A finite element dynamical nonlinear subscale approximation for the low Mach number flow equations

Matias Avila [a], Javier Principe [b,*], Ramon Codina [b]

[a] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain
[b] Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

ARTICLE INFO

ABSTRACT

In this work we propose a variational multiscale finite element approximation of thermally coupled low speed flows. The physical model is described by the low Mach number equations, which are obtained as a limit of the compressible Navier–Stokes equations in the small Mach number regime. In contrast to the commonly used Boussinesq approximation, this model permits to take volumetric deformation into account. Although the former is more general than the latter, both systems have similar mathematical structure and their numerical approximation can suffer from the same type of instabilities.

We propose a stabilized finite element approximation based on the variational multiscale method, in which a decomposition of the approximating space into a coarse scale resolvable part and a fine scale subgrid part is performed. Modeling the subscale and taking its effect on the coarse scale problem into account results in a stable formulation. The quality of the final approximation (accuracy, efficiency) depends on the particular model.

The distinctive features of our approach are to consider the subscales as transient and to keep the scale splitting in all the nonlinear terms. The first ingredient permits to obtain an improved time discretization scheme (higher accuracy, better stability, no restrictions on the time step size). The second ingredient permits to prove global conservation properties. It also allows us to approach the problem of dealing with thermal turbulence from a strictly numerical point of view.

Numerical tests show that nonlinear and dynamic subscales give more accurate solutions than classical stabilized methods.

1. Introduction

The general description of a fluid flow involves the solution of the compressible Navier–Stokes equations. It is widely accepted that these equations provide an accurate description of any problem in fluid mechanics which may present many different nonlinear physical mechanisms. Depending on the physics of the problem under consideration, different simplified models describing some of these mechanisms can be derived from the compressible Navier–Stokes equations.

Our application is directed to low speed strongly thermally coupled flows which are described by the compressible Navier–Stokes equations in the low-Mach number limit. This limit is derived by an asymptotic expansion of the problem variables as power series of the small parameter $\gamma \mathrm{Ma}^2 \ll 1$, where γ denotes the specific heat ratio and Ma the Mach number of the problem. For details of this asymptotic expansion procedure, see [27,29,33]. As a particular result of this process, the total pressure is split into two parts, the thermodynamic part $p^{th}(t)$ which is uniform in space, and the hydrodynamic part

* Corresponding author.
E-mail addresses: mavila@cimne.upc.edu (M. Avila), principe@cimne.upc.edu (J. Principe), ramon.codina@upc.edu (R. Codina).

$p(\boldsymbol{x}, t)$ which is several orders of magnitude smaller than p^{th} and is therefore omitted in the state and energy equations. This leads to a removal of the acoustic modes but large variations of density due to temperature variations are allowed. This system of equations is commonly used to describe problems of combustion in the form of deflagrations (i.e., flames at low speed).

Despite this important difference in the treatment of the incompressibility, the low Mach number equations present the same mathematical structure as the incompressible Navier–Stokes equations, in the sense that the mechanical pressure is determined from the mass conservation constraint. Consequently the same type of numerical instabilities can be found, namely the problem of compatibility conditions between the velocity and pressure finite element spaces, and the instabilities due to convection dominated flows. These instabilities can be avoided by the use of stabilization techniques. A Galerkin finite element method can be used with mixed LBB stable elements, avoiding stabilization techniques when convection is not dominant [18,30]. Stabilized finite element methods (FEM) have been initially developed for the Stokes [22] and for the convection diffusion reaction (CDR) problems [6]. Later they have been extended to incompressible Navier–Stokes equations [7,26], and for the low Mach approximation [32] but the nonlinearity of the problem was not considered in their design. These extensions were essentially the application to nonlinear transient problems of a technique developed for linear steady ones.

The design of stabilization techniques considering the transient nonlinear nature of the problems began with the introduction of dynamic nonlinear subscales in [8,12]. Developed in the context of the variational multiscale (VMS) concept introduced by Hughes [21], the idea is to consider the subgrid scale time dependent and to consider its effect on all the nonlinear terms, resulting in extra terms in the final discrete scheme. Important improvements in the discrete formulation of the incompressible Navier–Stokes problem have been observed. From a theoretical point of view, the use of transient subgrid scales explains how the stabilization parameter should depend on the time step size and makes space and time discretizations commutative. The tracking of the subscales along the nonlinear process provides global momentum conservation for incompressible flows. From a practical point of view, the use of time dependent nonlinear subscales results in a more robust and more accurate method (an unusual combination) as shown by numerical experiments [8,12].

These developments also opened the door to the use of numerical techniques to cope with the potential instabilities and to model turbulence at the same time, as pointed out in [8,12]. This is a natural step as turbulence is originated by the presence of the nonlinear convective term, as it is well known. The idea of a large eddy simulation (LES) approach to turbulence modeling using only numerical ingredients actually goes back at least to [5], and the possibility to use the VMS framework for that purpose to [8]. It was fully developed for incompressible flows in [3] and for low Mach number flows recently in [15], where quantitative comparisons against direct numerical simulations are presented. It is important to point out, however, that not all the terms arising from the nonlinear scale splitting are considered in these works. Apart from these results, a careful analysis of the dissipative structure of the variational multiscale method with nonlinear time dependent subscales was presented in [17,34], showing the physical interpretation of the method. This analysis was extended to thermally coupled flows using the Boussinesq approximation in [10].

In this article we consider time dependent subscales and their effect in *all* the nonlinear terms in the low Mach number flow equations. It is shown that the method does not only provide the necessary stabilization of the formulation but also enables to obtain more accurate solutions than the classical linear approach for an equivalent mesh as it happened for incompressible flows. It is also shown that global conservation properties for mass, momentum and energy are obtained from the final discrete scheme.

The paper is organized as follows. In Section 2, the Low Mach number equations and their variational formulation are given. Afterwards the VMS formulation through dynamic scale splitting is derived in Section 3. It is shown in Section 4 that this formulation provides global mass, momentum and energy conservation when using equal interpolation spaces for the velocity, pressure and temperature equations. Time integration schemes are discussed in Section 5. The treatment of the nonlinear terms is described in detail in Section 6. The formulation is tested for both stationary and dynamic problems in Section 7. Conclusions are drawn in Section 8.

2. The low Mach number equations

2.1. Initial and boundary value problem

Let $\Omega \subset \mathbb{R}^d$, with $d = 2,3$ be the computational domain in which the flow takes place during the time interval $[0, t_{end}]$, and let $\partial\Omega$ be its boundary. The initial and boundary value problem to be considered consists of finding a velocity field $\boldsymbol{u}$, a hydrodynamic pressure field p, a temperature field T, and the thermodynamic pressure p^{th} such that

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \text{ in } \Omega, \quad t \in (0, t_{end}) \tag{1}$$

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot (2\mu \boldsymbol{\varepsilon}'(\boldsymbol{u})) + \nabla p = \rho \boldsymbol{g} \text{ in } \Omega, \quad t \in (0, t_{end}) \tag{2}$$

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \boldsymbol{u} \cdot \nabla T - \nabla \cdot (k \nabla T) - \alpha T \frac{dp^{th}}{dt} = Q \text{ in } \Omega, \quad t \in (0, t_{end}) \tag{3}$$

where ρ denotes the density, μ the viscosity, $\boldsymbol{\varepsilon}'(\boldsymbol{u}) = \boldsymbol{\varepsilon}(\boldsymbol{u}) - \frac{1}{3}(\nabla \cdot \boldsymbol{u})\boldsymbol{I}$ the deviatoric part of the rate of deformation tensor $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \nabla^s \boldsymbol{u} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$, $\boldsymbol{I}$ the identity tensor, $\boldsymbol{g}$ the gravity force vector, c_p the specific heat coefficient at constant pressure, k the thermal conductivity, Q the heat source, and $\alpha = -\frac{1}{\rho}\frac{\partial \rho}{\partial T}|_p$ the thermal expansion coefficient. Eqs. (1)–(3) represent the mass, momentum and energy conservation respectively. Additionally the system must be closed by a state equation relating density ρ, thermodynamic pressure p^{th} and temperature T of the form

$$\rho = \rho(T, p^{th}) \tag{4}$$

These equations must be supplied with initial and boundary conditions. Initial conditions are

$$\boldsymbol{u} = \boldsymbol{u}_0 \text{ in } \Omega, \quad t = 0$$
$$T = T_0 \text{ in } \Omega, \quad t = 0$$
$$p^{th} = p_0^{th} \text{ in } \Omega, \quad t = 0$$

whereas Dirichlet and Neumann boundary conditions for Eqs. (2) and (3) are

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \text{ on } \Gamma_D^{\boldsymbol{u}}$$
$$T = \widehat{T} \text{ on } \Gamma_D^T$$
$$(-p\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}'(\boldsymbol{u})) \cdot \boldsymbol{n} = \boldsymbol{t}_n \text{ on } \Gamma_N^{\boldsymbol{u}}$$
$$k\nabla T \cdot \boldsymbol{n} = q_n \text{ on } \Gamma_N^T$$

where $\boldsymbol{n}$ is the outer unit normal on the boundary and it is assumed that $\Gamma_D^f \cup \Gamma_N^f = \partial\Omega$, and $\Gamma_D^f \cap \Gamma_N^f = \emptyset$ for $f = T, \boldsymbol{u}$.

Determination of the thermodynamic pressure. The time dependence of thermodynamic pressure $p^{th}(t)$ has to be determined independently of Eqs. (1)–(3). For open flows $(\Gamma_N^{\boldsymbol{u}} \neq \emptyset)$ the thermodynamic pressure is given by the boundary conditions. For closed flows $(\Gamma_N^{\boldsymbol{u}} = \emptyset)$ the thermodynamic pressure is determined through global conservation equations over domain Ω, taking advantage of the uniformity of pth.

In a closed system without inflow–outflow, the total mass remains constant over time, and p^{th} may be obtained at each time subject to an integral form of the state equation

$$\int_\Omega \rho(T, p^{th}) d\Omega = \int_\Omega \rho_0 d\Omega \tag{5}$$

where $\rho_0 = \rho(T_0, p_0^{th})$ is the initial density field.

In a closed system with inflow–outflow, the thermodynamic pressure may be determined by an equation obtained as a result of combining Eqs. (1), (3) and (4), given by

$$\frac{\alpha T}{\gamma - 1}\frac{dp^{th}}{dt} + \frac{\gamma}{\gamma - 1}\frac{\alpha T}{K}\nabla \cdot \boldsymbol{u} - \nabla \cdot (k\nabla T) = Q \tag{6}$$

where γ, α and the compressibility coefficient $K = \frac{1}{\rho}\frac{\partial \rho}{\partial p}|_T$ are thermodynamic functions, depending on p^{th} and T. Integrating Eq. (6) over domain Ω yields an ordinary differential equation for p^{th} as

$$\frac{dp^{th}}{dt}\int_\Omega \frac{\alpha T}{\gamma - 1} d\Omega + \int_\Omega \frac{\gamma}{\gamma - 1}\frac{\alpha T}{K}\nabla \cdot \boldsymbol{u} d\Omega = \int_\Omega Q d\Omega + \int_{\partial\Omega} \boldsymbol{n} \cdot k\nabla T d\Gamma \tag{7}$$

which is subject to the initial condition $p^{th}(t = 0) = p_0^{th}$.

Ideal gases. For ideal gases, the state equation is $\rho = p^{th}/RT$, with $R = \frac{\mathcal{R}}{M}$, where $\mathcal{R}$ is the universal gas constant and M the mean molecular mass. The thermal expansion and the compressibility coefficients for ideal gases are $\alpha = 1/T$ and $K = 1/p^{th}$, respectively. Considering also uniform mean molecular mass (no combustion), Eqs. (5) and (7) take the form

$$p^{th} = p_0^{th}\frac{\int_\Omega \frac{1}{T_0} d\Omega}{\int_\Omega \frac{1}{T} d\Omega}$$

and

$$\frac{|\Omega|}{(\gamma - 1)}\frac{dp^{th}}{dt} + \frac{\gamma}{\gamma - 1}p^{th}\int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{u} d\Gamma = \int_\Omega Q d\Omega + \int_{\partial\Omega} \boldsymbol{n} \cdot k\nabla T d\Gamma \tag{8}$$

respectively, $|\Omega|$ being the measure of Ω.

2.2. Variational formulation

To obtain a variational formulation for the system (1)–(3), let us denote by $\boldsymbol{V}$, Q, W the functional spaces where the solution is sought. When the Boussinesq approximation is considered they are given by $\boldsymbol{V} = \boldsymbol{L}^2(0, T; H^1(\Omega)^d)$, $Q = \mathcal{D}'(0, T; L^2(\Omega))$, and $W = L^2(0, T; H^1(\Omega))$ [10], where $\mathcal{D}'(0, T; L^2(\Omega))$ is the set of $L^2(\Omega)$ functions in space which are distributions in time. For the

low Mach number equations, the minimum regularity required is only known in very particular cases [27]. The corresponding space of (time independent) test functions will be denoted by V_0, Q_0, W_0. Functions belonging to these spaces vanish on the part of the boundary where Dirichlet conditions are imposed. We also introduce the notation $(\cdot, \cdot) \equiv (\cdot, \cdot)_\Omega$ and $(\cdot, \cdot)_\Gamma$ for the L^2-inner product on Ω and Γ, respectively, or for the integral of the product of two functions if they are not square integrable but their product can be integrated (the product of two functions becomes the contraction when vectors or tensors are considered).

Using this notation the weak form of the problem consists of finding $(\boldsymbol{u}, p, T) \in (\boldsymbol{V}, Q, W)$ such that

$$\left(\frac{\partial \rho}{\partial t}, q \right) + (\nabla \cdot (\rho \boldsymbol{u}), q) = 0 \quad \forall q \in Q_0 \tag{9}$$

$$\left(\rho \frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v} \right) + (\rho \boldsymbol{u} \cdot \nabla \boldsymbol{u}, \boldsymbol{v}) + (2\mu \varepsilon'(\boldsymbol{u}), \nabla^s \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v}) = (\rho \boldsymbol{g}, \boldsymbol{v}) + (\boldsymbol{t}_n, \boldsymbol{v})_{\Gamma_N^u} \quad \forall \boldsymbol{v} \in \boldsymbol{V}_0 \tag{10}$$

$$\left(\rho c_p \frac{\partial T}{\partial t}, w \right) + (\rho c_p \boldsymbol{u} \cdot \nabla T, w) + (k \nabla T, \nabla w) - \left(\alpha T \frac{dp^{th}}{dt}, w \right) = (Q, w) + (q_n, w)_{\Gamma_N^T} \quad \forall w \in W_0 \tag{11}$$

3. Space discretization by scale splitting

Let us consider a finite element partition $\{K\}$ with n_e elements of the computational domain Ω, from which we can construct finite element spaces for the velocity, pressure and temperature in the usual manner. We will denote them by $\boldsymbol{V}_h \subset \boldsymbol{V}$, $Q_h \subset Q$ and $W_h \subset W$, respectively. We will assume that they are all built from continuous piecewise polynomials of the same degree k. In this section we assume zero Dirichlet boundary conditions to simplify the presentation.

Let us split the continuous space $\boldsymbol{Y} = \boldsymbol{V} \times Q \times W$ as $\boldsymbol{Y} = \boldsymbol{Y}_h \oplus \widetilde{\boldsymbol{Y}}$, where $\widetilde{\boldsymbol{Y}} = \widetilde{\boldsymbol{V}} \times \widetilde{Q} \times \widetilde{W}$ is the subgrid space, that can be in principle any space to complete $\boldsymbol{Y}_h = \boldsymbol{V}_h \times Q_h \times W_h$ in $\boldsymbol{Y}$. The continuous unknowns are split as

$$\boldsymbol{u} = \boldsymbol{u}_h + \tilde{\boldsymbol{u}} \tag{12}$$
$$p = p_h + \tilde{p} \tag{13}$$
$$T = T_h + \widetilde{T} \tag{14}$$

where the components with subscripts h belong to the corresponding finite element spaces, and the components with the ~ correspond to the subgrid space. These additional components are what we will call subscales.

