ELSEVIER

Comput. Methods Appl. Mech. Engrg. 191 (2002) 4295–4321

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Stabilized finite element approximation of transient incompressible flows using orthogonal subscales

Ramon Codina

Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

Received 5 December 2000; received in revised form 15 April 2002

Abstract

In this paper we present a stabilized finite element method to solve the transient Navier–Stokes equations based on the decomposition of the unknowns into resolvable and subgrid scales. The latter are approximately accounted for, so as to end up with a stable finite element problem which, in particular, allows to deal with convection-dominated flows and the use of equal velocity–pressure interpolations. Three main issues are addressed. The first is a method to estimate the behavior of the stabilization parameters based on a Fourier analysis of the problem for the subscales. Secondly, the way to deal with transient problems discretized using a finite difference scheme is discussed. Finally, the treatment of the nonlinear term is also analyzed. A very important feature of this work is that the subgrid scales are taken as orthogonal to the finite element space. In the transient case, this simplifies considerably the numerical scheme.

1. Introduction

Let Ω be an open, bounded and polyhedral domain of $\mathbb{R}^d$, where $d = 2$ or 3 is the number of space dimensions, $\Gamma = \partial\Omega$ its boundary and $[0, T]$ the time interval of analysis. The Navier–Stokes problem consists in finding a velocity $\boldsymbol{u}$ and a pressure p such that

$$\partial_t \boldsymbol{u} - \nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \quad t \in]0, T[, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \quad t \in]0, T[, \tag{2}$$

where ν is the kinematic viscosity and $\boldsymbol{f}$ is the force vector. These equations must be supplied with an initial condition of the form $\boldsymbol{u} = \boldsymbol{u}^0$ in Ω, $t = 0$, and a boundary condition which, for simplicity, will be taken as $\boldsymbol{u} = \boldsymbol{0}$ on Γ, $t \in]0, T[$.

Let us introduce some standard notation. The space of functions whose p power ($p \geqslant 1$) is integrable in a domain ω is denoted by $L^p(\omega)$, and the space of functions whose distributional derivatives of order up to

E-mail address: ramon.codina@upc.es (R. Codina).
URL: http://www.rmee.upc.es/homes/codina.

0045-7825/02/$ - see front matter
PII: S0045-7825(02)00337-7

$m \geqslant 0$ (integer) belong to $L^2(\omega)$ by $H^m(\omega)$. The space $H_0^1(\omega)$ consists of functions in $H^1(\omega)$ vanishing on $\partial\omega$. The topological dual of $H_0^1(\Omega)$ is denoted by $H^{-1}(\Omega)$, and the duality pairing by $\langle \cdot, \cdot \rangle$. A bold character is used to denote the vector counterpart of all these spaces. The L^2 inner product in ω (for scalars, vectors or tensors) is denoted by $(\cdot, \cdot)_\omega$, and the norm in a Banach space X by $\| \cdot \|_X$. This notation is simplified in some cases as follows: $(\cdot, \cdot)_\Omega \equiv (\cdot, \cdot)$, $\| \cdot \|_{L^2(\Omega)} \equiv \| \cdot \|$, and if K is the domain of an element (see below) $\| \cdot \|_{L^2(K)} \equiv \| \cdot \|_K$.

Using this notation, the velocity and pressure finite element spaces for the continuous problem are $\boldsymbol{L}^2(0, T; \mathscr{V}_0)$ and $L^1(0, T; \mathscr{Q}_0)$, respectively, where $\mathscr{V}_0 := \boldsymbol{H}_0^1(\Omega)$, $\mathscr{Q}_0 := L^2(\Omega)/\mathbb{R}$. We shall be interested also in the spaces $\mathscr{W}_0 := \mathscr{V}_0 \times \mathscr{Q}_0$, $\mathscr{V} := \boldsymbol{H}^1(\Omega)$, $\mathscr{Q} := L^2(\Omega)$, $\mathscr{W} := \mathscr{V} \times \mathscr{Q}$. The weak form of the problem consists in finding $[\boldsymbol{u}, p] \in \boldsymbol{L}^2(0, T; \mathscr{V}_0) \times L^1(0, T; \mathscr{Q}_0)$ such that

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) + \nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) + (\boldsymbol{u} \cdot \nabla \boldsymbol{u}, \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v}) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle, \tag{3}$$

$$(q, \nabla \cdot \boldsymbol{u}) = 0, \tag{4}$$

for all $[\boldsymbol{v}, q] \in \mathscr{V}_0 \times \mathscr{Q}_0$, and satisfying the initial condition in a weak sense.

