential energies.

Remark II.1. A quantity of interest for atmospheric problems is the potential temperature

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

i.e., the temperature that a parcel of dry air would have if it were expanded or compressed adiabatically to standard pressure $p_0 = 10^5$ Pa, which is the atmospheric pressure at the ground. In addition,

we define the potential temperature fluctuation θ' as the difference between θ and its mean hydrostatic value θ_0 ,

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

Note that the hydrostatic reference state is a function of the vertical coordinate z only. See, e.g., Ref. 23 for more details.

B. LES models

The numerical discretization of the model presented in Subsection II A would lead to an accurate description of atmospheric

flow if one could afford a discretization mesh able to capture all the scales of turbulent structures. Since the Kolmogorov scale of a typical atmospheric problem is about 10^{-4} m, a Direct Numerical Simulation is currently beyond reach. One way to keep the computational cost affordable is to adopt a coarser mesh and model the effects of the small scales that are not directly solved. This can be carried out via Large Eddy Simulation.

We will focus on two LES models: the classical Smagorinsky model²¹ and the one equation eddy-viscosity model.²² Both approaches are equivalent to introducing additional terms in Eqs. (6) and (8) of the following form:

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p' + gz \nabla \rho - \nabla \cdot (2\mu_a \boldsymbol{\epsilon}(\mathbf{u})) + \nabla \cdot \left(\frac{2}{3} \mu_a \nabla \cdot \mathbf{u} \right) = 0 \quad \text{in } \Omega \times (0, t_f], \quad (11)$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) + \frac{\partial(\rho K)}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) - \frac{\partial p}{\partial t} + \rho g \mathbf{u} \cdot \hat{\mathbf{k}} - \nabla \cdot \left(\frac{\mu_a}{\text{Pr}} \nabla h \right) = 0 \quad \text{in } \Omega \times (0, t_f], \quad (12)$$

where μ_a is an artificial viscosity, $\boldsymbol{\epsilon}(\mathbf{u}) = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$ 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 definition of μ_a is what distinguishes the different LES models.

The artificial dynamics viscosity, also called subgrid-scale eddy viscosity, introduced by the Smagorinsky model is given by

$$\mu_a = \rho (C_s \delta)^2 \sqrt{2\boldsymbol{\epsilon} : \boldsymbol{\epsilon}}, \quad C_s^2 = C_k \sqrt{\frac{C_k}{C_\epsilon}}, \quad (13)$$

where δ is the filter width, and C_k and C_ϵ are model parameters. The values of such parameters are not universal and need to be tuned for each specific application. For the results in Sec. IV, we compute δ for each cell by taking the maximum distance between the cell center and a face center multiplied by 2. The Smagorinsky model has experienced many achievements for several reasons. It is relatively simple and computationally inexpensive compared to other turbulence models. It is easy to implement and can be used with a wide range of numerical methods. It has been successfully used in many practical applications (also thanks to the tunable parameters), which include weather forecasting and climate modeling. The Smagorinsky model has a few important shortcomings too. In fact, it introduces large amounts of artificial viscosity for certain flows that are not turbulent, e.g., laminar shear flow (where velocity gradients are constant but large). Moreover, it has difficulty in accurately simulating flows with strong turbulence or near-wall regions. However, its main limitation lies in the assumption of local balance between the subgrid-scale energy production and dissipation. The one equation eddy-viscosity model (kEqn model, for short) in Ref. 22 was developed to overcome this limitation.

The artificial dynamics viscosity introduced by this model is given by

$$\mu_a = \rho C_k \sqrt{k_{\text{sgs}}} \delta, \quad (14)$$

where C_k and δ represent the same quantities as in the Smagorinsky model, and k_{sgs} is the subgrid-scale kinetic energy computed from a transport equation that accounts for production, dissipation, and diffusion. See Ref. 22 for more details. The kEqn model shares some advantages of the Smagorinsky model, namely, it is easy to implement (as it requires only one additional equation, which can be easily incorporated into existing codes) and it is computationally efficient. The limitations of this approach include that it can be inaccurate for flows with strong streamline curvature, swirling flows, and flows with large-scale turbulence structures. Moreover, inaccuracies might arise for complex turbulent flows, such as those with large-scale anisotropy or separation. Finally, the kEqn model is sensitive to boundary conditions and can lead to inaccurate results if the boundary conditions are not carefully chosen.

III. A SPLITTING APPROACH

This section presents space and time discretization for the generic LES models (1), (4), (5), (7), (11), and (12).

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. Problems (1), (4), (5), (7), (11), and (12) discretized in time read: given ρ^0 , $\mathbf{u}^0$, h^0 , p^0 , and T^0 , set $K^0 = |\mathbf{u}^0|^2/2$, and for $n \geq 0$, find the solution $(\rho^{n+1}, \mathbf{u}^{n+1}, h^{n+1}, K^{n+1}, p^{n+1}, p'^{n+1}, T^{n+1})$ of system,

$$\frac{\rho^{n+1}}{\Delta t} + \nabla \cdot (\rho^{n+1} \mathbf{u}^{n+1}) = b_\rho^{n+1}, \quad (15)$$

$$\begin{aligned} \frac{\rho^{n+1} \mathbf{u}^{n+1}}{\Delta t} + \nabla \cdot (\rho^{n+1} \mathbf{u}^{n+1} \otimes \mathbf{u}^{n+1}) + \nabla p'^{n+1} + gz \nabla \rho^{n+1} \\ - \nabla \cdot (2\mu_a^{n+1} \boldsymbol{\epsilon}(\mathbf{u}^{n+1})) + \nabla \cdot \left(\frac{2}{3} \mu_a^{n+1} \nabla \cdot \mathbf{u}^{n+1} \right) = \mathbf{b}_u^{n+1}, \end{aligned} \quad (16)$$

$$\begin{aligned} & \frac{\rho^{n+1} h^{n+1}}{\Delta t} + \nabla \cdot (\rho^{n+1} \mathbf{u}^{n+1} h^{n+1}) + \frac{\rho^{n+1} K^{n+1}}{\Delta t} \\ & + \nabla \cdot (\rho^{n+1} \mathbf{u}^{n+1} K^{n+1}) - \frac{p^{n+1}}{\Delta t} + \rho^{n+1} g \mathbf{u}^{n+1} \\ & \times \widehat{\mathbf{k}} - \nabla \cdot \left(\frac{\mu_a^{n+1}}{\text{Pr}} \nabla h^{n+1} \right) = b_e^{n+1}, \end{aligned} \quad (17)$$

$$p^{n+1} = p'^{n+1} + \rho^{n+1} g z, \quad (18)$$

$$p^{n+1} = \rho^{n+1} R T^{n+1}, \quad (19)$$

$$h^{n+1} - h^n = c_p (T^{n+1} - T^n), \quad (20)$$

$$K^{n+1} = \frac{|\mathbf{u}^{n+1}|^2}{2}, \quad (21)$$

where $b_\rho^{n+1} = \rho^n / \Delta t$, $\mathbf{b}_u^{n+1} = \rho^n \mathbf{u}^n / \Delta t$, and $b_e^{n+1} = (\rho^n h^n + \rho^n K^n - \rho^n) / \Delta t$. Note that, in Eq. (20), we have chosen to update the value of the specific enthalpy incrementally.

A monolithic approach for coupled problems (15)–(21) would lead to high computational costs. Thus, to save computational time, we adopt a splitting approach. While the splitting applies to the partial differential equations, non-differential equations in problems (15)–(21) are used as predictors or to compute end-of-step variables. The splitting scheme is 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

- Step 1: First, find intermediate density $\rho^{n+\frac{1}{3}}$, intermediate velocity $\mathbf{u}^{n+\frac{1}{3}}$, and associated kinetic energy density $K^{n+\frac{1}{3}}$ such that

$$\frac{\rho^{n+\frac{1}{3}}}{\Delta t} = b_\rho^{n+1} - \nabla \cdot (\rho^n \mathbf{u}^n), \quad (22)$$

$$\begin{aligned} & \frac{\rho^{n+\frac{1}{3}} \mathbf{u}^{n+\frac{1}{3}}}{\Delta t} + \nabla \cdot (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+\frac{1}{3}}) + \nabla p^n \\ & - \nabla \cdot \left(2\mu_a^n \epsilon(\mathbf{u}^{n+\frac{1}{3}}) \right) + \nabla \cdot \left(\frac{2}{3} \mu_a^n \nabla \cdot \mathbf{u}^n \right) = \mathbf{b}_u^{n+1}, \end{aligned} \quad (23)$$

$$K^{n+\frac{1}{3}} = \frac{|\mathbf{u}^{n+\frac{1}{3}}|^2}{2}. \quad (24)$$

Note that Eq. (24) is a predictor step, and the associated corrector step is Eq. (32).

- Step 2: Find specific enthalpy h^{n+1} , temperature T^{n+1} , and second intermediate density $\rho^{n+\frac{2}{3}}$ such that

$$\begin{aligned} & \frac{\rho^{n+\frac{1}{3}} h^{n+1}}{\Delta t} + \nabla \cdot (\rho^n \mathbf{u}^n h^{n+1}) - \nabla \cdot \left(\frac{\mu_a^n}{\text{Pr}} \nabla h^{n+1} \right) \\ & = \tilde{b}_e^n - \frac{\rho^{n+\frac{1}{3}} K^{n+\frac{1}{3}}}{\Delta t} - \nabla \cdot (\rho^n \mathbf{u}^n K^{n+\frac{1}{3}}) + \frac{p^n}{\Delta t} \\ & - \rho^{n+\frac{1}{3}} g \mathbf{u}^{n+\frac{1}{3}} \cdot \widehat{\mathbf{k}}, \end{aligned} \quad (25)$$

$$h^{n+1} - c_p T^{n+1} = h^n - c_p T^n, \quad (26)$$

$$\rho^{n+\frac{2}{3}} R T^{n+1} = p^n, \quad (27)$$

where $\tilde{b}_e^n = (\rho^n h^n + \rho^n K^{n-1} - p^{n-1}) / \Delta t$. Equation (26) will be used to compute the end-of-step temperature (see Sec. III A), whereas Eq. (27) is a predictor step for density. The associated correction will come from Eq. (30) as explained in Sec. III A.