Each particular variational multiscale method will depend on the way the subscales are approximated. As presented in [21], there are many possibilities such as hierarchical order refinement, (residual free) bubbles or approximation to the element Green functions. In fact, two level schemes in which the subscales themselves are expanded in terms of bases functions have also been used to model turbulent flows [23–25]. However, it is also common to perform approximations to the differential operator in the subscale equations in order to obtain closed expressions for them, usually in terms of the residuals of the finite element equations, as will be explained in Section 3. This way permits to recover classical stabilized methods developed for linear equations (e.g. the Douglas–Wang, method originally developed for the Stokes problem in [14]). *Our particular approach* is to keep time dependency of these subscales and keep the previous decompositions (12)–(14) in *all* the terms of the variational problem (9)–(11) even if the differential operator is approximated.

The only approximation we will make for the moment is to assume that the subscales vanish on the interelement boundaries, ∂K. This happens for example if one assumes subscales as bubble functions, or that their Fourier modes correspond to high wave numbers, as it is explained in [8]. Ideas in the direction of relaxing this condition can be found in [11].

Substituting decompositions (12)–(14) in the variational problem (9)–(11), taking the tests functions in the corresponding finite element spaces and integrating some terms by parts, the problem consists in finding $(\boldsymbol{u}_h, T_h, p_h) \in \boldsymbol{V}_h \times Q_h \times W_h$ such that

$$\left(\frac{\partial \rho^h}{\partial t}, q_h \right) - (\rho^h \boldsymbol{u}_h, \nabla q_h) + (\rho^h \boldsymbol{n} \cdot \boldsymbol{u}_h, q_h)_{\partial \Omega} - (\rho^h \tilde{\boldsymbol{u}}, \nabla q_h) = 0 \tag{15}$$

$$\left(\rho^h \frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}_h \right) + (\rho^h (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}_h, \boldsymbol{v}_h) + (2\mu \varepsilon'(\boldsymbol{u}_h), \nabla^s \boldsymbol{v}_h) - (p_h, \nabla \cdot \boldsymbol{v}_h) + \left(\rho^h \frac{\partial \tilde{\boldsymbol{u}}}{\partial t}, \boldsymbol{v}_h \right)$$
$$- \left(\tilde{\boldsymbol{u}}, -\frac{\partial \rho^h}{\partial t} \boldsymbol{v}_h + \rho^h (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla \boldsymbol{v}_h + \nabla^h \cdot (2\mu \varepsilon(\boldsymbol{v}_h)) \right) - (\tilde{p}, \nabla \cdot \boldsymbol{v}_h) = (\rho^h \boldsymbol{g}, \boldsymbol{v}_h) + (\boldsymbol{t}_n, \boldsymbol{v}_h)_{\Gamma_N^u} \tag{16}$$

$$\left(\rho^h c_p \frac{\partial T_h}{\partial t}, w_h \right) + (\rho^h c_p (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla T_h, w_h) + (k \nabla T_h, \nabla w_h) - \left(\alpha (T_h + \widetilde{T}) \frac{dp^{th}}{dt}, w_h \right) + \left(\rho^h c_p \frac{\partial \widetilde{T}}{\partial t}, w_h \right)$$
$$- \left(\widetilde{T}, -c_p \frac{\partial \rho^h}{\partial t} w_h + \rho^h c_p (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla w_h + \nabla^h \cdot (k \nabla w_h) \right) = (Q, w_h) + (q_n, w_h)_{\Gamma_N^T} \tag{17}$$

for any test functions $(\boldsymbol{v}_h, q_h, w_h) \in (\boldsymbol{V}_{0,h}, Q_{0,h}, W_{0,h})$, where

$$\rho^h = \rho(T_h + \widetilde{T}, p^{th}) \tag{18}$$

is obtained applying the scale splitting to the state Eq. (4). Notation ρ^h indicates that the obtained density for the discrete problem is different from that in the continuous problem. The use of a superscript instead of a subscript is because density does not belong to any of the introduced finite element spaces but it is a nonlinear function evaluated using the finite element and subgrid temperatures (and it therefore depends, indirectly, on the mesh). The symbol ∇^h in Eqs. (16) and (17) indicates that the integral is carried over the finite element interiors, and not over the edges, for example

$$(\widetilde{T}, \nabla^h \cdot (k\nabla w_h)) = \sum_K (\widetilde{T}, \nabla \cdot (k\nabla w_h))_K$$

where $(\cdot, \cdot)_K$ is the $L^2(K)$ inner product.

In order to give a closure to system (15)–(18) we need to define how the subscales $\tilde{\boldsymbol{u}}, \tilde{p}$ and $\widetilde{T}$ are computed, which will be discussed in the rest of the section. As mentioned before, we will finally provide closed expressions for them in terms of the residuals of the finite element equations at each (integration point of each) element, and therefore this will provide an implicit definition of the spaces of subscales (in terms of the finite element spaces). However, we would like to point out that, once the velocity subscale is approximated in the momentum Eq. (16), it provides additional terms than those that appear in classical stabilized finite element methods. These are non standard terms in the sense that they are usually neglected and appear because we keep the scale splitting also in nonlinear terms. The terms involving the velocity subgrid scale arising from the convective term in the momentum equation $(\rho^h\tilde{\boldsymbol{u}} \cdot \nabla\boldsymbol{u}_h, \boldsymbol{v}_h) - (\tilde{\boldsymbol{u}}, \rho^h(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla\boldsymbol{v}_h)$ can be understood as the contribution from the Reynolds- and cross-stress terms of a LES approach. Therefore, modeling $\tilde{\boldsymbol{u}}$ implies modeling the subgrid scale tensor. The last row in (16) comes from the contribution of the pressure subscale, that reinforces mass balance, and the contribution from the external forces. Similar comments to those made for the momentum equation apply to the energy Eq. (17). Once the temperature subscale is approximated it provides additional terms than those that appear in classical stabilized methods. The terms involving the velocity and temperature subgrid scale arising from the convective term $(\rho^h c_p \tilde{\boldsymbol{u}} \cdot \nabla T_h, w_h) - (\widetilde{T}, \rho^h c_p(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla w_h)$ can be understood as the contribution from the Reynolds- and cross-stress terms of a LES approach, see [10]. In the mass conservation Eq. (15) the fourth term provides pressure stability once the velocity subscale is approximated.

To get the final numerical scheme we approximate the subscales in the element interiors. We do that below, but let us first write the subgrid equations before any approximation (apart from assuming that the subscales vanish on the interelement boundaries). In the same way the finite element equations can be understood as the projection of the original equations onto the finite element spaces, the equations for the subscales are obtained by projecting the original equations onto their corresponding spaces $\widetilde{\boldsymbol{Y}}$. If $\widetilde{P}$ denotes the projection onto any of these spaces, the subscale equations are written as

$$\widetilde{P}(\rho^h\nabla \cdot \tilde{\boldsymbol{u}} - \rho^h\alpha(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla\widetilde{T}) = \widetilde{P}(R_c) \tag{19}$$

$$\widetilde{P}\left(\rho^h\frac{\partial\tilde{\boldsymbol{u}}}{\partial t} + \rho^h(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla\tilde{\boldsymbol{u}} - \nabla \cdot (2\mu\boldsymbol{\varepsilon}'(\tilde{\boldsymbol{u}})) + \nabla\tilde{p} \right) = \widetilde{P}(\boldsymbol{R}_m) \tag{20}$$

$$\widetilde{P}\left(\rho^h c_p\frac{\partial\widetilde{T}}{\partial t} + \rho^h c_p(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla\widetilde{T} - \nabla \cdot (k\nabla\widetilde{T}) \right) = \widetilde{P}(R_e) \tag{21}$$

where

$$R_c = -\frac{\partial\rho^h}{\partial t} - \rho^h\nabla \cdot \boldsymbol{u}_h + \rho^h\alpha(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla T_h \tag{22}$$

$$\boldsymbol{R}_m = \rho^h\boldsymbol{g} - \rho^h\frac{\partial\boldsymbol{u}_h}{\partial t} - \rho^h(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla\boldsymbol{u}_h + \nabla \cdot (2\mu\boldsymbol{\varepsilon}'(\boldsymbol{u}_h)) - \nabla p_h \tag{23}$$

$$R_e = Q + \alpha(T_h + \widetilde{T})\frac{dp^{th}}{dt} - \rho^h c_p\frac{\partial T_h}{\partial t} - \rho^h c_p(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla T_h + \nabla \cdot (k\nabla T_h) \tag{24}$$

are the residuals of the finite element unknowns in the momentum, continuity and heat equation, respectively.

Let us remark that writing some terms in the left or right-hand-side we are not assuming that they are known or unknown but just introducing a definition of the residuals (22)–(24). Eqs. (19)–(21) are actually coupled to the finite element Eqs. (15)–(18) forming a system whose linearization will be discussed in Section 6 after performing the approximations needed to obtain a numerical method. This definition of the residuals is motivated by the following observation: modeling the gradients of the subscales is much more involved than modeling the subscales themselves. Note that although all the unknowns are being split in Eqs. (15)–(18), those equations do not contain any subscale gradient nor any density gradient. This has been purposely achieved by a proper integration by parts of the continuous problem (9)–(11).

Approximation of the subscales. Up to this point the only approximation introduced is to assume that the subscales vanish on the element boundaries. This approximation is not sufficient to obtain a numerical method because the space of subscales is still infinite dimensional (the "broken" space $\cup_K H_0^1(K)$, for example) and therefore the subscale problem (19)–(21) is as difficult as the original continuous problem. To overcome this problem we adopt a simple approximation which consists in replacing the (spatial) differential operator by an algebraic operator which can be easily inverted. This old strategy, which can be understood as an approximation of the element Green's function by a Dirac distribution [21], has been already used, for example, in [8,31] and references therein, and it is briefly described below.

The differential Eqs. (19)–(21) over each element domain K can be written in vectorial form as

$$\widetilde{P}\left(\boldsymbol{M}\frac{\partial\widetilde{\boldsymbol{U}}}{\partial t} + \mathcal{L}\widetilde{\boldsymbol{U}}\right) = \widetilde{P}(\boldsymbol{R}) \text{ in } K$$

where $\widetilde{\boldsymbol{U}} \equiv [\tilde{\boldsymbol{u}}, \tilde{p}, \widetilde{T}]$, $\mathcal{L}$ is a spacial differential vector operator, $\boldsymbol{M}$ is the $(d+2) \times (d+2)$ diagonal matrix $\boldsymbol{M} = \mathrm{diag}(\rho^h\boldsymbol{I}_d, 0, \rho^h c_p)$, where $\boldsymbol{I}_d$ is the $d \times d$ identity matrix, and $\boldsymbol{R} \equiv [\boldsymbol{R}_m, R_c, R_e]$. In the present work we will consider the space of subscales as that of the residuals, that is, we will consider $\widetilde{P} = I$ (the identity) when applied to the finite element residuals. Another possibility, advocated in [8], consists in taking $\widetilde{P}$ as the projection onto the space orthogonal to the finite element space. Even though this method presents some advantages like better accuracy, a clear identification of the energy transfer mechanisms between the finite element scales and the subscales [34], as well as improved stability and convergence estimates for transient Stokes and incompressible flows [1,2], its use in the case of low Mach number equations requires further developments. Among them we can mention the problem of approximating a weighted projection that would guarantee exact $L^2(\Omega)$ orthogonality by the standard $L^2(\Omega)$ projection, an issue discussed in [8] in the case of incompressible flows, as well as the approximation of the $L^2(\Omega)$ projection by a lumped mass matrix inversion. Both problems can be circumvented introducing an appropriate iterative procedure but this procedure would interact with the linearization scheme which is one of the points in which we are interested herein. We will therefore restrict our attention to the choice $\widetilde{P} = I$, which is what can be considered as the standard approach in stabilized finite element methods.

We consider the algebraic approximation $\mathcal{L} \approx \boldsymbol{\tau}^{-1}$ in each K, where $\boldsymbol{\tau}$ is an $(d+2) \times (d+2)$ diagonal matrix. Taking $\boldsymbol{\tau} = \mathrm{diag}(\tau_m\boldsymbol{I}_d, \tau_c, \tau_e)$ the approximation to the subscales Eqs. (19)–(21) within each element of the finite element partition reads

$$\frac{1}{\tau_c}\tilde{p} = R_c \tag{25}$$

$$\rho^h\frac{\partial\tilde{\boldsymbol{u}}}{\partial t} + \frac{1}{\tau_m}\tilde{\boldsymbol{u}} = \boldsymbol{R}_m \tag{26}$$

$$\rho^h c_p\frac{\partial\widetilde{T}}{\partial t} + \frac{1}{\tau_e}\widetilde{T} = R_e \tag{27}$$

It is important to remark that, after the approximation $\mathcal{L} \approx \boldsymbol{\tau}^{-1}$, Eqs. (25)–(27) become a set of ordinary differential equations at each integration point. Nevertheless, we keep the notation $\frac{\partial}{\partial t}$ for the temporal derivative to emphasize the fact that the subscales depend on the position.

The stabilization parameters can be motivated by an approximated Fourier analysis performed in [8], but they are usually obtained by scaling arguments. The same analysis can be performed for the present variable-density equation system to obtain

$$\tau_c = \frac{h^2}{c_1\rho^h\tau_m} = \frac{\mu}{\rho^h} + \frac{c_2}{c_1}|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}|h \tag{28}$$

$$\tau_m = \left(c_1\frac{\mu}{h^2} + c_2\frac{\rho^h|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}|}{h}\right)^{-1} \tag{29}$$

$$\tau_e = \left(c_1\frac{k}{h^2} + c_2\frac{\rho^h c_p|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}|}{h}\right)^{-1} \tag{30}$$

where h is the element size and c_1 and c_2 are algorithmic constants whose values are $c_1 = 4$ and $c_2 = 2$ in the numerical experiments of Section 7 using linear elements.

It is important to remark that (26) and (27) are nonlinear equations as the velocity subscale contributes to the advection velocity in momentum and energy residuals and also in the stabilization parameters τ_c, τ_m, τ_e. The temperature subscale contributes through ρ^h to the residuals and to the coefficients of the stabilization parameters.

When the time derivative of the subscales is neglected, we will call them *quasi-static*, whereas otherwise we will call them *dynamic*. However, using quasi-static subscales in combination with the nonlinear model presented here is an approach we do not favor for the following reason. Solutions of the nonlinear Eqs. (26) and (27) display a dynamic behavior which may be radically different from the linear one. The values of $\tilde{\boldsymbol{u}}$ and $\widetilde{T}$ for which these equations are satisfied when $\partial\tilde{\boldsymbol{u}}/\partial t = 0$ and $\partial\widetilde{T}/\partial t = 0$ constitute the equilibria or fixed points of the dynamical system, which can be one or more, stable

or unstable. Therefore, *quasi-static subscales can only be considered when solving linear subscale equations. To solve the nonlinear subscale Eqs.* (26) *and* (27) *the subscales must be dynamic* in order to avoid possible lack of uniqueness in their calculations.

The space discretized formulation is now complete, but contrarily to what happens with linear quasi-static subscales it is not possible to obtain a closed-form expression for dynamic subscales and insert them into (15)–(17) to obtain a problem for the finite element unknowns only. Before discretizing in time we cannot go any further than saying that the problem consists in solving (15)–(17) together with (25)–(27). This final semidiscrete system of equations is highly nonlinear, even more when nonlinear subscales are considered. Some possibilities to linearize the (fully discrete) problem are described in Section 6.

4. Global conservation properties

The aim of this section is to obtain global conservation statements similar to those holding for the continuous problem (1)–(3), but for the semidiscrete one. To do that it is necessary to consider the finite element spaces without Dirichlet boundary conditions and an augmented problem that also contains the tractions at the Dirichlet boundaries as unknowns [20]. This permits to take constant test functions (see below) and to arrive to conservation statements in terms of the tractions and flows at the boundaries. We shall see that the following conservation statements hold *when equal interpolation is used for all variables*.

