Journal of Computational Physics 459 (2022) 111137

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A VMS–based fractional step technique for the compressible Navier–Stokes equations using conservative variables

Samuel Parada [a,b], Ramon Codina [a,b,*], Joan Baiges [b]

[a] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Campus Nord UPC, Ed. C1, C/Gran Capità s/n, 08034 Barcelona, Spain
[b] Universitat Politècnica de Catalunya – Barcelona Tech, Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports de Barcelona (ETSECCPB), Campus Nord UPC, Ed. C1, C/Jordi Girona 1, 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 23 September 2021
Received in revised form 4 March 2022
Accepted 5 March 2022
Available online 14 March 2022

Keywords:
Compressible flow
Variational Multiscale Method (VMS)
Fractional step schemes
Finite element method
Supersonic flows

ABSTRACT

In this paper we address the compressible Navier–Stokes equations written in the so-called conservative formulation. In particular, we focus on the possibility of uncoupling the computation of the problem unknowns, namely, density, linear momentum and total energy, a technique usually labeled as fractional step method, which allows to reduce the associated computational cost. The proposed methodology is a finite-element solver supplemented with a stabilization technique within the Variational Multi-Scale framework. In this regard, we consider orthogonal and dynamic definitions for the subscales. This discretization in space shows an adequate stability, permitting in particular the use of equal interpolation for all variables in play. However, we complement it with a shock-capturing operator in order to solve problems involving shocks. Several representative benchmark flow simulations are performed, which demonstrate the suitability of the proposed algorithm for a vast range of regimes.

1. Introduction

The so-called compressible Navier–Stokes equations are commonly used to model flow problems where compressibility effects become relevant, e.g., in the aerodynamic and aeroacoustic fields, with applications ranging from classical turbo-machinery design [1] to modern speech therapy simulations [2]. The general mathematical setting consists of the momentum, mass and energy conservation equations together with thermodynamic properties, constitutive relations and proper initial and boundary conditions, which are appended to close the mathematical description ensuring the well-posedness of the problem. Such a set of partial differential equations describes a wide range of scales and, in general, computing its solution is a challenge in itself. One could proceed either by choosing small mesh and time step sizes or by using high precision numerical schemes. Regardless of the selected approach, obtaining a representative solution is particularly demanding from the computational viewpoint. This fact still remains as one of the main limitations in compressible flow simulations in spite of the increasing amount of computing facilities available for the scientific and engineering communities.

In this work, we focus on the finite element approximation of the Navier–Stokes compressible flow problem. Particularly, we aim at solving it in a segregated manner, that is to say, by uncoupling the calculation of the problem unknowns. These are the well-known conservative variables, i.e., density, momentum, and total energy. However, when these equations are

* Corresponding author.
E-mail addresses: samuel.parada@upc.edu (S. Parada), ramon.codina@upc.edu (R. Codina), joan.baiges@upc.edu (J. Baiges).

https://doi.org/10.1016/j.jcp.2022.111137
0021-9991/

approximated by the classical raw Galerkin approach, numerical instabilities may appear mainly due to the hyperbolic nature of the equations. These inconveniences can be overcome by resorting to a stabilized finite element formulation. In this regard, the Streamline Upwind Petrov Galerkin (SUPG) method [3] appeared as one of the first approaches, which was originally introduced as an extension of the previously developed stabilization methods for convection-diffusion flow problems. The foundation of the method was to introduce numerical diffusion along the streamlines in an optimal manner by defining a certain stabilization term. Such stabilization term contained a matrix of algorithmic parameters, an operator applied to the test function, and the residual of the differential equation (see [4]). Later, certain modifications to that operator were introduced, giving raise to the commonly named Galerkin Least Squares (GLS) method (see e.g., [5,6]).

The multi-scale concept was first presented in the context of compressible Navier–Stokes formulations as a turbulence model. For this reason, it originally played the role of a numerical artifact to account for the effect of the unresolved scales rather than a stabilization method per se. In this regard, [7] introduced a mixed formulation to approximate the solution by separating in advance the resolved and unresolved turbulent scales, whose effect is modeled using a Reynolds stress tensor. This initial work was contrasted in [8], specifically in the definition of the unresolved scales, proposing a Fourier-Spectral projector, which could be implemented with a discontinuous Galerkin method. More examples on this topic are [9,10] which also include turbulence energy spectra results for turbulent compressible flows. However, the Variational Multi-Scale (VMS) framework serves here as a source for the design of a stabilization technique with different capabilities. The idea of understanding VMS methods as a stabilization technique dates back to the original works of [11,12] and more recently to [13]. The cornerstone of this approach is the splitting of the unknowns of the problem into two components, namely, a coarse-scale that belongs to the finite element space and a subgrid scale or *subscale*, which is the remainder. Therefore, the original problem can be divided into two separated subproblems, one involving the finite element scales, and another for the subscales. For an in-depth review of the application of the VMS framework in CFD we refer the reader to [14]. Specifically in the context of compressible flows, [15] presents a review of different stabilized methods for compressible flow computations with a historical perspective from initial developments to modern approaches. In addition, [16] was the first attempt to introduce stabilization techniques strictly in the frame of VMS methods. In these references, VMS-based formulations previously developed for the incompressible case are satisfactorily extended to its compressible counterpart. More recently, in [17] the VMS method was applied in order to stabilize the Euler equations, where the authors demonstrated the convergence of the numerical method in a wide range of stratified flows, yet they restricted the explicit formulation to a linear Euler time integration scheme.

Although global stability is ensured by a VMS-based stabilized formulation, some localized oscillations may arise from sharp gradients in the solution, particularly at supersonic regimes. This effect is innate to compressible flows involving shocks (discontinuities). Hence, stabilized formulations usually need to be complemented with a local shock-capturing methodology. A possible approach to model shocks is the residual-based shock-capturing technique, first introduced in [18] and later tested in [19] for a SUPG compressible flow formulation. Another popular option is the so-called "YZβ" method from [20]. In contrast to the previous formulations, our strategy is to introduce the numerical diffusion in a "physical manner", that is to say, we shall modify the diffusion of the momentum and energy equations, but we avoid introducing artificial diffusion into the mass equation.

Nevertheless, in this work our main goal is the development of segregation methods for the transient compressible Navier–Stokes equations using a finite element approximation for the space discretization. As a reference in the comparisons, we shall take the solution of the so-called *monolithic* problem, i.e., the standard coupled calculation involving all the problem unknowns. The fully discrete and linearized monolithic scheme leads to an algebraic system, whose structure may be exploited in order to solve independently for the density, momentum and energy degrees of freedom. Referring to the time integration, on the following we shall concentrate on first and second order implicit finite difference schemes. The backward Euler method will be used for the former, whereas for second order methods we will stick to backward differentiation schemes, also known as Gear schemes. However, our developments are not restricted and, in principle, any other discretization methods might be used to advance the solution in time.

The technique we will discuss here corresponds to the classical *fractional step* algorithms, which might be seen as an alternative methodology to solve transient problems. Fractional step methods, commonly referred in the literature as splitting or projection methods, started to gain adepts after the original works published in [21] and [22]. In these publications, the uncoupling of velocity and pressure unknowns at the continuous level is attempted for the incompressible equations. Since then, many works have been devoted to a proper understanding of the original schemes, their numerical analysis, their extension to higher order approximations and to the design of adequate boundary conditions (see e.g. [23] and references therein). Nonetheless, our approach in this work is to present a splitting of the equations at the algebraic level once the equations have already been discretized in space and in time. This way to face the problem emerged after the identification in [24] of the classical pressure segregation method as an inexact factorization of the final algebraic system. This algebraic viewpoint is generally simpler and it makes possible to obviate a discussion on specific boundary conditions for the different stages of the fractional step scheme (see e.g. [25,26]). Furthermore, it has been extensively and effectively applied to a wide range of different test cases in computational physics, including incompressible [25,27], viscoelasticy [28], isentropic [29] and compressible (primitive variables) [30] problems. The literature regarding fractional step schemes for compressible flow applications is not as vast as for the incompressible problem. However, many innovative and diverse collections of uncoupling techniques [5,31–35] have been published, which involve distinct spatial discretizations (finite differences,

finite volumes, finite elements), temporal schemes (explicit, implicit, semi-implicit) and also different sets of unknowns (conservative, primitive and even combinations of both types of variables).

The article is structured as follows: in Section 2 we introduce the compressible Navier–Stokes problem together with its variational formulation. In Section 3 we present the numerical approximation including different features such as the VMS-based stabilized formulation and the shock-capturing technique. Section 4 is devoted to the design of the fractional step scheme. Numerical experiments are conducted in Section 5, and, finally, we close the paper with some concluding statements in Section 6.

2. Problem statement

2.1. Initial and boundary value problem

Let Ω be an open, bounded and polyhedral domain of $\mathbb{R}^{n_{\text{sd}}}$ ($n_{\text{sd}} = 2$ or 3 refers to the number of space dimensions) and $[0, t_{\text{f}}]$ the time interval of analysis. The Navier–Stokes equations, governing unsteady viscous compressible flows are formulated as the following system of partial differential equations:

$$\partial_t \rho + \operatorname{div}(\rho \mathbf{u}) = 0 \qquad \text{in } \Omega \times (0, t_{\text{f}}), \tag{1a}$$

$$\partial_t (\rho \mathbf{u}) + \operatorname{div}(\rho \mathbf{u} \otimes \mathbf{u} - \boldsymbol{\sigma}^{\text{d}}) + \mathbf{grad}\, p = \rho \mathbf{f} \qquad \text{in } \Omega \times (0, t_{\text{f}}), \tag{1b}$$

$$\partial_t (\rho e) + \operatorname{div}((\rho e + p)\mathbf{u} + \mathbf{q} - \boldsymbol{\sigma}^{\text{d}} \mathbf{u}) = \rho(\mathbf{f} \cdot \mathbf{u} + r) \qquad \text{in } \Omega \times (0, t_{\text{f}}), \tag{1c}$$

where we have made use of the short notation for the partial time derivative $\partial(\diamond)/\partial t := \partial_t(\diamond)$. Here and in what follows, $\rho(\mathbf{x}, t)$, $\mathbf{u}(\mathbf{x}, t)$, $p(\mathbf{x}, t)$, $\boldsymbol{\sigma}^{\text{d}}(\mathbf{x}, t)$, $e(\mathbf{x}, t)$ and $\mathbf{q}(\mathbf{x}, t)$ denote, respectively, density, velocity, pressure, viscous stress tensor, total energy per unit mass, and heat flux vector. In addition to this, $\mathbf{f}(\mathbf{x}, t)$ is a possible external acceleration and $r(\mathbf{x}, t)$ is a possible heat source, which may include chemical reactions or even electromagnetic effects.

The viscous stress tensor is defined as

$$\boldsymbol{\sigma}^{\text{d}} = \boldsymbol{\sigma}^{\text{d}}(\mathbf{u}(\mathbf{x}, t)) := 2\mu \boldsymbol{\varepsilon} + (\mu_{\text{B}} + \lambda)(\operatorname{div} \mathbf{u}) \mathbf{I}_{n_{\text{sd}}}, \tag{2}$$

where $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, t)) := \mathbf{grad}^{\text{s}}(\mathbf{u})$ is the strain rate tensor computed as the symmetric gradient of velocity, i.e., $\mathbf{grad}^{\text{s}}(\diamond) := (\mathbf{grad}(\diamond) + \mathbf{grad}^{\intercal}(\diamond))/2$, where $(\bullet)^{\intercal}$ denotes the transpose of $(\bullet)$. Likewise, $\mu(\mathbf{x}, t)$, $\mu_{\text{B}}(\mathbf{x}, t)$ and $\lambda(\mathbf{x}, t)$ are respectively, the so-called *molecular* (or *shear*), *bulk* (or *volumetric*) and *second* viscosity coefficients, non-constant in the general case and positive.

The heat flux vector is calculated using Fourier's law of heat conduction, which in general can be written as $\mathbf{q} = -\mathbf{grad}\ g(\vartheta)$ where g is a nonlinear function of the temperature $\vartheta(\mathbf{x}, t)$. Although nonlinear diffusion problems are often found in practice, we will restrict ourselves to the classical linear relation in order to ease the discussion. Hence we write

$$\mathbf{q} := -\mathbf{k}\, \mathbf{grad}\, \vartheta, \tag{3}$$

where $\mathbf{k}$ is the (symmetrical second-order) tensor of thermal conductivity, which is a property of the fluid. For the isotropic case, the thermal conductivity tensor is a spherical tensor $\mathbf{k} = \kappa \mathbf{I}_{n_{\text{sd}}}$ and depends on the scalar parameter $\kappa(\mathbf{x}, t)$, which is the thermal conductivity of the fluid.

Henceforth, we will make use of the most frequently encountered form of the thermal equation of state, i.e., the ideal-gas law,

$$p = \rho R_{\text{g}} \vartheta, \tag{4}$$

where R_{g} is the constant of the gas under consideration, defined as $R_{\text{g}} = R^0 M_{\text{w}}^{-1}$ being $R^0 = 8.31$ J/(mol K) the universal gas constant and M_{w} the molecular weight. For a calorically perfect gas, the internal energy ι is a sole function of the temperature, i.e.,

$$\iota = \iota(\vartheta) := c_v \vartheta, \tag{5}$$

and hence,

$$e = c_v \vartheta + \frac{1}{2} \|\mathbf{u}\|^2. \tag{6}$$

In the sequel, c_v denotes the specific heat at constant volume, and c_p stands for the specific heat at constant pressure. We also define the ratio of specific heats as $\gamma := c_p / c_v$.

Remark 2.1. A common practice in the analysis of the motion of compressible fluids is to make use of the well-known Stokes hypothesis, i.e., $\lambda + \frac{2}{3}\mu = \mu_B = 0$. Setting $\mu_B = 0$ is supported by the kinetic theory of gases and such assumption renders the mathematical treatment of compressible flows notably easier, yet it has been the object of long-lasting discussions on compressible flows simulations.

Remark 2.2. According to experimental evidence, only in very particular conditions will the term $\mu_B(\operatorname{div}\mathbf{u})$ be of practical significance. This may happen for instance when the fluid exhibits large values of μ_B (e.g., CO_2), or the motion is such that extremely large values of div $\mathbf{u}$ occur, for example in hypersonic flows, which are out of the scope of the present article.

Remark 2.3. Although μ and κ might be assumed to be constant to ease the discussion, several models can be introduced in order to reproduce more realistic conditions. Apart from the classical Sutherland law which makes the variables temperature dependent, other expressions based on the kinetic theory can be considered such as the Chapman-Cowling relation for μ or the modified Eucken correction formula for κ (see e.g. [36–38]).