- Step 3: Find end-of-step velocity $\mathbf{u}^{n+1}$ and associated kinetic energy density K^{n+1} , pressure p^{n+1} and pressure fluctuation p'^{n+1} , and end-of-step density ρ^{n+1} such that

$$\begin{aligned} & \frac{\rho^{n+\frac{1}{3}} \mathbf{u}^{n+1}}{\Delta t} + \nabla \cdot (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1}) + \nabla p^{n+1} - \nabla \cdot (2\mu_a^n \epsilon(\mathbf{u}^{n+1})) \\ & + \nabla \cdot \left(\frac{2}{3} \mu_a^n \nabla \cdot \mathbf{u}^n \right) = \mathbf{b}_u^{n+1}, \end{aligned} \quad (28)$$

$$p^{n+1} - p'^{n+1} = \rho^{n+\frac{2}{3}} g z, \quad (29)$$

$$p^{n+1} - \rho^{n+1} R T^{n+1} = 0, \quad (30)$$

$$\frac{\rho^{n+\frac{2}{3}}}{\Delta t} + \nabla \cdot (\rho^{n+\frac{1}{3}} \mathbf{u}^{n+1}) = b_\rho^{n+1}, \quad (31)$$

$$K^{n+1} = \frac{|\mathbf{u}^{n+1}|^2}{2}. \quad (32)$$

Note that in Eq. (28), one diffusive term is kept implicit, whereas the other is treated explicitly. This is customary in OpenFOAM.

Section III A will describe the space discretization of the equations at each step and explain how to further reduce the computational cost associated with Step 3.

A. Space discretization

For the space discretization of Eqs. (22)–(32), we adopt a finite volume method. This requires partitioning 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 $j = 1, \dots, M$.

Let us start with Step 1. After applying the Gauss-divergence theorem, the integral form of Eq. (22) for each volume Ω_i is given by

$$\frac{1}{\Delta t} \int_{\Omega_i} \rho^{n+\frac{1}{3}} d\Omega = \int_{\Omega_i} b_\rho^{n+1} d\Omega - \int_{\partial\Omega_i} \rho^n \mathbf{u}^n \cdot d\mathbf{A}. \quad (33)$$

Let us denote with $(\rho^n \mathbf{u}^n)_i$ and $\rho_i^{n+\frac{1}{3}}$ the average flux and intermediate density in control volume Ω_i , respectively. Then, Eq. (33) is approximated as follows:

$$\frac{1}{\Delta t} \rho_i^{n+\frac{1}{3}} + \sum_j \varphi_j^n = b_{\rho,i}^{n+1}, \quad \varphi_j^n = (\rho^n \mathbf{u}^n)_{i,j} \cdot \mathbf{A}_j, \quad (34)$$

where ϕ_j^n denotes the convective flux through face j of Ω_i , and $b_{\rho,i}^{n+1}$ is the average right-hand side term in Ω_i . The convective flux at the cell faces is computed by a linear interpolation of the values from the adjacent cells.

Next, we consider Eq. (23). Its integral form for each volume Ω_i , it is given by

$$\begin{aligned} & \frac{1}{\Delta t} \int_{\Omega_i} \rho^{n+\frac{1}{3}} \mathbf{u}^{n+\frac{1}{3}} d\Omega + \int_{\partial\Omega_i} (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+\frac{1}{3}}) \cdot d\mathbf{A} \\ & + \int_{\Omega_i} \nabla p^n d\Omega - \int_{\partial\Omega_i} 2\mu_a^n \epsilon(\mathbf{u}^{n+\frac{1}{3}}) \cdot d\mathbf{A} \\ & + \int_{\partial\Omega_i} \left(\frac{2}{3} \mu_a^n \nabla \cdot \mathbf{u}^n \right) d\mathbf{A} = \int_{\Omega_i} \mathbf{b}_u^{n+1} d\Omega, \end{aligned} \quad (35)$$

after the application of the Gauss-divergence theorem. For the approximation of most of the terms in Eq. (35), we follow Ref. 24. Therefore, we write the discretized form of Eq. (35) divided by the control volume Ω_i as

$$\begin{aligned} & \frac{1}{\Delta t} \rho_i^{n+\frac{1}{3}} \mathbf{u}_i^{n+\frac{1}{3}} + \sum_j \phi_j^n \mathbf{u}_{i,j}^{n+\frac{1}{3}} + \nabla p_i^n - \sum_j 2\mu_a^n \epsilon(\mathbf{u}_i^{n+\frac{1}{3}})_j \cdot \mathbf{A}_j \\ & + \sum_j \left(\frac{2}{3} \mu_a^n (\nabla \cdot \mathbf{u}^n)_j \right) \mathbf{A}_j = \mathbf{b}_{u,i}^{n+1}, \end{aligned} \quad (36)$$

where $\mathbf{u}_i^{n+\frac{1}{3}}$, p_i^n , and $\mathbf{b}_{u,i}^{n+1}$ are the average intermediate velocity, pressure, and source term in control volume Ω_i , respectively, and $\mathbf{u}_{i,j}^{n+\frac{1}{3}}$ denotes the intermediate velocity associated with the centroid of face j normalized by the volume of Ω_i . This approximation of $\mathbf{u}^{n+\frac{1}{3}}$ at cell face j is obtained with a third-order interpolation scheme.¹⁰ We will use the same scheme for all the flux terms in the equations below.

For term $\epsilon(\mathbf{u}_i^{n+\frac{1}{3}})_j$, we need to approximate the gradient of $\mathbf{u}_i^{n+\frac{1}{3}}$ at face j . We choose to do it with second-order accuracy. See Ref. 25 for more details. The pressure term is treated with a second-order face flux reconstruction in order to suppress spurious oscillations.²⁶ We discretize all the pressure terms in the following equations in the same way.

To complete Step 1, we need the discretized form of Eq. (24), which is given by

$$K_i^{n+\frac{1}{3}} = \frac{|\mathbf{u}_i^{n+\frac{1}{3}}|^2}{2}. \quad (37)$$

Let us now consider Step 2. After the application of the Gauss-divergence theorem, the integral form of Eq. (25) for each volume Ω_i becomes

$$\begin{aligned} & \frac{1}{\Delta t} \int_{\Omega_i} \rho^{n+\frac{1}{3}} h^{n+1} d\Omega + \int_{\partial\Omega_i} (\rho^n \mathbf{u}^n h^{n+1}) \cdot d\mathbf{A} \\ & - \int_{\partial\Omega_i} \left(\frac{\mu_a^n}{\text{Pr}} \nabla h^{n+1} \right) \cdot d\mathbf{A} = \int_{\Omega_i} \tilde{b}_e^n d\Omega \\ & - \frac{1}{\Delta t} \int_{\Omega_i} \rho^{n+\frac{1}{3}} K^{n+\frac{1}{3}} d\Omega - \int_{\partial\Omega_i} (\rho^n \mathbf{u}^n K^{n+\frac{1}{3}}) \cdot d\mathbf{A} \\ & + \frac{1}{\Delta t} \int_{\Omega_i} p^n d\Omega - \int_{\Omega_i} \rho^{n+\frac{1}{3}} g \mathbf{u}^{n+\frac{1}{3}} \cdot \hat{\mathbf{k}} d\Omega. \end{aligned} \quad (38)$$

The discretized form of Eq. (38) can be written as

$$\begin{aligned} & \frac{1}{\Delta t} \rho_i^{n+\frac{1}{3}} h_i^{n+1} + \sum_j \phi_j^n h_{i,j}^{n+1} - \sum_j \frac{\mu_a^n}{\text{Pr}} (\nabla h_i^{n+1})_j \cdot \mathbf{A}_j = \tilde{b}_{e,i}^n \\ & - \frac{1}{\Delta t} \rho_i^{n+\frac{1}{3}} K_i^{n+\frac{1}{3}} - \sum_j \phi_j^n K_{i,j}^{n+\frac{1}{3}} \\ & + \frac{1}{\Delta t} p_i^n - \rho_i^{n+\frac{1}{3}} g \mathbf{u}_i^{n+\frac{1}{3}} \cdot \hat{\mathbf{k}}, \end{aligned} \quad (39)$$

where h_i^{n+1} and $\tilde{b}_{e,i}^n$ are the average specific enthalpy and source term in control volume Ω_i , respectively, while $h_{i,j}^{n+1}$ and $K_{i,j}^{n+\frac{1}{3}}$ denote the specific enthalpy and kinetic energy density associated with the centroid of face j normalized by the volume of Ω_i , respectively. Finally, $(\nabla h_i^{n+1})_j$ is the gradient of h_i^{n+1} at face j .

To complete Step 2, we need the discretized form of Eqs. (26) and (27), which are given by

$$h_i^{n+1} - c_p T_i^{n+1} = h_i^n - c_p T_i^n, \quad (40)$$

$$\rho_i^{n+\frac{2}{3}} R T_i^{n+1} = p_i^n. \quad (41)$$

Note that the specific enthalpy is computed from Eq. (39), so one computes the end-of-step temperature from Eq. (40) and the second intermediate density from Eq. (41) in a completely decoupled fashion.