4.1. Mass conservation

Taking the test function $q_h = 1$ in the mass Eq. (15) global mass conservation follows immediately:

$$\int_\Omega \frac{\partial \rho^h}{\partial t} d\Omega = -\int_{\partial\Omega} \boldsymbol{n} \cdot \rho^h \boldsymbol{u}_h d\Gamma \tag{31}$$

4.2. Momentum conservation

Suppose that a canonical Cartesian coordinate system is used. Taking the tests function $\boldsymbol{v}_h = (1,0,0), (0,1,0)$ and $(0,0,1)$ in (the augmented problem corresponding to) the finite element momentum Eq. (16), we get

$$\int_\Omega \rho^h \frac{\partial}{\partial t}(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) d\Omega + \int_\Omega \rho^h (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}_h d\Omega + \int_\Omega \tilde{\boldsymbol{u}} \frac{\partial \rho^h}{\partial t} d\Omega = \int_\Omega \rho^h \boldsymbol{g} d\Omega + \int_{\partial\Omega} \boldsymbol{t}_n d\Gamma \tag{32}$$

When using equal interpolation spaces for the velocity components and pressure equations, we can take the test function equal to velocity components $q_h = u_{h,i}, i = 1, \ldots, d$, in the discrete mass Eq. (15). Therefore, we get the relation

$$\int_\Omega \rho^h (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}_h d\Omega = \int_\Omega \left(\frac{\partial \rho^h}{\partial t}\right) \boldsymbol{u}_h d\Omega + \int_{\partial\Omega} (\boldsymbol{n} \cdot \boldsymbol{u}_h) \rho^h \boldsymbol{u}_h d\Gamma$$

Replacing in (32) we arrive to

$$\int_\Omega \rho^h \frac{\partial}{\partial t}(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) d\Omega + \int_\Omega (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \frac{\partial \rho^h}{\partial t} d\Omega = \int_\Omega \rho^h \boldsymbol{g} d\Omega + \int_{\partial\Omega} (\boldsymbol{t}_n - (\boldsymbol{n} \cdot \boldsymbol{u}_h) \rho^h \boldsymbol{u}_h) d\Gamma$$

If momentum is defined as $\boldsymbol{p} = \rho^h \boldsymbol{u}_h + \rho^h \tilde{\boldsymbol{u}}$, with the contributions due to the finite element and the subscale components, we arrive to the following conservation statement

$$\int_\Omega \frac{\partial \boldsymbol{p}}{\partial t} d\Omega = \int_\Omega \rho^h \boldsymbol{g} d\Omega + \int_{\partial\Omega} (\boldsymbol{t}_n - (\boldsymbol{n} \cdot \boldsymbol{u}_h) \rho^h \boldsymbol{u}_h) d\Gamma \tag{33}$$

This equality indicates that the change of the total momentum over the system is equal to the total force over the system plus the traction and momentum fluxes over the boundary $\partial\Omega$. Note that Eq. (33) holds independently of the subscale approximation. If, as we have formally assumed, it is zero on the element boundaries, $\rho^h \boldsymbol{u}_h|_{\partial\Omega} = \boldsymbol{p}|_{\partial\Omega}$. Otherwise, boundary subscale terms would need to be added.

4.3. Energy conservation

Taking the test function $w_h = 1$ in (the augmented problem corresponding to) the finite element energy Eq. (17) we get

$$\int_\Omega \left(\rho^h c_p \frac{\partial}{\partial t}(T_h + \tilde{T}) + \rho^h c_p (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla T_h + \tilde{T} c_p \frac{\partial \rho^h}{\partial t}\right) d\Omega = \int_\Omega \left(Q + \alpha(T_h + \tilde{T}) \frac{dp^{th}}{dt}\right) d\Omega + \int_{\partial\Omega} q_n d\Gamma \tag{34}$$

When using equal interpolation spaces for the temperature and pressure equations ($W_h = Q_h$), we can take $q_h = T_h$ in the discrete mass Eq. (15) to obtain

$$\int_\Omega \rho^h (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}) \cdot \nabla T_h d\Omega = \int_\Omega \left(\frac{\partial \rho^h}{\partial t} \right) T_h d\Omega + \int_{\partial\Omega} \rho^h \boldsymbol{n} \cdot \boldsymbol{u}_h T_h d\Gamma$$

Replacing this equality in (34) we get the relation

$$\int_\Omega c_p \frac{\partial}{\partial t} (\rho^h (T_h + \tilde{T})) d\Omega = \int_\Omega \left(Q + \alpha (T_h + \tilde{T}) \frac{dp^{th}}{dt} \right) d\Omega + \int_{\partial\Omega} (q_n - \boldsymbol{n} \cdot \boldsymbol{u}_h \rho^h c_p T_h) d\Gamma \tag{35}$$

which is the discrete counterpart of energy conservation Eq. (3) integrated over domain Ω. So, Eq. (35) implies energy conservation.

In the case of *ideal gases* (taking $p^{th} = \rho^h R(T_h + \tilde{T})$ with $R = \frac{\gamma-1}{\gamma} c_p$ and $\alpha = 1/(T_h + \tilde{T})$) Eq. (35) can be written as

$$\frac{|\Omega|}{\gamma-1} \frac{dp^{th}}{dt} + \frac{\gamma p^{th}}{\gamma-1} \int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{u}_h d\Gamma = \int_\Omega Q d\Omega + \int_{\partial\Omega} q_n d\Gamma \tag{36}$$

which is the discrete version of Eq. (8), implying global energy conservation. For ideal gases the internal energy per unit mass is $e = c_v T = \frac{1}{\gamma-1} p^{th}/\rho$, where $c_v \equiv c_p/\gamma$. So, for the low Mach approximation internal energy per unit volume ρe is uniform and directly proportional to thermodynamic pressure p^{th}. According to that, we define at the discrete level, the discrete internal energy per unit volume as $\rho^h e^h = \frac{1}{\gamma-1} p^{th} = \rho^h c_v (T_h + \tilde{T})$. Replacing this definition in Eq. (36), we arrive to the first law of thermodynamics for open systems in terms of the internal energy

$$\int_\Omega \frac{\partial(\rho^h e_h)}{\partial t} d\Omega = \int_\Omega Q d\Omega + \int_{\partial\Omega} (q_n - \boldsymbol{n} \cdot \boldsymbol{u}_h (\rho^h e^h + p^{th})) d\Gamma \tag{37}$$

that indicates that the change of internal energy of the system is equal to the heat power added to the system plus the work done over the system ($W = -\int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{u}_h p^{th} d\Gamma = -p^{th} \int_\Omega \nabla \cdot \boldsymbol{u}_h d\Omega$) plus the boundary fluxes of heat and internal energy q_n and $\boldsymbol{n} \cdot \boldsymbol{u}_h \rho^h e^h$. It has been proved recently in [10] that for the Boussinesq approximation global conservation of energy is obtained when nonlinear and orthogonal subscales are used.

5. Time discretization

Any time integration scheme could now be applied to discretize in time the finite element Eqs. (15)–(17), together with Eqs. (25)–(27). As it is discussed in [12] the time integration for the subscales could be one order less accurate than for the finite element equations without affecting the accuracy of the scheme. To be specific we will consider backward differencing schemes, of second order for the finite element functions and of first order for subscale functions. The choice of a lower order accurate scheme for the subscales results in important memory savings.

Let δt be the time step size of a uniform partition of the time interval $[0, t_{end}]$, $0 = t^0 < t^1 < \cdots < t^N = t_{end}$. Functions approximated at time t^n will be identified with the superscript n. Temporal derivatives in Eqs. (15)–(17) and (25)–(27) as well as in (22)–(24), will be approximated by

$$\left. \frac{\partial \phi}{\partial t} \right|_{t^{n+1}} \approx \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\delta t} := \delta_t \phi^{n+1}$$

$$\left. \frac{\partial \varphi}{\partial t} \right|_{t^{n+1}} \approx \frac{\varphi^{n+1} - \varphi^n}{\delta t} := \tilde{\delta}_t \varphi^{n+1}$$

where operator $\delta_t \phi$ indicates discrete temporal derivatives over functions $\phi = \boldsymbol{u}_h$, $\phi = T_h$, $\phi = \rho^h$, $\phi = p^{th}$ and the operator $\tilde{\delta}_t \varphi$ indicates discrete temporal derivatives over the subscales $\varphi = \tilde{\boldsymbol{u}}$ and $\varphi = \tilde{T}$ whereas, as usual, all the rest of the terms in these equations are evaluated at t^{n+1} (eventually $\tilde{\delta}_t = \delta_t$ if the same integration scheme is wished to be used for both the finite element and subscale component of an unknown). After doing this, the subscale Eqs. (25)–(27) yield

$$\tilde{p}^{n+1} = \tau_c^{n+1} R_c^{n+1} \tag{38}$$

$$\tilde{\boldsymbol{u}}^{n+1} = \left(\frac{\rho^h}{\delta t} + \frac{1}{\tau_m^{n+1}} \right)^{-1} \left(\boldsymbol{R}_m^{n+1} + \frac{\rho^h \tilde{\boldsymbol{u}}^n}{\delta t} \right) \tag{39}$$

$$\tilde{T}^{n+1} = \left(\frac{\rho^h c_p}{\delta t} + \frac{1}{\tau_e^{n+1}} \right)^{-1} \left(R_e^{n+1} + \frac{\rho^h c_p \tilde{T}^n}{\delta t} \right) \tag{40}$$

From these expressions, we see that the residual of the momentum and energy equations are multiplied respectively by

$$\tau_{tm} = \left(\frac{\rho^h}{\delta t} + \frac{1}{\tau_m^{n+1}} \right)^{-1} \tag{41}$$

$$\tau_{te} = \left(\frac{\rho^h c_p}{\delta t} + \frac{1}{\tau_e^{n+1}}\right)^{-1} \tag{42}$$

These can be considered the effective stabilization parameters for the transient Low Mach equations. Expressions with asymptotic behavior similar to coefficients τ_{tm}, τ_{te} in terms of h, μ, $|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}|$, and δt can often be found in the literature (see e.g. [16]). It is important to note that if the stabilization parameter depends on δt and subscales are not considered time dependent, *the steady-state solution will depend on the time step size*. This does not happen if expressions (39) and (40) are used. It can be checked that, when steady state is reached the usual expressions employed for stationary problems are recovered, namely $\tilde{\boldsymbol{u}} = \tau_m \boldsymbol{R}_m$, and $\widetilde{T} = \tau_e R_e$.

After introducing the temporal discretization described above one faces a coupled nonlinear problem for $\boldsymbol{u}_h^{n+1}$, p_h^{n+1}, T_h^{n+1}, $\tilde{\boldsymbol{u}}^{n+1}$, $\tilde{p}^{n+1}$ and $\widetilde{T}^{n+1}$ which consists of the time discrete version of (15)–(18), (38)–(40) together with the definition of the residuals (22)–(24). The linearization of this system is described in the following section.

6. Linearization strategy

In order to solve the strongly coupled nonlinear problem given by the time discrete version of (15)–(18), (38)–(40) together with the definition of the residuals (22)–(24), we perform an iterative loop of index i, at each time step n. In this loop, given $\boldsymbol{u}_h^{n+1,i}$, $T_h^{n+1,i}$, $\tilde{\boldsymbol{u}}^{n+1,i}$ and $\widetilde{T}^{n+1,i}$, we first compute the finite element unknowns $\boldsymbol{u}_h^{n+1,i+1}$, $p_h^{n+1,i+1}$ and $T_h^{n+1,i+1}$ from the equations obtained when replacing the subscale expressions (38)–(40) into the time discretized finite element Eqs. (15)–(17). Likewise, temporal derivatives of velocity and temperature subscales are written in terms of known iterates of subscales and residuals using (26) and (27) and replaced into the time discretized finite element Eqs. (15)–(17). Finally, it is worth noting that the fourth term in (17) (the one involving the thermodynamic pressure) is treated differently: the temperature subscale is evaluated as $\widetilde{T}^{n+1,i}$ because in this way, when an ideal gas is considered, $\alpha^i\left(T_h^{n+1,i} + \widetilde{T}^{n+1,i}\right) = 1$, where

$$\alpha^i = -\frac{1}{\rho}\frac{\partial\rho}{\partial T}\bigg|_{T=T_h^{n+1,i}+\widetilde{T}^{n+1,i}}$$

The thermodynamic pressure itself is also evaluated at the end of each iteration, so this term can be considered as part of the right-hand-side.

In order to write the equations obtained in this way, we also introduce the notation

$$\boldsymbol{a}^i = \boldsymbol{u}_h^{n+1,i} + \tilde{\boldsymbol{u}}^{n+1,i}$$

$$\rho^i = \rho\left(T_h^{n+1,i} + \widetilde{T}^{n+1,i}, p^{th,n+1,i}\right)$$

and we use these expressions to compute $\tau_c^{n+1,i}, \tau_m^{n+1,i}, \tau_e^{n+1,i}$ using (28)–(30) and $\tau_{tm}^{n+1,i}, \tau_{te}^{n+1,i}$ using (41) and (42). Let us also define perturbations of test functions as

$$P_c^i(q_h) = \tau_{tm}^{n+1,i}\rho^i\nabla q_h$$

$$\boldsymbol{P}_m^i(\boldsymbol{v}_h) = \tau_{tm}^{n+1,i}\left[-\frac{\rho^i}{\delta t}\boldsymbol{v}_h - \delta_t\rho^i\boldsymbol{v}_h + \rho^i\boldsymbol{a}^i\cdot\nabla\boldsymbol{v}_h + \nabla^h\cdot(2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}_h))\right]$$

$$\boldsymbol{P}_{tm}^i(\boldsymbol{v}_h) = \tau_{tm}^{n+1,i}\left[\frac{1}{\tau_m^{n+1,i}}\boldsymbol{v}_h - \delta_t\rho^i\boldsymbol{v}_h + \rho^i\boldsymbol{a}^i\cdot\nabla\boldsymbol{v}_h + \nabla^h\cdot(2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}_h))\right]$$

$$P_e^i(w_h) = \tau_{te}^{n+1,i}\left[-\frac{\rho^i c_p}{\delta t}w_h - c_p\delta_t\rho^i w_h + \rho^i c_p\boldsymbol{a}^i\cdot\nabla w_h + \nabla^h\cdot(k\nabla w_h)\right]$$

$$P_{te}^i(w_h) = \tau_{te}^{n+1,i}\left[\frac{1}{\tau_e^{n+1,i}}w_h - c_p\delta_t\rho^i w_h + \rho^i c_p\boldsymbol{a}^i\cdot\nabla w_h + \nabla^h\cdot(k\nabla w_h)\right]$$