The numerical approximation of problem (3) and (4) is in principle straightforward. However, apart for the difficulties associated to the extremely complex physical phenomena that it may represent, there are several well known numerical problems due to the mathematical structure of the equations.

The discretization strategy adopted in this work consists of two steps. First, the equations are discretized in time using a finite difference time integration scheme, and then a finite element approximation is performed in space. This procedure uncouples errors coming from the temporal discretization from those of the spatial one. Nevertheless, it has to be remarked that the most common approach is to proceed the other way around, that is to say, by discretizing first in space and then approximating the resulting system of ordinary differential equations in time.

Concerning the temporal discretization, we will use here the generalized trapezoidal rule, which is the simplest single step finite difference method. The stability of this scheme is analyzed for example in [1]. Convergence in the particular case of the backward Euler method is proven in [2] (see also [3,4]). However, the ideas presented here can be applied to any other finite difference scheme.

Referring to the spatial discretization, it is well known that the standard Galerkin method may fail basically for two reasons: the dominance of the (nonlinear) convective term over the viscous one when ν is small and the compatibility required for the velocity and pressure finite element spaces posed by the inf–sup condition. Both can be overcome by resorting to a *stabilized* formulation. The one adopted in this work is based on the subgrid scale concept and, in particular, in the approach introduced by Hughes in [5,6] for the scalar convection–diffusion equation (see also [7,8] for related methods). The basic idea is to approximate the *effect* of the component of the continuous solution which cannot be resolved by the finite element mesh on the discrete finite element solution. An important feature of the formulation developed herein is that the unresolved component, hereafter referred to as *subgrid scale* or *subscale*, is assumed to be L^2 orthogonal to the finite element space, in a sense to be explained later. This idea was first introduced in [9] as an extension of a stabilization method originally introduced for the Stokes problem in [10] and fully analyzed for the stationary Navier–Stokes equations in [11]. The method of these papers is only intended to stabilize the pressure, and not to deal with convection dominated flows. This, together with the goal of dealing with transient problems, is what the formulation elaborated in this work permits.

A question that arises when a stabilized finite element method is applied to the equations coming from a temporal discretization is how does this discretization in time affect the stabilized method. This is one of the fundamental issues analyzed in this paper. It is shown here that the subgrid scale decomposition allows to answer this question in a simple manner. In our particular case we start from a finite difference time discretization, but exactly the same ideas can be applied if a space-time finite element approximation is chosen.

As we will see, the final conclusion is that the subgrid scales satisfy an *evolution* equation which needs to be approximated to obtain a closed-form expression for them. We will call this process *modeling of the subscales*, in analogy to what is done in turbulence.

Our modeling assumption is based on a Fourier analysis of the problem for the subscales, but of course other options are possible. For example, the numerical formulations presented in [12,13] are based on adding numerical diffusion to the subgrid components of the solution, which is another type of modeling approach. Likewise, when subscales are approximated by bubble functions, as in [7,14–19], solving for them is what can be considered the modeling step.

A related approach is the nonlinear Galerkin method and its variants (see [20–25] and references therein). The idea now is to split the solution into large and small scales *in space* and use a different treatment for them in time, generally with a coarser time approximation for the small scales. The goal again is to obtain a numerical model (or a continuous decomposition as in [21]) whose computational cost be driven by the large scales but in which the subgrid scales are approximately accounted for.

It has to be remarked that both in the development of the stabilized formulation for the stationary problem and in its extension to the transient case, our arguments are *heuristic*. They are intended to lead to a fully discrete problem from which improved properties with respect to the standard Galerkin method can be expected. The validity of these properties can only be verified through the numerical analysis of the method (which is open for the general nonlinear and transient problem) and the numerical experiments.