The treatment of Step 3 requires careful attention in order to contain the computational cost. We start by plugging Eq. (29) into Eq. (28). The integral form of the resulting equation for each volume Ω_i is given by

$$\begin{aligned} & \frac{1}{\Delta t} \int_{\Omega_i} \rho^{n+\frac{1}{3}} \mathbf{u}^{n+1} d\Omega + \int_{\partial\Omega_i} (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1}) \cdot d\mathbf{A} \\ & + \int_{\Omega_i} \nabla p'^{n+1} d\Omega + \int_{\Omega_i} g z \nabla \rho^{n+\frac{2}{3}} d\Omega \\ & - \int_{\partial\Omega_i} 2\mu_a^n \epsilon(\mathbf{u}^{n+1}) \cdot d\mathbf{A} + \int_{\partial\Omega_i} \left(\frac{2}{3} \mu_a^n \nabla \cdot \mathbf{u}^n \right) d\mathbf{A} \\ & = \int_{\Omega_i} \mathbf{b}_u^{n+1} d\Omega, \end{aligned} \quad (42)$$

after the application of the Gauss-divergence theorem. Each term in Eq. (42) is approximated like the corresponding term in Eq. (35). Hence, the discretized form of Eq. (42), divided by the control volume Ω_i , is given by

$$\begin{aligned} & \frac{1}{\Delta t} \rho_i^{n+\frac{1}{3}} \mathbf{u}_i^{n+1} + \sum_j \phi_j^n \mathbf{u}_{i,j}^{n+1} + \nabla p_i'^{n+1} + g z_i \nabla \rho_i^{n+\frac{2}{3}} \\ & - \sum_j 2\mu_a^n \epsilon(\mathbf{u}_i^{n+1})_j \cdot \mathbf{A}_j + \sum_j \left(\frac{2}{3} \mu_a^n (\nabla \cdot \mathbf{u}^n)_j \right) \mathbf{A}_j = \mathbf{b}_{u,i}^{n+1}, \end{aligned} \quad (43)$$

where $\mathbf{u}_i^{n+1}$, $p_i'^{n+1}$, and $\mathbf{b}_{u,i}^{n+1}$ are the average end-of-step velocity, pressure fluctuation, and source term in control volume Ω_i , respectively, while $\mathbf{u}_{i,j}^{n+1}$ denotes the end-of-step velocity associated with the centroid of face j normalized by the volume of Ω_i . In Eq. (43), z_i is the vertical coordinate of the centroid of cell Ω_i .

Following Ref. 25, we now write Eq. (43) in semi-discretized form, i.e., with some terms in the continuous form while all other terms (grouped in $\mathbf{H}$) are in the discrete form,

$$\begin{aligned} \mathbf{u}^{n+1} &= \frac{\Delta t}{\rho^{n+\frac{1}{3}}} \left(\mathbf{H}(\mathbf{u}_i^{n+1}) - gz \nabla \rho^{n+\frac{2}{3}} - \nabla p'^{n+1} \right), \\ \mathbf{H}(\mathbf{u}_i^{n+1}) &= - \sum_j \varphi_j^n \mathbf{u}_{i,j}^{n+1} + \sum_j 2\mu_a^n \epsilon(\mathbf{u}_i^{n+1})_j \\ &\quad \times \mathbf{A}_j - \sum_j \left(\frac{2}{3} \mu_a^n (\nabla \cdot \mathbf{u}^n)_j \right) \mathbf{A}_j + \mathbf{b}_{u,i}^{n+1}. \end{aligned} \quad (44)$$

Next, we plug Eq. (44) into Eq. (31) to obtain

$$\begin{aligned} \frac{1}{\Delta t} \rho^{n+\frac{2}{3}} + \nabla \cdot \left(\rho^{n+\frac{2}{3}} \left(\frac{\Delta t}{\rho^{n+\frac{1}{3}}} \left(\mathbf{H}(\mathbf{u}^{n+1}) - gz \nabla \rho^{n+\frac{2}{3}} - \nabla p'^{n+1} \right) \right) \right) \\ = b_{\rho}^{n+1}. \end{aligned} \quad (45)$$

By integrating Eq. (45) over the control volume Ω_i , applying the Gauss-divergence theorem, and dividing by the control volume, we obtain

$$\begin{aligned} \sum_j \rho_j^{n+\frac{2}{3}} (\nabla p_i'^{n+1})_j \cdot \mathbf{A}_j \\ = \sum_j \frac{\rho_j^{n+\frac{2}{3}} \Delta t}{\rho_j^{n+\frac{1}{3}}} \left(\mathbf{H}(\mathbf{u}_i^{n+1})_j - gz_j (\nabla \rho_i^{n+\frac{2}{3}})_j \right) \\ \times \mathbf{A}_j - b_{\rho,i}^{n+1} + \frac{1}{\Delta t} \rho_i^{n+\frac{2}{3}}, \end{aligned} \quad (46)$$

which is used to compute the pressure fluctuation. In Eq. (46), $(\nabla p_i'^{n+1})_j$ and $(\nabla \rho_i^{n+\frac{2}{3}})_j$ are the gradients of p'^{n+1} and $\rho_i^{n+\frac{2}{3}}$ at faces j , respectively. In OpenFOAM, there are a few partitioned algorithms that decouple the computation of the pressure from the computation of the velocity, namely, SIMPLE²⁷ for steady-state problems, and PISO²⁸ and PIMPLE²⁹ for time-dependent problems. In this work, we use the PISO algorithm.

Now, we have computed the pressure fluctuation and the end-of-step velocity, and we can obtain the pressure and the end-of-step kinetic energy density using the discretized forms of Eqs. (29) and (32),

$$\begin{aligned} p_i^{n+1} &= p_i'^{n+1} + \rho_i^{n+\frac{2}{3}} gz_i, \\ K_i^{n+1} &= \frac{|\mathbf{u}_i^{n+1}|^2}{2}. \end{aligned}$$

Finally, we compute the end-of-step density with the space-discrete version of Eq. (30),

$$\rho_i^{n+1} = \frac{p_i^{n+1}}{RT_i^{n+1}}. \quad (47)$$

IV. NUMERICAL RESULTS

We test our OpenFOAM-based solver for the Euler equation with two standard benchmarks that have been widely used to assess atmospheric dynamical cores, i.e., the rising thermal bubble and the

density current. Both test cases involve a perturbation of a neutrally stratified atmosphere with uniform background potential temperature over a flat terrain. Therefore, before reporting the results for the two benchmarks, in Sec. IV A, we show that an unperturbed stratified atmosphere with uniform background potential temperature over a flat terrain remains unchanged up to a certain tolerance. In Sec. IV B, we present our results for the rising thermal bubble benchmark. There exist several variations of this benchmark, featuring different geometries and/or initial conditions. We use the settings from Ref. 30. See also Ref. 31 for a recent study using this variation. Our results for the classical density current test^{32,33} are shown in Sec. IV C. Both benchmarks are two-dimensional. Although our solver is designed to deal with fully three-dimensional benchmarks, two-dimensional problems can be solved by generating a three-dimensional mesh with just one cell in the third dimension.

For each benchmark, we will report the Reynolds number Re and the Mach number M defined as

$$Re = \frac{\rho_r U_r L_r}{\mu_r}, \quad M = \frac{U_r}{\sqrt{\gamma R T_r}}, \quad (48)$$

where ρ_r , μ_r , U_r , L_r , and T_r are reference density, dynamic viscosity, velocity, length, and temperature, respectively. Moreover, $\gamma = c_p/c_v$.

We would like to point out that neither the rising bubble nor the density current benchmark has an exact solution. Hence, one can only have a relative comparison with other numerical data available in the literature.

A. Hydrostatic atmosphere

The goal of this first test case is to verify that an initial resting atmosphere over a flat terrain remains still within a reasonable accuracy for a long time interval.

The computational domain in the xz -plane is $\Omega = [0, 16000] \times [0, 800] \text{ m}^2$. In this domain, the hydrostatic atmosphere, initially at rest, is free to evolve until $t = 25$ days.³⁴ Impenetrable, free-slip boundary conditions are imposed at all the boundaries. We consider a uniform mesh with mesh sizes $h = \Delta x = \Delta z = 250 \text{ m}$ and we set the time step to $\Delta t = 0.1 \text{ s}$.

Figure 1 shows the time evolution of the maximal vertical velocity $w_{\max}$. We observe that $w_{\max}$ has an initial, rather fast, growth but subsequently stabilizes around $1 \times 10^{-5} \text{ m/s}$. Thus, we conclude that

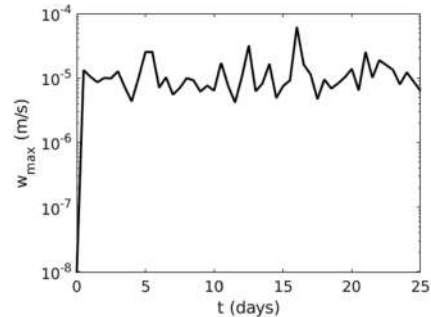


FIG. 1. Hydrostatic atmosphere: time evolution of the maximal vertical velocity $w_{\max}$.

the hydrostatic equilibrium is preserved with reasonably good accuracy, especially in view of the two benchmarks we are going to study next.

B. Rising thermal bubble in a neutrally stratified atmosphere

The computational domain in the xz -plane is $\Omega = [0, 5000] \times [0, 10000]$ m², and the time interval of interest is $(0, 1020]$ s. Impenetrable, free-slip boundary conditions are imposed on all walls. The initial density is given by

$$\rho^0 = \frac{p_g}{R\theta_0} \left(\frac{p}{p_g} \right)^{c_v/c_p}, \quad p = p_g \left(1 - \frac{gz}{c_p\theta^0} \right)^{c_p/R}, \quad (49)$$

with $c_p = R + c_v$, $c_v = 715.5$ J/(Kg K), and $R = 287$ J/(Kg K). In Eq. (49), θ^0 is the initial potential temperature, which is defined as

$$\theta^0 = 300 + 2 \left[1 - \frac{r}{r_0} \right] \text{ if } r \leq r_0 = 2000 \text{ m}, \quad \theta^0 = 300 \text{ otherwise}, \quad (50)$$

where $r = \sqrt{(x - x_c)^2 + (z - z_c)^2}$, and $(x_c, z_c) = (5000, 2000)$ m.^{30,35} Note that Eqs. (49) and (50) represent a neutrally