2.2. Compact and quasi-linear form of the problem

The Navier–Stokes equations of compressible flow can be compactly written as

$$\partial_t \boldsymbol{\mathcal{U}} + \operatorname{div}\Big(\boldsymbol{\mathcal{F}}^{\mathrm{c}}(\boldsymbol{\mathcal{U}}) + \boldsymbol{\mathcal{F}}^{\mathrm{d}}(\boldsymbol{\mathcal{U}}, \mathbf{grad}\,\boldsymbol{\mathcal{U}})\Big) = \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{U}}) \quad \text{in } \Omega \times (0, t_{\mathrm{f}}), \tag{7}$$

where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_{\mathrm{sd}}+2}$ is the vector of conservative variables, $\boldsymbol{\mathcal{F}}^{\mathrm{c}}(\boldsymbol{\mathcal{U}})$ and $\boldsymbol{\mathcal{F}}^{\mathrm{d}}(\boldsymbol{\mathcal{U}}, \mathbf{grad}\,\boldsymbol{\mathcal{U}}) \in \mathbb{R}^{(n_{\mathrm{sd}}+2)\times n_{\mathrm{sd}}}$ are the convective and diffusion flux tensors, and $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{U}}) \in \mathbb{R}^{n_{\mathrm{sd}}+2}$ is the vector of external forcing terms, all of them respectively given by

$$\boldsymbol{\mathcal{U}} := \begin{Bmatrix} \rho \\ \rho\mathbf{u} \\ \rho e \end{Bmatrix}, \qquad \boldsymbol{\mathcal{F}}^{\mathrm{c}}(\boldsymbol{\mathcal{U}}) = \begin{bmatrix} \rho\mathbf{u}^{\mathsf{T}} \\ \rho\mathbf{u}\otimes\mathbf{u} + p\mathbf{I}_{n_{\mathrm{sd}}} \\ (\rho e + p)\mathbf{u}^{\mathsf{T}} \end{bmatrix}, \tag{8a}$$

$$\boldsymbol{\mathcal{F}}^{\mathrm{d}}(\boldsymbol{\mathcal{U}}, \mathbf{grad}\,\boldsymbol{\mathcal{U}}) = \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\sigma}^{\mathrm{d}} \\ \left(\mathbf{q} - \boldsymbol{\sigma}^{\mathrm{d}}\mathbf{u}\right)^{\mathsf{T}} \end{bmatrix}, \qquad \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{U}}) = \begin{Bmatrix} \mathbf{0} \\ \rho\mathbf{f} \\ \rho(r + \mathbf{f}^{\mathsf{T}}\cdot\mathbf{u}) \end{Bmatrix}. \tag{8b}$$

However, the divergence of the convective and diffusive flux tensors from the original system, Eq. (7), can be written in a more convenient manner upon the definition of the so-called Euler Jacobian and diffusion matrices. Making use of index notation, those are related to the convective and diffusive flux tensors as follows:

$$\boldsymbol{\mathcal{A}}_j(\boldsymbol{\mathcal{U}}) := \frac{\partial \boldsymbol{\mathcal{F}}^{\mathrm{c}}_j(\boldsymbol{\mathcal{U}})}{\partial \boldsymbol{\mathcal{U}}} \qquad \forall\, j = 1, \ldots, n_{\mathrm{sd}}, \tag{9a}$$

$$\frac{\partial \boldsymbol{\mathcal{F}}^{\mathrm{d}}_j(\boldsymbol{\mathcal{U}})}{\partial x_j} := -\frac{\partial}{\partial x_k}\left(\boldsymbol{\mathcal{K}}_{kj}(\boldsymbol{\mathcal{U}})\frac{\partial \boldsymbol{\mathcal{U}}}{\partial x_j}\right) \qquad \forall\, j,k = 1, \ldots, n_{\mathrm{sd}}. \tag{9b}$$

The expressions of $\boldsymbol{\mathcal{A}}_j(\boldsymbol{\mathcal{U}})$ and $\boldsymbol{\mathcal{K}}_{kj}(\boldsymbol{\mathcal{U}})$ are given in Appendix A. The last term in Eq. (7), the vector of sources, can also be rewritten by means of a reactive-like term of the form,

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{U}}) := \boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{U}}. \tag{10}$$

Therefore, taking all this information into account, the original compressible Navier–Stokes system can be now stated as: find the set of conservative unknowns $\boldsymbol{\mathcal{U}}$ such that the following is satisfied,

$$\partial_t \boldsymbol{\mathcal{U}} + \mathcal{L}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}}) = \mathbf{0} \quad \text{in } \Omega \times (0, t_{\mathrm{f}}). \tag{11}$$

The second-order nonlinear operator $\mathcal{L}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}})$ includes the previous definitions for the convective, diffusive, and reactive terms and it is written as

$$\mathcal{L}(\boldsymbol{\mathcal{U}}_0; \boldsymbol{\mathcal{U}}) := \boldsymbol{\mathcal{A}}_j(\boldsymbol{\mathcal{U}}_0)\frac{\partial \boldsymbol{\mathcal{U}}}{\partial x_j} - \frac{\partial}{\partial x_k}\left(\boldsymbol{\mathcal{K}}_{kj}(\boldsymbol{\mathcal{U}}_0)\frac{\partial \boldsymbol{\mathcal{U}}}{\partial x_j}\right) - \boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{U}} \qquad \forall\, j,k = 1, \ldots, n_{\mathrm{sd}}, \tag{12}$$

being non–linear in the first argument due to the dependency of the Euler Jacobian and diffusivity matrices on the unknowns.

The compressible Navier–Stokes problem is a non-linear initial and boundary value problem of parabolic–hyperbolic type which needs to be complemented with appropriate initial and boundary conditions. In general, these can be expressed in vector form as

$$n_k\left(\mathcal{K}_{kj}(\mathcal{U})\frac{\partial \mathcal{U}}{\partial x_j}\right)=\mathbf{h} \qquad \text{on } \partial\Omega_{\mathrm{N}},\ t\in(0,t_{\mathrm{f}}), \tag{13a}$$

$$\mathcal{D}\mathcal{U}=\mathcal{D}\mathcal{U}_{\mathrm{D}} \qquad \text{on } \partial\Omega_{\mathrm{D}},\ t\in(0,t_{\mathrm{f}}), \tag{13b}$$

$$\mathcal{U}=\mathcal{U}_0(\mathbf{x}), \qquad \text{in } \Omega,\ t=0. \tag{13c}$$

Neumann boundary conditions $\mathbf{h}$ are diffusive fluxes. For the sake of simplicity, let us assume $\mathbf{h}=\mathbf{0}$ in the following. The Dirichlet boundary operator $\mathcal{D}$ is used to impose the Dirichlet boundary conditions from given values $\mathcal{U}_{\mathrm{D}}$. Although we have grouped Neumann conditions on $\partial\Omega_{\mathrm{N}}$ and Dirichlet conditions on $\partial\Omega_{\mathrm{D}}$, a mixed type of conditions could be applied to different variables on the same part of the boundary, and hence they may overlap. We shall explicitly indicate in our examples how the initial and boundary conditions are prescribed.

The description of the problem can be further complemented with the definition of the Reynolds, Prandtl and Mach numbers, respectively given as follows:

$$\mathrm{Re}=\frac{\rho_\infty\|\mathbf{u}_\infty\|L}{\mu_\infty}, \tag{14a}$$

$$\mathrm{Pr}=\frac{c_p\mu_\infty}{\kappa_\infty}, \tag{14b}$$

$$\mathrm{Ma}=\frac{\|\mathbf{u}_\infty\|}{a_\infty}. \tag{14c}$$

In these equations, $a:=\sqrt{\gamma p/\rho}$ denotes the speed of sound and L is a characteristic length. Variables are expressed in terms of reference free-stream values, indicated by the subscript ∞.

Remark 2.4. The non-dimensional Mach number defines the compressibility regime. It can range from subsonic ($\mathrm{Ma}<0.8$), transonic ($0.8<\mathrm{Ma}<1.2$), supersonic ($\mathrm{Ma}>1.2$), and hypersonic flow ($\mathrm{Ma}\gg 1$). Boundary conditions need to be prescribed appropriately according to the compressibility regime. In this paper, all conservative variables are imposed at the inflow part of the Dirichlet boundary, i.e., $\partial\Omega_{\mathrm{in}}:=\{\mathbf{x}\in\partial\Omega|(\mathbf{u}\cdot\mathbf{n})<0\}$, regardless the compressibility regime. For the supersonic case, no Dirichlet conditions need to be imposed at the outflow, defined as $\partial\Omega_{\mathrm{out}}:=\{\mathbf{x}\in\partial\Omega|(\mathbf{u}\cdot\mathbf{n})>0\}$. In the case of subsonic flow, only density (pressure) might be imposed at the outflow boundary $\partial\Omega_{\mathrm{out}}$.

Remark 2.5. Solid boundaries can be represented as a slip condition $\mathbf{u}\cdot\mathbf{n}=0$, as a no-slip condition $\mathbf{u}=\mathbf{0}$, as an isothermal wall with a given temperature value or an adiabatic wall $-\kappa\mathbf{n}\cdot\mathbf{grad}\ \vartheta=0$.

2.3. Variational formulation

Let V be an appropriate trial functional space where the solution is to be sought. The weak form of the problem is obtained by testing Eq. (11) against an arbitrary set of test functions $\delta\mathcal{U}=[\delta\rho,\delta(\rho\mathbf{u}),\delta(\rho e)]^{\intercal}$ which we consider time-independent and such that they vanish separately on the Dirichlet part of the boundary. The weak form can be generally written after integration by parts of the second order terms as: find $\mathcal{U}:(0,t_{\mathrm{f}})\to V$ such that

$$\int_\Omega \delta\mathcal{U}^{\intercal}\cdot\partial_t\mathcal{U}\,\mathrm{d}\Omega+\mathcal{B}(\mathcal{U},\delta\mathcal{U})=0 \quad \forall\,\delta\mathcal{U}, \tag{15}$$

where the following form is introduced,

$$\mathcal{B}(\mathcal{U},\delta\mathcal{U})=\int_\Omega \delta\mathcal{U}^{\intercal}\cdot\left(\mathcal{A}_j(\mathcal{U})\frac{\partial\mathcal{U}}{\partial x_j}-\mathcal{S}\mathcal{U}\right)\mathrm{d}\Omega+\int_\Omega\frac{\partial\delta\mathcal{U}^{\intercal}}{\partial x_k}\cdot\left(\mathcal{K}_{kj}(\mathcal{U})\frac{\partial\mathcal{U}}{\partial x_j}\right)\mathrm{d}\Omega. \tag{16}$$

3. Numerical approximation

3.1. Galerkin finite element discretization

The finite element approximation of the continuous variational problem Eq. (15) can be performed via the standard Galerkin method. For this purpose, let $\mathcal{T}_h$ be a shape-regular and conforming partition of Ω, such that $\overline{\Omega}=\cup_{e=1}^{n_{\mathrm{el}}}\overline{\Omega}^{(e)}$ being n_{el} the total number of elements in the partition. This triangulation is described by the characteristic mesh size, defined as $h:=\max\{h^{(e)}\mid\Omega^{(e)}\in\mathcal{T}_h(\Omega)\}$ with $h^{(e)}=\mathrm{diam}\left(\Omega^{(e)}\right)$. In the following, finite element functions will be identified with a subscript h. The conforming functional space $V_h\subset V$ is defined as a continuous piecewise set of polynomials. Hence, the Galerkin finite element approximation of the problem consists now in finding, $\mathcal{U}_h:(0,t_{\mathrm{f}})\to V_h$ such that

$$\int_\Omega \delta\boldsymbol{\mathcal{U}}_h^\intercal \cdot \partial_t \boldsymbol{\mathcal{U}}_h \,\mathrm{d}\Omega + \mathcal{B}(\boldsymbol{\mathcal{U}}_h, \delta\boldsymbol{\mathcal{U}}_h) = 0 \quad \forall\, \delta\boldsymbol{\mathcal{U}}_h, \tag{17}$$

together with the appropriate initial and boundary conditions.

However, it is well known that raw Galerkin formulations may suffer from different types of numerical instabilities when applied to flow problems. The main ones arise, precisely, from the non-elliptic nature of the equations and also due to an inf-sup-like condition which restrains the compatibility of the interpolation spaces of the different variables in play. Both inconveniences can be overcome by resorting to a stabilized finite element formulation. We shall further discuss this point later.

3.2. Time discretization

The problem can be discretized in time using a uniform partition of the time interval $(0, t_\mathrm{f})$. Given a time step δt, we set $t^0 = 0$, $t^n = t^0 + n\delta t$ (for any $n \geq 1$). Denoting by f^n an approximation to a time-dependent function $f(t)$ at time t^n, we consider backward difference (BDF) schemes with the operator,

$$D_k f^{n+1} = \frac{1}{\phi_k}\left(f^{n+1} - \sum_{i=0}^{k-1} \xi_k^i f^{n-i}\right),$$

being ϕ_k and ξ_k^i numerical parameters depending on the order of the temporal approximation. In particular, for the first and second order schemes, i.e., $k = 1, 2$, it is found that:

$$D_1 f^{n+1} = \delta f^{n+1} = f^{n+1} - f^n, \tag{18a}$$

$$D_2 f^{n+1} = \frac{3}{2}\left(f^{n+1} - \frac{4}{3} f^n + \frac{1}{3} f^{n-1}\right). \tag{18b}$$

For the design of fractional step schemes, it is particularly useful to make use of the backward extrapolation operators,

$$\hat{f}_0^{n+1} = 0, \tag{19a}$$

$$\hat{f}_1^{n+1} = f^n, \tag{19b}$$

$$\hat{f}_2^{n+1} = 2f^n - f^{n-1}, \tag{19c}$$

or, in general,

$$\hat{f}_k^{n+1} := f^{n+1} - \delta^{(k)} f^{n+1} = f^{n+1} + \mathcal{O}(\delta t^k),\ k = 0, 1, 2 \ldots \tag{20}$$

with $\delta^{(i+1)} f^{n+1} := \delta^{(i)} f^{n+1} - \delta^{(i)} f^n$, $i = 1, 2, 3, \ldots$.

Therefore, the fully discrete problem finally reads: find $\boldsymbol{\mathcal{U}}_h^{n+1} \in V_h$ such that

$$\int_\Omega \delta\boldsymbol{\mathcal{U}}_h^\intercal \cdot \frac{D_k \boldsymbol{\mathcal{U}}_h^{n+1}}{\delta t} \,\mathrm{d}\Omega + \mathcal{B}(\boldsymbol{\mathcal{U}}_h^{n+1}, \delta\boldsymbol{\mathcal{U}}_h) = 0 \quad \forall\, \delta\boldsymbol{\mathcal{U}}_h. \tag{21}$$

3.3. Residual-based stabilized formulation

In this subsection we present a stabilized formulation for the compressible Navier–Stokes equations based on the VMS approach. As stated in the introduction, the key idea behind this framework is to approximate the effect of the part of the solution of the continuous problem which cannot be reproduced by the finite element space.

3.3.1. VMS framework

The starting idea of VMS methods is to split the continuous space of the problem as $V = V_h \oplus \check{V}$, where V_h is the finite element space (and hence finite dimensional) and $\check{V}$ is any complementary space which completes V_h in V, usually termed subgrid space and which is in principle infinite dimensional. Such splitting of the space V induces a scale separation of unknowns of the form $\boldsymbol{\mathcal{U}} := \boldsymbol{\mathcal{U}}_h + \check{\boldsymbol{\mathcal{U}}}$ (and same for the test functions). It is readily checked that the continuous problem can be written as the following system of equations:

$$\int_\Omega \delta\boldsymbol{\mathcal{U}}_h^\intercal \cdot \partial_t \boldsymbol{\mathcal{U}} \,\mathrm{d}\Omega + \mathcal{B}(\boldsymbol{\mathcal{U}}, \delta\boldsymbol{\mathcal{U}}_h) = 0 \quad \forall\, \delta\boldsymbol{\mathcal{U}}_h, \tag{22a}$$

$$\int_\Omega \delta\check{\boldsymbol{\mathcal{U}}}^\intercal \cdot \partial_t \boldsymbol{\mathcal{U}} \,\mathrm{d}\Omega + \mathcal{B}(\boldsymbol{\mathcal{U}}, \delta\check{\boldsymbol{\mathcal{U}}}) \,\mathrm{d}\Omega = 0 \quad \forall\, \delta\check{\boldsymbol{\mathcal{U}}}, \tag{22b}$$

where the first row Eq. (22a) is termed finite element scale equation and the second row Eq. (22b) is labeled as subgrid scale equation in the literature. From this point, the goal is to find an approximation of the subscales in order to end up with a problem which should depend just on the finite element scale.