This paper is organized as follows. In the following section the bases of the stabilized formulation are explained. For this, it is enough to consider the stationary and linearized form of problem (1) and (2), that is, replacing the momentum Eq. (1) by

$$-\nu\Delta\boldsymbol{u} + \boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \tag{5}$$

(and of course dropping the initial condition), where $\boldsymbol{a}$ is a given solenoidal velocity field. The finite element approximation of the linear problem (5) and (2) (Oseen problem) may suffer from instabilities due to the velocity–pressure interpolation and from the dominance of the convective term over the viscous one. Both are eliminated by using the stabilized formulation presented in Section 2, where it is motivated in some detail. After presenting the problem to be solved, the subgrid scale decomposition is described. The subscales, which are solution of a differential equation, are approximated as proportional to the independent term of this equation. The coefficients of proportionality are called stabilization parameters. A method to determine these parameters based on a Fourier analysis of the equation for the subscales is then presented, which is applicable to any forcing term in this equation. The next step is precisely to select this term so that the subscales be orthogonal to the finite element space, which leads to the formulation that we propose. It is shown in [26] that the stability and convergence properties of this method are similar to those of related formulations aiming to stabilize the pressure interpolation and the convective term, as those analyzed in [27–32], among others. However, in our case the analysis is significantly more complex due to the fact that the bilinear form associated to the problem is not coercive. Using appropriate norms, only an inf–sup condition can be proven.

The second important point to be treated is how to apply the stabilized formulation to the equations resulting from the temporal discretization of a transient problem. This is done in Section 3. Again, the effects of the nonlinearity are irrelevant for this discussion, and it suffices to consider the linear equation

$$\partial_t\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \quad t\in]0,T[, \tag{6}$$

instead of Eq. (1). After presenting the basic time discretization, it is shown that the decomposition into large and subgrid scales leads to an evolution equation for the latter that can be modeled using the same ideas as for the stationary case. Two implementation issues are then discussed. The first concerns how to track the subscales in practice, and the second is a discussion on the possibility of treating some terms explicitly.

Finally, the stabilization concepts developed in the linear case are applied to the fully transient and nonlinear problem (1) and (2) in Section 4. This is can be easily done after a linearization of the problem. The resulting scheme, which summarizes the method proposed in this paper, is presented in Box 1.

Some numerical results are presented in Section 5. The main conclusion that can be drawn from them is that the formulation presented here is in general as stable as other related methods but less diffusive. From the computational point of view, it may be advantageous in some cases. These and other conclusions are finally stated in Section 6.

2. Stationary Oseen equations

2.1. Problem statement

In this section we consider the linear and stationary problem (5) and (4), supplied with the homogeneous Dirichlet condition for the velocity field.

Let $\boldsymbol{U} \equiv [\boldsymbol{u}, \boldsymbol{p}] \in \mathscr{W}_0$. The equations to be solved can be written as

$$\mathscr{L}(\boldsymbol{U}) := \begin{bmatrix} -\nu\Delta\boldsymbol{u} + \boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p \\ \nabla\cdot\boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} =: \boldsymbol{F} \tag{7}$$

in the domain Ω and $\boldsymbol{u} = 0$ on Γ. Let $\boldsymbol{V} \equiv [\boldsymbol{v}, q] \in \mathscr{W}_0$. The variational statement for problem (7) can be written in terms of the bilinear form defined on $\mathscr{W}_0 \times \mathscr{W}_0$ as

$$B(\boldsymbol{U}, \boldsymbol{V}) := \nu(\nabla\boldsymbol{u}, \nabla\boldsymbol{v}) + (\boldsymbol{a}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}) - (p, \nabla\cdot\boldsymbol{v}) + (q, \nabla\cdot\boldsymbol{u}) \tag{8}$$

and the linear form $L(\boldsymbol{V}) := \langle \boldsymbol{f}, \boldsymbol{v}\rangle$. Problem (7) with the homogeneous Dirichlet condition consists then in finding $\boldsymbol{U} \in \mathscr{W}_0$ such that

$$B(\boldsymbol{U}, \boldsymbol{V}) = L(\boldsymbol{V}), \quad \forall \boldsymbol{V} \in \mathscr{W}_0. \tag{9}$$

The standard Galerkin approximation of this abstract variational problem is now straightforward. Let $\mathscr{P}_h$ denote a finite element partition of the domain Ω. The diameter of an element domain $K \in \mathscr{P}_h$ is denoted by h_K and the diameter of the finite element partition by $h = \max\{h_K | K \in \mathscr{P}_h\}$. We can now construct conforming finite element spaces $\mathscr{V}_h \subset \mathscr{V}$, $\mathscr{Q}_h \subset \mathscr{Q}$ and $\mathscr{W}_h = \mathscr{V}_h \times \mathscr{Q}_h$ in the usual manner, as well as the corresponding subspaces $\mathscr{V}_{h,0}$, $\mathscr{Q}_{h,0}$ and $\mathscr{W}_{h,0} = \mathscr{V}_{h,0} \times \mathscr{Q}_{h,0}$. In principle, functions in $\mathscr{V}_h$ are continuous, whereas functions in $\mathscr{Q}_h$ not necessarily. Likewise, the polynomial orders of these spaces may be different.