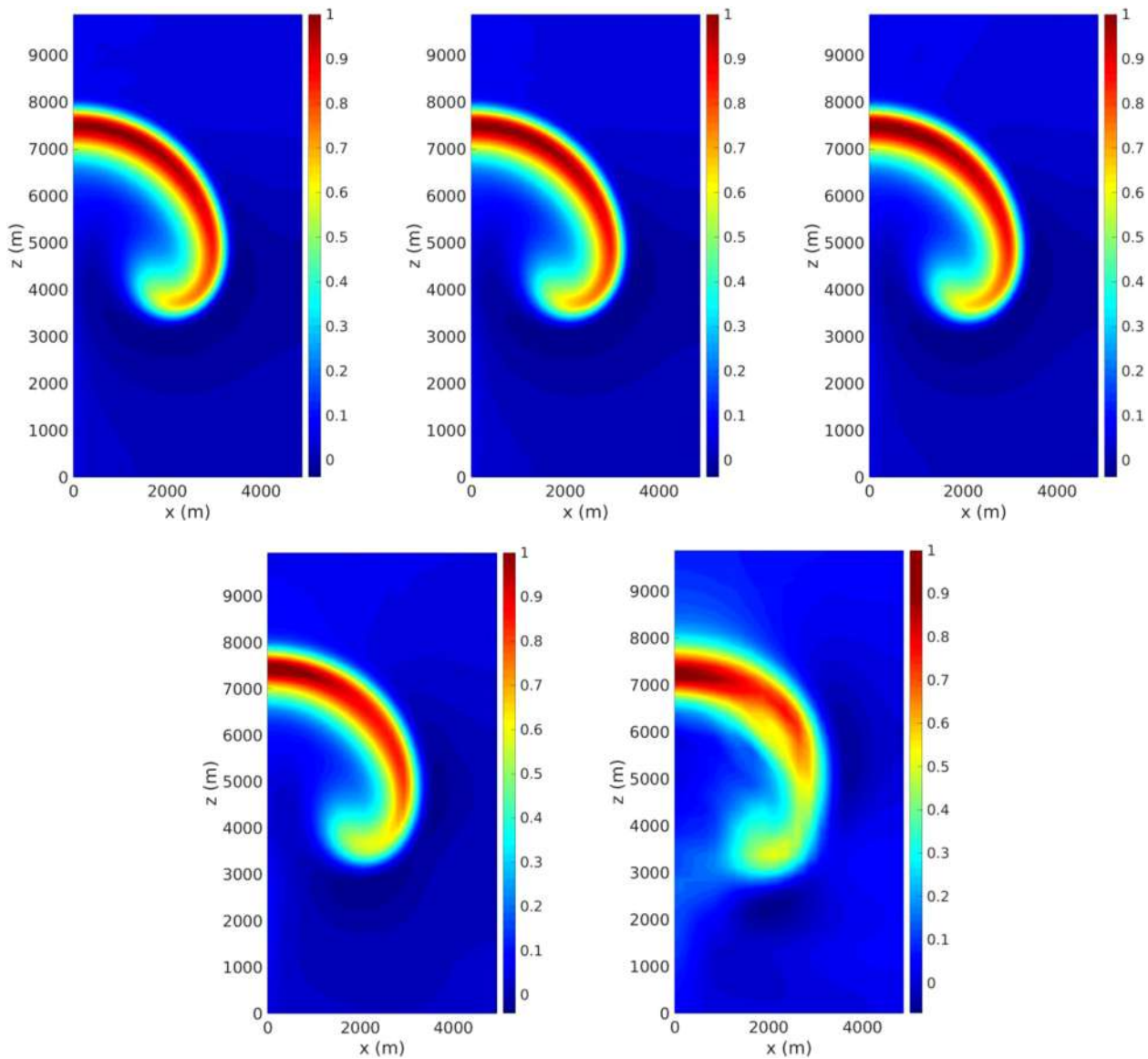


FIG. 2. Rising thermal bubble, AV15 model: perturbation of potential temperature computed with mesh $h = 15.625$ m (top-left), $h = 32.25$ m (top-center), $h = 62.5$ m (top-right), $h = 125$ m (bottom-left), and $h = 250$ m (bottom-right).

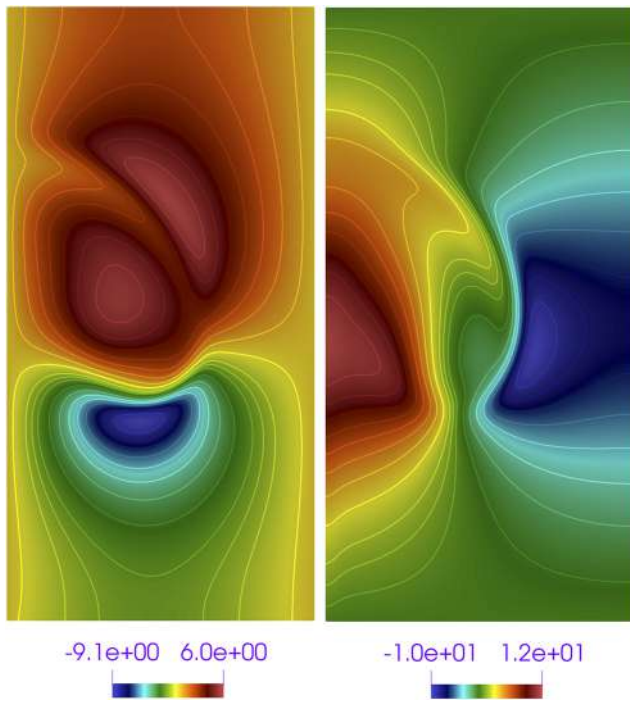


FIG. 3. Rising thermal bubble, AV15 model: contour plots of the horizontal velocity component u (left) and the vertical velocity component w (right) at $t = 1020$ s computed with mesh $h = 125$ m.

stratified atmosphere with a uniform background potential temperature of 300 K perturbed by a circular bubble of warmer air. The initial velocity field is zero everywhere. Finally, the initial specific enthalpy is given by

$$h^0 = c_p \theta^0 \left(\frac{p}{p_g} \right)^{\frac{R}{c_p}}. \quad (51)$$

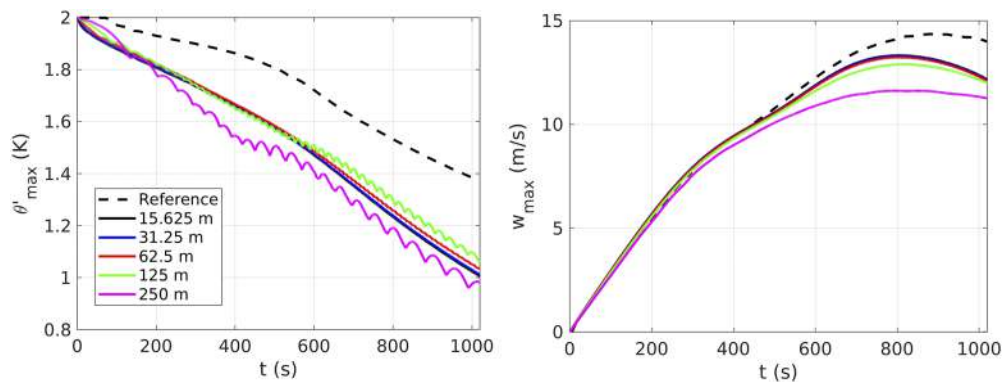


FIG. 4. Rising thermal bubble, AV15 model: time evolution of the maximum perturbation of potential temperature $\theta'_{\max}$ (left) and the maximum vertical component of the velocity $w_{\max}$ (right) computed with all the meshes under consideration. The reference values are taken from Ref. 30 and refer to resolution 125 m.

This benchmark is characterized by $Re \approx 1.8 \times 10^9$ and $M \approx 0.04$ when we set the characteristic quantities as follows: $L_r = 2000$ m (i.e., the initial bubble radius), $\rho_r = 1.2$ Kg/m³ and $\mu_r = 1.8 \times 10^{-5}$ Pa s (i.e., the air density and viscosity at 300 K), $T_r = 300$ K, and $U_r = 13.3$ m/s (i.e., the maximum vertical component of the velocity).

For this test, we use five different meshes with uniform resolution $h = \Delta x = \Delta z = 250, 125, 62.5, 31.25, 15.625$ m. The time step is set to $\Delta t = 0.1$ s for all the simulations. For stabilization, we consider two strategies. First, following Ref. 30, we set $\mu_a = 15$ and $Pr = 1$. Hereinafter, we refer to this model as AV15, where AV stands for artificial viscosity. We note that both of these are *ad hoc* values chosen by the authors of Ref. 30 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., Ref. 36). Other authors have chosen other arbitrary values, like $Pr = 0.1$ in Ref. 37. For a qualitative analysis of the results for the rising thermal bubble as Pr varies we refer to the reader to Ref. 38. After AV15, we consider the Smagorinsky model as described in Sec. II B.

Figure 2 reports the perturbation of potential temperature θ' at $t = 1020$ s computed by the AV15 model with all the meshes under consideration. By $t = 1020$, the air warmer than the ambient has risen due to buoyancy and deformed due to shearing motion. As a result, the bubble has evolved into a mushroom shape. From Fig. 2, we observe that the no substantial change in the computed θ' when the mesh is refined past $h = 125$ m. We remark that to facilitate the comparison of the panels in Fig. 2, we have forced the color bar to range from 0 to 1. Qualitatively, these results are in very good agreement with those reported in the literature. For example, see Refs. 30, 35, and 38.

For further qualitative assessment, Fig. 3 displays velocity components u and w at $t = 1020$ s computed by the AV15 model with mesh $h = 125$ m. These contour plots are in very good agreement with those reported in the literature. See, e.g., Fig. 7 in Ref. 30.

For a more quantitative comparison, in Fig. 4, we show the time evolution of the maximum perturbation of potential temperature $\theta'_{\max}$ and maximum vertical component of the velocity $w_{\max}$ computed by the AV15 model with all the meshes, together with the corresponding results from Ref. 30. We observe that the evolution

TABLE I. Rising thermal bubble, AV15 model: minimum and maximum vertical velocity w and potential temperature perturbation θ' at $t = 1020$ s compared with the values extracted from the figures in Ref. 30.