After integration by parts in Eq. (22a), which avoids introducing a model for the derivatives of the subscales, one finds

$$\int_\Omega \delta\boldsymbol{\mathcal{U}}_h^\intercal \cdot \partial_t \boldsymbol{\mathcal{U}}\, \mathrm{d}\Omega + \mathcal{B}(\boldsymbol{\mathcal{U}}_h, \delta\boldsymbol{\mathcal{U}}_h) + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^{(e)}} \left[\mathcal{L}^*(\boldsymbol{\mathcal{U}}; \delta\boldsymbol{\mathcal{U}}_h)\right]^\intercal \check{\boldsymbol{\mathcal{U}}}\, \mathrm{d}\Omega = 0 \quad \forall\, \delta\boldsymbol{\mathcal{U}}_h, \tag{23}$$

where inter-element terms are neglected by supposing that the subscales vanish at the element boundaries, though this assumption could be relaxed by including the subscales on the element boundaries, as discussed in [39]. Furthermore, we have introduced the formal adjoint operator of $\mathcal{L}(\boldsymbol{\mathcal{U}};\boldsymbol{\mathcal{U}})$ denoted with the asterisk superscript. Such adjoint operator satisfies, up to boundary terms,

$$\int_\Omega \delta\boldsymbol{\mathcal{U}}^\intercal \mathcal{L}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{W}})\, \mathrm{d}\Omega = \int_\Omega \left[\mathcal{L}^*(\boldsymbol{\mathcal{U}}; \delta\boldsymbol{\mathcal{U}})\right]^\intercal \boldsymbol{\mathcal{W}}\, \mathrm{d}\Omega \quad \forall\, \boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}, \delta\boldsymbol{\mathcal{U}}. \tag{24}$$

The outcome of the adjoint of the non-linear operator when applied to the test functions vector is

$$\begin{aligned}\mathcal{L}^*(\boldsymbol{\mathcal{U}}; \delta\boldsymbol{\mathcal{U}}_h) := -\Bigg\{ &\frac{\partial}{\partial x_j}\left[\boldsymbol{\mathcal{A}}_j^\intercal(\boldsymbol{\mathcal{U}})\delta\boldsymbol{\mathcal{U}}_h\right] + \frac{\partial}{\partial x_j}\left[\boldsymbol{\mathcal{K}}_{kj}^\intercal(\boldsymbol{\mathcal{U}})\frac{\partial \delta\boldsymbol{\mathcal{U}}_h}{\partial x_k}\right] \\ &+ \boldsymbol{\mathcal{S}}^\intercal \delta\boldsymbol{\mathcal{U}}_h\Bigg\} \quad \forall\, j,k = 1,\dots,n_{\mathrm{sd}}.\end{aligned} \tag{25}$$

It is important to remark the contribution of the derivative of the Euler Jacobian matrix in the first term on the right hand side. In this work we linearize the derivatives of the first and second terms on the right hand side of the previous expression, respectively as:

$$\frac{\partial}{\partial x_j}\left[\boldsymbol{\mathcal{A}}_j^\intercal(\boldsymbol{\mathcal{U}})\delta\boldsymbol{\mathcal{U}}_h\right] \approx \boldsymbol{\mathcal{A}}_j^\intercal(\boldsymbol{\mathcal{U}})\frac{\partial \delta\boldsymbol{\mathcal{U}}_h}{\partial x_j} + \frac{\partial \boldsymbol{\mathcal{A}}_j^\intercal(\boldsymbol{\mathcal{U}})}{\partial \boldsymbol{\mathcal{U}}}\frac{\partial \boldsymbol{\mathcal{U}}_h}{\partial x_j}\delta\boldsymbol{\mathcal{U}}_h \qquad \forall\, j = 1,\dots,n_{\mathrm{sd}}, \tag{26}$$

$$\frac{\partial}{\partial x_j}\left[\boldsymbol{\mathcal{K}}_{kj}^\intercal(\boldsymbol{\mathcal{U}})\frac{\partial \delta\boldsymbol{\mathcal{U}}_h}{\partial x_k}\right] \approx \boldsymbol{\mathcal{K}}_{kj}^\intercal(\boldsymbol{\mathcal{U}})\frac{\partial^2 \delta\boldsymbol{\mathcal{U}}_h}{\partial x_j \partial x_k} + \frac{\partial \boldsymbol{\mathcal{K}}_{kj}^\intercal(\boldsymbol{\mathcal{U}})}{\partial \boldsymbol{\mathcal{U}}}\frac{\partial \boldsymbol{\mathcal{U}}_h}{\partial x_j}\frac{\partial \delta\boldsymbol{\mathcal{U}}_h}{\partial x_k} \qquad \forall\, j,k = 1,\dots,n_{\mathrm{sd}}. \tag{27}$$

The finite element scale equation can be seen as the projection of the original problem onto the finite element space. Similarly, the subgrid scale equation is nothing but a projection of the equations onto the space of subscales. If $\check{\Pi}$ denotes the L^2-projection onto the space of subscales, Eq. (22b) can be rewritten as

$$\partial_t \check{\boldsymbol{\mathcal{U}}} + \mathcal{L}(\boldsymbol{\mathcal{U}}; \check{\boldsymbol{\mathcal{U}}}) = \check{\Pi}\left[\mathcal{R}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}}_h)\right], \tag{28}$$

where the finite element residual,

$$\mathcal{R}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}}_h) := -\partial_t \boldsymbol{\mathcal{U}}_h - \mathcal{L}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}}_h), \tag{29}$$

has been introduced.

Since the subscales cannot be represented by the finite element mesh, the effect of the non-linear operator applied to the subscales in Eq. (28) needs to be modeled somehow. However, the goal of this paper is not to propose a new stabilization technique, but rather to present the application of a well established approach. This method can be obtained by solving for the subscales as a result of the following nonlinear evolution problem,

$$\partial_t \check{\boldsymbol{\mathcal{U}}} + \boldsymbol{\tau}^{-1}(\boldsymbol{\mathcal{U}})\check{\boldsymbol{\mathcal{U}}} = \check{\Pi}\left[\mathcal{R}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}}_h)\right], \tag{30}$$

where $\boldsymbol{\tau}(\boldsymbol{\mathcal{U}})$ is the so-called matrix of stabilization parameters. There are different approximations in order to calculate $\check{\boldsymbol{\mathcal{U}}}$ from the previous equation. The most common approach is to take $\check{\Pi} = I$, the identity, when applied to the finite element residual. Another possibility, which is the one that we favor in this work, is to take $\check{\Pi}$ as the projection onto the space orthogonal to V_h, i.e., the orthogonal projection to the finite element space.

3.3.2. Orthogonal-dynamic subgrid scale formulation

When the space of subscales is enforced to be L^2-orthogonal to the finite element space V_h, the method is termed Orthogonal SubGrid-Scale (OSGS or simply OSS). It corresponds to taking $\check{\Pi} = \Pi_h^\perp = I - \Pi_h$ where Π_h is the projection operator onto the appropriate finite element space without boundary conditions. The particularity of this method is that it makes the subscales active in regions which cannot be resolved by the finite element mesh [40]. As a result, the OSGS method allows certain type of simplifications, which we state next.

- The orthogonal projection of the external forces might be neglected. External loads are assumed to belong to the finite element space or are approximated by an element of the corresponding space. Hence,

$$\Pi_h^{\perp}(\rho r) \approx 0, \quad \Pi_h^{\perp}(\rho \mathbf{f}) \approx 0.$$

- The orthogonal projection of terms involving temporal derivatives can be neglected, taking into account that,

$$\Pi_h^{\perp}(\partial_t f_h) = \partial_t f_h - \Pi_h(\partial_t f_h) = 0, \quad \forall\, f_h \in V_h,$$

due to the orthogonality property.

In addition to the previous developments, we will also consider the time tracking of the subscales, which has become an effective feature in order to eliminate numerical oscillations originated by initial transients while minimizing numerical dissipation [41]. In this regard, Eq. (28) can be for instance integrated in time with a classical backward Euler scheme and this yields

$$\breve{\mathcal{U}}^{n+1} = \boldsymbol{\tau}_{\text{dyn}}(\mathcal{U}^{n+1})\breve{\Pi}\left[\boldsymbol{\mathcal{R}}(\mathcal{U}^{n+1}; \mathcal{U}_h^{n+1})\right] + \boldsymbol{\tau}_{\text{dyn}}(\mathcal{U}^{n+1})\frac{\breve{\mathcal{U}}^{n}}{\delta t}, \tag{31}$$

with $\boldsymbol{\tau}_{\text{dyn}} = \left[(1/\delta t)\,\mathbf{I}_{n_{\text{sd}}+2} + \boldsymbol{\tau}^{-1}(\mathcal{U}^{n+1})\right]^{-1}$. The rationale behind using such a dissipative scheme for the time integration of the fine scales is precisely the assumption of bubble functions. However, it can be shown that this choice would not modify a second-order accuracy in time of the finite element solution (large scales) [42].

Upon substitution of Eq. (31) into the time discretized version of Eq. (23), the final problem to be solved in this work finally reads: find $\mathcal{U}_h \in V_h$ such that

$$\begin{aligned}
&\int_{\Omega} \delta\mathcal{U}_h^{\intercal} \cdot \frac{D_k\mathcal{U}_h^{n+1}}{\delta t}\,\mathrm{d}\Omega + \mathcal{B}(\mathcal{U}_h^{n+1}, \delta\mathcal{U}_h) \\
&\quad + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^{(e)}} [\mathcal{L}^*(\mathcal{U}^{n+1}; \delta\mathcal{U}_h)]^{\intercal}\boldsymbol{\tau}_{\text{dyn}}(\mathcal{U}^{n+1})\Pi_h^{\perp}\left[\boldsymbol{\mathcal{R}}(\mathcal{U}^{n+1}; \mathcal{U}_h^{n+1})\right]\mathrm{d}\Omega \\
&\quad - \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^{(e)}} [\mathcal{L}^*(\mathcal{U}^{n+1}; \delta\mathcal{U}_h)]^{\intercal}\boldsymbol{\tau}_{\text{dyn}}(\mathcal{U}^{n+1})\frac{\breve{\mathcal{U}}^{n}}{\delta t}\,\mathrm{d}\Omega = 0 \quad \forall\, \delta\mathcal{U}_h.
\end{aligned} \tag{32}$$

3.3.3. On the stabilization parameters

Although the application of the VMS method to the compressible Navier–Stokes problem has already been discussed, the stabilization technique is not completed until one introduces a definition to compute the matrix of stabilization parameters $\boldsymbol{\tau}(\mathcal{U})$ (generally nonlinear).

Up to our knowledge, there is no general rule to define it for systems of equations. It must be designed for each particular problem taking into account its stability deficiencies or even scaling requirements. The usual definition of the compressible stabilization parameters includes a local sound velocity that arises from a linearization of the characteristic compressible flow problem. In this paper we follow our previous findings published in [43] (see section 3.3 from that paper for a complete and detailed exposition) which are based on a Fourier analysis of the non-linear operator of the problem.

Therefore, the stabilization matrix for the 3D case is computed here as

$$\boldsymbol{\tau}^{-1} = \begin{bmatrix} \tau_{\rho}^{-1} & \mathbf{0}^{\intercal} & 0 \\ \mathbf{0} & \boldsymbol{\tau}_{\rho\mathbf{u}}^{-1}\mathbf{I}_{n_{\text{sd}}} & \mathbf{0} \\ 0 & \mathbf{0}^{\intercal} & \tau_{\rho e}^{-1} \end{bmatrix} \tag{33}$$

with the following non-zero entries,

$$\tau_{\rho}^{-1} = c_2\frac{(\|\mathbf{u}\| + a)}{h} + c_3\left(\frac{r^2 + 2\|\mathbf{f}\|^2a^2 + \sqrt{r^4 + 4a^2\|\mathbf{f}\|^2r^2}}{2a^4}\right)^{1/2}, \tag{34a}$$

$$\tau_{\rho\mathbf{u}}^{-1} = c_1\frac{4\mu}{3\rho h^2} + c_2\frac{(\|\mathbf{u}\| + a)}{h} + c_3\left(\frac{r^2 + 2\|\mathbf{f}\|^2a^2 + \sqrt{r^4 + 4a^2\|\mathbf{f}\|^2r^2}}{2a^4}\right)^{1/2}, \tag{34b}$$

$$\tau_{\rho e}^{-1} = c_1 \frac{\kappa}{\rho c_p h^2} + c_2 \frac{(\|\mathbf{u}\| + a)}{h} + c_3 \left(\frac{r^2 + 2\|\mathbf{f}\|^2 a^2 + \sqrt{r^4 + 4a^2\|\mathbf{f}\|^2 r^2}}{2a^4} \right)^{1/2} . \tag{34c}$$

It is understood that these expressions are evaluated element by element. The numerical constants c_1 and c_2 are independent of the physical parameters of the problem. In the numerical calculations we take them as $c_1 = 12\omega^4$, $c_2 = 2\omega$, $c_3 = 1$, ω being the order of the finite element interpolation.

3.4. Discontinuity capturing technique

Although the previous stabilized finite element formulation yields a globally stable solution, i.e., norms of the unknowns over the whole domain Ω are bounded, local stability is not guaranteed in the vicinity of shocks or regions with sharp gradients. In order to mitigate these possible local oscillations, an artificial shock-capturing (SC) term is added to the numerical approximation of the problem. The main idea of a shock capturing technique is to increase the amount of numerical dissipation in the proximity of sharp gradients. Many different approaches can be adopted to introduce artificial dissipation. Here, two different alternatives are presented, which introduce numerical diffusion only in the momentum and energy equations.

The first non-linear method that we implement is a classical residual-based technique, which is consistent, that is to say, when it is applied to the exact solution, the added diffusion is zero. For this technique we calculate the artificial kinematic viscosity as

$$\nu_{\text{sc}} = \frac{C_{\text{sc}} h}{2} \frac{|\boldsymbol{\mathcal{R}}_{\rho\mathbf{u}}(\boldsymbol{\mathcal{U}}; \boldsymbol{\mathcal{U}}_h)|}{|\mathbf{grad}\ (\rho\mathbf{u})_h|}, \quad \text{if } |\mathbf{grad}\ (\rho\mathbf{u})_h| \neq 0, \tag{35a}$$

$$\nu_{\text{sc}} = 0, \quad \text{otherwise}, \tag{35b}$$

where C_{sc} is an algorithmic constant to be set before the simulation, h is the characteristic length that gives dimensional consistency to the expression, and $|\mathbf{grad}(\rho\mathbf{u})_h|$ is the Frobenius norm of the gradient of the momentum finite element solution. In a similar manner, for the energy equation we introduce an artificial thermal diffusivity computed as

$$\alpha_{\text{sc}} = \frac{C_{\text{sc}} h}{2} \frac{|\boldsymbol{\mathcal{R}}_{\rho e}(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{U}}_h)|}{\|\mathbf{grad}\ (\rho e)_h\|}, \quad \text{if } \|\mathbf{grad}\ (\rho e)_h\| \neq 0, \tag{36a}$$

$$\alpha_{\text{sc}} = 0, \quad \text{otherwise}, \tag{36b}$$

where $\|\mathbf{grad}\ (\rho e)_h\|$ is the norm of the gradient of the total energy finite element solution.

An alternative to introduce the numerical diffusion is the weakly consistent orthogonal projection technique from [44]. It is based on the orthogonal projection onto the finite element space of the gradient of the unknown, instead of the common residual approach. Mathematically speaking, we take the artificial viscosity as follows:

$$\nu_{\text{sc}} = \frac{C_{\text{sc}} h \|\mathbf{u}\|}{2} \frac{|\Pi_h^{\perp}[\mathbf{grad}\ (\rho\mathbf{u})_h)]|}{|\mathbf{grad}\ (\rho\mathbf{u})_h|}, \quad \text{if } |\mathbf{grad}\ (\rho\mathbf{u})_h| \neq 0, \tag{37a}$$

$$\nu_{\text{sc}} = 0, \quad \text{otherwise}, \tag{37b}$$

and,

$$\alpha_{\text{sc}} = \frac{C_{\text{sc}} h \|\mathbf{u}\|}{2} \frac{\left\| \Pi_h^{\perp}(\mathbf{grad}\ (\rho e)_h) \right\|}{\|\mathbf{grad}\ (\rho e)_h\|}, \quad \text{if } \|\mathbf{grad}\ (\rho e)_h\| \neq 0, \tag{38a}$$

$$\alpha_{\text{sc}} = 0, \quad \text{otherwise}, \tag{38b}$$

where the norm of the velocity gives dimensional consistency.

In practice, the added numerical diffusion is introduced into the diffusive Galerkin term by computing a modified viscous stress tensor $\tilde{\boldsymbol{\sigma}}^d$ and heat flux $\tilde{\mathbf{q}}$ vector in the following isotropic manner,

$$\tilde{\sigma}_{ij}^{\text{d}} = \left(1 + \frac{\rho \nu_{\text{sc}}}{\mu}\right) \sigma_{ij}^{\text{d}}, \qquad \forall\, i, j = 1, \ldots, n_{\text{sd}} \tag{39a}$$

$$\tilde{q}_i = \left(1 + \frac{\rho c_v \alpha_{\text{sc}}}{\kappa}\right) q_i \qquad \forall\, i = 1, \ldots, n_{\text{sd}}. \tag{39b}$$

3.5. On the linearization of the discrete problem

Apart from the classical non-linearities appearing in the convective terms due to the advection velocity, there are others inherent to the nature of the compressible Navier–Stokes equations. In any fractional step scheme, the time step of the computation δt cannot be taken very large for the method to be effective, when it is compared for example to the critical

time step of an explicit time integration scheme. This fact is of remarkable importance in compressible flow simulations, as the time step should be sufficiently small in order to reproduce the wide range of different scales that the compressible Navier–Stokes problem spans. Thus, for a given solution at a particular non-linear iteration, the next solution should be particularly close and our experience shows that the fixed-point option is usually enough. Otherwise one should expand the equations by means of a Taylor series. Although this may be appealing, and it is actually a preferred approach in the literature, it definitely introduces some burden in the formulation since the resulting terms involve rational expressions which are in general arduous to integrate numerically.

Let us also explain how we manage the orthogonal projection calculation. When compared to the raw Galerkin method, the matrices emerging from the orthogonal projection of the unknowns show a wide stencil. In order to avoid dealing with them, at the i-th iteration of the n-th time step, we may approximate $\Pi_h^{\perp}\left(f^{n,(i)}\right) \approx f^{n,(i)} - \Pi_h\left(f^{n,(i-1)}\right)$ or $\Pi_h^{\perp}\left(f^{n,(i)}\right) \approx f^{n,(i)} - \Pi_h\left(f^{n-1}\right)$ for any generic function f. In other words, we perform the projection by means of known values from either the previous iteration or time step. Numerical experiments reveal that both options are effective, the latter being chosen in the simulations presented in Section 5.