The discrete version of problem (9) is: find $\boldsymbol{U}_h \in \mathscr{W}_{h,0}$ such that

$$B(\boldsymbol{U}_h, \boldsymbol{V}_h) = L(\boldsymbol{V}_h), \quad \forall \boldsymbol{V}_h \in \mathscr{W}_{h,0}. \tag{10}$$

The well posedness of this problem relies on the ellipticity of the viscous term and the inf–sup or Babuška–Brezzi condition (see [33]), which can be shown to hold for the continuous problem. The first property is automatically inherited by its discrete counterpart. However, the inf–sup condition needs to be explicitly required. This leads to the need of using mixed interpolations, that is, different for $\boldsymbol{u}$ and p, and verifying

$$\inf_{q_h \in \mathscr{Q}_{h,0}} \sup_{\boldsymbol{v}_h \in \mathscr{V}_{h,0}} \frac{(q_h, \nabla\cdot\boldsymbol{v}_h)}{\|q_h\| \|\boldsymbol{v}_h\|_1} \geqslant \beta > 0, \tag{11}$$

for a constant β independent of h.

Convenient velocity–pressure interpolations, such as *equal* interpolation, turn out to violate condition (11). This is why many of the so called *stabilized formulations* have been proposed to approximate problem (9). The idea is to replace (10) by another discrete variational problem in which the bilinear form B is

replaced by a possibly mesh dependent bilinear form B_h with enhanced stability properties. Likewise, it has already been mentioned that instability problems may arise when the convective term dominates the viscous one. Both this and the need to satisfy (11) can be overcome by using the finite element formulation described next.

2.2. The subgrid scale approach

Let $\mathscr{W} = \mathscr{W}_h \oplus \tilde{\mathscr{W}}$, where $\tilde{\mathscr{W}}$ is any space to complete $\mathscr{W}_h$ in $\mathscr{W}$. Obviously, $\tilde{\mathscr{W}}$ is infinite-dimensional, but once the final method will be formulated, it will be approximated by a finite-dimensional space (cf. Remark 2 below), although we will keep the same symbol $\tilde{\mathscr{W}}$ for it. The elements of this space are denoted by $\tilde{\boldsymbol{V}} = [\tilde{\boldsymbol{v}}, \tilde{q}]$. Likewise, let $\mathscr{W}_0 = \mathscr{W}_{h,0} \oplus \tilde{\mathscr{W}}_0$, with $\tilde{\mathscr{W}}_0$ any complement of $\mathscr{W}_{h,0}$ in $\tilde{\mathscr{W}}_0$. The space $\tilde{\mathscr{W}}_0$ will be called the space of *subgrid scales* or *subscales*.

The continuous problem is equivalent to find $\boldsymbol{U}_h \in \mathscr{W}_{h,0}$ and $\tilde{\boldsymbol{U}} \in \tilde{\mathscr{W}}_0$ such that

$$B(\boldsymbol{U}_h, \boldsymbol{V}_h) + B(\tilde{\boldsymbol{U}}, \boldsymbol{V}_h) = L(\boldsymbol{V}_h) \quad \forall \boldsymbol{V}_h \in \mathscr{W}_{h,0}, \tag{12}$$

$$B(\boldsymbol{U}_h, \tilde{\boldsymbol{V}}) + B(\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{V}}) = L(\tilde{\boldsymbol{V}}) \quad \forall \tilde{\boldsymbol{V}} \in \tilde{\mathscr{W}}_0. \tag{13}$$