Model	Resolution (m)	$w_{\min}$ (m/s)	$w_{\max}$ (m/s)	$\theta'_{\min}$ (K)	$\theta'_{\max}$ (K)
AV15	15.625	-10.02	12.17	-0.038	1
AV15	31.25	-10.05	12.13	-0.037	1.01
AV15	62.5	-10.08	12.09	-0.037	1.03
AV15	125	-10.02	11.97	-0.042	1.06
AV15	250	-9.59	11.23	-0.069	0.89
Reference 30	125	-7.75	13.95	-0.01	1.4

of $\theta'_{\max}$ computed with meshes $h = 250$ m and $h = 125$ m is affected by spurious oscillations. Such oscillation disappears at higher resolution and they are not present at all in the evolution of $w_{\max}$. We see that $\theta'_{\max}$ and $w_{\max}$ computed with meshes $h = 31.25$ m and $h = 15.625$ m are practically overlapped for the entire time interval, indicating that we are close to convergence. The “converged” $w_{\max}$ overlaps with the reference value until about $t = 500$ s and it remains close to it until about $t = 800$ s. The agreement of the “converged” $\theta'_{\max}$ with the results from Ref. 30 is not as good. These trends are confirmed from Table I, which reports the extrema for the vertical

velocity w and potential temperature perturbation θ' at $t = 1020$ s obtained with the AV15 model, together with the values extracted from the figures in Ref. 30. The results from Ref. 30 are obtained with mesh resolution of 125 m and a density-based approach developed from a Godunov-type scheme that employs flux-based wave decompositions for the solution of Riemann problem. The authors of Ref. 30 start from the Euler equations written in density, velocity, and potential temperature. Other differences with our methodology include the orders of space and time discretization and the different treatment of the hydrostatic term. Given all these differences, we believe that our results cannot be considered off with respect to the reference. In addition, other references, such as Ref. 38 (see Fig. 1, top-left panel), report a $\theta'_{\max}$ at $t = 1020$ s computed with resolution 125 m closer to 1 K (like in our case) than to 1.4 K (like in Ref. 30).

Finally, we run this test with the Smagorinsky model. We set the parameter $C_s = 0.094$, with $C_k = 0.21$ and $C_e = 1.048$. Details on the choice of these parameters are reported in Sec. IV C. Figure 5 depicts the spatial distribution of θ' at $t = 1020$ s computed with meshes 15.625 and 32.25 m. As expected, with these fine meshes, the Smagorinsky model can capture a larger amount of vortical structures than the AV15 model. The beautiful vortex strip created by the Rayleigh–Taylor instability at the edge of the bubble shown in Fig. 5 (left) matches well with that shown in Ref. 38 (see Fig. 1, bottom-right panel). Table II reports the extrema for the vertical

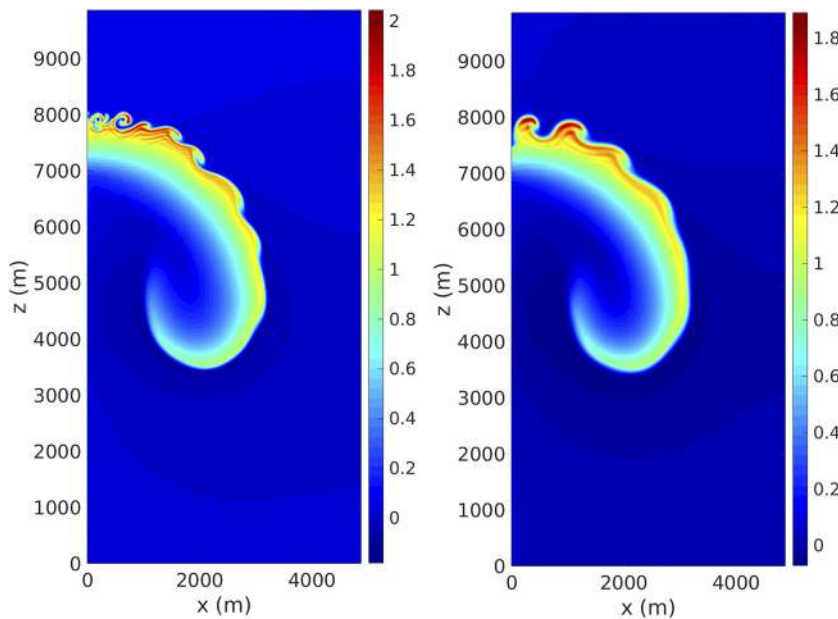


FIG. 5. Rising thermal bubble, Smagorinsky model: perturbation of potential temperature computed with mesh $h = 15.625$ m (left) and $h = 32.25$ m (right).

TABLE II. Rising thermal bubble, Smagorinsky model: minimum and maximum vertical velocity w and potential temperature perturbation θ' at $t = 1020$ s.

Model	Resolution (m)	$w_{\min}$ (m/s)	$w_{\max}$ (m/s)	$\theta'_{\min}$ (K)	$\theta'_{\max}$ (K)
Smagorinsky (LES)	15.625	-12.24	15.42	-0.18	2.04
Smagorinsky (LES)	31.25	-11.54	15.04	-0.072	1.89

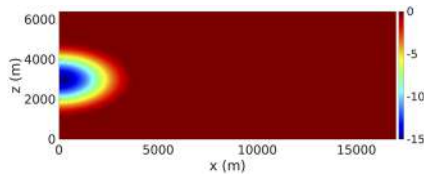


FIG. 6. Density current: initial potential temperature fluctuation θ' on part of the computational domain.

velocity w and potential temperature perturbation θ' at $t = 1020$ s obtained with the Smagorinsky model. Since we could not find any data obtained with an LES model for the exact setting of Ref. 30, Table II does contain reference data for comparison.

We conclude this section with a comment on the computational time required by the simulations to indicate the solver efficiency. We ran all the simulations using an 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50 GHz processor. The full simulation with the coarsest mesh we considered ($h = 250$ m) takes about 14 s. It was only run with the AV15 model. With the finest mesh ($h = 15.625$ m), the full simulation with the AV15 model takes about 35 min, while it takes roughly 57 min with the Smagorinsky model.

C. Density current

The computational domain in the xz -plane is $\Omega = [0, 25\,600] \times [0, 6400]$ m², 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 Eq. (49) with initial potential temperature,

$$\theta^0 = 300 - \frac{15}{2}[1 + \cos(\pi r)] \text{ if } r \leq 1, \quad \theta^0 = 300 \text{ otherwise,} \quad (52)$$

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 potential temperature fluctuation

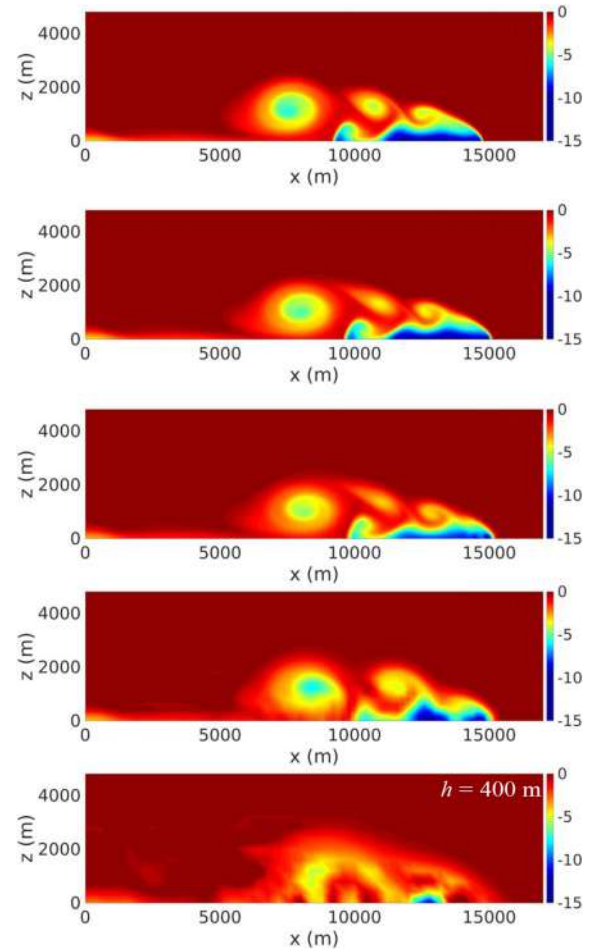


FIG. 8. Density current, AV75 model: potential temperature fluctuation θ' computed at $t = 900$ s with meshes $h = 25, 50, 100, 200, 400$ m. The mesh size is increasing from top to bottom.

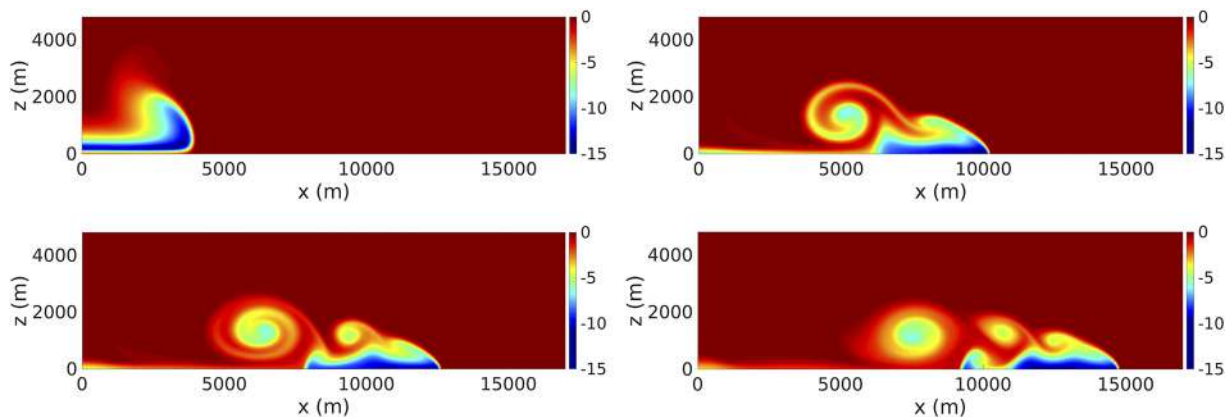


FIG. 7. Density current, AV75 model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 25$ m.