4. Fractional step methods

Instead of solving the monolithic system, an alternative is to use a fractional step method in time, in which various equations need to be solved for the different variables in an uncoupled way, probably with the addition of some correction steps. The splitting of the equations introduced in fractional step methods has an additional temporal error, which has to be *at least* of the order of the time integration scheme used to approximate time derivatives. Otherwise, time accuracy is broken. In this section we develop a novel algebraic fractional step method for the compressible Navier–Stokes problem. The basic procedure entails a calculation of an intermediate momentum with a guess of the density and total energy. After solving for density and energy (in that order), we will finally correct the intermediate momentum calculation, so as to ensure that the global time accuracy of the method is maintained.

4.1. Algebraic problem

We assume that ρ_h^n, $(\rho\mathbf{u})_h^n$ and $(\rho e)_h^n$ are constructed using the standard finite element interpolation from the nodal values, which we denote hereinafter as $\boldsymbol{\varrho}^n$, $\boldsymbol{U}^n$ and $\boldsymbol{E}^n$ respectively. These are computed as the solution of a non-linear algebraic problem, which is obtained directly from Eq. (21). We shall skip the stabilization terms for the sake of conciseness. Their addition is straightforward once the fractional step method is developed.

The structure of the final system can be written in a compact matrix form as $\mathbf{K}\boldsymbol{\Phi} = \mathbf{f}$ with

$$\mathbf{K} = \begin{bmatrix} \boldsymbol{M}_{\rho}^{\rho}\frac{D_k}{\delta t} & \boldsymbol{C}_{\rho}^{\rho\mathbf{u}} & \mathbf{0} \\ \boldsymbol{A}_{\rho\mathbf{u}}^{\rho}(\boldsymbol{\varrho},\boldsymbol{U}) & \boldsymbol{M}_{\rho\mathbf{u}}^{\rho\mathbf{u}}\frac{D_k}{\delta t} + \boldsymbol{A}_{\rho\mathbf{u}}^{\rho\mathbf{u}}(\boldsymbol{\varrho},\boldsymbol{U}) & \boldsymbol{C}_{\rho\mathbf{u}}^{\rho e} \\ \boldsymbol{A}_{\rho e}^{\rho}(\boldsymbol{\varrho},\boldsymbol{U},\boldsymbol{E}) & \boldsymbol{A}_{\rho e}^{\rho\mathbf{u}}(\boldsymbol{\varrho},\boldsymbol{U},\boldsymbol{E}) & \boldsymbol{M}_{\rho e}^{\rho e}\frac{D_k}{\delta t} + \boldsymbol{A}_{\rho e}^{\rho e}(\boldsymbol{\varrho},\boldsymbol{U}) \end{bmatrix}, \tag{40a}$$

$$\boldsymbol{\Phi} = [\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1}, \boldsymbol{E}^{n+1}]^{\intercal}, \tag{40b}$$

$$\mathbf{f} = [\boldsymbol{F}_{\rho}^{n+1}, \boldsymbol{F}_{\rho\mathbf{u}}^{n+1}, \boldsymbol{F}_{\rho e}^{n+1}]^{\intercal}, \tag{40c}$$

and,

$$\boldsymbol{A}_{\rho\mathbf{u}}^{\rho\mathbf{u}} := \boldsymbol{C}_{\rho\mathbf{u}}^{\rho\mathbf{u}}(\boldsymbol{\varrho},\boldsymbol{U}) + \boldsymbol{D}_{\rho\mathbf{u}}^{\rho\mathbf{u}}(\boldsymbol{\varrho}) \qquad \boldsymbol{A}_{\rho\mathbf{u}}^{\rho} := \boldsymbol{C}_{\rho\mathbf{u}}^{\rho}(\boldsymbol{\varrho},\boldsymbol{U}) + \boldsymbol{D}_{\rho\mathbf{u}}^{\rho}(\boldsymbol{\varrho},\boldsymbol{U}) \tag{41a}$$

$$\boldsymbol{A}_{\rho e}^{\rho} := \boldsymbol{C}_{\rho e}^{\rho}(\boldsymbol{\varrho},\boldsymbol{U},\boldsymbol{E}) + \boldsymbol{D}_{\rho e}^{\rho}(\boldsymbol{\varrho},\boldsymbol{U},\boldsymbol{E}) \qquad \boldsymbol{A}_{\rho e}^{\rho\mathbf{u}} := \boldsymbol{C}_{\rho e}^{\rho\mathbf{u}}(\boldsymbol{\varrho},\boldsymbol{U},\boldsymbol{E}) + \boldsymbol{D}_{\rho e}^{\rho\mathbf{u}}(\boldsymbol{\varrho},\boldsymbol{U}) \tag{41b}$$

$$\boldsymbol{A}_{\rho e}^{\rho e} := \boldsymbol{C}_{\rho e}^{\rho e}(\boldsymbol{\varrho},\boldsymbol{U}) + \boldsymbol{D}_{\rho e}^{\rho e}(\boldsymbol{\varrho}) \tag{41c}$$

The subscript on the arrays refers to the mass (ρ), momentum ($\rho\mathbf{u}$) and energy (ρe) equations, whereas the superscript stands for the unknown to which the term refers to. The high non-linear character of the problem is made explicit in the system by including the dependence of the arrays on the variables in the parenthesis, which are all evaluated at time step $n+1$. Mass matrices as labeled with the symbol $\boldsymbol{M}$, convective-like matrices with $\boldsymbol{C}$ and diffusive-like matrices with the symbol $\boldsymbol{D}$. The latter comes from the discretization of the terms inside $\mathcal{K}_{kj}$ and the convective matrices from those in $\mathcal{A}_j$ (see Section 2.2). All the arrays in the system are computed from the local assembly of the elemental contributions. Furthermore, the right hand side terms in $\mathbf{f}$ contain known terms such as external forces or known values of the unknowns from previous times steps due to temporal discretization.

In order to derive the fractional step method, let us proceed as follows: if $\widetilde{\boldsymbol{U}}^{n+1}$ denotes an intermediate momentum variable, $\widehat{\boldsymbol{\varrho}}_{k-1}^{n+1}$ and $\widehat{\boldsymbol{E}}_{k-1}^{n+1}$ are respectively density and energy extrapolated values of order $k-1$ according to Eq. (20) and we split the former momentum equation into two parts, the previous system can be rewritten in the following *equivalent* form,

$$\boldsymbol{M}_{\rho\mathbf{u}}^{\rho\mathbf{u}}\frac{D_k}{\delta t}\widetilde{\boldsymbol{U}}^{n+1} + \boldsymbol{C}_{\rho\mathbf{u}}^{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1},\widetilde{\boldsymbol{U}}^{n+1})\widetilde{\boldsymbol{U}}^{n+1} + \boldsymbol{D}_{\rho\mathbf{u}}^{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1})\widetilde{\boldsymbol{U}}^{n+1}$$

$$+ \boldsymbol{C}^{\rho}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1})\widehat{\boldsymbol{\varrho}}^{n+1}_{k-1} + \boldsymbol{D}^{\rho}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1})\widehat{\boldsymbol{\varrho}}^{n+1}_{k-1} + \boldsymbol{C}^{\rho e}_{\rho\mathbf{u}}\widehat{\boldsymbol{E}}^{n+1}_{k-1} = \boldsymbol{F}^{n+1}_{\rho\mathbf{u}}, \tag{42a}$$

$$\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\frac{1}{\phi_k\delta t}\left(\boldsymbol{U}^{n+1} - \widetilde{\boldsymbol{U}}^{n+1}\right) + \boldsymbol{N}^{n+1}_{\rho\mathbf{u}} + \boldsymbol{N}^{n+1}_{\rho} + \boldsymbol{N}^{n+1}_{\rho e} = \boldsymbol{0}, \tag{42b}$$

$$\boldsymbol{M}^{\rho}_{\rho}\frac{D_k}{\delta t}\boldsymbol{\varrho}^{n+1} + \boldsymbol{C}^{\rho\mathbf{u}}_{\rho}\boldsymbol{U}^{n+1} = \boldsymbol{F}^{n+1}_{\rho}, \tag{42c}$$

$$\begin{aligned}\boldsymbol{M}^{\rho e}_{\rho e}\frac{D_k}{\delta t}\boldsymbol{E}^{n+1} &+ \boldsymbol{C}^{\rho e}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1})\boldsymbol{E}^{n+1} + \boldsymbol{D}^{\rho e}_{\rho e}(\boldsymbol{\varrho}^{n+1})\boldsymbol{E}^{n+1}\\ &+ \boldsymbol{C}^{\rho}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1}, \boldsymbol{E}^{n+1})\boldsymbol{\varrho}^{n+1} + \boldsymbol{D}^{\rho}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1}, \boldsymbol{E}^{n+1})\boldsymbol{\varrho}^{n+1}\\ &+ \boldsymbol{C}^{\rho\mathbf{u}}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1}, \boldsymbol{E}^{n+1})\boldsymbol{U}^{n+1} + \boldsymbol{D}^{\rho\mathbf{u}}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1})\boldsymbol{U}^{n+1} = \boldsymbol{F}^{n+1}_{\rho e},\end{aligned} \tag{42d}$$

with the following definitions,

$$\begin{aligned}\boldsymbol{N}^{n+1}_{\rho\mathbf{u}} := &\left[\boldsymbol{C}^{\rho\mathbf{u}}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1}) + \boldsymbol{D}^{\rho\mathbf{u}}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1})\right]\boldsymbol{U}^{n+1}\\ &- \left[\boldsymbol{C}^{\rho\mathbf{u}}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1}, \widetilde{\boldsymbol{U}}^{n+1}) + \boldsymbol{D}^{\rho\mathbf{u}}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1})\right]\widetilde{\boldsymbol{U}}^{n+1},\end{aligned} \tag{43a}$$

$$\boldsymbol{N}^{n+1}_{\rho} := \left[\boldsymbol{C}^{\rho}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1}) + \boldsymbol{D}^{\rho}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1}, \boldsymbol{U}^{n+1})\right]\left(\boldsymbol{\varrho}^{n+1} - \widehat{\boldsymbol{\varrho}}^{n+1}_{k-1}\right), \tag{43b}$$

$$\boldsymbol{N}^{n+1}_{\rho e} := \boldsymbol{C}^{\rho e}_{\rho\mathbf{u}}\left(\boldsymbol{E}^{n+1} - \widehat{\boldsymbol{E}}^{n+1}_{k-1}\right), \tag{43c}$$

and where $D_k\widetilde{\boldsymbol{U}}^{n+1}$ is computed as $D_k\boldsymbol{U}^{n+1}$ but replacing $\boldsymbol{U}^{n+1}$ by a yet undetermined function $\widetilde{\boldsymbol{U}}^{n+1}$ (intermediate momentum). The reader should note that adding up Eqs. (42a)–(42b) with the definitions in Eqs. (43a)–(43c), we obtain the former momentum equation, i.e., the second row of the original system $\mathbf{K}\boldsymbol{\Phi} = \mathbf{f}$ in Eqs. (40a)–(40c). We shall refer to Eq. (42a) as the *intermediate momentum equation* and Eq. (42b) as the *momentum correction equation*. The purpose of the latter is precisely to ensure that the global time accuracy is not broken.

If we denote the node indexes with superscripts a, b, and the standard shape function of node a by φ^a, the components of the matrix $\boldsymbol{C}^{\rho\mathbf{u}}_{\rho}$ are

$$\left[\boldsymbol{C}^{\rho\mathbf{u}}_{\rho}\right]^{ab}_{j} = \mathop{\mathbb{A}}_{e=1}^{n_{\mathrm{el}}}\int_{\partial\Omega^{(e)}}\varphi^a\partial_j\varphi^b\,\mathrm{d}\Omega \qquad \forall\, j = 1, \ldots, n_{\mathrm{sd}}, \tag{44}$$

where $\mathbb{A}$ symbolizes the assembly of the local contributions (addition plus injection) to the global equation. Let us now proceed as follows: first, integration by parts over that term in the mass equation would momentarily yield

$$\boldsymbol{M}^{\rho}_{\rho}\frac{D_k}{\delta t}\boldsymbol{\varrho}^{n+1} - \check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\boldsymbol{U}^{n+1} = \boldsymbol{F}^{n+1}_{\rho} - \check{\boldsymbol{F}}^{n+1}_{\rho}, \tag{45}$$

where

$$\left[\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\right]^{ab}_{j} = \mathop{\mathbb{A}}_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^{(e)}}\partial_j\varphi^a\varphi^b\,\mathrm{d}\Omega \quad \forall\, j = 1, \ldots, n_{\mathrm{sd}}, \tag{46}$$

$$\left[\check{\boldsymbol{F}}^{n+1}_{\rho}\right]^{a} = \mathop{\mathbb{A}}_{e=1}^{n_{\mathrm{el}}}\int_{\partial\Omega^{(e)}}\varphi^a(\mathbf{n}\cdot\boldsymbol{U}^{n+1})\,\mathrm{d}\partial\Omega. \tag{47}$$

Now, solving for $\boldsymbol{U}^{n+1}$ from Eq. (42b) and multiplying by $\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}$ it gives

$$\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\boldsymbol{U}^{n+1} = \check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\widetilde{\boldsymbol{U}}^{n+1} - \phi_k\delta t\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\left[\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\right]^{-1}\left[\boldsymbol{N}^{n+1}_{\rho} + \boldsymbol{N}^{n+1}_{\rho\mathbf{u}} + \boldsymbol{N}^{n+1}_{\rho e}\right]. \tag{48}$$

This new expression can be now used in Eq. (45). Hence,

$$\boldsymbol{M}^{\rho}_{\rho}\frac{D_k}{\delta t}\boldsymbol{\varrho}^{n+1} + \phi_k\delta t\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\left[\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\right]^{-1}\left[\boldsymbol{N}^{n+1}_{\rho} + \boldsymbol{N}^{n+1}_{\rho\mathbf{u}} + \boldsymbol{N}^{n+1}_{\rho e}\right] = \boldsymbol{F}^{n+1}_{\rho} - \check{\boldsymbol{F}}^{n+1}_{\rho} + \check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\widetilde{\boldsymbol{U}}^{n+1}. \tag{49}$$

At this point, we have modified the original mass equation by introducing some burden that we need to solve. One should notice that the resulting matrix from $\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\left[\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\right]^{-1}\boldsymbol{C}^{\rho}_{\rho\mathbf{u}}$ can be viewed as an approximation to the discrete version of a Laplacian-like operator, [45]. In order to avoid dealing with this matrix, which is in general dense and might still be expensive to compute even when $\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}$ is lumped, we use the approximation $\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}\left[\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\right]^{-1}\boldsymbol{C}^{\rho}_{\rho\mathbf{u}} \approx \boldsymbol{L}$ where $\boldsymbol{L}$ is a Laplacian matrix computed using the gradient of the standard shape functions and defined at the element level as $[\boldsymbol{L}^{(e)}]^{ab} = \int_{\Omega^{(e)}} 0.5(\mathbf{u}\cdot\mathbf{u})(\gamma - 1)\partial_j\varphi^a\partial_j\varphi^b\,\mathrm{d}\Omega$, where the first factors in the integrand are introduced to keep the proper scaling (see Appendix A). Being able to perform this approximation is what led us to obtain matrix $\check{\boldsymbol{C}}^{\rho\mathbf{u}}_{\rho}$ instead of working directly with $\boldsymbol{C}^{\rho\mathbf{u}}_{\rho}$.

Remark 4.1. Recalling the definition of the extrapolation operators, Eq. (20), note that the difference $\left\| \boldsymbol{\varrho}^{n+1} - \widehat{\boldsymbol{\varrho}}_{k-1}^{n+1} \right\|$ or $\left\| \boldsymbol{E}^{n+1} - \widehat{\boldsymbol{E}}_{k-1}^{n+1} \right\|$ is of order $\mathcal{O}(\delta t^{k-1})$. Therefore, it is easy to see from Eq. (42b) that $\mathcal{O}(||\boldsymbol{U}^{n+1} - \widetilde{\boldsymbol{U}}^{n+1}||) = \mathcal{O}(\delta t^k)$, and thus, the global accuracy of the temporal integrator is formally maintained for the intermediate momentum variable.

Remark 4.2. Note that if we substitute $\boldsymbol{U}^{n+1}$ by $\widetilde{\boldsymbol{U}}^{n+1}$ in the definition of $\check{\boldsymbol{F}}_{\rho}^{n+1}$, which is supported by the previous remark, the last two terms in Eq. (45) can be grouped as $-\boldsymbol{C}_{\rho}^{\rho\mathbf{u}}\widetilde{\boldsymbol{U}}^{n+1}$ (integration by parts).