Using these definitions we can write mass conservation as

$$-\left(\rho^i\boldsymbol{u}_h^{n+1,i+1},\nabla q_h\right) + \left(\rho^i\boldsymbol{n}\cdot\boldsymbol{u}_h^{n+1,i+1},q_h\right)_{\partial\Omega}$$

$$+ \left(\rho^i\delta_t\boldsymbol{u}_h^{n+1,i+1} + \rho^i\boldsymbol{a}^i\cdot\nabla\boldsymbol{u}_h^{n+1,i+1} + \nabla p_h^{n+1,i+1} - \nabla^h\cdot\left(2\mu\boldsymbol{\varepsilon}'\left(\boldsymbol{u}_h^{n+1,i+1}\right)\right), P_c^i(q_h)\right) = -(\delta_t\rho^i,q_h) + \left(\rho^i\boldsymbol{g} + \frac{\rho^i\tilde{\boldsymbol{u}}^n}{\delta t}, P_c^i(q_h)\right) \tag{43}$$

momentum conservation as

$$\left(\rho^i\delta_t\boldsymbol{u}_h^{n+1,i+1} + \rho^i\boldsymbol{a}^i\cdot\nabla\boldsymbol{u}_h^{n+1,i+1}, \boldsymbol{v}_h + \boldsymbol{P}_m^i(\boldsymbol{v}_h)\right) - \left(p_h^{n+1,i+1},\nabla\cdot\boldsymbol{v}_h\right) + \left(2\mu\boldsymbol{\varepsilon}'\left(\boldsymbol{u}_h^{n+1,i+1}\right),\nabla^s\boldsymbol{v}_h\right)$$

$$+ \left(\nabla p_h^{n+1,i+1} - \nabla^h\cdot\left(2\mu\boldsymbol{\varepsilon}'\left(\boldsymbol{u}_h^{n+1,i+1}\right)\right),\boldsymbol{P}_m^i(\boldsymbol{v}_h)\right) + \left(\rho^i\nabla\cdot\boldsymbol{u}_h^{n+1,i+1} - \rho^i\alpha^i\boldsymbol{a}^i\cdot\nabla T_h^{n+1,i},\tau_c^{n+1,i}\nabla\cdot\boldsymbol{v}_h\right)$$

$$= (\boldsymbol{t}_n,\boldsymbol{v}_h)_{\Gamma_N^{\boldsymbol{u}}} - \left(\delta_t\rho^i,\tau_c^{n+1,i}\nabla\cdot\boldsymbol{v}_h\right) + \left(\rho^i\boldsymbol{g},\boldsymbol{v}_h + \boldsymbol{P}_m^i(\boldsymbol{v}_h)\right) + \left(\frac{\rho^i}{\delta t}\tilde{\boldsymbol{u}}^n,\boldsymbol{P}_{tm}^i(\boldsymbol{v}_h)\right) \tag{44}$$

and energy conservation as

$$\left(\rho^i c_p \delta_t T_h^{n+1,i+1} + \rho^i c_p \boldsymbol{a}^i \cdot \nabla T_h^{n+1,i+1}, w_h + P_e^i(w_h)\right) - \left(\nabla^h \cdot \left(k \nabla T_h^{n+1,i+1}\right), P_e^i(w_h)\right) + \left(k \nabla T_h^{n+1,i+1}, \nabla w_h\right)$$

$$= (q_n, w_h)_{\Gamma_N^T} + \left(Q + \alpha^i \left(T_h^{n+1,i} + \widetilde{T}^{n+1,i}\right) \delta_t p^{th,n+1,i}, w_h + P_e^i(w_h)\right) + \left(\frac{\rho^i c_p}{\delta t} \widetilde{T}^n, P_{te}^i(w_h)\right) \qquad (45)$$

After solving these equations, the finite element unknowns $\boldsymbol{u}_h^{n+1,i+1}$, $p_h^{n+1,i+1}$ and $T_h^{n+1,i+1}$ are obtained.

Now we proceed to solve (38)–(40) together with the definition of the residuals (22)–(24) to obtain the subscales $\tilde{\boldsymbol{u}}^{n+1,i+1}$ and $\widetilde{T}^{n+1,i+1}$. Note that Eqs. (38)–(40) are nonlinear in $\tilde{\boldsymbol{u}}^{n+1,i+1}$ (through the stabilization parameters) and in $\widetilde{T}^{n+1,i+1}$ (through the density). These dependencies are made explicit by rewriting (39) and (40) as

$$\rho^h(\widetilde{T}) \tilde{\delta}_t \tilde{\boldsymbol{u}} + \tau_m^{-1}(\tilde{\boldsymbol{u}}, \widetilde{T}) \tilde{\boldsymbol{u}} + \rho^h(\widetilde{T}) \tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{u}_h = \boldsymbol{R}_{ml}(\widetilde{T}) \qquad (46)$$

$$\rho^h(\widetilde{T}) c_p \tilde{\delta}_t \widetilde{T} + \tau_e^{-1}(\tilde{\boldsymbol{u}}, \widetilde{T}) \widetilde{T} + \rho^h(\widetilde{T}) c_p \tilde{\boldsymbol{u}} \cdot \nabla T_h = R_{el}(\widetilde{T}) \qquad (47)$$

where all variables are evaluated at time step $n + 1$ iteration $i + 1$ and the residuals $\boldsymbol{R}_{ml}$ and R_{el} (not depending on $\tilde{\boldsymbol{u}}$) are given by

$$\boldsymbol{R}_{ml}(\widetilde{T}) = \rho^h \boldsymbol{g} - \rho^h \delta_t \boldsymbol{u}_h - \rho^h \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h + \nabla \cdot (2\mu \boldsymbol{\varepsilon}'(\boldsymbol{u}_h)) - \nabla p_h$$

$$R_{el}(\widetilde{T}) = Q + \alpha(T_h + \widetilde{T}) \delta_t p^{th} - \rho^h c_p \delta_t T_h - \rho^h c_p \boldsymbol{u}_h \cdot \nabla T_h + \nabla \cdot (k \nabla T_h)$$

The difference between $\boldsymbol{R}_{ml}$ and $\boldsymbol{R}_m$ (resp. R_e and R_{el}) is precisely the last term of the left-hand-side in (46) (resp. (47)). To obtain $\tilde{\boldsymbol{u}}^{n+1,i+1}$ and $\widetilde{T}^{n+1,i+1}$ at each integration point of each element we need to solve Eqs. (46) and (47) iteratively. Their linearization will be discussed in the following, considering either fixed point (Picard) or Newton–Raphson's methods, and also either a monolithic or a segregated calculation of the velocity and temperature subscales. We will present the different methods and then their performance will be evaluated in a simple test case in Section 7.

The final time evolution and iterative scheme is presented in Algorithm 1.

Algorithm 1. Time evolution and iterative strategy

1: Set $\boldsymbol{u}_h^0$ and $\tilde{\boldsymbol{u}}^0$.
2: Set $n = 0$.
3: **for** $n < N$ **do**
4:    Set $\boldsymbol{u}_h^{n+1,0} = \boldsymbol{u}_h^n$ and $T_h^{n+1,0} = T_h^n$
5:    Set $\tilde{\boldsymbol{u}}^{n+1,0} = \tilde{\boldsymbol{u}}^n$ and $\widetilde{T}^{n+1,0} = \widetilde{T}^n$
6:    Set $i = 0$.
7:    **while** not converged **do**
8:        Solve (43)–(45) to obtain $\boldsymbol{u}_h^{n+1,i+1}$, $T_h^{n+1,i+1}$ and $p_h^{n+1,i+1}$
9:        **for** each integration point **do**
10:           Solve iteratively (46) and (47) to obtain $\tilde{\boldsymbol{u}}^{n+1,i+1}$ and $\widetilde{T}^{n+1,i+1}$
11:       **end for**
12:       Solve (5) or (7) to obtain $p^{th,n+1,i+1}$
13:       $i = i + 1$.
14:    **end while**
15: **end for**

6.1. Picard's linearization

Let us denote with a superscript k or $k + 1$ the iteration counter for the subscales (remember that all unknowns need to be computed at time step $n + 1$ and global iteration $i + 1$). The finite element unknowns are assumed to be given as we need to solve problem (46) and (47). If the subscales are known at iteration k, the simplest way to approximate them at iteration $k + 1$ is by solving

$$\rho^h \tilde{\delta}_t \tilde{\boldsymbol{u}}^{k+1} + \tau_m^{-1}(\tilde{\boldsymbol{u}}^k, \widetilde{T}^k) \tilde{\boldsymbol{u}}^{k+1} + \rho^h \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla \boldsymbol{u}_h = \boldsymbol{R}_{ml}(\widetilde{T}^k) \qquad (48)$$

$$\rho^h c_p \tilde{\delta}_t \widetilde{T}^{k+1} + \tau_e^{-1}(\tilde{\boldsymbol{u}}^k, \widetilde{T}^k) \widetilde{T}^{k+1} + \rho^h c_p \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla T_h = R_{el}(\widetilde{T}^k) \qquad (49)$$

where $\rho^h = \rho\left(p^{th}, T_h + \widetilde{T}^k\right)$ is computed from the unknowns at the last iteration. Note that solving (48) implies the solution of a $d \times d$ linear system at each integration point. This is due to the last term in the left-hand-side of (48) which was evaluated in iteration k in [9,12,19] (doing that permits to obtain a closed expression for $\tilde{\boldsymbol{u}}^{k+1}$, but the iterative scheme may fail to converge).

6.2. Picard's linearization, Gauss–Seidel-type $\tilde{\boldsymbol{u}} - \widetilde{T}$ calculation

It is observed that Eq. (48) does not depend on $\widetilde{T}^{k+1}$, and therefore this equation may be used to compute $\tilde{\boldsymbol{u}}^{k+1}$. Once it is known, instead of using $\tilde{\boldsymbol{u}}^k$ in Eq. (49) we could think of using this updated value, $\tilde{\boldsymbol{u}}^{k+1}$, that is to say, using a Gauss-Seidel-type coupling of (48) and (49). The resulting problem is

$$\rho^h \tilde{\delta}_t \tilde{\boldsymbol{u}}^{k+1} + \tau_m^{-1}(\tilde{\boldsymbol{u}}^k, \widetilde{T}^k)\tilde{\boldsymbol{u}}^{k+1} + \rho^h \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla \boldsymbol{u}_h = \boldsymbol{R}_{ml}(\widetilde{T}^k)$$

$$\rho^h c_p \tilde{\delta}_t \widetilde{T}^{k+1} + \tau_e^{-1}(\tilde{\boldsymbol{u}}^{k+1}, \widetilde{T}^k)\widetilde{T}^{k+1} + \rho^h c_p \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla T_h = R_{el}(\widetilde{T}^k)$$

This scheme has been proved to converge faster than (48) and (49). Note that the only difference between both methods is the way to evaluate τ_e.

6.3. Newton–Raphson's linearization, monolithic $\tilde{\boldsymbol{u}} - \widetilde{T}$ calculation

An alternative to fixed point schemes is to use the Newton–Raphson method. When applied to (46) and (47), the problem to be solved at each iteration is

$$\rho^h \tilde{\delta}_t \tilde{\boldsymbol{u}}^{k+1} + \rho^h \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla \boldsymbol{u}_h + \tau_m^{-1}(\tilde{\boldsymbol{u}}^k, \widetilde{T}^k)\tilde{\boldsymbol{u}}^{k+1} + \frac{c_2 \rho^h}{h}\left[\frac{(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k)}{|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k|} \cdot (\tilde{\boldsymbol{u}}^{k+1} - \tilde{\boldsymbol{u}}^k)\right]\tilde{\boldsymbol{u}}^k$$

$$- \left(\frac{c_2}{h}|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k|\tilde{\boldsymbol{u}}^k + (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k) \cdot \nabla \boldsymbol{u}_h - \boldsymbol{g} + \delta_t(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k)\right)\frac{\rho^h(\widetilde{T}^{k+1} - \widetilde{T}^k)}{T_h + \widetilde{T}^k} = \boldsymbol{R}_{ml}(\widetilde{T}^k)$$

$$\rho^h c_p \tilde{\delta}_t \widetilde{T}^{k+1} + \tau_e^{-1}(\tilde{\boldsymbol{u}}^k, \widetilde{T}^k)\widetilde{T}^{k+1} + \rho^h c_p \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla T_h + \frac{c_2 \rho^h c_p}{h}\left[\frac{(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k)}{|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k|} \cdot (\tilde{\boldsymbol{u}}^{k+1} - \tilde{\boldsymbol{u}}^k)\right]\widetilde{T}^k$$

$$- \left(\frac{c_2}{h}|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k|\widetilde{T}^k + (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^{k+1}) \cdot \nabla T_h + \delta_t(T_h + \widetilde{T}^k)\right)\frac{\rho^h c_p(\widetilde{T}^{k+1} - \widetilde{T}^k)}{T_h + \widetilde{T}^k} = R_{el}(\widetilde{T}^k)$$

In our numerical tests we have observed that this scheme is the most robust, always achieving convergence but needing sometimes more iterations than the other schemes. Note that it implies to solve for $\tilde{\boldsymbol{u}}^{k+1}$ and $\widetilde{T}^{k+1}$ in a monolithic (coupled) way.

6.4. Newton–Raphson's linearization, segregated $\tilde{\boldsymbol{u}} - \widetilde{T}$ calculation

An alternative to the monolithic scheme presented above would be to use $\widetilde{T}^k$ in the equation for $\tilde{\boldsymbol{u}}^{k+1}$, so as to allow for a segregated calculation of $\tilde{\boldsymbol{u}}^{k+1}$ and $\widetilde{T}^{k+1}$. The result is

$$\rho^h \tilde{\delta}_t \tilde{\boldsymbol{u}}^{k+1} + \rho^h \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla \boldsymbol{u}_h + \tau_m^{-1}(\tilde{\boldsymbol{u}}^k, \widetilde{T}^k)\tilde{\boldsymbol{u}}^{k+1} + \frac{c_2 \rho^h}{h}\left[\frac{(\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k)}{|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^k|} \cdot (\tilde{\boldsymbol{u}}^{k+1} - \tilde{\boldsymbol{u}}^k)\right]\tilde{\boldsymbol{u}}^k = \boldsymbol{R}_{ml}(\widetilde{T}^k)$$

$$\rho^h c_p \tilde{\delta}_t \widetilde{T}^{k+1} + \rho^h c_p \tilde{\boldsymbol{u}}^{k+1} \cdot \nabla T_h + \tau_e^{-1}(\tilde{\boldsymbol{u}}^{k+1}, \widetilde{T}^k)\widetilde{T}^{k+1}$$

$$- \left(\frac{c_2}{h}|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^{k+1}|\widetilde{T}^k + (\boldsymbol{u}_h + \tilde{\boldsymbol{u}}^{k+1}) \cdot \nabla T_h + \delta_t(T_h + \widetilde{T}^k)\right)\frac{\rho^h c_p\left(\widetilde{T}^{k+1} - \widetilde{T}^k\right)}{T_h + \widetilde{T}^k} = R_{el}(\widetilde{T}^k)$$

Note that $\tilde{\boldsymbol{u}}^{k+1}$ is used in the evaluation of τ_e, using the same Gauss–Seidel-type update as for Picard's method.

7. Numerical examples

In this section we present three examples. The first example, extensively studied in the literature, is a stationary flow in a differentially heated cavity and permits us to perform a detailed study of the behavior of the nonlinear subscale models and their iterative solution. The second one is a transient bidimensional closed flow with increasing thermodynamic pressure and permits us to test the temporal accuracy of the proposed scheme. The third one is a transient three dimensional fire compartment room in which, apart from testing the accuracy of the method we also present some preliminary conclusions on its computational efficiency.

Whereas in the first example different subscale models are compared, in the second and third we compare the formulation presented in this paper against the algebraic subgrid scale (ASGS) method, as presented in [32] , which is reduced to the Galerkin least squares method (GLS) [18] when linear elements are used. Bilinear (2D) and trilinear (3D) interpolated Q_1 elements are used in all the numerical examples for velocity, pressure and temperature. In the first two examples the algebraic system corresponding to the finite element problem (line 6 in Algorithm 1) is solved using a direct method whereas in the

last one it is solved using an iterative one (GMRES) whose convergence criteria is that any residual (for mass, momentum or energy equations) cannot be bigger than 10^{-6} of the initial one. The convergence criteria for the nonlinear iterations (loop starting at line 7 in Algorithm 1) was set as $\left|\phi_h^{n+1,i+1} - \phi_h^{n+1,i}\right| < \epsilon_1\left|\phi_h^{n+1,i+1}\right|$ for any finite element unknown ϕ_h, where the tolerance ϵ_1 was set to 10^{-9} in the first example and to $5\cdot10^{-4}$ in the second and third ones. The subscales were computed at each integration point until $|\tilde{\phi}^{n+1,i+1,k+1} - \tilde{\phi}^{n+1,i+1,k}| < \epsilon_2|\tilde{\phi}^{n+1,i+1,k+1}|$, for any subscale $\tilde{\phi}$. The tolerance was set to $\epsilon_2 = 10^{-3}\left|\phi_h^{n+1,i+1} - \phi_h^{n+1,i}\right|/\left|\phi_h^{n+1,i+1}\right|$ in all the examples.

In all examples we consider an ideal gas with constant values of $R = 287.0$ J kg^{-1} K^{-1} and $c_p = 1004.5$ J kg^{-1} K^{-1}.