(10) on part of the domain Ω is shown in Fig. 6. Note that, in this case, the initial bubble is cold, while the bubble in Eq. (50) is warm. The initial velocity field is zero everywhere and the initial specific enthalpy is given by Eq. (51).

This benchmark is characterized by $Re \approx 9 \times 10^9$ and $M \approx 0.097$ when we set the characteristic quantities as follows: $L_r = 4000$ m (i.e., the semi-major axis of the initial perturbation), $\rho_r = 1.2$ Kg/m³ and $\mu_r = 1.8 \times 10^{-5}$ Pa s (i.e., the air density and viscosity at 300 K),

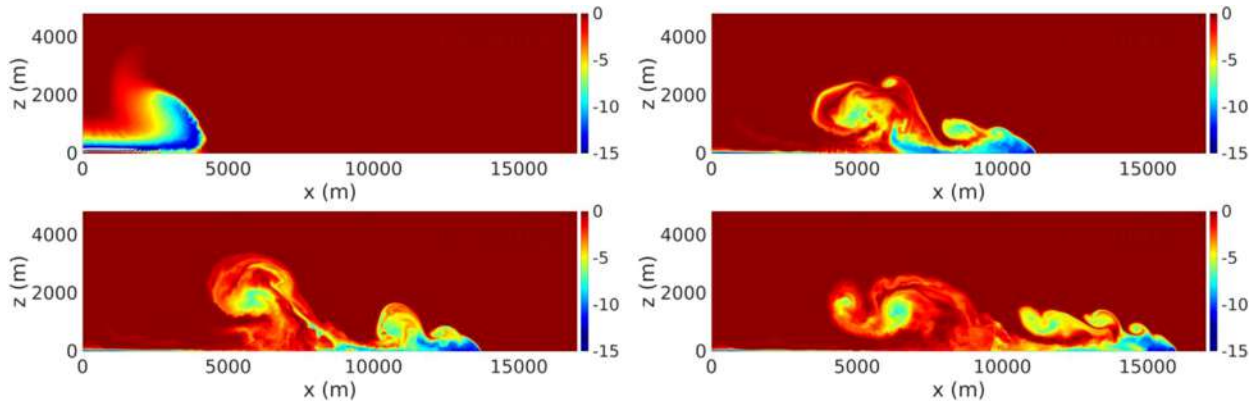


FIG. 9. Density current, Smagorinsky model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 12.5$ m.

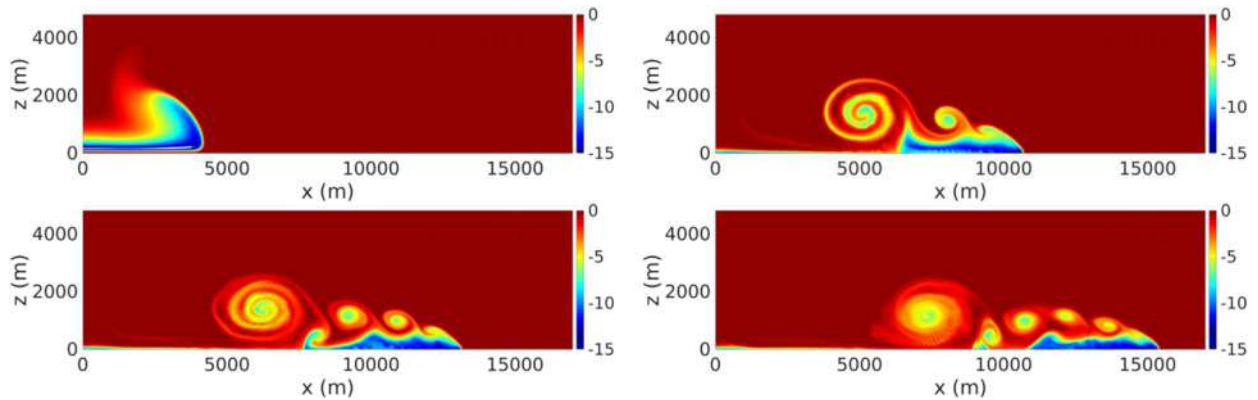


FIG. 10. Density current, Smagorinsky model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 25$ m.

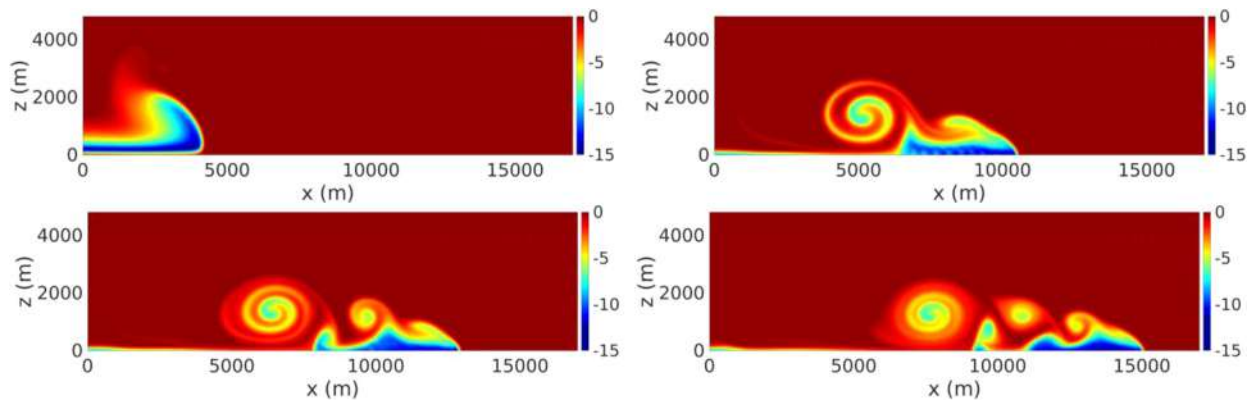


FIG. 11. Density current, Smagorinsky model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m.

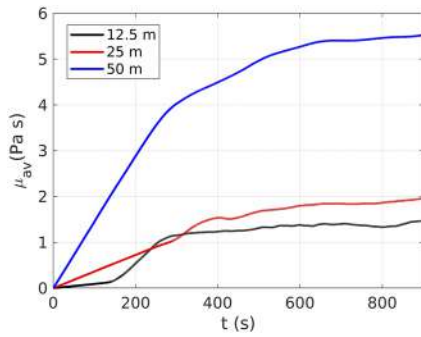


FIG. 12. Density current, Smagorinsky model: time evolution of the average eddy viscosity (53) for meshes $h = 12.5, 25, 50$ m.

$T_r = 300$ K, and $U_r = 33.7$ m/s (i.e., the maximum velocity magnitude at $t = 900$ s, which is the time the flow is fully developed).

We first consider uniform meshes with commonly used^{20,30,33,38–39} mesh sizes $h = \Delta x = \Delta z = 400, 200, 100, 50, 25$ m. The time step is set to $\Delta t = 0.1$ s. Since this benchmark features more complex vortical structures than the rising thermal bubble, we consider three stabilization strategies. Following Refs. 30 and 33, we first set $\mu_a = 75$ in Eqs. (11) and (12) and $\text{Pr} = 1$ in Eq. (12). We will refer to this model as AV75 since 75 is an *ad hoc* artificial value. Then, we consider the Smagorinsky and kEqn models as described in Sec. II B.

Let us start with the AV75 model and a qualitative illustration of the flow evolution. Figure 7 shows θ' computed with the AV75 model 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 Ref. 33, which were obtained with the same resolution. Indeed, we see that, as expected, the cold air descends due to negative buoyancy and strong downdrafts develop at the center of the cold bubble. When the cold air reaches the ground, it rolls up and forms a front. As this front propagates, shear is generated at its top boundary with a resulting

Kelvin–Helmholtz-type instability that leads to a three-rotor structure at $t = 900$ for the resolution under consideration.

In Fig. 8, we report θ' computed at $t = 900$ s by the AV75 model with all the meshes mentioned above. We observe the emergence of a clear three-rotor structure when the resolution is equal to or smaller than $h = 100$ m, with more definition when the mesh is finer. Also, the results in Fig. 8 are in very good agreement with those reported in the literature. See, e.g., Refs. 30, 33,³⁸ and ³⁹.

Next, we focus on the Smagorinsky model. Given the poor quality of the solutions computed with the AV75 model and meshes coarser than $h = 100$ m, we will use only the meshes finer than $h = 100$ m. Moreover, following Ref. 38 for the LES models, we will also consider a very fine mesh with size $h = 12.5$ m. Figures 9, 10, and 11 display the time evolution of the potential temperature fluctuation computed with the Smagorinsky model and meshes $h = 12.5, 25, 50$ m, respectively. For all these simulations, the parameter C_s in Eq. (13) was set to 0.454, with $C_k = 0.6$ and $C_e = 1.048$. We note that the ratio of the values of C_s used for this benchmark and for the rising thermal bubble is about 5, which is the ratio of the ad-hoc artificial viscosities. From Figs. 9–11, we clearly see that more vortical structures appear as the mesh size decreases. Indeed, we can see that with $h = 50$ m, there is a three-rotor structure at $t = 900$ s (see Fig. 11, bottom-right panel), and with $h = 25$ m, a quadri-rotor structure is present (see Fig. 10, bottom-right panel). The largest recirculation in the bottom-right panels of 11 and 10 is broken into two recirculations when mesh $h = 12.5$ m is used. See Fig. 9, bottom-right panel.