Up to our knowledge, there is not a possible approximation for the rest of the products involved in $\check{\boldsymbol{C}}_{\rho}^{\rho\mathbf{u}}\left[\boldsymbol{M}_{\rho\mathbf{u}}^{\rho\mathbf{u}}\right]^{-1}\left[\boldsymbol{N}_{\rho}^{n+1} + \boldsymbol{N}_{\rho\mathbf{u}}^{n+1} + \boldsymbol{N}_{\rho e}^{n+1}\right]$. However, the presence of such extra products involving density and energy could be avoided by performing an extrapolation of the same order of that used to approximate time derivatives in the intermediate momentum equation. Then, we shall use $\boldsymbol{D}_{\rho\mathbf{u}}^{\rho}\widehat{\boldsymbol{\varrho}}_{k}^{n+1}$ instead of $\boldsymbol{D}_{\rho\mathbf{u}}^{\rho}\widehat{\boldsymbol{\varrho}}_{k-1}^{n+1}$ and $\boldsymbol{C}_{\rho\mathbf{u}}^{\rho e}\widehat{\boldsymbol{E}}_{k}^{n+1}$ instead of $\boldsymbol{C}_{\rho\mathbf{u}}^{\rho e}\widehat{\boldsymbol{E}}_{k-1}^{n+1}$ in Eq. (42a). Therefore, there is no need to include such terms in the correction equation (42b) since $\left\| \boldsymbol{\varrho}^{n+1} - \widehat{\boldsymbol{\varrho}}_{k}^{n+1} \right\|$ and $\left\| \boldsymbol{E}^{n+1} - \widehat{\boldsymbol{E}}_{k}^{n+1} \right\|$ are already of order $\mathcal{O}\left(\delta t^k\right)$.

Remark 4.3. Using directly $\boldsymbol{D}_{\rho\mathbf{u}}^{\rho}\widehat{\boldsymbol{\varrho}}_{k}^{n+1}$ and $\boldsymbol{C}_{\rho\mathbf{u}}^{\rho e}\widehat{\boldsymbol{E}}_{k}^{n+1}$ involves an explicit treatment of density and energy in the momentum equation that could imply the introduction of a critical time step to ensure stability. However, in practice we have not observed such time step limitation.

4.2. Fractional step algorithm

Taking all the previous information into account, the fractional step approach that we favor to solve the fully compressible Navier–Stokes problem in conservative variables is composed of four main steps:

i. Compute an intermediate momentum from Eq. (42a) making use of $\widehat{\boldsymbol{\varrho}}_{k}^{n+1}$ and $\widehat{\boldsymbol{E}}_{k}^{n+1}$.
ii. Compute an approximation to the density from Eq. (42c), neglecting $\boldsymbol{N}_{\rho\mathbf{u}}^{n+1}$, replacing $\boldsymbol{U}^{n+1}$ by $\widetilde{\boldsymbol{U}}^{n+1}$ and taking into account the previous Laplacian approximation.

$$\left[\boldsymbol{M}_{\rho}^{\rho}\frac{D_k}{\delta t} + \phi_k \delta t \boldsymbol{L}\right]\boldsymbol{\varrho}^{n+1} = \boldsymbol{F}_{\rho}^{n+1} - \boldsymbol{C}_{\rho}^{\rho\mathbf{u}}\widetilde{\boldsymbol{U}}^{n+1} + \phi_k \delta t \boldsymbol{L}\widehat{\boldsymbol{\varrho}}_{k-1}^{n+1}. \tag{50}$$

iii. Compute an approximation to the total energy from Eq. (42d), replacing $\boldsymbol{U}^{n+1}$ by $\widetilde{\boldsymbol{U}}^{n+1}$ (already known as a result of the first step).
iv. Update the end-of-step momentum with Eq. (42b) neglecting $\boldsymbol{N}_{\rho\mathbf{u}}^{n+1}$.

This procedure will make possible to segregate the calculation of the unknowns of the problem and we shall refer to it as density/energy-correction algorithm, using a similar nomenclature as it is usually done for incompressible/low Mach algorithms. See Algorithm 1.

Once we introduce the VMS-based stabilization terms, new coupling terms involving the three unknown conservative variables appear. Hence, some extra information is required in order to achieve the complete uncoupling process. When needed, we may replace $\boldsymbol{U}^{n+1}$ by $\widetilde{\boldsymbol{U}}^{n+1}$, $\boldsymbol{\varrho}^{n+1}$ by $\widehat{\boldsymbol{\varrho}}_{k}^{n+1}$ and $\boldsymbol{E}^{n+1}$ by $\widehat{\boldsymbol{E}}_{k}^{n+1}$. These approximations are supported by Remark 4.1. The remaining terms which could still couple the problem variables can be evaluated by taking the known values of the unknowns from the previous iteration, time step or from the intermediate equations. In particular, note that density and energy equations become coupled, an important fact for the development of the whole methodology. The coupling blocks are taken to the right-hand side and treated in an explicit manner, generally with the most up-to-date known values. However, when evaluating the residuals and the projections, we have found of critical importance to perform the computations with all the terms evaluated at the same time instant. Otherwise, convergence problems may appear which we associate with the strong coupling between thermodynamic variables.

Remark 4.4. In the design of fractional step methods for incompressible flows there are two equations in play taking velocity and pressure as main variables. The classic and well-established pressure-correction procedure [25,28] allows to split the calculation of the unknowns after performing some extrapolation in the momentum equation, prior to solving the continuity one. A similar design was carried out in [29] for isentropic flows. However, for the case of compressible flows, as the problem includes also the energy equation, there are more possibilities to perform the splitting and hence to design the algorithm. Our intention in this paper was to follow the same pattern as in previous developments in our group [25–30,46] where the first equation to be solved is the intermediate momentum equation. Nevertheless, other approaches might be explored.

Algorithm 1 Fractional-step method, order $k = 1, 2$.

(1) Compute the intermediate momentum with density and energy extrapolations as follows:

$$\left[\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\frac{D_k}{\delta t} + \boldsymbol{C}^{\rho\mathbf{u}}_{\rho\mathbf{u}}(\widehat{\boldsymbol{\varrho}}^{n+1}_k, \widetilde{\boldsymbol{U}}^{n+1}) + \boldsymbol{D}^{\rho\mathbf{u}}_{\rho\mathbf{u}}(\widehat{\boldsymbol{\varrho}}^{n+1}_k)\right]\widetilde{\boldsymbol{U}}^{n+1} = \boldsymbol{F}^{n+1}_{\rho\mathbf{u}} - \boldsymbol{C}^{\rho}_{\rho\mathbf{u}}(\widehat{\boldsymbol{\varrho}}^{n+1}_k, \widetilde{\boldsymbol{U}}^{n+1})\boldsymbol{\varrho}^{n+1}_{k-1} - \boldsymbol{D}^{\rho}_{\rho\mathbf{u}}(\widehat{\boldsymbol{\varrho}}^{n+1}_k, \widetilde{\boldsymbol{U}}^{n+1})\widehat{\boldsymbol{\varrho}}^{n+1}_k - \boldsymbol{C}^{\rho e}_{\rho\mathbf{u}}\widehat{\boldsymbol{E}}^{n+1}_k$$

(2) Compute density using the previous intermediate momentum values:

$$\left[\boldsymbol{M}^{\rho}_{\rho}\frac{D_k}{\delta t} + \phi_k\delta t\boldsymbol{L}\right]\boldsymbol{\varrho}^{n+1} = \boldsymbol{F}^{n+1}_{\rho} - \boldsymbol{C}^{\rho\mathbf{u}}_{\rho}\widetilde{\boldsymbol{U}}^{n+1} + \phi_k\delta t\boldsymbol{L}\widehat{\boldsymbol{\varrho}}^{n+1}_{k-1}$$

(3) Compute energy using intermediate momentum and density solutions:

$$\left[\boldsymbol{M}^{\rho e}_{\rho e}\frac{D_k}{\delta t} + \boldsymbol{C}^{\rho e}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \widetilde{\boldsymbol{U}}^{n+1}) + \boldsymbol{D}^{\rho e}_{\rho e}(\boldsymbol{\varrho}^{n+1})\right]\boldsymbol{E}^{n+1} = \boldsymbol{F}^{n+1}_{\rho e} - \boldsymbol{C}^{\rho}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \widetilde{\boldsymbol{U}}^{n+1}, \widehat{\boldsymbol{E}}^{n+1}_k)\boldsymbol{\varrho}^{n+1} - \boldsymbol{D}^{\rho}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \widetilde{\boldsymbol{U}}^{n+1}, \widehat{\boldsymbol{E}}^{n+1}_k)\boldsymbol{\varrho}^{n+1} - \boldsymbol{C}^{\rho\mathbf{u}}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \widetilde{\boldsymbol{U}}^{n+1}, \widehat{\boldsymbol{E}}^{n+1}_k)\widetilde{\boldsymbol{U}}^{n+1} - \boldsymbol{D}^{\rho\mathbf{u}}_{\rho e}(\boldsymbol{\varrho}^{n+1}, \widetilde{\boldsymbol{U}}^{n+1})\widetilde{\boldsymbol{U}}^{n+1}$$

(4) Momentum correction, i.e., end-of-step momentum calculation:

$$\boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\frac{\boldsymbol{U}^{n+1}}{\phi_k\delta t} = \boldsymbol{M}^{\rho\mathbf{u}}_{\rho\mathbf{u}}\frac{\widetilde{\boldsymbol{U}}^{n+1}}{\phi_k\delta t} - \boldsymbol{C}^{\rho}_{\rho\mathbf{u}}(\boldsymbol{\varrho}^{n+1} - \boldsymbol{\varrho}^{n+1}_{k-1})$$

5. Numerical examples

In this section, a set of numerical examples is presented to show the capabilities of the proposed fractional step method for the simulation of compressible flows at different regimes.

All the implementations of the algorithms in this paper have been carried out into our in-house code `FEMUSS` (Finite Element Method Using Subgrid Scales). `FEMUSS` is an object-oriented `Fortran`-based finite element code which follows a modular approach for multiphysics interaction and performs parallel computations under `MPI` directives. In order to solve the final underlying systems of linear equations, and if nothing else is stated, we make use of an iterative algorithm based on the stabilized version of the BiConjugate Gradient method `BiCGstab` [47], which is already included as a part of the parallel solver library `PETSc` [48], which has been coupled with `FEMUSS`.

We consider an ideal gas with $\gamma = 1.4$ ($R_g = 287$ J/(kg K)) and $c_p = 1004.5$ J/(kg K) and $\mathrm{Pr} = 0.72$. A maximum of 10 iterations is set to solve each non-linear problem, and the relative numerical tolerance for the L^2-norm is 1×10^{-5}. All the plots are in SI units.

5.1. Convergence test

In this first example we consider a simple convergence test whose goal is to check numerically the time rate of convergence for the proposed fractional step algorithm. We recall that as time integration scheme we use backward differences, of the same order as the fractional step to be tested. Here we make use of the manufactured solutions method, which serves as a tool to quantify the numerical errors of PDE solvers. In particular, exact solutions for density, momentum and energy are specified a priori, which are composed of smooth functions, although they have no physical meaning. The compressible Navier–Stokes problem is solved over the unit square $\overline{\Omega} = [0, 1] \times [0, 1]$ and the force term is set so that the exact solution of the problem is

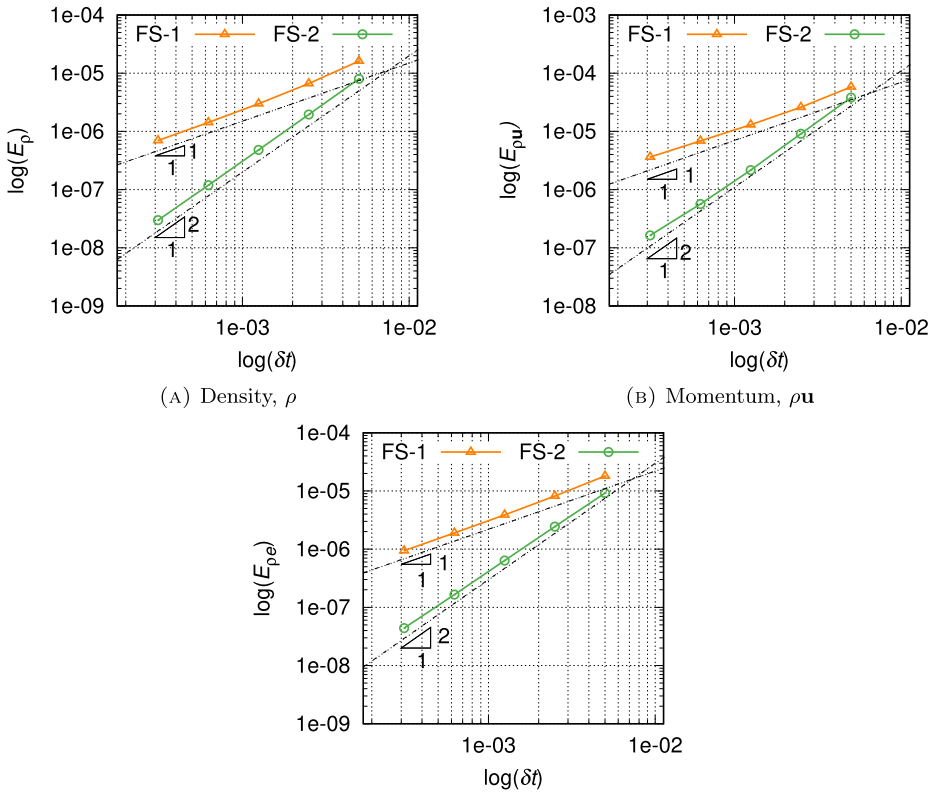
$$\rho(x, y, t) := \pi + x\cos(\sin t) + y\sin(\sin t), \tag{51a}$$

$$\rho u(x, y, t) := -y\cos(t), \tag{51b}$$

$$\rho v(x, y, t) := x\cos(t), \tag{51c}$$

$$\rho e(x, y, t) := 4\pi + x\cos(\sin(t)) + y\sin(\sin(t)). \tag{51d}$$

The finite element partition is structured and uniform. It contains $\mathrm{Q}_1/\mathrm{Q}_1/\mathrm{Q}_1$ finite elements of size $h = 1/100$. Both the boundary and initial conditions are evaluated from the previous equations, and particularized for each of the sides of the square at each time step and for $t = 0$, respectively. We select a range of time step sizes and, in addition, we set $\mu = 1 \times 10^{-5}$ kg/(m s), $\kappa = 0.0015$ W/(m K), and we make use of a sparse direct solver from the `MUMPS` library [49,50]. The



(A) Density, ρ (B) Momentum, $\rho\mathbf{u}$

(C) Energy, ρe

Fig. 1. Convergence test: time convergence of the relative errors of the three unknown variables measured in the $\ell^2(0,t_f,L^2(\Omega))$-norm. The number after the dash symbol stands for first (1) or second (2) order scheme in time.

error between the exact solution and the numerical one is measured in the ℓ^2-norm of the sequence of spatial L^2-norms of the solutions, i.e.,

$$E_f := \left(\delta t \sum_{n=1}^{N} \frac{\| f_h^n - f(t^n) \|_{L^2}^2}{\| f(t^n) \|_{L^2}^2} \right)^{1/2},$$

for $f = \rho$, $\rho\mathbf{u}$ or ρe, respectively.

Fig. 1 shows the convergence results obtained with the fractional step algorithm for both first and second order time integration schemes. The expected convergence rate can be clearly seen for both temporal approximations and for all the time step sizes.

5.2. Supersonic viscous flow over a flat plate

The so–called Carter's flat plate is a simple and classical benchmark to examine the performance of the solver in a problem involving shock waves, boundary layers and the interaction between them. This example is based on a 2D viscous supersonic flow over a flat plate with conditions, $\mathrm{Re}_\infty = 1,000$ and $\mathrm{Ma}_\infty = 3.0$.

Fig. 2 shows the problem setup. If the coordinate origin is placed and the tip of the plate, the computational domain is the rectangle covering from -0.25 m $\leq x \leq 1.2$ m and $0 \leq y \leq 0.8$ m. Density, velocity and temperature are set at the left and top boundaries of the domain. These prescribed values are, respectively, $\rho_\infty = 1$ kg/m^3, $u_\infty = 1$ m/s, $v_\infty = 0$ m/s and $\vartheta_\infty = 2.769 \times 10^{-4}$ K. The no-slip boundary condition $\mathbf{u} = \mathbf{0}$ m/s is enforced at the plate wall, together with the stagnation temperature, which is computed as

$$\vartheta_{\mathrm{stag}} = \vartheta_\infty \left(1 + \frac{\gamma - 1}{2} \mathrm{Ma}_\infty^2 \right). \tag{52}$$

On the "symmetric wall", i.e., the boundary prior to the plate, normal velocity, tangential traction, and heat flux are all set to zero. No prescriptions are made at the outflow, and the computations are initialized with the free-stream values for each

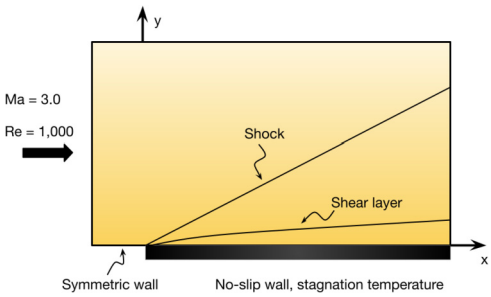


Fig. 2. Supersonic flow over a flat plate. Problem setup and boundary conditions.