7.1. Natural convection in a cavity

7.1.1. Problem description and numerical results

The present flow example has been proposed in [35] as a benchmark problem for natural convection flows with large temperature gradients. This example was also considered in, e.g. [16,18,32]. The problem domain is $\Omega = [0,L] \times [0,L]$ with $L = 1$ m. Adiabatic boundary conditions are prescribed for upper and lower walls ($q_n = k\mathbf{n}\cdot\nabla T_h = 0$). The left wall is maintained at a fixed temperature T_H and the right wall at temperature T_C. The initial thermodynamic pressure and temperatures are $p_0^{th} = 101325$ Pa, $T_0 = 600$ K, yielding an initial uniform density of $\rho_0 = 0.58841$ kg/m^3. In contrast to the Boussinesq approximation, in the low Mach approximation the stationary solution depends on the initial thermodynamic pressure p_0^{th}. The dimensionless Prandtl and Rayleigh numbers are fixed to $\text{Pr} = \frac{c_p\mu}{k} = 0.71, \text{Ra} = 2\frac{|\mathbf{g}|\rho_0^2}{\mu^2}\varepsilon = 10^6$, where $\varepsilon = \frac{T_H-T_C}{T_H+T_C} = 0.6$. The viscosity is $\mu = 10^{-3}$ kg/(m s). Boundary left and right wall temperatures are $T_H = 960$ K and $T_C = 240$ K, satisfying the relation $(T_H + T_C)/2 = T_0$. Zero Dirichlet boundary conditions for the velocity are assumed on all boundaries.

Numerical results were obtained using meshes from 20×20 to 240×240 uniform elements. The reference solution was obtained using a grid of 720×720 uniform elements using both the ASGS and the DSS methods. The relative difference between these two solutions is 2.8×10^{-4} in the discrete $L^2(\Omega)$-norm of the velocities and 4.4×10^{-5} in the discrete $L^2(\Omega)$-norm of the temperatures. The discrete $L^2(\Omega)$-norm (denoted by $|\cdot|_h$) has been computed for any variable ϕ as

$$|\phi|_h^2 = \frac{1}{N_n} \sum_a [\phi(\mathbf{x}^a)]^2 \tag{50}$$

where the summation extends to all N_n mesh nodes a. As shown below, relative differences between results obtained with any other mesh and method with respect to the one obtained using the ASGS method in the 720×720 uniform mesh are, at least, one order of magnitude bigger. This proves that the comparisons against either the solution obtained in the 720×720 uniform mesh using the ASGS or the DSS method will give rise to the same conclusions. We choose the ASGS method as reference because it is a standard method extensively tested and compared against direct numerical simulations in the literature [3,16].

The obtained velocities and temperatures are depicted in Fig. 1.

The compared stabilization methods are (note that the first three methods only differ in the subscale model):

- The method in the present paper containing full nonlinear subscales, labeled as "DSS" in Figs. 2 and 3.
- The method presented in this paper, but neglecting the temperature subscale in the state equation, that is ρ^h is computed as $\rho^h = \frac{p^{th}}{RT_h}$, labeled as "SemiSGS" in Figs. 2 and 3.
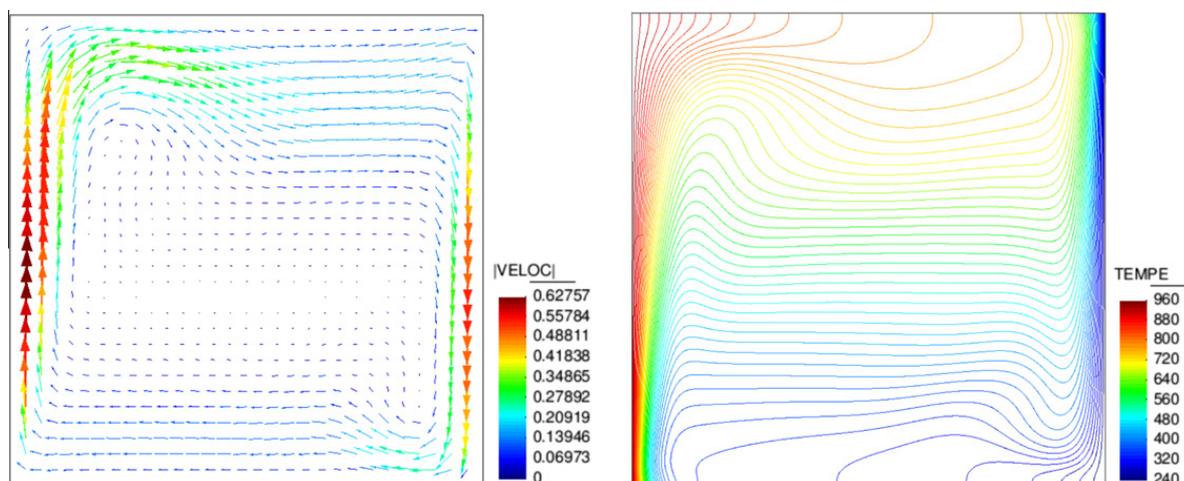

Fig. 1. Velocity vectors and temperature distribution.

- The method presented in this paper, but using a linear approximation to the subscale Eqs. (19)–(21), that is, taking τ_m, τ_c, τ_e and $\boldsymbol{R}_m$, R_c, R_e independent of the velocity and temperature subscales $\tilde{\boldsymbol{u}}$, $\widetilde{T}$. In all cases $|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}|$ is replaced by $|\boldsymbol{u}_h|$ as it is usually done, for example in [16] or [3]. As in the previous case the temperature subscale is neglected in the state equation. This method is labeled as "LinSGS" in Figs. 2 and 3.
- The ASGS method, a linear stabilization method as described in [32].
- The reference solution, labeled as "Ref" in Fig. 3.

The discrete $L^2(\Omega)$-norm of the error (with respect to the reference solution) as a function of the mesh size h is shown in Fig. 2 for the different stabilization methods. Second order convergence of the error for the smallest values of h is obtained for all the methods. However, the highest accuracy is obtained when subscales are kept in all the nonlinear terms, both in the finite element equations and in the subscale equations. As it can be seen in Fig. 2 the error obtained using the full nonlinear method is smaller than the one obtained using the ASGS method over all the mesh size range. When using the full nonlinear
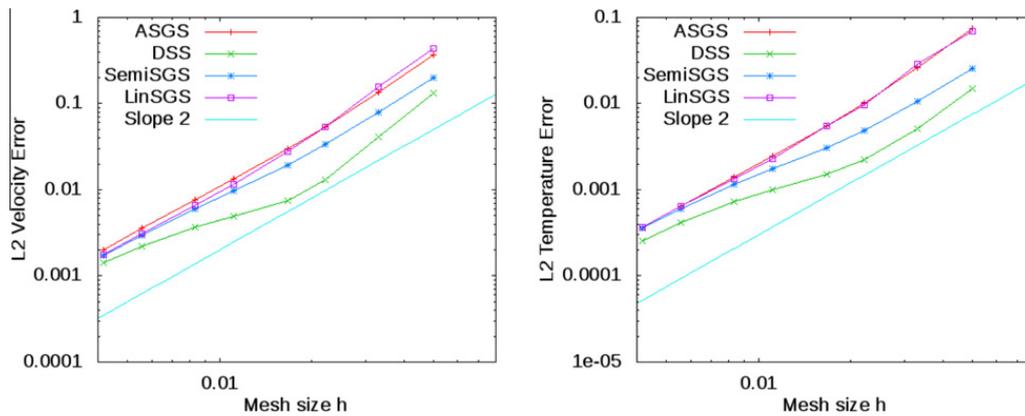
Fig. 2. Discrete $L^2(\Omega)$ error of the solutions obtained using different methods against mesh size h.
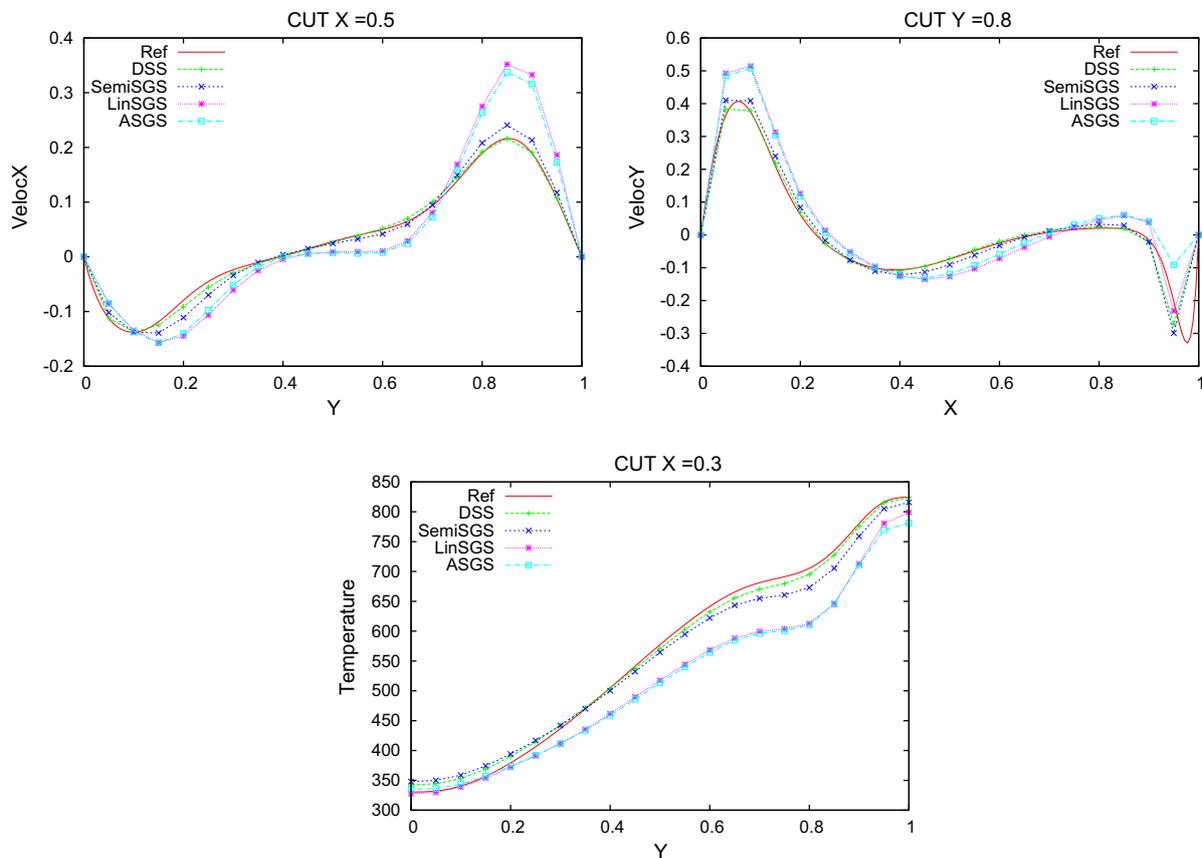

Fig. 3. Solutions obtained on a 20×20 uniform grid using different stabilization methods against the reference solution.

method a convergence rate of order greater than 2 is observed on the coarsest meshes. As will be justified in the next Section 7.1.2 the full nonlinear method will converge to the ASGS method for h small enough and this implies the there must be a region in which this slope decreases below 2. When the temperature subscale is neglected in the state equation the solution is less accurate, the gain in accuracy respect to the ASGS method is lower, becoming negligible over the finer grids. Finally, when a linear approximation to the subscale Eqs. (25)–(27) is considered, even less accurate solutions are obtained, the gain in accuracy respect to the ASGS method being negligible over all the mesh size range. A possible explanation for this phenomenon arises from the analysis of the subgrid velocity and temperature fields given in the following Section 7.1.2.

The higher accuracy of the fully nonlinear approximation for the subscales is further illustrated in Fig. 3 where the variation of the velocity and temperature solutions for different stabilization methods are depicted along three different lines cutting the domain.

Another quantity of interest is the heat transfer from the hot to the cold wall, represented by the Nusselt number, defined as

$$\mathrm{Nu}(\boldsymbol{x}) = \frac{L}{T_H - T_C} \boldsymbol{n} \cdot \nabla T(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega$$

The discrete $L^2(\partial\Omega)$-norm error of the Nusselt distribution against mesh size h is shown in Fig. 4 for the ASGS and DSS methods. When using finer grids both methods give similar errors. For the coarsest grids the DSS method gives slightly more accurate distributions, although the difference is not as significant as for the errors in the interior of the computational domain. In this case, the discrete $L^2(\partial\Omega)$-norm error is calculated as in (50), taking

$$\mathrm{Error} = \frac{\sum_a (\mathrm{Nu}_h(\boldsymbol{x}^a) - \mathrm{Nu}(\boldsymbol{x}^a))^2}{\sum_a (\mathrm{Nu}(\boldsymbol{x}^a))^2}$$

where now a refers to boundary nodes, $\mathrm{Nu}_h(\boldsymbol{x}^a)$ is obtained variationally from T_h (as explained in Section 4 and [20]) and at node $\boldsymbol{x}^a$, and $\mathrm{Nu}(\boldsymbol{x}^a)$ is the variational Nusselt number obtained from the reference solution at this node. Once again, this reference solution was obtained using the ASGS method over a grid of 720×720 elements. The convergence of the error against mesh size h is between linear and quadratic.

Even though the quality of the thermal fluxes is similar using the ASGS and the DSS methods, let us remark that due to the property of global conservation of energy (37) when using the DSS method, the average Nusselt number over all the boundary $\partial\Omega$ is identically zero. Instead it differs from zero when using ASGS method.

Reference solutions for the average Nusselt numbers over hot and cold walls can be found in [35]. Those differ in less than 0.02% from the Nusselt obtained in our reference solution. Let us mention that the ASGS yields better results on the hot wall, whereas the Nusselt number is better approximated using the DSS method on the cold wall. These observations are however inconclusive since, as we have seen, the global discrete $L^2(\partial\Omega)$ error is very similar in both methods.

In Fig. 4 the thermodynamic pressure convergence is depicted against the mesh size h using the ASGS and the DSS methods. The reference value for p^{th} is taken from [35]. As for the Nusselt number, convergence is found to be between linear and quadratic for smaller h. When using coarser grids the DSS method yields more accurate values for p^{th} than the ASGS method. However, convergence is not monotone for coarser grids.

7.1.2. Analysis of subscale models

Velocity and temperature subscales are depicted in Fig. 5. As expected, higher values for the subscales are obtained on the boundary layer, where velocity and temperature gradients are larger. In fact, the temperature subscale can be significant when compared to the finite element temperature and the velocity subscale can be even higher than the finite element
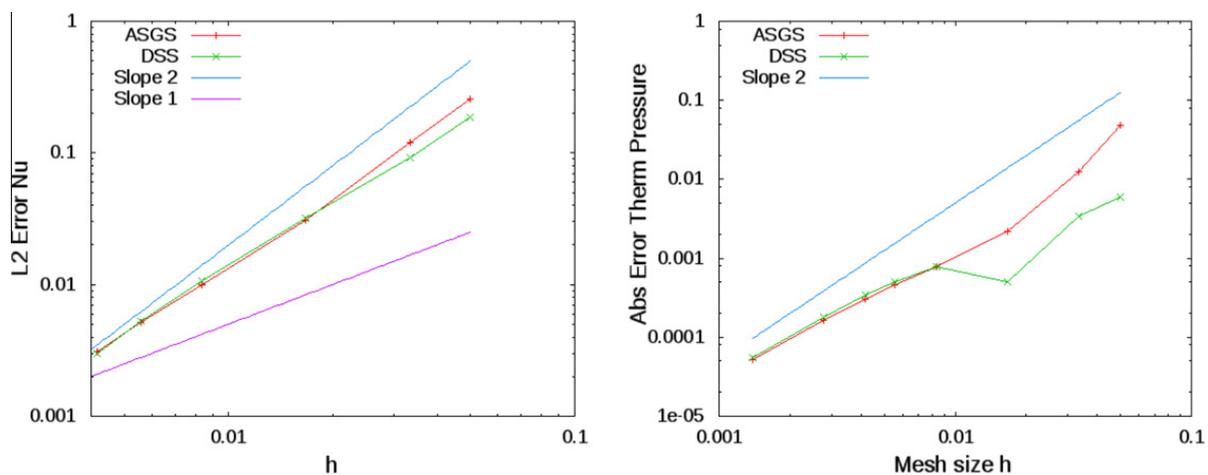

Fig. 4. $L^2(\partial\Omega)$-norm error of the Nusselt numbers distributions (left) and absolute error for the thermodynamic pressure (right) against mesh size h.

velocity, as can be seen comparing Fig. 5 with Fig. 1. This is actually the case in boundary layers, as can be seen in Fig. 6, where the logarithmic ratio between subscale and finite element velocities is shown for two different meshes. This shows the importance of keeping the scale splitting everywhere in nonlinear terms. A common argument to develop linear subscale models [3,16], is to assume that subscales are small compared to finite element components (i.e. $|\tilde{\boldsymbol{u}}| \ll |\boldsymbol{u}_h|$) thus neglecting terms which are quadratic in the subscales (although nonlinear terms are kept in the finite element equations in [3,16]). It is tempting to model subscales as linear and quasi-static because in this case there is no need to store them, and thus there would be no memory increase due to the subscale evaluation. However, more accurate solutions are obtained using the non-linear model, as shown above.