To further justify our choice of C_s , we report in Fig. 12 the time evolution of the space-averaged eddy viscosity,

$$\mu_{av} = \frac{1}{\Omega} \int_{\Omega} \mu_a d\Omega, \quad (53)$$

for meshes $h = 12.5, 25, 50$ m. We see that μ_{av} decreases with increasing resolution over most of the time interval. This is expected since the eddy viscosity (13) is a quadratic function of the filter width, which is related to the mesh size. However, even with the coarsest mesh (i.e., $h = 50$ m), the space-averaged eddy viscosity is much smaller than 75, i.e., the value used by the AV75 model. Values of

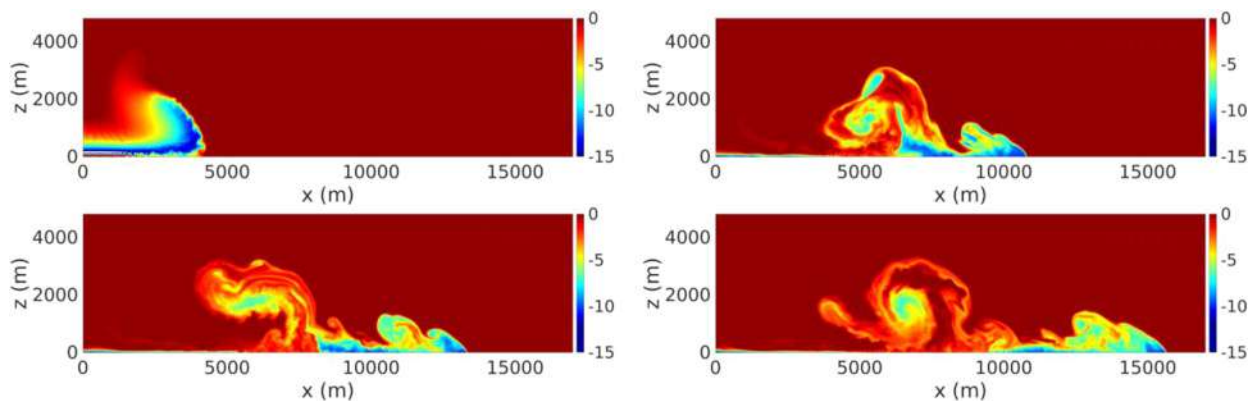


FIG. 13. Density current, kEqn model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 12.5$ m.

C_s smaller than 0.454 do not provide enough artificial dissipation to stabilize the solution, leading to under-resolved regions and a significant amount of noise above the recirculations. The value $C_s = 0.454$ ensures a good compromise between accuracy and stability and allows us to obtain a very good qualitative agreement with the solutions reported in Refs. 30, 33, and 38–39.

We now switch to the kEqn model. Figures 13, 14, and 15 display the time evolution of the potential temperature fluctuation computed with the kEqn model and meshes $h = 12.5, 25$, and 50 m, respectively. We observe a small difference between the solutions computed by the Smagorinsky model and the kEqn model with mesh $h = 50$ m for the entire duration of the time interval. Compare Fig. 11 with Fig. 15. The differences in the solutions given by the two LES models with mesh $h = 25$ m become visible around $t = 750$ s when the largest recirculation given by the kEqn model has a more flattened top boundary. In addition, the quadri-rotor structure at $t = 900$ s given by the Smagorinsky model is more defined. Compare the bottom panels in Fig. 10 with Fig. 14. This is expected since the kEqn model can be inaccurate for flows with strong streamline curvature and flows with large-scale turbulence structures as mentioned in Sec. II B. The differences in the solutions given the two LES models

with mesh $h = 12.5$ m are pretty remarkable already at $t = 600$ s. Compare Fig. 9 with Fig. 13.

Figure 16 shows the time evolution of the space-averaged eddy viscosity (53) given by the kEqn model for meshes $h = 12.5, 25$, and 50 m. By comparing Fig. 16 with Fig. 12, we observe that the two LES models have a very similar evolution of μ_{av} for mesh $h = 12.5$ m. For the other two meshes, the Smagorinsky model ramps up the value of μ_{av} faster than the kEqn model. Finally, we note that μ_{av} introduced by both models keeps increasing over the time interval $[0, 900]$ s since more and more vortical structures develop.

For a more quantitative comparison, we consider the front location, defined as the location on the ground where $\theta' = -1$ K, at $t = 900$ s. In Table III, we report the space interval that contains the front location for our implementations of the AV75, Smagorinsky, and kEqn models. We compare our values with the data in Ref. 33 (Table IV), which refer to the AV75 model and 14 different approaches for the numerical solution of this benchmark problem. We note that our results fall well within the values reported in Ref. 33. Furthermore, from Table III, we see that with our implementation of the Smagorinsky and kEqn models, the front becomes faster as the mesh is refined. This is more evident with the Smagorinsky

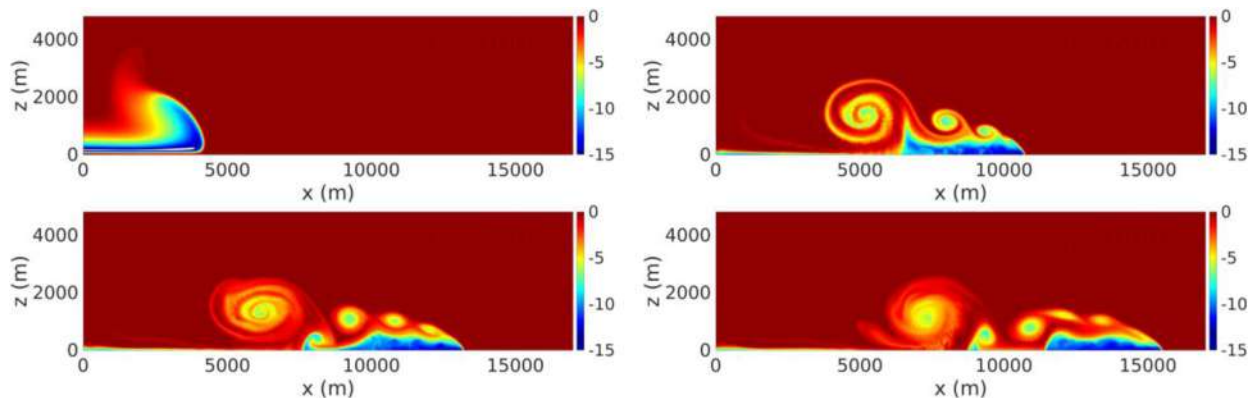


FIG. 14. Density current, kEqn model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 25$ m.

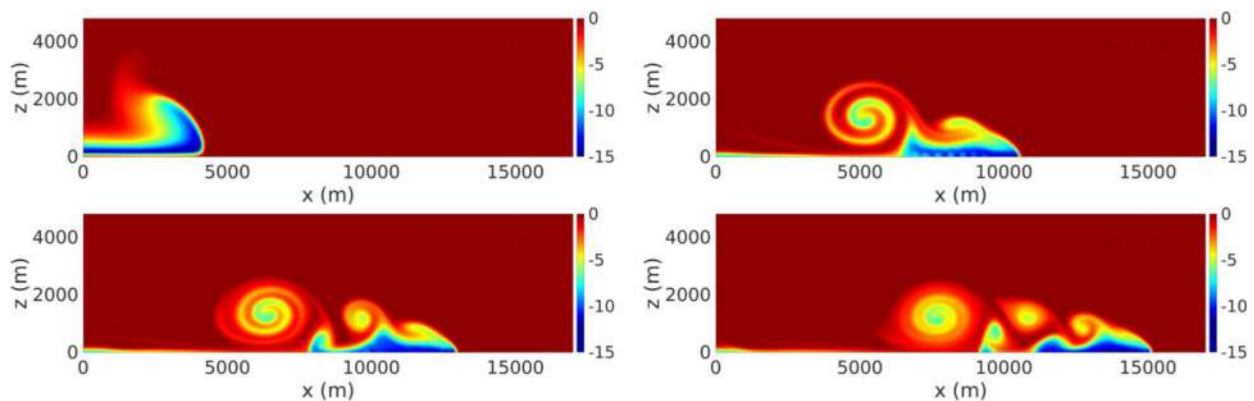


FIG. 15. Density current, kEqn model: time evolution of potential temperature fluctuation θ' computed with mesh $h = 50$ m.

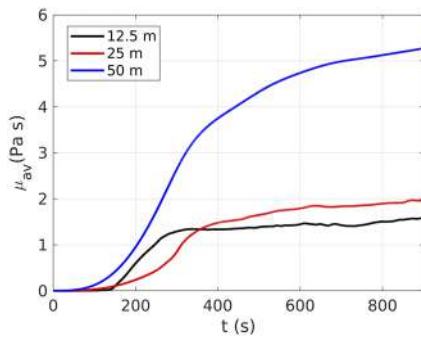


FIG. 16. Density current, kEqn model: time evolution of the average eddy viscosity (53) for meshes $h = 12.5, 25$, and 50 m.

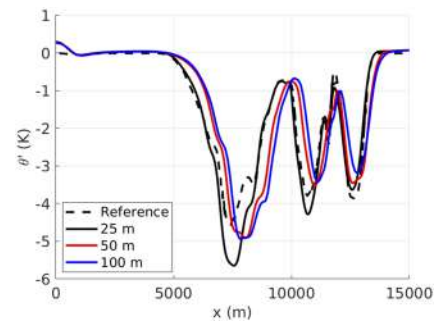


FIG. 17. Density current: potential temperature perturbation θ' at $t = 900$ s along the horizontal direction at a height of $z = 1200$ m given the AV75 model on meshes $h = 100, 50$, and 25 m and compared with data from Ref. 20 and resolution 25 m (denoted as "Reference").

TABLE III. Density current: our results for the front location at $t = 900$ s obtained with the AV75, Smagorinsky, and kEqn models and different meshes compared against results reported in Refs. 33 and 38. For Ref. 33, we reported the range of mesh sizes and front location values obtained with different methods. For Ref. 38, we report only the front location computed with the finest resolution.

Model	Resolution (m)	Front Location (m)
AV75	100	[15 269, 15 295]
AV75	50	[15 141, 15 167]
AV75	25	[14 783, 14 808]
Smagorinsky (LES)	50	[15 104, 15 130]
Smagorinsky (LES)	25	[15 411, 15 437]
Smagorinsky (LES)	12.5	[16 000, 16 026]
kEqn (LES)	50	[15 181, 15 206]
kEqn (LES)	25	[15 514, 15 539]
kEqn (LES)	12.5	[15 642, 15 667]
Reference 33	(25, 200)	(14 533, 17 070)
Reference 38	12.5	15 056

model than with the kEqn model, while the AV75 model shows the opposite trend. The front locations obtained with the AV75 model and different meshes are within about 500 m from each other. The same is true for the kEqn model, while the Smagorinsky model gives a larger variation in front location (about 900 m) as the mesh size varies. Since, in Ref. 33, no results are reported for mesh $h = 12.5$ m, in Table III, we list the front location found with an LES model and resolution 12.5 m in Ref. 38. Such front location is within about 500 m of the value we obtain with the kEqn model and about 900 m of the value we obtain with the Smagorinsky model.