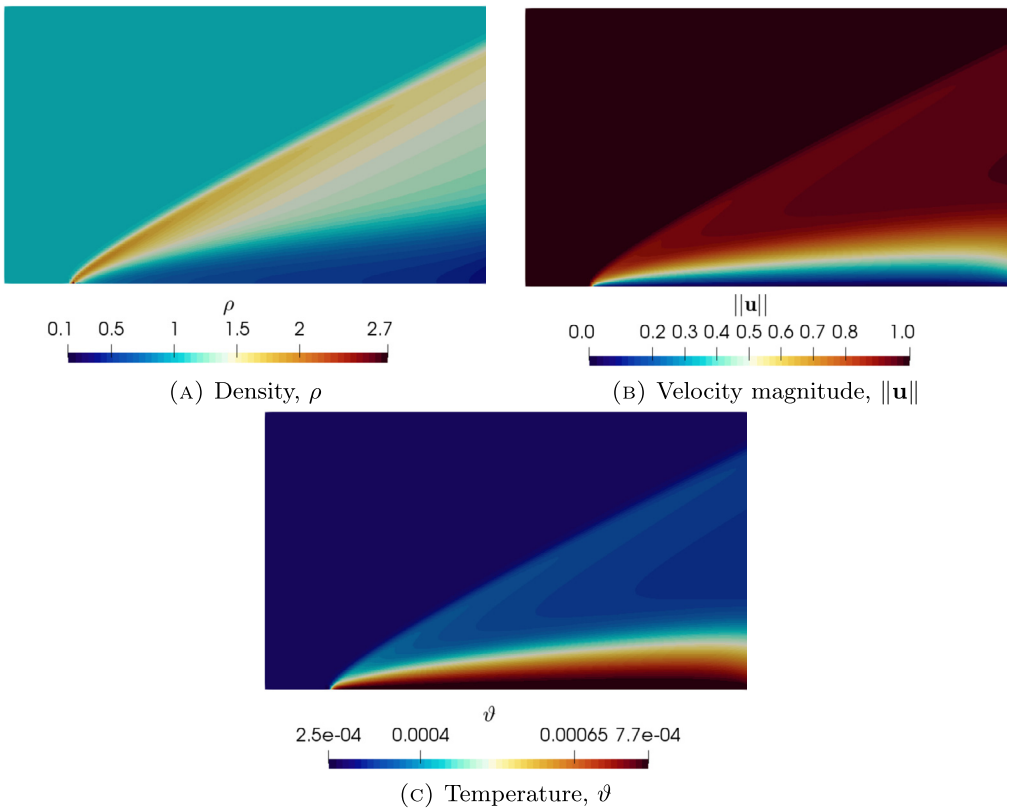


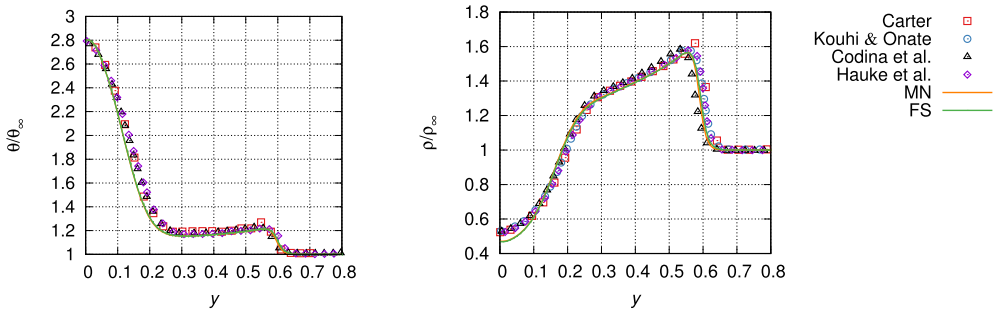
Fig. 3. Carter's flow problem: (a) density, (b) velocity magnitude and (c) temperature contours computed with the monolithic algorithm. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

degree of freedom in the entire domain. The temperature-dependent viscosity is computed according to the following form of the Sutherland's law,

$$\mu = \mu(\vartheta) = C_1 \frac{\vartheta^{1.5}}{\vartheta + S}, \tag{53}$$

where $S = 0.0001406$, and $C_1 = 0.0906$ is a scaling factor chosen so as to yield the desired free-stream conditions. The simulation was carried out using a time step corresponding to a CFL of 4. Here, we estimate the critical time step for the explicit scheme as the minimum of the stabilization parameters, i.e., $\delta t = \min(\tau_\rho, \tau_{\rho\mathbf{u}}, \tau_{\rho e})$.

In addition, we discretize the domain with a structured mesh of square elements of size $h = 0.01$. Fig. 3 displays the obtained contours of density, velocity and temperature. The results are in general agreement with the literature, e.g. [5,51–54]. The solution is computed with the gradient-based shock capturing operator. Several values for the operator constant C_{sc} were tested, $C_{sc} = \{0.25, 0.5, 0.75\}$ although the results were all really similar. We shall use $C_{sc} = 0.25$ in the upcoming examples.



(A) Normalized temperature, $\vartheta/\vartheta_\infty$

(B) Normalized density, ρ/ρ_∞

Fig. 4. Carter's flow problem: comparison of the obtained (a) normalized temperature and (b) normalized density along the line $x = 1.2$ m with results reported in the literature [5,51,52,54]. Hereinafter, MN stands for monolithic results and FS for fractional step results.

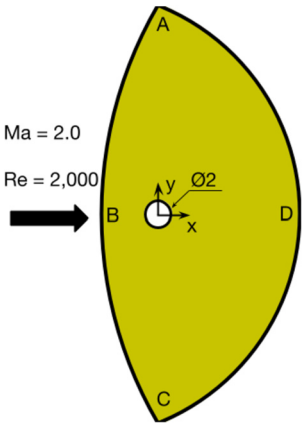


Fig. 5. Supersonic flow over a cylinder: domain specification. The computational domain is defined as the area enclosed within two arcs passing, respectively, through points A(0,15), B(-4,0), C(0,-15), and A, D(10,0) and C. The cylinder is placed at the origin of coordinates with diameter equal to 2 m.

In Fig. 4, we plot the normalized density and temperature profiles along the line $x = 1.2$ for the stationary solution, in order to further compare our solutions with the literature. Although the obtained peak point values are not coincident with the reference ones, an overall good agreement with the reference results can be observed.

Since the solution is stationary, this example serves to demonstrate that the approximations in the matrices introduced in the design of the fractional step scheme maintain the accuracy and the ability to model shocks of the monolithic formulation.

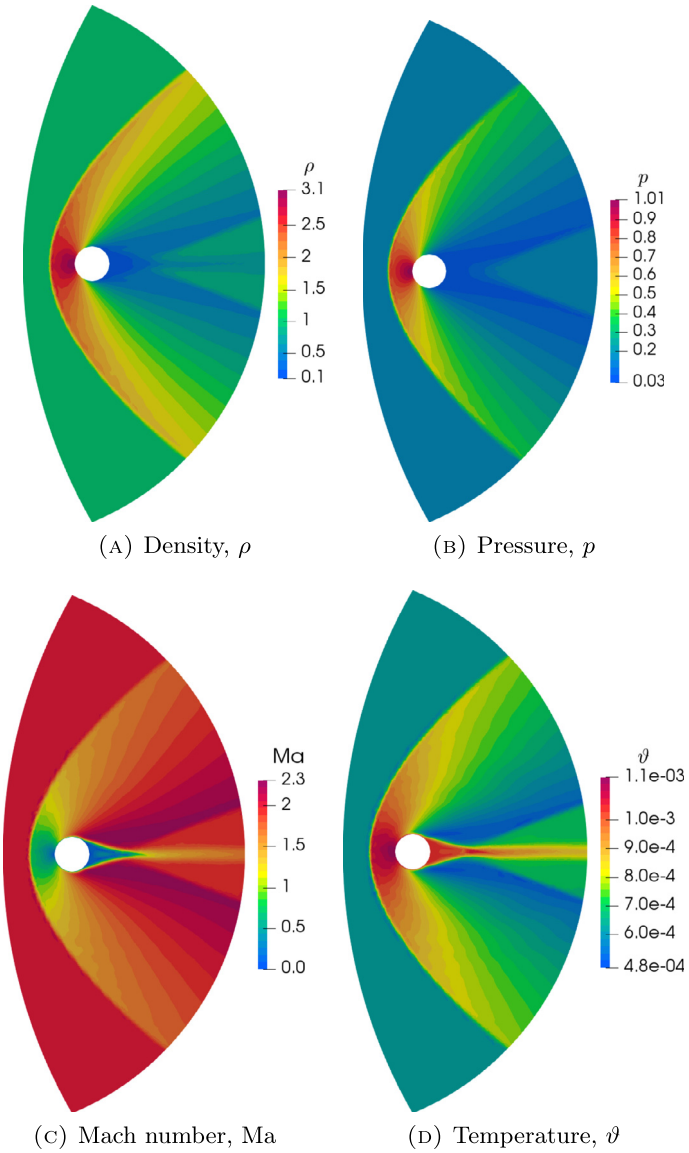
5.3. Supersonic viscous flow over a cylinder

Another classical example in compressible flows is the supersonic viscous flow over a cylinder, as sketched in Fig. 5, with free stream conditions $\mathrm{Re}_\infty = 2,000$ and $\mathrm{Ma}_\infty = 2$.

The specification of boundary conditions is as follows: all variables are specified on the upstream boundary matching those conditions. The cylinder wall is assumed to be adiabatic, no-slip condition is specified for the velocity on its surface, and at the downstream boundary, no conditions are imposed. The computations are initialized with the free-stream values for each degree of freedom in the entire domain. Likewise, we set $\mu = 0.001$ kg/(m s) and $\kappa = 1.39514$ W/(m K).

The finite element mesh is unstructured and it is composed by 31,288 linear triangular elements. Smaller elements are used near the wall cylinder, whereas the mesh is coarser in the rest of the domain. The mesh size was fixed to $h = 0.005$ m in the finer region near the cylinder wall.

Comparing the solution obtained with monolithic and fractional step formulations, the field contours are almost identical, what shows, as in the previous example, that the space accuracy is not affected by the design of the fractional step scheme. Density, pressure, Mach number and temperature distributions are shown in Fig. 6. A supersonic expansion develops from the cylinder surface, while in the wake the flow pattern is characterized by a recirculation zone and a weak tail shock. The presented contours for the steady state solution are in accordance with the ones presented in [16,43,55], although neither Sutherland law nor power law were used for viscosity in this example.



(A) Density, ρ (B) Pressure, p

(C) Mach number, Ma (D) Temperature, ϑ

Fig. 6. Supersonic flow over a cylinder: (a) density, (b) pressure, (c) Mach number and (d) temperature contours. The solution is obtained using the fractional step formulation together with the isotropic gradient-based shock capturing method.

Since no noticeable qualitative differences were observed in our results, some aerodynamic integral values of the flow were computed. In particular, we calculate the lift and drag non-dimensional coefficients, i.e.,

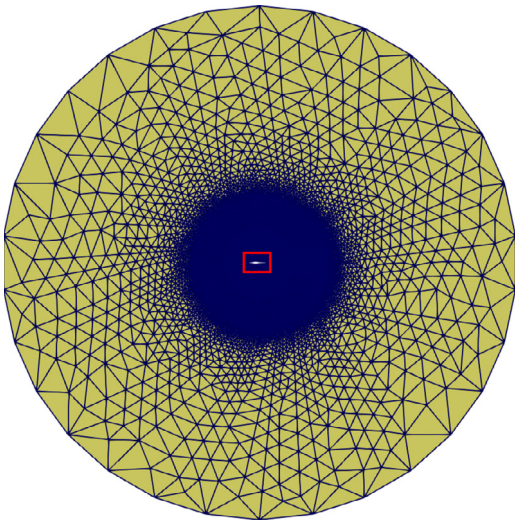
$$C_d = \frac{\mathrm{f}_x}{\frac{1}{2}\rho_\infty \|\mathbf{u}\|_\infty^2 D}, \tag{54a}$$

$$C_l = \frac{\mathrm{f}_y}{\frac{1}{2}\rho_\infty \|\mathbf{u}\|_\infty^2 D}, \tag{54b}$$

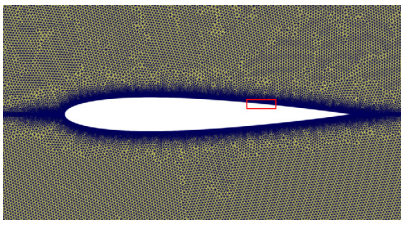
where the exerted force of the fluid over the cylinder is

$$\mathbf{f} = -\int_{\partial\Omega_{\text{Cylinder}}} \left(-p\mathbf{I}_{n_{\text{sd}}} + \boldsymbol{\sigma}^{\text{d}}\right)\cdot\mathbf{n}\,\mathrm{d}\partial\Omega. \tag{55}$$

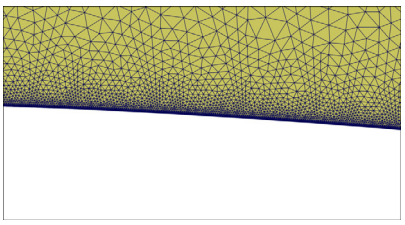
The results are $C_d = 1.439$ for the monolithic case and $C_d = 1.442$ for the fractional step counterpart, which have an accuracy comparable to that reported in [55], $C_d = 1.44$.



(A) General view of the mesh.



(B) Mesh around the airfoil.



(C) Airfoil surface refinement.

Fig. 7. Mesh refinement for the NACA 0012 simulations.

5.4. NACA0012 airfoil

In this section we consider the well-known geometry of the NACA0012 airfoil with chord length $c = 1$ m and a sharp trailing edge. The objective now is to test our methodology in a wider range of regimes. For this purpose, we consider two cases with angle of attack AoA $= 0°$: first a subsonic case with $\mathrm{Ma}_\infty = 0.5$ and $\mathrm{Re}_\infty = 5,000$, and later $\mathrm{Ma}_\infty = 0.85$ and $\mathrm{Re}_\infty = 10,000$, a transonic problem.

In order to perform the simulation, we define a circular O-type domain with the mid-chord point of the airfoil located at the coordinate origin. The far-field boundary is placed at 15 chord lengths from the airfoil. In the inflow part of the boundary, velocity and temperature are fixed according to the selected Reynolds and Mach numbers, whereas in the outflow part only density is prescribed. A no slip adiabatic wall condition is imposed at the airfoil surface and the computations are initialized with free-stream conditions in the entire domain. Furthermore, we use an unstructured mesh of triangular elements, including non-uniform refinement towards the airfoil surface. Fig. 7 reports the details of the mesh, which features 254,186 elements. The problem is solved with a time step corresponding to a CFL of 12.

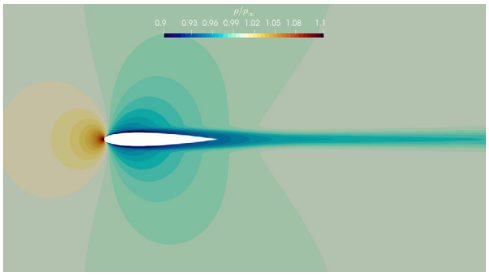
5.4.1. Ma=0.5, Re=5,000

First we present the subsonic viscous flow simulation. Fig. 8 displays density, Mach number and temperature distributions for the steady-state solution of this problem. Likewise, Fig. 9 contains the chord-wise distributions of pressure and skin friction coefficients, comparing both fractional step and monolithic results with other researchers [51,55,56]. These are respectively computed from

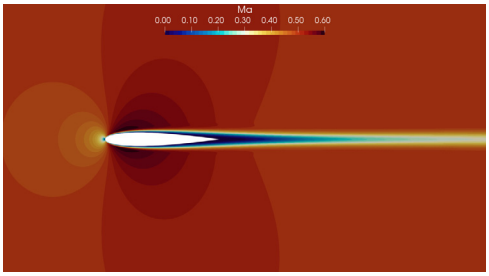
$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty \|\mathbf{u}\|_\infty^2}, \tag{56a}$$

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho_\infty \|\mathbf{u}\|_\infty^2}, \tag{56b}$$

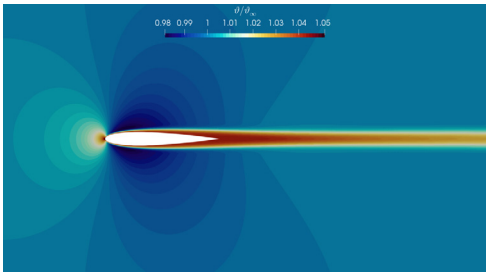
where p_∞ is the inflow static pressure and $\tau_w = (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{t}$ is the wall stress, with $\mathbf{t}$ the tangent vector to the surface. It is observed that the results exhibit a clear agreement with the literature.