Integrating by parts within each element in (12) and (13), it is found that these two equations can be written as

$$B(\boldsymbol{U}_h, \boldsymbol{V}_h) + \sum_K \int_K \tilde{\boldsymbol{U}} \cdot \mathscr{L}^*(\boldsymbol{V}_h)\,\mathrm{d}\Omega + \sum_K \int_{\partial K} \tilde{\boldsymbol{u}} \cdot (q_h \boldsymbol{n} + \nu \boldsymbol{n} \cdot \nabla \boldsymbol{v}_h)\,\mathrm{d}\Gamma = L(\mathbf{V}_h), \tag{14}$$

$$\sum_K \int_{\partial K} \tilde{\boldsymbol{v}} \cdot (p\boldsymbol{n} + \nu \boldsymbol{n} \cdot \nabla \boldsymbol{u})\,\mathrm{d}\Gamma + \sum_K \int_K \tilde{\boldsymbol{V}} \cdot \mathscr{L}(\tilde{\boldsymbol{U}})\,\mathrm{d}\Gamma = \sum_K \int_K \tilde{\boldsymbol{V}} \cdot [\boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h)]\,\mathrm{d}\Omega, \tag{15}$$

where $\sum_K$ stands for the summation over all $K \in \mathscr{P}_h$, $\boldsymbol{n}$ is unit normal exterior to the integration domain, and $\mathscr{L}^*$ is the formal adjoint of $\mathscr{L}$, given by

$$\mathscr{L}^*(\boldsymbol{V}_h) = \begin{bmatrix} -\nu \Delta \boldsymbol{v}_h - \boldsymbol{a} \cdot \nabla \boldsymbol{v}_h - \nabla q_h \\ -\nabla \cdot \boldsymbol{v}_h \end{bmatrix}. \tag{16}$$

Assuming that the exact tractions are continuous across inter-element boundaries, the first term of (15) vanishes. This equation is then equivalent to

$$\mathscr{L}(\tilde{\boldsymbol{U}}) = \boldsymbol{R} := \boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h) + \boldsymbol{V}_{h,\mathrm{ort}} \quad \text{in } K \in \mathscr{P}_h, \quad \boldsymbol{V}_{h,\mathrm{ort}} \in \tilde{\mathscr{W}}_0^{\perp}, \tag{17}$$

which must be satisfied together with boundary conditions on ∂K that are unknown, but who must ensure in particular the continuity of the diffusive fluxes across interior boundaries.

The element $\boldsymbol{V}_{h,\mathrm{ort}}$ appearing in (17) is responsible to guarantee that $\mathscr{L}(\tilde{\boldsymbol{U}}) - [\boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h)]$ belongs to $\tilde{\mathscr{W}}_0^{\perp}$, that is what (15) (without the first term) implies. Its expression depends on the choice of the space $\tilde{\mathscr{W}}_0$, and will be determined for the particular option that we will use later on. Here and below, orthogonality is understood with respect to the L^2 inner product, unless otherwise specified.

The idea now is to approximate the solution of (17) with the appropriate boundary conditions by

$$\tilde{\boldsymbol{U}} \approx \boldsymbol{\tau}_K \boldsymbol{R} \quad \text{in } K \in \mathscr{P}_h, \tag{18}$$

where $\boldsymbol{\tau}_K$ is a matrix of algorithmic parameters depending on K and the coefficients of the operator $\mathscr{L}$. This approximation for $\tilde{\boldsymbol{U}}$ is intended to mimic the effect of the exact subscales in the volume integral of (14), whereas the integral over the element faces will be neglected. Observe that the *pointwise values of* $\tilde{\boldsymbol{U}}$ *are not*

needed, and thus (18) needs not to be understood point-wise. If the coefficients of $\mathscr{L}$ are constant, only the moments of $\tilde{\boldsymbol{U}}$ appear in (14) ($\mathscr{L}^*(\boldsymbol{V}_h)$ is a polynomial).

Matrix $\boldsymbol{\tau}_K$ in (18) will be called the *matrix of stabilization parameters*. Its design is one of the cornerstones in the development of stabilized finite element methods, many of which can be formulated in the previous framework [5,34]. The heuristic approach proposed in this work is the subject of the following subsection.

2.3. Behavior of the stabilization parameters from a Fourier analysis

To simplify the discussion, we may consider first the previous procedure applied to the convection–diffusion-reaction equation

$$-\nu\Delta u + \boldsymbol{a}\cdot\nabla u + \sigma u = f \quad \text{in } \Omega,$$

where u and f are now scalar and $\sigma \geqslant 0$ is a reaction coefficient. The counterpart of (17) is

$$\begin{aligned} &-\nu\Delta\tilde{\boldsymbol{u}} + \boldsymbol{a}\cdot\nabla\tilde{\boldsymbol{u}} + \sigma\tilde{\boldsymbol{u}} = r \quad \text{in } K \in \mathscr{P}_h,\\ &r := f - (-\nu\Delta u_h + \boldsymbol{a}\cdot\nabla u_h + \sigma u_h), \end{aligned} \tag{19}$$

which must be solved approximately for the subscale $\tilde{\boldsymbol{u}}$, u_h being the finite element approximation to u. In what follows, it is understood that all the quantities are referred to the element K under consideration. Likewise, for simplicity we will assume now that $\boldsymbol{a}$ is constant over element K.