For a further quantitative comparison, we consider the potential temperature perturbation θ' at $t = 900$ s along the horizontal direction at a height of $z = 1200$ m. Figure 17 displays a comparison between the results given the AV75 model on meshes $h = 100, 50, 25$ m and the results obtained with the same model and resolution 25 m in Ref. 20. Two numerical methods were used in Ref. 20 to approximate the solution: a spectral element method and a discontinuous Galerkin method, which provide curves that are superimposed in Fig. 17 and are denoted as "Reference." We observe that our results are slightly out of phase for meshes $h = 50, 100$ m

with respect to the data in Ref. 20. The potential temperature fluctuation obtained with mesh $h = 25$ m is in phase but has larger negative peaks, especially in correspondence of the first recirculation. However, given all the differences between our approach and the one in Ref. 20, the comparison is satisfactory.

As for the previous benchmark, we conclude by providing the computational times. The full simulation with the coarsest mesh ($h = 400$ m) takes about 14 s. It was run only with the AV75 model. The full simulation with the finest mesh ($h = 25$ m) takes about 53 min with the AV15 model, 59 min with the Smagorinsky model, and 1 h with the kEqn model.

V. CONCLUDING REMARKS

The goal of this paper is to show that OpenFOAM provides a reliable and accurate framework for the simulation of non-hydrostatic atmospheric flows. To achieve this goal, we developed a pressure-based solver for the Euler equations written in conservative form using density, momentum, and total energy as variables and tested it against numerical data available in the literature for two well-known benchmarks, namely, the rising thermal bubble and the density current. For the rising bubble and the density current tests, we obtain good qualitative and quantitative comparisons when the classical Smagorinsky model and/or the one equation eddy-viscosity model are adopted for stabilization. We observed a small difference between the solutions computed by the Smagorinsky model and the one equation eddy-viscosity model for coarser meshes. The differences become visible for finer meshes, and the finer the mesh is, the earlier they appear. For both LES models, the front in the density current benchmark becomes faster as the mesh is refined. However, while the front locations obtained with the one equation eddy-viscosity model and different meshes are within about 500 m from each other, a larger variation in front location (about 900 m) is given by the Smagorinsky model as the mesh size varies.

ACKNOWLEDGMENTS

We thank Dr. S. Marras for the fruitful conversations. 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 the U.S. National Science Foundation through Grant No. DMS-1953535 (PI A. Quaini). A. Quaini acknowledges the support from the Radcliffe Institute for Advanced Study at Harvard University, where she has been the 2021–2022 William and Flora Hewlett Foundation Fellow.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Michele Girfoglio: Investigation (equal); Methodology (equal); Validation (equal). **Annalisa Quaini:** Investigation (equal); Methodology (equal); Validation (equal). **Gianluigi Rozza:** Funding acquisition (lead); Supervision (lead).

DATA AVAILABILITY

The code created for this paper is available within open-source package Geophysical and Environmental Applications (GEA).⁷

REFERENCES

- Community Earth system model, <https://www.cesm.ucar.edu/>.
- Energy exascale Earth system model, <https://e3sm.org/about/>.
- Massachusetts Institute of Technology general circulation model, <http://mitgcm.org/>.
- Climate modeling alliance, <https://github.com/CliMA>.
- Weather research and forecasting, <https://www.mmm.ucar.edu/models/wrf>.
- Atlas—A library for numerical weather prediction and climate modelling, <https://sites.ecmwf.int/docs/atlas/>.
- GEA—Geophysical and environmental applications, <https://github.com/GEA-Geophysical-and-Environmental-Apps/GEA>.
- M. Girfoglio, A. Quaini, and G. Rozza, “A novel large eddy simulation model for the quasi-geostrophic equations in a finite volume setting,” *J. Comput. Appl. Math.* **418**, 114656 (2023).
- M. Girfoglio, A. Quaini, and G. Rozza, “A linear filter regularization for POD-based reduced order models of the quasi-geostrophic equations,” *C. R. Mec.* **351**, 1–21 (2023).
- H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques,” *Comput. Phys.* **12**, 620–631 (1998).
- N. Godinaud, P. Boivin, P. Freton, J.-J. Gonzalez, and F. Camy-Peyret, “Development of a new OpenFOAM solver for plasma cutting modeling,” *Comput. Fluids* **241**, 105479 (2022).
- G. Maragkos, S. Verma, A. Trounev, and B. Merci, “Evaluation of OpenFOAM’s discretization schemes used for the convective terms in the context of fire simulations,” *Comput. Fluids* **232**, 105208 (2022).
- V. Skurić, P. D. Jaeger, and H. Jasak, “Lubricated elastoplastic contact model for metal forming processes in OpenFOAM,” *Comput. Fluids* **172**, 226–240 (2018).
- G. Kristóf, N. Rácz, and M. Balogh, “Adaptation of pressure based CFD solvers for mesoscale atmospheric problems,” *Bound.-Layer Meteorol.* **131**, 85–103 (2009).
- M. Balogh, A. Parente, and C. Benocci, “RANS simulation of ABL flow over complex terrains applying an enhanced $k - \epsilon$ model and wall function formulation: Implementation and comparison for fluent and OpenFOAM,” *J. Wind Eng. Ind. Aerodyn.* **104–106**, 360–368 (2012), 13th International Conference on Wind Engineering.
- F. Flores, R. Garraud, and R. C. Muñoz, “CFD simulations of turbulent buoyant atmospheric flows over complex geometry: Solver development in OpenFOAM,” *Comput. Fluids* **82**, 1–13 (2013).
- H. Weller, W. McIntyre, and D. Shipley, “Multifluids for representing subgrid-scale convection,” *J. Adv. Model. Earth Syst.* **12**, e2019MS001966 (2020).
- H. Yamazaki, H. Weller, C. J. Cotter, and P. A. Browne, “Conservation with moving meshes over orography,” *J. Comput. Phys.* **461**, 111217 (2022).
- AtmosFOAM, <https://github.com/AtmosFOAM/AtmosFOAM>.
- F. X. Giraldo and M. Restelli, “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.* **227**, 3849–3877 (2008).
- J. Smagorinsky, “General circulation experiments with the primitive equations: I. The basic experiment,” *Mon. Weather Rev.* **91**, 99–164 (1963).
- A. Yoshizawa and K. Horiuti, “A statistically-derived subgrid-scale kinetic energy model for the large-eddy simulation of turbulent flows,” *J. Phys. Soc. Jpn.* **54**, 2834–2839 (1985).
- J. F. Kelly and F. X. Giraldo, “Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: Limited-area mode,” *J. Comput. Phys.* **231**, 7988–8008 (2012).
- M. Girfoglio, A. Quaini, and G. Rozza, “A finite volume approximation of the Navier–Stokes equations with nonlinear filtering stabilization,” *Comput. Fluids* **187**, 27–45 (2019).
- H. Jasak, “Error analysis and estimation for the finite volume method with applications to fluid flows,” Ph.D. thesis, Imperial College, University of London, 1996.
- H. J. Aguerre, C. I. Pairetti, C. M. Venier, S. Márquez Damián, and N. M. Nigro, “An oscillation-free flow solver based on flux reconstruction,” *J. Comput. Phys.* **365**, 135–148 (2018).
- S. V. Patankar and D. B. Spalding, “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows,” *Int. J. Heat Mass Transfer* **15**, 1787–1806 (1972).
- R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting,” *J. Comput. Phys.* **62**, 40–65 (1986).
- F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*, 1st ed. (Springer Publishing Company, Incorporated, 2015).
- N. Ahmad and J. Lindeman, “Euler solutions using flux-based wave decomposition,” *Int. J. Numer. Methods Fluids* **54**, 47–72 (2007).
- Y. Feng, J. Miranda-Fuentes, J. Jacob, and P. Sagaut, “Hybrid lattice Boltzmann model for atmospheric flows under anelastic approximation,” *Phys. Fluids* **33**, 036607 (2021).
- R. L. Carpenter, K. K. Droegemeier, P. R. Woodward, and C. E. Hane, “Application of the piecewise parabolic method (PPM) to meteorological modeling,” *Mon. Weather Rev.* **118**, 586–612 (1990).
- J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier, “Numerical solution of a nonlinear density current: A benchmark solution and comparisons,” *Int. J. Numer. Methods Fluids* **17**, 1–22 (1993).
- N. Botta, R. Klein, S. Langenberg, and S. Lützenkirchen, “Well balanced finite volume methods for nearly hydrostatic flows,” *J. Comput. Phys.* **196**, 539–565 (2004).
- N. N. Ahmad, “High-resolution wave propagation method for stratified flows,” in *AIAA Aviation Forum* (AIAA, Atlanta, GA, 2018).

³⁶M. Restelli and F. X. Giraldo, “A conservative discontinuous Galerkin semi-implicit formulation for the Navier–Stokes equations in nonhydrostatic mesoscale modeling,” *SIAM J. Sci. Comput.* **31**, 2231–2257 (2009).

³⁷M. Nazarov and J. Hoffman, “Residual-based artificial viscosity for simulation of turbulent compressible flow using adaptive finite element methods,” *Int. J. Numer. Methods Fluids* **71**, 339–357 (2013).

³⁸S. Marras, M. Nazarov, and F. X. Giraldo, “Stabilized high-order Galerkin methods based on a parameter-free dynamic SGS model for LES,” *J. Comput. Phys.* **301**, 77–101 (2015).

³⁹S. Marras, M. Moragues, M. Vázquez, O. Jorba, and G. Houzeaux, “A variational multiscale stabilized finite element method for the solution of the Euler equations of nonhydrostatic stratified flows,” *J. Comput. Phys.* **236**, 380–407 (2013).