Similar comments can be made regarding the temperature coupling. If we consider ρ^h not depending on $\widetilde{T}$ and τ_m, τ_e not depending on $\tilde{\boldsymbol{u}}$ (i.e. replacing $|\boldsymbol{u}_h + \tilde{\boldsymbol{u}}|$ by $|\boldsymbol{u}_h|$ in (29) and (30)) the subscales are modeled by a linear system of equations, although $\tilde{\boldsymbol{u}}$ and $\widetilde{T}$ are still taken into account in the finite element residuals (22)–(24). This method does not require additional memory cost to store the subscales. However we did not find this method better than the method labeled "LinSGS" above and considered in [16,3] in terms of accuracy, at least in the differentially heated cavity problem presented here. Further, the assumption that the density does not depend on $\widetilde{T}$ leads to a nonlinear system of equations where the velocity subscale $\tilde{\boldsymbol{u}}$ is decoupled from the temperature subscale $\widetilde{T}$. This nonlinear system has one less unknown, and the temperature subscale can be obtained later from the linear Eq. (49). However, it has been observed in the present example that the effect of considering the state equation depending on the temperature subscale improves the accuracy of the method as shown in Fig. 2.

We proceed now to present heuristic arguments showing why the nonlinear subscale model can be more accurate than a linear one. As shown in Fig. 6 one region in which the subscale velocity is bigger than the finite element velocity is the (left) boundary layer. As shown in Figs. 1 and 5, the finite element and subgrid velocities are parallel there and therefore (assuming that the finite element velocity only depends on the coordinate normal to the wall), from Eq. (48), we have $\tilde{u}_1 = 0$ and $\tilde{u}_2 = \tau_m R_{ml,2}$. Then, the ($y$ component of the) finite element velocity, the subscale velocity and the residual have the same sign, which we assume positive. Omitting subscript 2 and defining the linear parameter
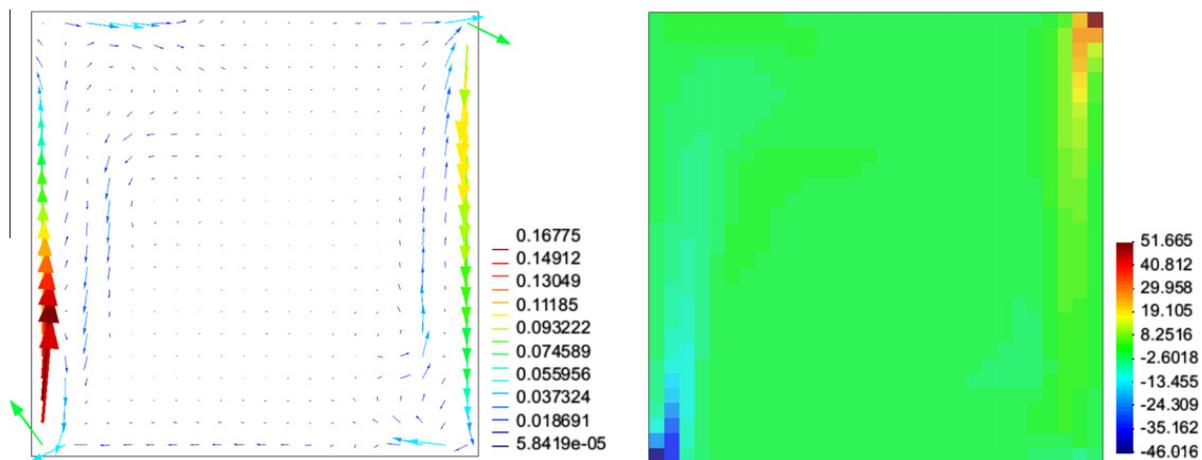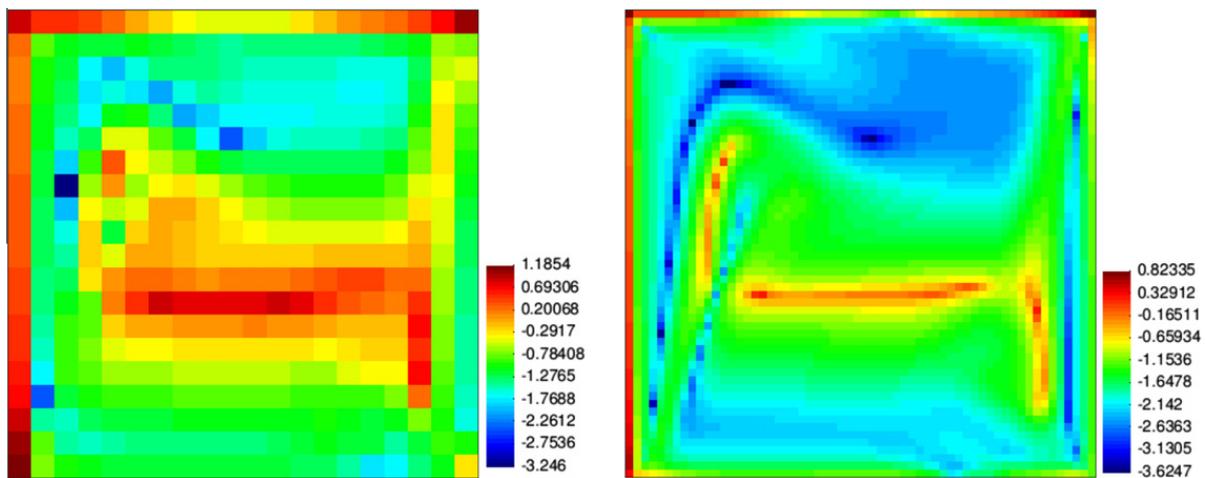

Fig. 5. Velocity and temperature subscales.

Fig. 6. Logarithmic relation between velocity subscale and finite element velocity norms, $\log_{10}(|\tilde{\boldsymbol{u}}|/|\boldsymbol{u}_h|)$, over 20×20 (left) and 60×60 (right) meshes.

$$\tau = \left(c_1 \frac{\mu}{h^2} + c_2 \frac{\rho^h u_h}{h} \right)^{-1}$$

this nonlinear equation can be written as

$$\tau^{-1} \tilde{u} + \frac{c_2 \rho \tilde{u}^2}{h} - R_{ml} = 0$$

whose solution is

$$\tilde{u} = \frac{h}{2c_2 \rho \tau} \left[-1 + \left(1 + \frac{4c_2 \rho \tau^2 R_{ml}}{h} \right)^{1/2} \right]$$

that behaves as

$$\tilde{u} \approx \begin{cases} (c_2 \rho)^{-1/2} (h R_{ml})^{1/2} & \text{when } 4c_2 \rho \tau^2 R_{ml}/h \gg 1 \\ \tau R_{ml} & \text{when } 4c_2 \rho \tau^2 R_{ml}/h \ll 1 \end{cases}$$

The parameter $\frac{4\rho \tau^2 R_{ml}}{h}$ tends to zero when the mesh is refined (expectedly as h^3 for linear elements) and that may explain why the LinSGS and the DSS methods give similar results when the mesh is fine enough. Whereas the LinSGS method approximates $\tilde{u} = \tau R_{ml}$ regardless of the parameter $\frac{4\rho \tau^2 R_{ml}}{h}$, the fully nonlinear DSS method gives a different solution when $\frac{4\rho \tau^2 R_{ml}}{h} \gg 1$ (coarse meshes) and the ratio between these solutions is, precisely,

$$\frac{\tilde{u}_{\text{LinSGS}}}{\tilde{u}_{\text{DSS}}} = \frac{\tau R_{ml}}{(c_2 \rho)^{-1/2} (h R_{ml})^{1/2}} = \frac{1}{2} \left(\frac{4c_2 \rho \tau^2 R_{ml}}{h} \right)^{1/2} \tag{51}$$

We have verified that the maximum value of this parameter (it varies over the domain and the maximum occurs in the boundary layer) is around 16 for the 20×20 mesh and is around 0.02 for the 240×240 mesh. We therefore expect the subscale obtained using the LinSGS method to be around twice times the subscale obtained using the DSS method in the 20×20 mesh but nearly the same in the 240×240 mesh. The norm of the velocity subscales obtained using LinSGS and DSS methods for both meshes shown in Fig. 7 confirms these expectations and helps to understand the results shown in Fig. 2 (smaller error using DSS for coarse meshes and the same result for finer meshes).

7.1.3. Testing the linearization methods

Let us describe the performance of the different methods presented in Section 6 to solve the subscale equations (line 10 in Algorithm 1) for this particular numerical example. In spite of the fact that the behavior of linearization methods is highly problem dependent, we have found the results to be presented representative of several test cases.

Consider thus the problem at Ra = 10^6 with a 20×20 mesh. We took as initial condition the stationary solution using the ASGS method. We then solved the problem as transient setting a time step $\delta t = 14 \delta t_c$, where δt_c is the critical time step that would be found using the forward Euler scheme for the viscous and convective terms, which behaves as the stabilization parameter (see [13]). We took as initial guess $\tilde{\boldsymbol{u}} = \boldsymbol{0}$ and $\widetilde{T} = 0$. Convergence of the subgrid scale equations using the linearization schemes described in Section 6 are compared in Fig. 8 (left). In this figure, the method of Section 6.1 is labeled "Picard" , the method of Section 6.3 is labeled "Newton Raphson" , the method of Section 6.4 is labeled "NewRaph Segregated" and the method of Section 6.2 is labeled "Picard Gauss Seidel".

We only show the velocity subscale convergence for simplicity, because the temperature subscale convergence has the same shape. It is observed that "Picard Gauss Seidel" converges faster than the original Picard method. The Newton–Raphson segregated method presents much faster convergence than Picard's methods. The Newton–Raphson monolithic method converges very slowly for some integration points in the first iterations and then converges much faster (quadratically) than all the other linearization methods, which only converge linearly, as expected.

The next test consists of taking the same initial conditions as in the first test, but duplicating the gravity force, (i.e. Ra = $2 \cdot 10^6$). The time step is set to $\delta t = 55 \delta t_c$. In Fig. 8 (right) the convergence of the subgrid equations using the different linearization methods is compared. The most remarkable result is that only the full Newton–Raphson scheme achieves convergence at all integration points, in our case corresponding to a Gauss–Legendre quadrature. From a total of 1600 of such integration points (four per element) the Newton–Raphson segregated scheme does not achieve convergence at 4 of them, the Picard method with a Gauss–Seidel-type coupling at 16 and the original Picard method at 34. The conclusion is that the full Newton–Raphson scheme is the most robust, but sometimes needs more iterations to achieve convergence.

It was observed in practice that if the convergence of the subgrid scales is not achieved at only one integration point, then the finite element solution gets strongly affected. If the subscales are not well converged then the convergence of the finite element solutions gets strongly deteriorated, a fact already observed in [19]. When using the Newton–Raphson monolithic scheme we have always achieved convergence.
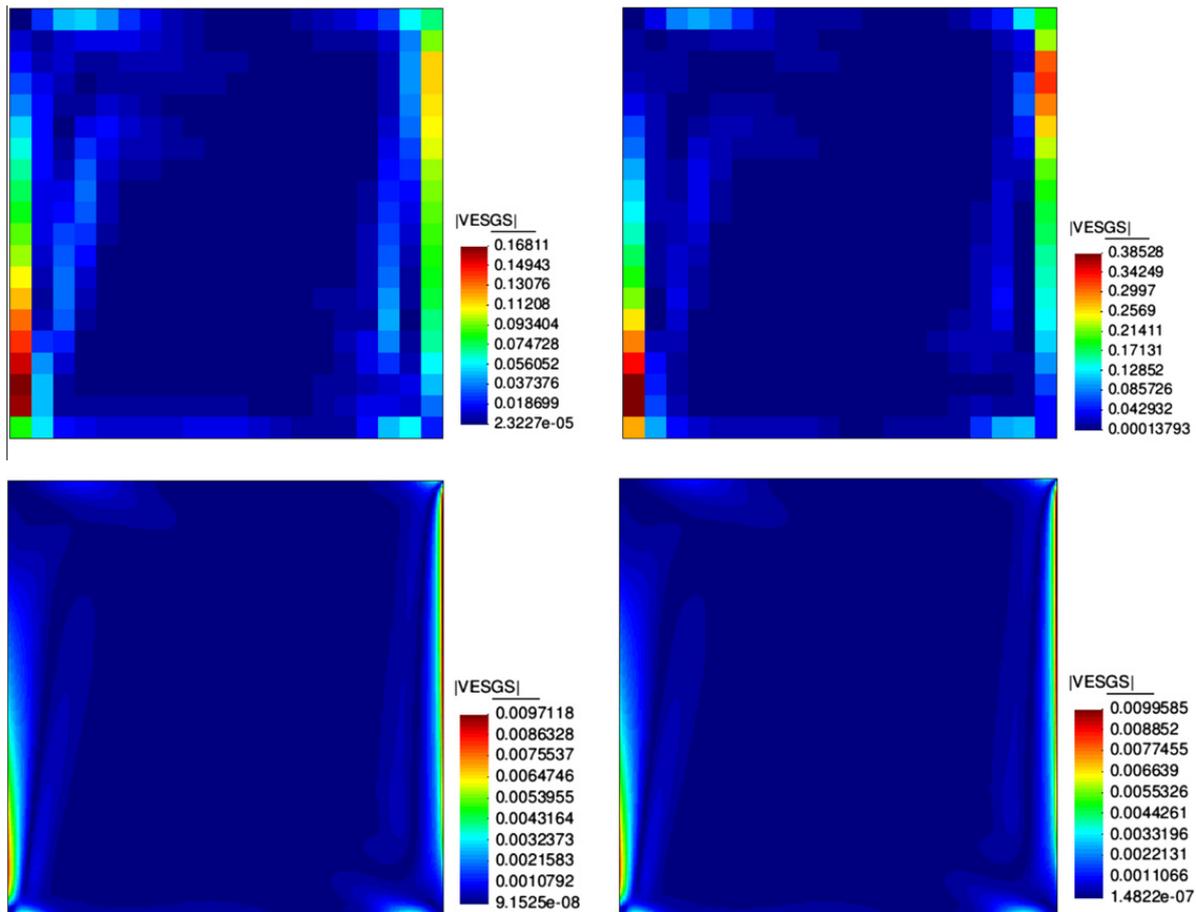
Fig. 7. Velocity subscales norm obtained using the DSS (left) and the LinSGS (right) method in a 20 × 20 (top) and a 240 × 240 mesh (bottom)
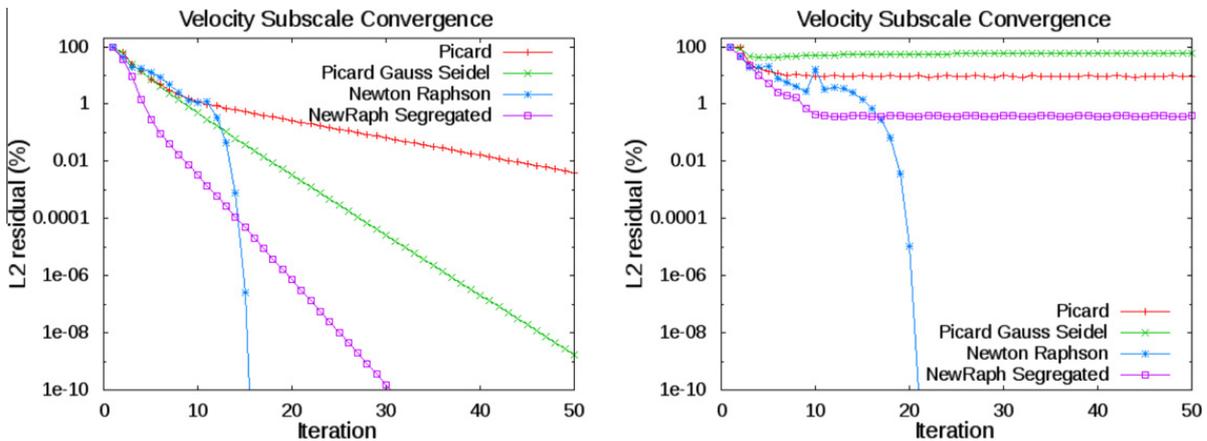

Fig. 8. Subscale convergence history. Comparisons of Picard and Newton–Raphson strategies. Ra = 10^6 (left). Ra = $2 \cdot 10^6$ (right).