As for the Oseen problem, the solution to Eq. (19) will be approximated by

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) \approx \tau r(\boldsymbol{x}), \tag{20}$$

where τ is a parameter to be determined. The purpose of what follows is to give an expression for τ and to precise in which sense τr approximates $\tilde{\boldsymbol{u}}$.

Let us consider the following Fourier transform of a generic function g defined on K:

$$\hat{g}(\boldsymbol{k}) := \int_K \mathrm{e}^{-\mathrm{i}\frac{\boldsymbol{k}\cdot\boldsymbol{x}}{h}} g(\boldsymbol{x})\,\mathrm{d}\Omega_x, \tag{21}$$

where $\mathrm{i} = \sqrt{-1}$, h is now the diameter of element K and $\boldsymbol{k} = (k_1, \ldots, k_d)$ is the dimensionless wave number.

The subscales $\tilde{\boldsymbol{u}}$ are the part of the continuous solution which cannot be approximated by the finite element discretization. This means that their Fourier representation will be dominated by the components with *high wave numbers*.

If n_j is the jth component of the normal exterior to K, it can be readily checked that

$$\widehat{\frac{\partial g}{\partial x_j}}(\boldsymbol{k}) = \int_{\partial K} n_j \mathrm{e}^{-\mathrm{i}\frac{\boldsymbol{k}\cdot\boldsymbol{x}}{h}} g(\boldsymbol{x})\,\mathrm{d}\Gamma_x + \mathrm{i}\frac{k_j}{h}\hat{g}(\boldsymbol{k}). \tag{22}$$

From this expression it is seen that if we are interested in high wave numbers, the second term in the right-hand side of this expression dominates the first one, *no matter which is the value of the function g on* ∂K. Thus, for functions with high wave numbers we may approximate Eq. (22) by

$$\widehat{\frac{\partial g}{\partial x_j}}(\boldsymbol{k}) \approx \mathrm{i}\frac{k_j}{h}\hat{g}(\boldsymbol{k}). \tag{23}$$

All the properties valid for functions of rapid decay defined on $\mathbb{R}^d$ apply to this case. We assume that this in particular can be applied to the subscales $\tilde{\boldsymbol{u}}$ and its derivatives. Observe that (23) would hold if we assume that the subscales vanish on the element boundaries, that is, they are *bubble functions*. This assumption is very often adopted *a priori*, and thus our argument provides some heuristic justification to it.

If we take the Fourier transform of Eq. (19) we will have that

$$\hat{\tilde{\boldsymbol{u}}}(\boldsymbol{k}) \approx \mathscr{T}(\boldsymbol{k})\hat{\boldsymbol{r}}(\boldsymbol{k}), \qquad \mathscr{T}(\boldsymbol{k}) := \left(\nu \frac{|\boldsymbol{k}^2|}{h^2} + \mathrm{i}\frac{\boldsymbol{a}\cdot\boldsymbol{k}}{h} + \sigma \right)^{-1}. \tag{24}$$

Plancherel's formula leads to

$$\|\tilde{\boldsymbol{u}}\|_K^2 \approx \frac{1}{(2\pi)^d}\|\hat{\tilde{\boldsymbol{u}}}\|_{L^2(\mathbb{R}^d)}^2 \approx \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} |\mathscr{T}(\boldsymbol{k})|^2 |\hat{\boldsymbol{r}}(\boldsymbol{k})|^2 \,\mathrm{d}k, \tag{25}$$

where both approximations come from neglecting the boundary values of $\tilde{\boldsymbol{u}}$. Since both $|\mathscr{T}(\boldsymbol{k})|^2$ and $|\hat{\boldsymbol{r}}(\boldsymbol{k})|^2$ are nonnegative, the mean value theorem implies that there exists a wavenumber $\boldsymbol{k}_0$ for which

$$\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} |\mathscr{T}(\boldsymbol{k})|^2 |\hat{\boldsymbol{r}}(\boldsymbol{k})|^2 \,\mathrm{d}K = \frac{1}{(2\pi)^d}|\mathscr{T}(\boldsymbol{k}_0)|^2 \int_{\mathbb{R}^d} |\hat{\boldsymbol{r}}(\boldsymbol{k})|^2 \,\mathrm{d}k.$$