(A) Density, ρ/ρ_∞

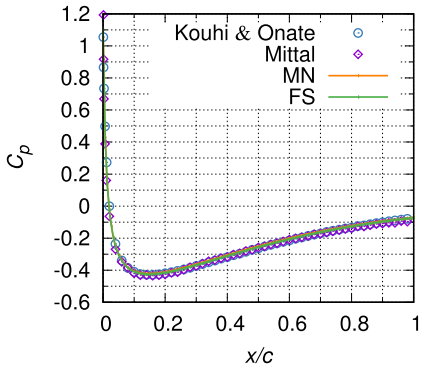


(B) Mach number

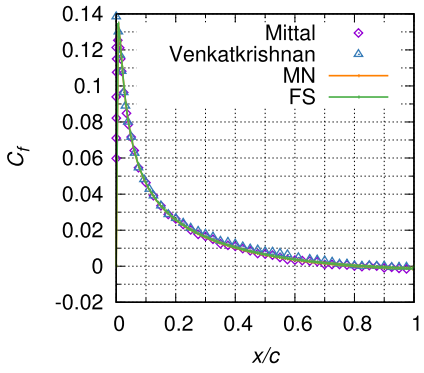


(C) Temperature, $\vartheta/\vartheta_\infty$

Fig. 8. Subsonic flow over a NACA 0012 profile: (a) density, (b) Mach number and (c) temperature distributions around the airfoil computed with the second order version of the fractional step scheme.



(A) Pressure coefficient, C_p



(B) Skin friction coefficient, C_f

Fig. 9. Subsonic flow over a NACA 0012 profile: (a) pressure coefficient, and (b) skin friction coefficient on the airfoil.

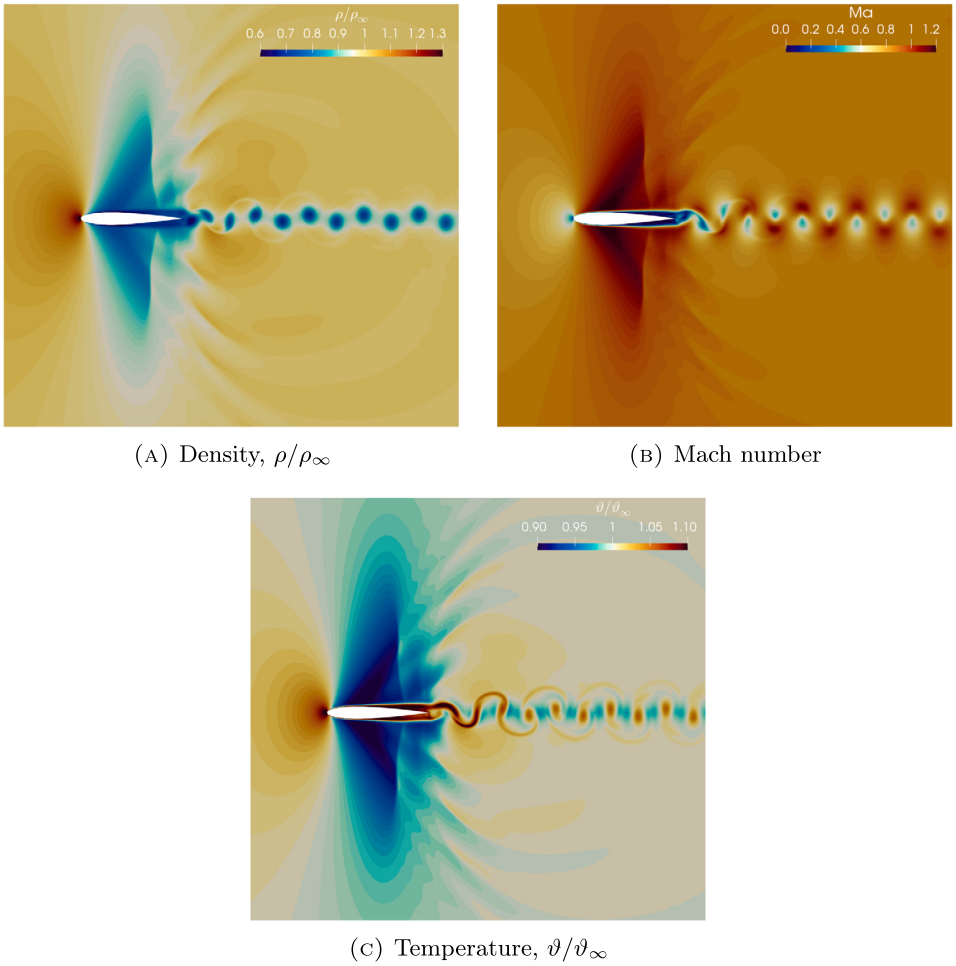
Table 1
Subsonic flow over a NACA 0012 profile: comparison of drag coefficient and separation point.

Ref.	C_d	x_s/c
Mittal [55]	0.0550	0.813
Venkatkrishnan [56]	0.0554	[0.810, 0.825]
Present-Monolithic	0.0548	0.827
Present-Fractional	0.0551	0.829

Additionally, the drag coefficient values are $C_d = 0.0548$ for the monolithic computations and $C_d = 0.0551$ for the fractional step counterpart, which are in line with the values reported by Mittal [55], $C_d = 0.0550$ and Venkatkrishnan [56], $C_d = 0.0554$. Finally, for the separation point, it is located at 82.7% and 82.9% of the chord from the leading edge, respectively for monolithic and fractional step algorithms. These results are summarized in Table 1.

5.4.2. Ma=0.85, Re=10,000

We switch now the conditions of the problem to a transonic regime with $\mathrm{Re} = 10,000$ and turn on the gradient-based shock capturing operator with $C_{sc} = 0.25$. After an initial transient period is overcome, a fully developed periodic solution



(A) Density, ρ/ρ_∞ (B) Mach number

(C) Temperature, $\vartheta/\vartheta_\infty$

Fig. 10. Transonic flow over a NACA 0012 profile: (a) density, (b) Mach number and (c) temperature distributions around the airfoil computed with the second order version of the fractional step scheme.

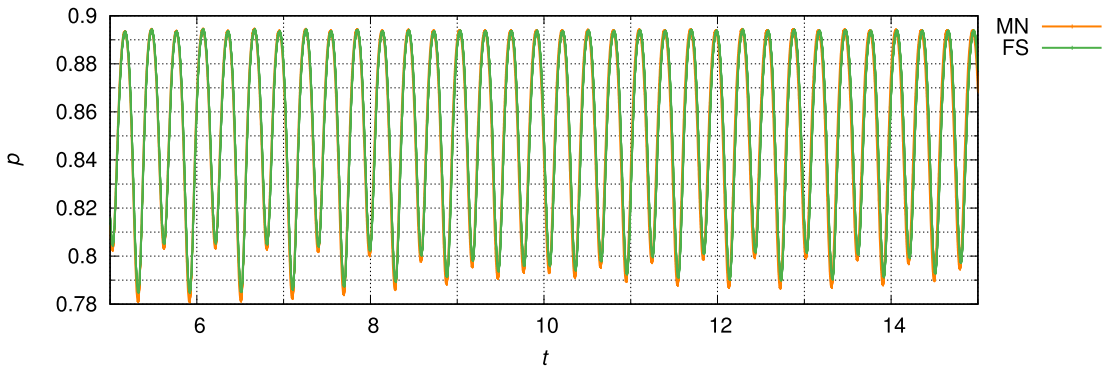


Fig. 11. Transonic flow over a NACA 0012 profile: pressure time history for both formulations at point (3.5,0) located at the wake of the airfoil.

is established. Fig. 10 displays density, Mach number and temperature distributions once this periodic flow has been fully developed, and Fig. 11 contains the pressure time history comparing fractional step and monolithic solutions. Both formulations obtain very similar results. The flow evolves as described in the literature [57]: it originally expands from the front stagnation point and then the boundary layer thickness begins to increase until the interaction between the layer and the shock wave results in the separation of flow. It is to be noticed that the vortices are shed from both, the upper and the lower surfaces of the airfoil. Moreover, two different instability mechanisms are active in the wake region, one mainly asso-

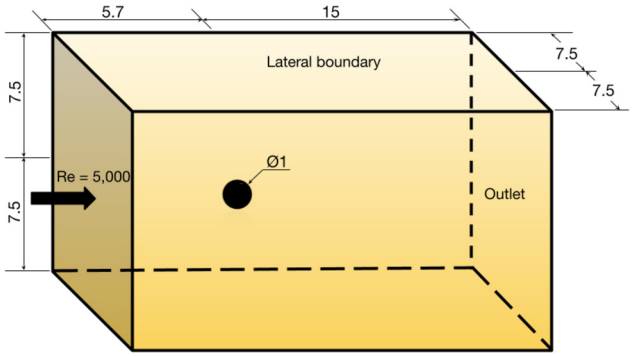


Fig. 12. Flow around a sphere. Problem setup and domain specification.

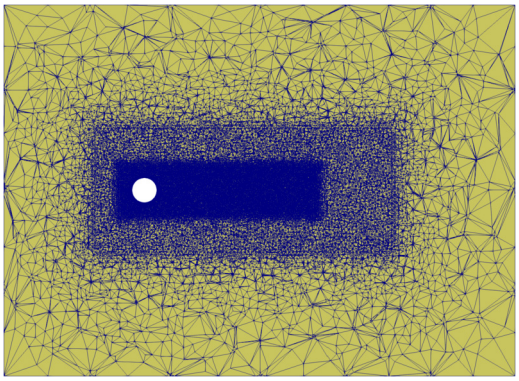


Fig. 13. Flow around a sphere. Mesh cut.

ciated with the shear layer (also known as Kelvin-Helmholtz mechanism), and the other one resulting from the interaction between the layer and the shock wave.

5.5. 3D flow over a sphere

In this final numerical example we model a 3D flow over a sphere. The purpose of this simulation is twofold. First, we use this example to show the applicability of the developed compressible formulation in complex 3D problems. Second, we will perform some simple tests in order to provide some insights on the actual computational savings that the fractional step implementation may offer with respect to the standard monolithic solver. We consider the flow around a sphere at $\mathrm{Re}_\infty = 5,000$ and two different Mach numbers, $\mathrm{Ma}_\infty = 0.25$ and $\mathrm{Ma}_\infty = 0.75$.

The problem setup is shown in Fig. 12. A uniform flow with the desired conditions impinges the left-inlet boundary, where density, velocity and temperature are prescribed. The lateral boundaries are defined as no-penetration and adiabatic, and the right-outflow boundary is traction free and adiabatic. The surface of the sphere is no-slip as well as adiabatic. As usual, the computations are initiated with free stream values. In order to build up the mesh, we first assign a size $h = 0.005$ over the sphere with uniform triangles. Likewise, two cylindrical refinement zones around and downstream of the sphere are introduced in order to better reproduce the wake of the flow, one with diameter of 2 m and the second one with diameter of 5 m. The remainder of the domain is filled with tetrahedral elements. A cut through the mesh is shown in Fig. 13 in order to illustrate the interior of the domain. The mesh accounts for a total of ~ 5.8 million elements. Fig. 14 shows streamlines just to give and idea of the features of the flow.

The left part in Table 2 shows the savings in CPU time of the fractional step algorithms (second order) for the two cases considered with respect to the monolithic formulation. A simple implementation over a series of time steps has been used. The savings are presented as the ratio between the CPU time of the fractional step scheme over the CPU time of the corresponding monolithic scheme. In addition, we collect the number of iterations needed by the solver to obtain the solution of the system of equations and the number of non-linear iterations used to obtain converged results. In fractional step schemes, each variable requires a different number of iterations to solve the corresponding linear system. Apart from the fact that the linear systems to be solved in the fractional step method are smaller and usually better conditioned, the main drawback of the monolithic formulation is that the total number of iterations of the linear system solver is driven by the slowest variable. This is at least our experience from our previous works on fractional step algorithms for incompressible, isentropic and compressible-primitive formulations (see the section of numerical results and conclusions in [28–30]). The general trend is that the slowest variable in those cases is the pressure. This point is illustrated by the first row of Table 2.

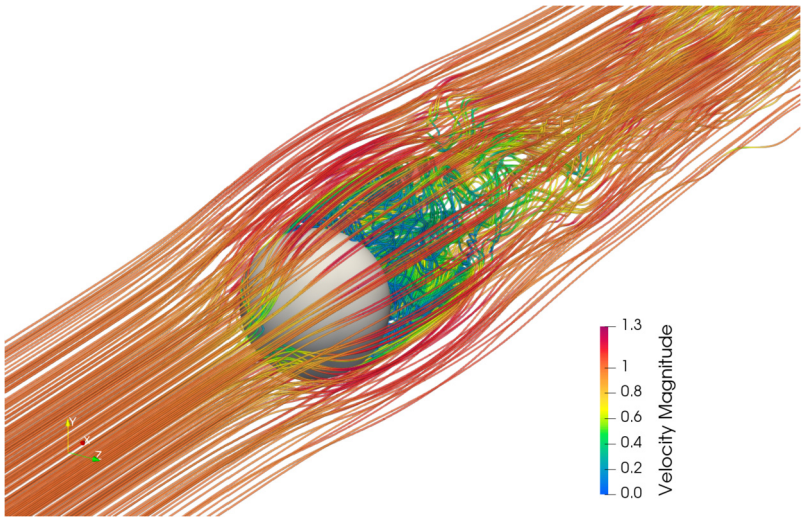


Fig. 14. Flow around a sphere: stream lines for $\mathrm{Ma} = 0.25$.

Table 2
Comparison between the number of non-linear and solver iterations of the monolithic and of the second order fractional step algorithm using the BDF2 time integrator. Here, $\overline{\mathrm{nni}}$ is the average number of nonlinear iterations to achieve convergence and $\overline{\mathrm{nsi}}$ stands for the average number of iterations needed by the iterative algebraic solver.

Case	Time ratio	Monolithic	Fractional step		
		$\overline{\mathrm{nni}}/\overline{\mathrm{nsi}}$	$\overline{\mathrm{nni}}_{\rho\mathbf{u}}/\overline{\mathrm{nsi}}_{\rho\mathbf{u}}$	$\overline{\mathrm{nsi}}_{\rho}$	$\overline{\mathrm{nni}}_{\rho e}/\overline{\mathrm{nsi}}_{\rho e}$
Ma=0.25	0.58	8/9	5/7	11	2/7
Ma=0.75	0.89	7/7	5/6	7	4/6

As expected, for the low Mach subsonic $\mathrm{Ma}_\infty = 0.25$ case the slowest variable is the density (pressure). Not only there is a reduction in the computational cost but also the non-linearities are solved in a better manner. However, the benefit of using a fractional step scheme in conservative variables degrades as the Mach number is progressively increased. As the problem is far from the incompressible behavior, the density stops being the slowest variable and then we observe that the number of iterations of the solver is basically the same for both monolithic and fractional step methodologies. This is why the obtained savings of the $\mathrm{Ma}_\infty = 0.75$ case are not very important, although more important savings should be expected in very large scale problems.

6. Conclusions

In this article, we have introduced a fractional step technique to solve the compressible Navier–Stokes equations using conservative variables. The development of this methodology, which is up to second order in time, has been designed at the algebraic level, departing from the fully discretized monolithic problem both in space and in time and by using the extrapolation concept.

From the numerical viewpoint, the herein proposed fractional step compressible model is based on a stabilized VMS method and an implicit scheme to advance the solution in time. In addition, other ingredients were appended, such as the orthogonal and dynamic definition of subscales and the shock-capturing operator which is calculated by using the orthogonal projection onto the finite element space of the gradient of the solution.

First, we have shown that the fractional step method introduces a splitting error but it maintains the general temporal accuracy of the time integration scheme. Furthermore, the supersonic viscous flow over a cylinder and the classical flow over a plate have been used to test the performance of the algorithm in the supersonic regime, where the interaction of shocks and shear layers is relevant. Similarly, the NACA0012 flow problem was used to evaluate the behavior of the implementation in subsonic and transonic cases also with adequate results. Finally we have shown an involved 3D case and also we have provided some insights on the possible savings that a fractional step scheme may offer.