7.2. Transient injection flow at low Mach regime

This flow example was recently proposed in [4], and also considered by [16]. The problem domain is $\Omega = [-L/2,L/2] \times [0,H]$, where $L = 3$ m and $H = 7$ m. The initial values are $T_0 = 300$ K and $p_0^{th} = 10^5$ Pa resulting in an initial density of $1.161 \frac{kg}{m^3}$. Furthermore $\mu = 0.005 \frac{kg}{m\,s}$ and Pr = 0.71 are taken. Zero Dirichlet boundary conditions for the velocity are assumed on all boundaries, except for a small hole in the bottom wall at $[-l/2,l/2]$ where $l = 0.2$ m. Through this hole fluid is injected subject to a parabolic inflow profile $\boldsymbol{u}_d = (0, 2.5830(1.0 - 100x^2))$ m/s. The temperature of the injected fluid is $T_D = 600$ K. Aside from this, adiabatic boundary conditions are prescribed on all boundaries. We consider a gravity $\boldsymbol{g} = (0, -9.81)$ m/s^2.

The domain is discretized with 60×60 elements, and the time step size is chosen to be $\delta t = 0.06$ s. The computation is advanced until $t_{\text{end}} = 6.0$ s. The second order time integration scheme BDF2 was used.

The results obtained with our stabilization method using dynamic subscales (DSS) are compared to those obtained using the ASGS method and a reference solution obtained using a mesh of 180×180 uniform elements and the ASGS method. Cuts of temperature and *x*-velocity fields at $y = 5.6$ m and $x = 0.2$ m when $t = 6.0$ s are depicted in Fig. 9. As in the previous example, a gain in accuracy is observed when the method of dynamic and nonlinear subscales is used. Time evolutions of thermodynamic pressure, velocities and temperature at point $(0.4, 4.0)$ m are compared in Fig. 10. This figure shows a higher
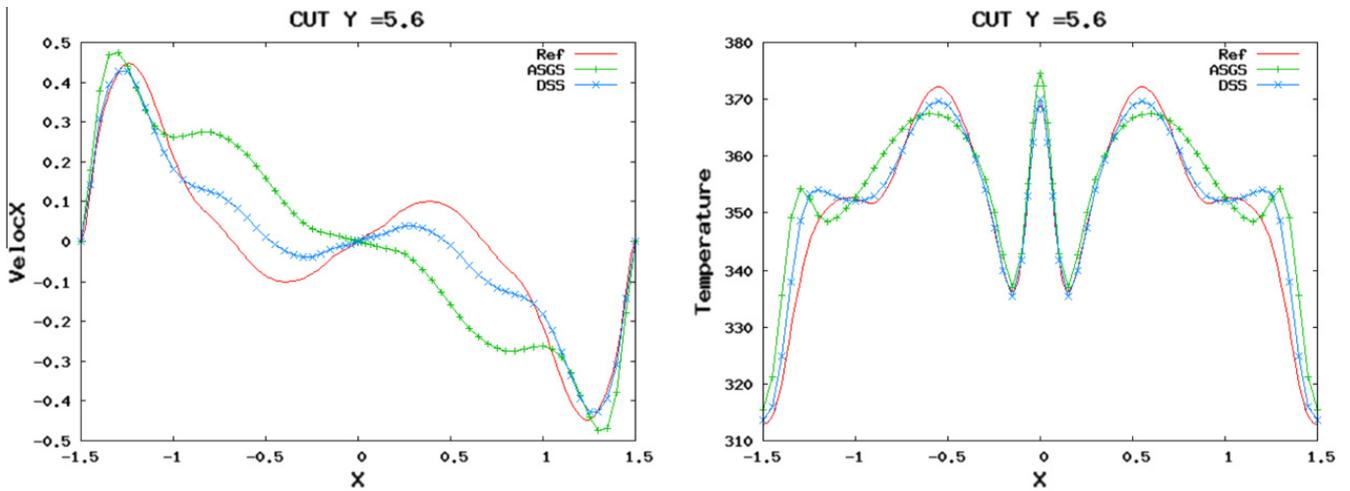
Fig. 9. Cuts of the solution at $y = 5.6$ m when $t = 6.0$ s for different stabilization methods against the reference solution.
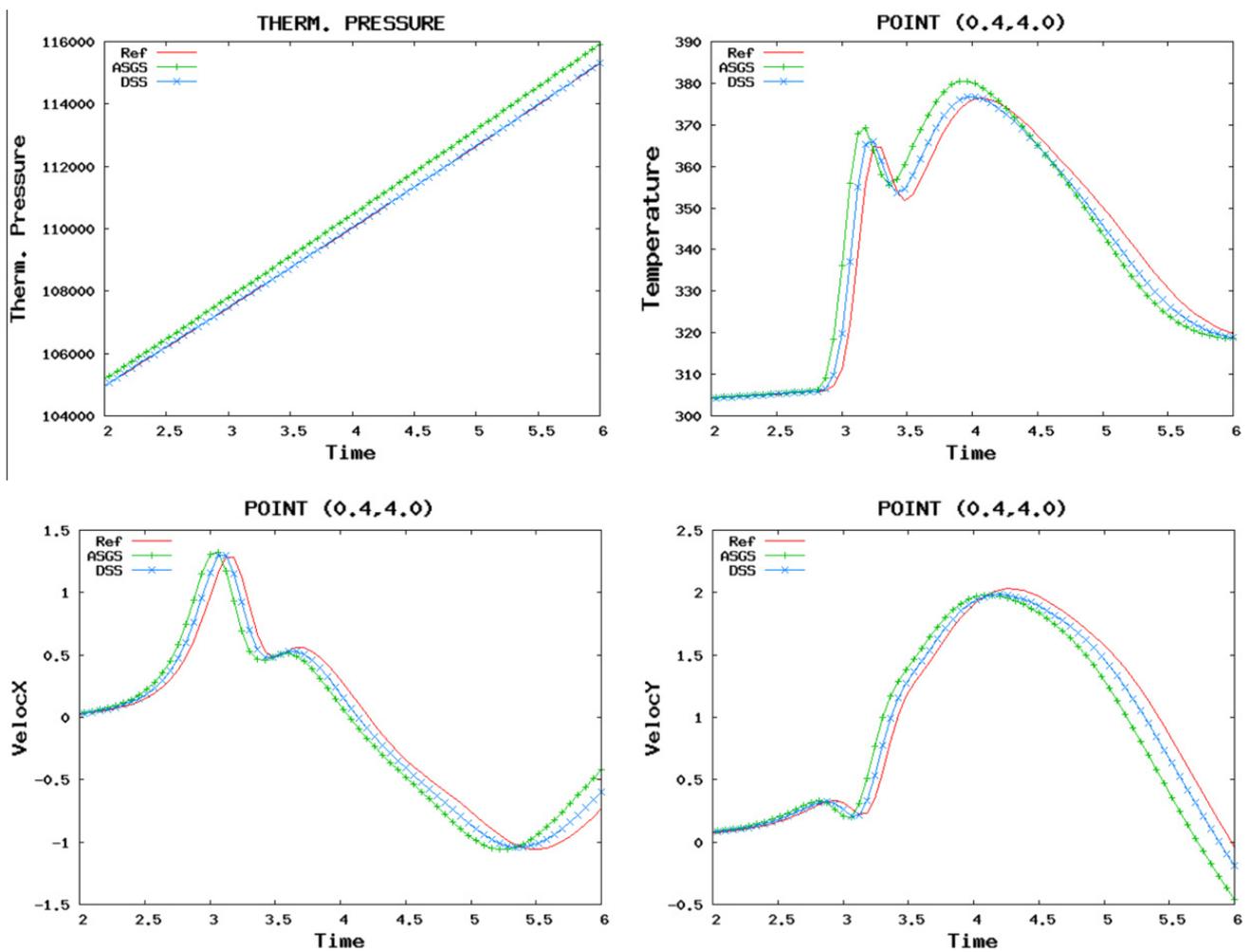

Fig. 10. Point evolution of the unknowns when using the ASGS and the DSS stabilization methods against the reference solution.
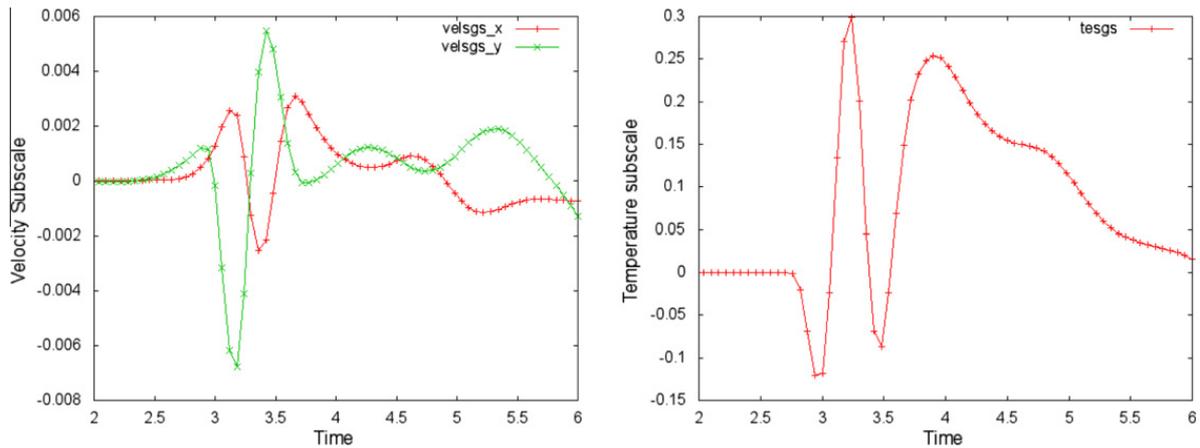
Fig. 11. Point evolution of the subgrid scale velocity (left) and temperature (right) at point $(0.4, 4.0)$ m. when using the DSS method.

temporal accuracy of the scheme when transient nonlinear subscales are used. Thermodynamic pressure evolution presents much higher accuracy when using the DSS method. This is due to global energy conservation (36). For both methods the thermodynamic pressure evolution has been determined using global mass conservation Eq. (31). Time evolution of the velocity and temperature subscales when using the DSS method are depicted in Fig. 11 at point $(0.4, 4.0)$ m . Although the relation between the subgrid scale and the finite element component is less than 0.5% for velocity and 0.1% for temperature, a significant improvement of the solution has been obtained, as observed in Fig. 10.

It is important to remark that the use of nonlinear subscales results in an increase in the cost of the solution of the nonlinear problem. The required number of nonlinear iterations increases a 25% with respect to the ASGS method. This is the price to be paid for the gain in accuracy.

The subscale equations were solved using the Newton–Raphson scheme. Convergence has been achieved at all integration points, needing at most four nonlinear iterations.

7.3. Fire in a 3D room with an open door

The third numerical test is a fire compartment similar to that considered in [28]. The problem domain is $\Omega = [0, L] \times [0, L] \times [0, H]$ where $L = 2.8$ m and $H = 2.18$ m. The compartment has an open door on the side wall of the room $(x = L)$ whose dimension is 0.7×1.853 m^2. The fire is modeled by a uniform heat source of 5 kW, located at the center of the room just over the floor, with dimensions $0.84 \times 0.84 \times 0.218$ m^3. Adiabatic boundary conditions are imposed on all the walls. Non slip boundary conditions for velocity are imposed on all the boundaries except the door, where atmospheric boundary condition is imposed, that is, a traction $\boldsymbol{t}_n = (-\rho |\boldsymbol{g}| z, 0, 0)$. As the flow is open $((\Gamma_N^{\boldsymbol{u}} \neq \emptyset))$, the thermodynamic pressure is set constant in time to $p^{th} = 101325$ Pa. The initial temperature and velocity values are $T_0 = 300$ K and $\boldsymbol{u}_0 = \boldsymbol{0}$ over all the domain Ω. Furthermore, the viscosity is $\mu = 0.0094 \frac{\text{kg}}{\text{m s}}$ and Pr = 0.71 m. The gravity is set to $\boldsymbol{g} = (0, 0, -9.8)$ m/s^2.

The compartment is meshed using grids of $20 \times 20 \times 20$ and $40 \times 40 \times 40$ uniform elements, stabilizing with the dynamic nonlinear subscale method (DSS) presented in this paper and with the ASGS method. We compare the results against a reference solution obtained using the ASGS method over a fine mesh of $80 \times 80 \times 80$ uniform elements. We solved the problem using two different time steps of $\delta t = 0.25$ s and $\delta t = 1.0$ s. The major difference between the solutions using those different time steps is in the initial transient. After some time the solutions are very similar. The computation is advanced until $t_{\text{end}} = 100.0$ s using the second order time integration scheme BDF2. The tolerance for the nonlinear iterations was set to $5 \cdot 10^{-4}$ in the relative norm of the difference between two iterates.

In Fig. 12 the temperature and vertical velocity distributions along the z direction using the DSS and ASGS methods at $t = t_{\text{end}}$ are shown, whereas in Fig. 13 the temperature and vertical velocity distributions along the x direction at $t = t_{\text{end}}$ are shown. In all those figures it is clearly observed the greater similarity of the results respect to the reference solution when using the DSS method.

The time evolution of temperature and vertical velocity (along the z direction) at point $(1.55, 1.4, 0.55)$ m are compared in Fig. 14. It is observed that over the $20 \times 20 \times 20$ grid using the ASGS method the solution differs much more from the reference solution than when using the DSS method. The same can be said over the $40 \times 40 \times 40$ grid.