CRediT authorship contribution statement

Samuel Parada: Writing – Original draft preparation, Methodology, Software, Numerical experiments. **Ramon Codina**: Conceptualization, Methodology, Writing – Reviewing and Editing. **Joan Baiges**: Software, Writing – Reviewing and Editing.

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.

Acknowledgements

Samuel Parada gratefully acknowledges the support received from the Agència de Gestió d'Ajuts Universitaris i de Recerca through the predoctoral FI grant 2019-FI-B-00607. Joan Baiges gratefully acknowledges the support of the Spanish Government through the Ramón y Cajal grant RYC-2015-17367. Ramon Codina gratefully acknowledges the support received from the ICREA Acadèmia Research Program of the Catalan Government. This work was partially funded through the TOP-FSI: RTI2018-098276-B-I00 project of the Spanish Government. CIMNE is a recipient of a "Severo Ochoa Programme for Centers of Excellence in R&D" grant (CEX2018-000797-S) by the Spanish Government.

Appendix A

In order to derive the complete expressions for the Euler and diffusion matrices, it is mandatory to express the constitutive relations in terms of the vector of conservative variables, namely $\boldsymbol{\mathcal{U}}$, defined in Eq. (8a).

In this section we will use index notation, with indexes i, j, k, l, b running from 1 to n_{sd}. Taking into account the definition of the viscous part of the Newtonian stress tensor in Eq. (2) and denoting by $\mathbf{m} := \rho\mathbf{u}$ the linear momentum, that tensor can be directly calculated using the conservative variables as

$$\sigma^{\text{d}}_{ij} = \nu\left(\frac{\partial m_i}{\partial x_j} + \frac{\partial m_j}{\partial x_i}\right) - \frac{2\nu}{3}\left(\frac{\partial m_b}{\partial x_b}\right)\delta_{ij} - \frac{\nu}{\rho}\left(m_i\frac{\partial \rho}{\partial x_j} + m_j\frac{\partial \rho}{\partial x_i}\right) + \frac{2\nu}{3\rho}\left(m_l\frac{\partial \rho}{\partial x_l}\right)\delta_{ij}. \tag{57}$$

In this equation, δ_{ij} is the Kronecker delta (i.e., $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$) and ν is the kinematic viscosity. In order to calculate the pressure and the speed of sound in terms of conservative variables, the caloric equation of state Eq. (5) and the ideal gas state equation (4) are used. It is then found that the pressure and the speed of sound can be written respectively as

$$p = (\gamma - 1)\left((\rho e) - \frac{\mathbf{m}\cdot\mathbf{m}}{2\rho}\right), \tag{58}$$

$$a = \sqrt{\gamma(\gamma-1)\left(\frac{(\rho e)}{\rho} - \frac{\mathbf{m}\cdot\mathbf{m}}{2\rho^2}\right)}. \tag{59}$$

In a similar fashion, applying the caloric and state equations to expand Eq. (3), the heat flux vector can be computed as

$$q_i = \left[\frac{\kappa(\rho e)}{c_v\rho^2} - \frac{\kappa(\mathbf{m}\cdot\mathbf{m})}{c_v\rho^3}\right]\frac{\partial\rho}{\partial x_i} + \frac{\kappa m_j}{c_v\rho^2}\frac{\partial m_j}{\partial x_i} - \frac{\kappa}{\rho c_v}\frac{\partial(\rho e)}{\partial x_i}. \tag{60}$$

The Euler matrix $\boldsymbol{\mathcal{A}}_j(\boldsymbol{\mathcal{U}})$ is developed using the spatial derivatives of the pressure using Eq. (58) and it gives

$$\boldsymbol{\mathcal{A}}_j(\boldsymbol{\mathcal{U}}) = \begin{bmatrix} 0 & \mathbf{e}_j^{\intercal} & 0 \\ \mathbf{u}u_j + a_1\mathbf{e}_j & \mathbf{I}_{n_{\text{sd}}}u_j + \mathbf{u}\otimes\mathbf{e}_j - (\gamma-1)(\mathbf{e}_j\otimes\mathbf{e}_j)\mathbf{u}^{\intercal} & (\gamma-1)(\mathbf{e}_j\otimes\mathbf{e}_j) \\ (a_1 - a_2)u_j & -(\gamma-1)\mathbf{u}^{\intercal}u_j + a_2\mathbf{e}_j^{\intercal} & \gamma u_j \end{bmatrix},$$

for $j = 1, \ldots, n_{\text{sd}}$, and where $\mathbf{e}_j$ stands for the unit vector in the j-th direction. In the previous definition, the thermodynamic relations a_1 and a_2 stand for

$$a_1 = \frac{1}{2}(\gamma-1)|\mathbf{u}|^2,$$
$$a_2 = \frac{(\rho e) + p}{\rho}.$$

Let us denote by $\mathbf{0}$ the vector of $\mathbb{R}^{n_{\text{sd}}}$ with null components. Using the viscous stress tensor and heat flux vector definitions based on conservative variables i.e. Eqs. (57) – (60) and the ideal gas law, each matrix component in the diffusive matrix can be respectively computed as

$$\boldsymbol{\mathcal{K}}_{jj}(\boldsymbol{\mathcal{U}}) = \begin{bmatrix} 0 & \mathbf{0}^{\intercal} & 0 \\ -\nu\mathbf{u} & \nu\mathbf{I}_{n_{\text{sd}}} & \mathbf{0} \\ (\alpha-\nu)|\mathbf{u}|^2 - \frac{\alpha}{\rho}(\rho e) & (\nu-\alpha)\mathbf{u}^{\intercal} & \alpha \end{bmatrix},$$

with $j = 1, \ldots, n_{\text{sd}}$ and,

$$\mathcal{K}_{kj}(\mathcal{U}) = \begin{bmatrix} 0 & \mathbf{0}^{\mathsf{T}} & 0 \\ -\nu u_k \mathbf{e}_j + \frac{2}{3}\nu u_j \mathbf{e}_k & \nu(\mathbf{e}_j \otimes \mathbf{e}_k) - \frac{2}{3}\nu(\mathbf{e}_k \otimes \mathbf{e}_j) & \mathbf{0} \\ -\frac{1}{3}\nu u_k u_j & \nu u_j \mathbf{e}_k^{\mathsf{T}} - \frac{2}{3}\nu u_k \mathbf{e}_j^{\mathsf{T}} & 0 \end{bmatrix},$$

with $j, k = 1, \ldots, n_{\text{sd}}$, and $k \neq j$. Finally, the reactive matrix can be written in terms of the external sources as

$$\mathcal{S} = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{f} & \mathbf{0} & 0 \\ r & \mathbf{f}^{\mathsf{T}} & 0 \end{bmatrix}. \tag{61}$$

References

[1] Y. Bazilevs, K. Takizawa, M.C.H. Wu, T. Kuraishi, R. Avsar, Z. Xu, T.E. Tezduyar, Gas turbine computational flow and structure analysis with isogeometric discretization and a complex-geometry mesh generation method, Comput. Mech. 67 (2021) 57–84.
[2] H. Lu, T. Yoshinaga, C. Li, K. Nozaki, A. Iida, M. Tsubokura, Numerical investigation of effects of incisor angle on production of sibilant /s/, Sci. Rep. 11 (2021) 16720.
[3] T. Tezduyar, T.J.R. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations, in: Proceedings of AIAA 21st Aerospace Sciences Meeting, AIAA Paper, vol. 83, 1983, pp. 1–25.
[4] A. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, in: Proceedings of FENOMEC'81, 1981.
[5] G. Hauke, A. Landaberea, I. Garmendia, J. Canales, A segregated method for compressible flow computation. Part II: general divariant compressible flows, Int. J. Numer. Methods Fluids 49 (2005) 183–209.
[6] G. Hauke, T.J. Hughes, A comparative study of different sets of variables for solving compressible and incompressible flows, Comput. Methods Appl. Mech. Eng. 153 (1) (1998) 1–44.
[7] B. Koobus, C. Farhat, A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes—application to vortex shedding, Comput. Methods Appl. Mech. Eng. 193 (15) (2004) 1367–1383.
[8] F.V.D. Bos, J.V.D. Vegt, B. Geurts, A multi-scale formulation for compressible turbulent flows suitable for general variational discretization techniques, Comput. Methods Appl. Mech. Eng. 196 (29) (2007) 2863–2875.
[9] W. Dahmen, T. Gotzen, S. Müller, R. Schäfer, Adaptive multiresolution finite volume discretization of the variational multiscale method. General framework, Inst. für Geometrie und Praktische Mathematik, 2011.
[10] V. Levasseur, P. Sagaut, F. Chalot, A. Davroux, An entropy-variable-based VMS/GLS method for the simulation of compressible flows on unstructured grids, Comput. Methods Appl. Mech. Eng. 195 (9) (2006) 1154–1179.
[11] T. Hughes, G. Feijóo, L. Mazzei, J. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Eng. 166 (1998) 3–24.
[12] T.J. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Eng. 127 (1) (1995) 387–401.
[13] T. Hughes, G. Scovazzi, L. Franca, Multiscale and stabilized methods, in: Encyclopedia of Computational Mechanics, Wiley, 2017.
[14] R. Codina, S. Badia, J. Baiges, J. Príncipe, Variational multiscale methods in computational fluid dynamics, in: Encyclopedia of Computational Mechanics, second edition, 2018, pp. 1–28.
[15] T.J.R. Hughes, G. Scovazzi, T.E. Tezduyar, Stabilized methods for compressible flows, J. Sci. Comput. 43 (2010) 343–368.
[16] F. Rispoli, R. Saavedra, A stabilized finite element method based on SGS models for compressible flows, Comput. Methods Appl. Mech. Eng. 196 (2006) 652–664.
[17] S. Marras, M. Moragues, M. Vázquez, O. Jorba, G. Houzeaux, Simulations of moist convection by a variational multiscale stabilized finite element method, J. Comput. Phys. 252 (2013) 195–218.
[18] T.J.R. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems, Comput. Methods Appl. Mech. Eng. 58 (3) (1986) 305–328.
[19] G.L. Beau, T. Tezduyar, Finite element computation of compressible flows with the SUPG formulation, Army High Performance Computing Research Center, 1991.
[20] T. Tezduyar, M. Senga, Stabilization and shock-capturing parameters in SUPG formulation of compressible flows, Comput. Methods Appl. Mech. Eng. 195 (13) (2006) 1621–1632.
[21] A.J. Chorin, A numerical method for solving incompressible viscous problems, J. Comput. Phys. 2 (1967) 12–26.
[22] R. Teman, Sur l'approximation de la solution des equations de Navier–Stokes par la méthode des pas fractionaires (I), Arch. Ration. Mech. Anal. 32 (1969) 135–153.
[23] J. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006) 6011–6045.
[24] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[25] S. Badia, R. Codina, Algebraic pressure segregation methods for the incompressible Navier–Stokes equations, Arch. Comput. Methods Eng. 15 (3) (2007) 1–52.
[26] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Comput. Phys. 170 (2001) 112–140.
[27] S. Badia, R. Codina, Pressure segregation methods based on a discrete pressure Poisson equation. An algebraic approach, Int. J. Numer. Methods Fluids 56 (2008) 351–382.
[28] E. Castillo, R. Codina, First, second and third order fractional step methods for the three-field viscoelastic flow problem, J. Comput. Phys. 296 (2015) 113–137.
[29] S. Parada, J. Baiges, R. Codina, A fractional step method for computational aeroacoustics using weak imposition of Dirichlet boundary conditions, Comput. Fluids 197 (2020).
[30] S. Parada, R. Codina, J. Baiges, Development of an algebraic fractional step scheme for the primitive formulation of the compressible Navier–Stokes equations, J. Comput. Phys. 433 (2021).
[31] J.M. Hérard, O. Hurisse, A fractional step method to compute a class of compressible gas-liquid flows, Comput. Fluids 55 (2012) 57–69.
[32] T. Gallouët, L. Gastaldo, J.C.L.R. Herbin, An unconditionally stable pressure correction scheme for the compressible barotropic Navier–Stokes equations, ESAIM: Math. Model. Numer. Anal. 42 (2) (2008) 303–331.

[33] K. Liu, R.H. Pletcher, A fractional step method for solving the compressible Navier–Stokes equations, J. Comput. Phys. 226 (2) (2007) 1930–1951.
[34] R. Codina, M. Vázquez, O. Zienkiewicz, A general algorithm for the compressible and incompressible flows. Part III: the semi–implicit form, Int. J. Numer. Methods Fluids 27 (1996) 13–32.
[35] D. Grapsas, R. Herbin, W. Kheriji, J.C. Latché, An unconditionally stable staggered pressure correction scheme for the compressible Navier–Stokes equations, SMAI J. Comput. Math. 2 (2016) 51–97.
[36] S. Chapman, T.G. Cowling, The Mathematical Theory of Non-uniform Gases, Cambridge University Press, Cambridge, 1991.
[37] J.O. Hirschfelder, C.F. Curtiss, R.B. Bird, Molecular Theory of Gases and Liquids, Wiley, Hoboken, 1954.
[38] B.R. Hollis, Real-gas flow properties for NASA Langley Research Center aerothermodynamic facilities complex wind tunnels, tech. rep., NASA Langley Research Center, Hampton, VA, United States, 1996.
[39] R. Codina, J. Príncipe, J. Baiges, Subscales on the element boundaries in the variational two-scale finite element method, Comput. Methods Appl. Mech. Eng. 198 (2009) 838–852.
[40] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Eng. 191 (2002) 4295–4321.
[41] R. Codina, J. Principe, O. Guasch, S. Badia, Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, Comput. Methods Appl. Mech. Eng. 196 (2007) 2413–2430.
[42] R. Codina, J. Principe, Dynamic subscales in the finite element approximation of thermally coupled incompressible flows, Int. J. Numer. Methods Fluids 54 (2007) 707–730.
[43] C. Bayona, J. Baiges, R. Codina, Variational multi-scale finite element solution of the compressible Navier–Stokes equations, Int. J. Numer. Methods Heat Fluid Flow 26 (2015) 1240–1271.
[44] R. Codina, Finite element approximation of the convection-diffusion equation: subgrid-scale spaces, local instabilities and anisotropic space-time discretizations, in: BAIL 2010 - Boundary and Interior Layers, Computational and Asymptotic Methods, Springer Berlin Heidelberg, 2011, pp. 85–97.
[45] R. Codina, A. Folch, A stabilized finite element predictor–corrector scheme for the incompressible Navier–Stokes equations using a nodal–based implementation, Int. J. Numer. Methods Fluids 44 (2004) 483–503.
[46] R. Codina, O. Soto, Approximation of the incompressible Navier–Stokes equations using orthogonal subscale stabilization and pressure segregation on anisotropic finite element meshes, Comput. Methods Appl. Mech. Eng. 193 (2004) 1403–1419.
[47] H.A.V. der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644.
[48] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, D.A. May, L.C. McInnes, R.T. Mills, T. Munson, K. Rupp, P. Sanan, B.F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc web page, http://www.mcs.anl.gov/petsc, 2015.
[49] P.R. Amestoy, I.S. Duff, J. Koster, J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 15–41.
[50] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Comput. 32 (2) (2006) 136–156.
[51] M. Kouhi, E. Oñate, An implicit stabilized finite element method for the compressible Navier–Stokes equations using finite calculus, Comput. Mech. 56 (2015) 113–129.
[52] O. Zienkiewicz, R. Codina, A general algorithm for the compressible and incompressible flows. Part I: the split, characteristic-based scheme, Int. J. Numer. Methods Fluids 20 (8–9) (1995) 869–885.
[53] F. Xu, G. Moutsanidis, D. Kamensky, M.-C. Hsu, M. Murugan, A. Ghoshal, Y. Bazilevs, Compressible flows on moving domains: stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling, Comput. Fluids 158 (2017) 201–220.
[54] J. Carter, Numerical solutions of the Navier–Stokes equations for the supersonic laminar flow over a two-dimensional compression corner, tech. rep., TR R-385, NASA, 1972.
[55] S. Mittal, Finite element computation of unsteady viscous compressible flows, Comput. Methods Appl. Mech. Eng. 157 (1998) 151–175.
[56] V. Venkatakrishnan, Viscous computations using a direct solver, Comput. Fluids 18 (2) (1990) 191–204.
[57] A. Bouhadji, M. Braza, Organised modes and shock–vortex interaction in unsteady viscous transonic flows around an aerofoil. Part I: Mach number effect, Comput. Methods Appl. Mech. Eng. 32 (2003) 1233–1260.