The total number of nonlinear iterations needed to solve the problem (i.e. the sum of the nonlinear iterations performed in all time steps) and the total CPU time are indicated in Tables 1 and 2 for time steps $\delta t = 0.25$ s and $\delta t = 1.0$ s respectively. The CPU time spent for assembly (which includes numerical integration and solution of the subscale problem, i.e. operations involving a loop over integration points, indicated in line 10 of Algorithm 1) and CPU time spent in (linear) solver procedures are also indicated. The mean CPU times spent in nonlinear iterations are also indicated.
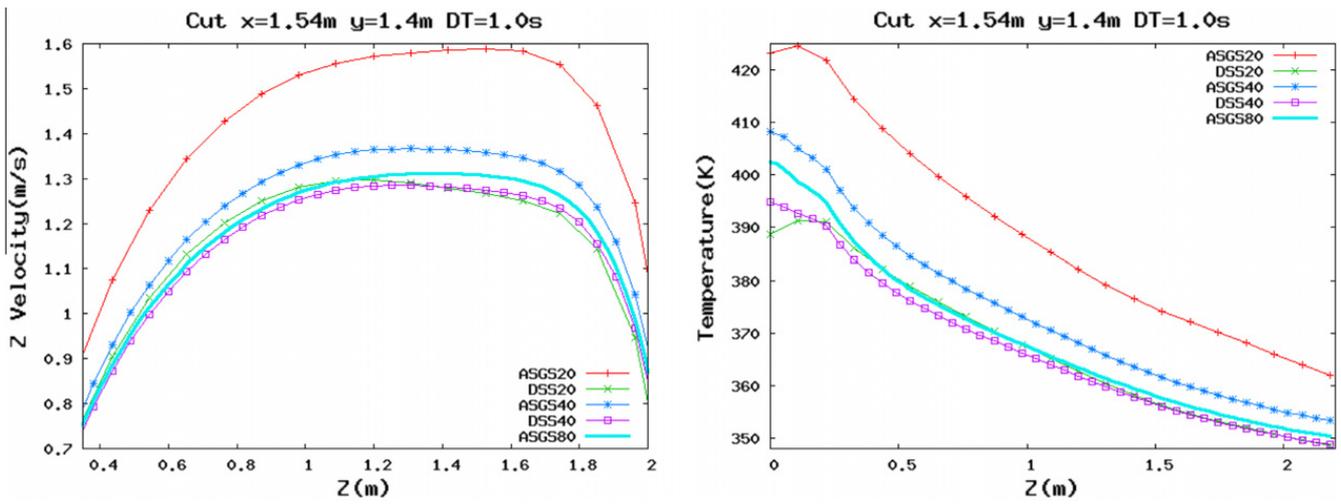
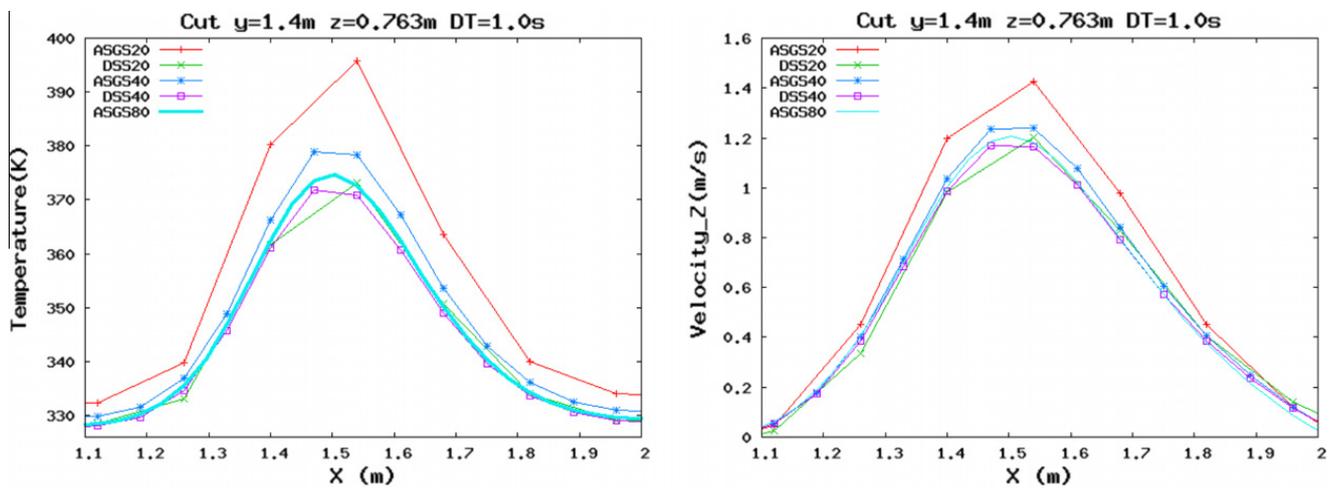**Fig. 12.** Temperature and vertical velocity distributions along *z* direction.

Fig. 13. Temperature and vertical velocity distributions along *x* direction.
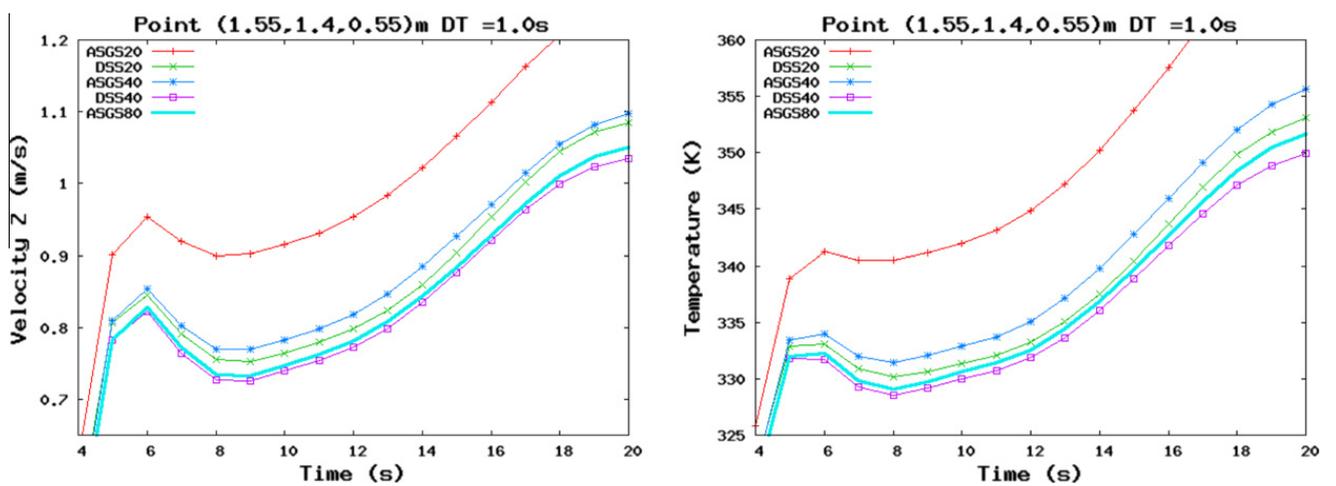

Fig. 14. Time evolution of temperature and vertical velocity (i.e. u_z) in point $(1.55, 1.4, 0.55)$ m.

The total number of nonlinear iterations do not differ more than 5% between both methods. The extra cost to evaluate the subgrid scale computation is about 15% of the total element calculations (including all operations at numerical integration points and assembly). However, we did not try to make these calculations as efficient as possible.

Table 1
Comparison of the required CPU time for the different methods over different grids using δt = 1.0 s.

δt = 1.0 s	# Iterations	CPU time (s)	CPU assem. (s)	CPU solv. (s)	CPU assem./# iterations	CPU solv./# iterations
ASGS20	574	2559	857	1699	1.49	2.96
DSS20	573	2870	1019	1846	1.78	3.22
ASGS40	600	25670	6112	19530	10.19	32.55
DSS40	573	25976	6823	19116	11.91	33.36

Table 2
Comparison of the required CPU time for the different methods over different grids using δt = 0.25 s.

δt = 0.25 s	# Iterations	CPU time (s)	CPU assem. (s)	CPU solv. (s)	CPU assem./# iterations	CPU solv./# iterations
ASGS20	1275	5216	1921	3285	1.51	2.57
DSS20	1307	6129	2296	3819	1.75	2.92
ASGS40	1298	43388	12561	30763	9.68	23.7
DSS40	1301	47608	14705	32817	11.3	25.2

The subscale equations were solved using the Newton–Raphson scheme, convergence was achieved at all integration points needing at most three nonlinear iterations.

8. Conclusions

In this article a finite element approximation of the low Mach number equations based on a splitting of the unknowns into finite element and unresolvable components has been developed. The main ingredients of the formulation are:

- To consider time dependent subscales.
- To keep the subscale components in all the nonlinear terms.

The effect of considering time dependent subscales is well known [12] and our experience with the low Mach number equations confirms the properties known for incompressible flows. The effect of considering the splitting of the unknowns in all the terms (including the state equation) leads to a more accurate solution than classical stabilization methods and provides global mass, momentum and energy conservation when using equal interpolation spaces for the velocity, pressure and temperature equations. This improvement in the quality of the solution is quite remarkable in the interior of the computational domain, as we have shown in the numerical experiments, although it is not so important for boundary fluxes. This may be due to the poor approximation of the subscales close to boundaries, since boundary subgrid scales have been neglected.

We would like to stress, once again, that we keep the splitting of the unknowns in all terms *also in the subscale equations*, and we have numerically verified that this makes a substantial difference in the accuracy of the scheme. In other words, considering the scale splitting in the nonlinear terms in the finite element equation only results in a small improvement compared to that obtained with the full nonlinear scheme presented here. A detailed strategy for solving the nonlinear subscale problem has been also presented in Section 6. From numerical experiments it turns out that a fully coupled Newton–Raphson is to be recommended.

This nonlinear and transient treatment of the subscales *also in the subscale equations* has a computational cost, in memory requirements and sometimes in the number of nonlinear iterations required to converge. Nevertheless, the extra amount of memory needed only grows linearly with the number of nodes (and will be usually dominated by the memory needed to solve the linear system).

The formulation intrinsically contains cross- and Reynolds-stress terms, and presents an open door to turbulence modeling. The present method remains unchanged irrespective of whether laminar, transitional and turbulent situations are present. Nevertheless, the adequacy of the present method for turbulent flows situations remains to be investigated. We intend to do this in our future work.

References

[1] S. Badia, R. Codina, On a multiscale approach to the transient Stokes problem. Transient subscales and anisotropic space-time discretization, Applied Mathematics and Computation 207 (2009) 415–433.
[2] S. Badia, R. Codina, J.V. Gutiérrez-Santacreu, Long term stability estimates and existence of global attractors in a finite element approximation of the Navier–Stokes equations with numerical sub-grid scale modeling, SIAM Journal on Numerical Analysis 48 (2010) 1013–1037.

[3] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Computer Methods in Applied Mechanics and Engineering 197 (2007) 173–201.

[4] A. Beccantini, E. Studer, S. Gounand, J.P. Magnaud, T. Kloczko, Numerical simulations of a transient injection flow at low Mach number regime, International Journal for Numerical Methods in Engineering 76 (2008) 662–696.

[5] J.P. Boris, F.F. Grinstein, E.S. Oran, R.L. Kolbe, New insights into large eddy simulation, Fluid Dynamics Research 10 (4–6) (1992) 199.

[6] R. Codina, Comparison of some finite element methods for solving the diffusion–convection–reaction equations, Computer Methods in Applied Mechanics and Engineering 156 (1998) 185–210.

[7] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Computer Methods in Applied Mechanics and Engineering 190 (20–21) (2001) 2681–2706.

[8] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Computer Methods in Applied Mechanics and Engineering 191 (2002) 4295–4321.

[9] R. Codina, J. Principe, Dynamic subscales in the finite element approximation of thermally coupled incompressible flows, International Journal of Numerical Methods in Fluids 54 (2007) 707–730.

[10] R. Codina, J. Principe, M. Avila, Finite element approximation of turbulent thermally coupled incompressible flows with numerical sub-grid scale modelling, International Journal of Numerical Methods for Heat and Fluid Flow 20 (5) (2010) 492–516.

[11] R. Codina, J. Principe, J. Baiges, Subscales on the element boundaries in the variational two-scale finite element method, Computer Methods in Applied Mechanics and Engineering 198 (2009) 838–852.

[12] R. Codina, J. Principe, O. Guasch, S. Badia, Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, Computer Methods in Applied Mechanics and Engineering 196 (2007) 2413–2430.

[13] R. Codina, O.C. Zienkiewicz, CBS versus GLS stabilization of the incompressible Navier–Stokes equations and the role of the time step as stabilization parameter, Communications in Numerical Methods in Engineering 18 (2002) 99–112.

[14] J. Douglas, J. Wang, An absolutely stabilized finite element method for the Stokes problem, Mathematics of Computation 52 (1989) 495–508.

[15] V. Gravemeier, W.A. Wall, An algebraic variational multiscale-multigrid method for large-eddy simulation of turbulent variable-density flow at low Mach number, Journal of Computational Physics 229 (2010) 6047–6070.

[16] V. Gravemeier, W.A. Wall, Residual-based variational multiscale methods for laminar transitional and turbulent variable-density flow at low Mach number, International Journal for Numerical Methods in Fluids (2011), doi:10.1002/fld.2242). n/a.

[17] O. Guasch, R. Codina, A heuristic argument for the sole use of numerical stabilization with no physical les modelling in the simulation of incompressible turbulent flows. Submitted, 2010.

[18] V. Heuveline, On higher-order mixed fem for low Mach number flows: Applications to a natural convection benchmark problem, International Journal for Numerical Methods in Fluids 41 (2003) 1339–1356.

[19] G. Houzeaux, J. Principe, A variational subgrid scale model for transient incompressible flows, International Journal of Computational Fluid Dynamics 22 (2008) 135–152.

[20] T.J.R. Hughes, G. Emgel, L. Mazzei, M.G. Larson, The continuous Galerkin method is locally conservative, Journal of Computational Physics 163 (2000) 467–488.

[21] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.B. Quincy, The variational multiscale method – a paradigm for computational mechanics, Computer Methods in Applied Mechanics and Engineering 166 (1998) 3–24.

[22] T.J.R. Hughes, L.P. Franca, M. Balestra, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: a stable Petrov Galerkin formulation of the stokes problem accomodating equal order interpolations, Computer Methods in Applied Mechanics and Engineering 59 (1) (1986) 85–99.

[23] T.J.R. Hughes, L. Mazzei, K.E. Jansen, Large eddy simulation and the variational multiscale method, Computing and Visualization in Science 3 (2000) 47–59.

[24] T.J.R. Hughes, L. Mazzei, A.A. Oberai, The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence, Physics of Fluids 13 (2) (2001) 505–512.

[25] T.J.R. Hughes, A.A. Oberai, L. Mazzei, Large eddy simulation of turbulent channel flows by the variational multiscale method, Physics of Fluids 13 (6) (2001) 1784–1799.

[26] T.J.R. Hughes, G. Scovazzi, L.P. Franca, Multiscale and stabilized methods, in: T.J.R. Hughes, E. Stein, R. de Borst (Eds.), Encyclopedia of Computational Mechanics, Willey, Chichester, 2004.

[27] P.L. Lions, Mathematical topics in fluid dynamics, Compressible Models, vol. 2, Oxford University Press, 1996.

[28] W. Liu, G. Makhviladze, An implicit finite element solution of thermal flows at low Mach number, Journal of Computational Physics 227 (2008) 2743–2757.

[29] A. Majda, J. Sethian, The derivation and numerical solution of the equations for zero Mach number combustion, Combustion Science and Technology 42 (1985) 185–205.

[30] M.J. Martinez, D.K. Gartling, A finite element method for low-speed compressible flows, Computer Methods in Applied Mechanics and Engineering 193 (21-22) (2004) 1959–1979.

[31] J. Principe, Subgrid Scale Stabilised Finite Elements for Low Speed Flows, PhD thesis, Universitat Politecnica de Catalunya, 2008.

[32] J. Principe, R. Codina, A stabilized finite element approximation of low speed thermally coupled flows, International Journal of Numerical Methods for Heat and Fluid Flow 18 (2007) 835–867.

[33] J. Principe, R. Codina, Mathematical models for thermally coupled low speed flows, Advances in Theoretical and Applied Mechanics 2 (2009) 93–112.

[34] J. Principe, R. Codina, F. Henke, The dissipative structure of variational multiscale methods for incompressible flows, Computer Methods in Applied Mechanics and Engineering 199 (2010) 791–801.

[35] P. Le Quere, C. Weisman, H. Paillere, J. Vierendeels, E. Dick, R. Becker, M. Braack, J. Locke, Modelling of natural convection flows with large temperature differences: a benchmark problem for low Mach number solvers. Part 1. Reference solutions, ESAIM: Mathematical Modelling and Numerical Analysis 39 (3) (2005) 609–616.