This, together with Eq. (25) and using again Plancherel's formula yields

$$\|\tilde{\boldsymbol{u}}\|_K \approx |\mathscr{T}(\boldsymbol{k}_0)|\|r\|_K. \tag{26}$$

This expression is what we were looking for. If we identify τ in Eq. (20) with $|\mathscr{T}(\boldsymbol{k}_0)|$, it allows us to conclude that *if we take*

$$\tau = \left[\left(c_1 \frac{\nu}{h^2} + \sigma \right)^2 + \left(c_2 \frac{|\boldsymbol{a}|}{h} \right)^2 \right]^{-1/2} \tag{27}$$

then, there exist values of c_1 and c_2 independent of h for which Eq. (20) *holds, in the sense that both $\tilde{u}$ and τr have* (*approximately*) *the same L^2-norm over element K.* Moreover, c_1 is independent of the coefficients ν, $\boldsymbol{a}$ and σ, whereas c_2 depends only on the direction of $\boldsymbol{a}$, but not on its magnitude. The constant c_1 can be identified with $|\boldsymbol{k}_0|^2$ and c_2 with $|\boldsymbol{k}_0||\cos\alpha|$, α being the angle between $\boldsymbol{a}$ and $\boldsymbol{k}_0$. This in particular implies that $c_2^2 \leqslant c_1$. Observe that $\boldsymbol{k}_0$ depends on the residual $r(\boldsymbol{x})$ in Eq. (19), and thus the constants c_1 and c_2 will also depend on it.

Let us consider now the application of these ideas to the Oseen problem, again taking the advection velocity constant. The equation for the subscales (17) can be written as

$$-\nu\nabla\tilde{\boldsymbol{u}} + \boldsymbol{a}\cdot\nabla\tilde{\boldsymbol{u}} + \nabla\tilde{p} = \boldsymbol{r}_1, \tag{28}$$

$$\nabla\cdot\tilde{\boldsymbol{u}} = r_2, \tag{29}$$

where $\boldsymbol{r}_1$ and r_2 are the components of $\boldsymbol{R}$ in (17). Without loss of generality, we may assume that $\boldsymbol{r}_1$ is divergence free, and that its potential component is included in the pressure subscale $\tilde{p}$.

The Fourier transform defined above applied to (28) and (29), neglecting again the behavior of the subscales on the element boundary, yields

$$\left(\nu\frac{|\boldsymbol{k}|^2}{h^2} + \mathrm{i}\frac{\boldsymbol{a}\cdot\boldsymbol{k}}{h} \right)\hat{\tilde{\boldsymbol{u}}}(\boldsymbol{k}) + \mathrm{i}\frac{\boldsymbol{k}}{h}\hat{\tilde{p}}(\boldsymbol{k}) = \hat{\boldsymbol{r}}_1(\boldsymbol{k}), \tag{30}$$

$$\mathrm{i}\frac{\boldsymbol{k}}{h}\cdot\hat{\tilde{\boldsymbol{u}}}(\boldsymbol{k}) = \hat{r}_2(\boldsymbol{k}). \tag{31}$$

Multiplying (30) by $\mathrm{i}\boldsymbol{k}/h$ and using (31) and the fact that $\boldsymbol{r}_1$ is divergence free, we get the Fourier-transformed pressure Poisson equation

$$\left(\nu\frac{|\boldsymbol{k}|^2}{h^2} + \mathrm{i}\frac{\boldsymbol{a}\cdot\boldsymbol{k}}{h} \right)\hat{r}_2(\boldsymbol{k}) - \frac{|\boldsymbol{k}|^2}{h^2}\hat{\tilde{p}}(\boldsymbol{k}) = 0, \tag{32}$$

from which we may approximate the pressure using the same reasoning as for the convection–diffusion equation, that is, by

$$\tilde{p} \approx \tau_2 r_2, \quad \tau_2 = \left[\nu^2 + \left(\frac{c_2}{c_1} \frac{|\boldsymbol{a}|}{h} \right)^2 \right]^{1/2}, \tag{33}$$

where the constants c_1 and c_2 have the same interpretation as in (27).

Using the values of $\hat{\tilde{p}}(\boldsymbol{k})$ obtained from (32) in (30) it is found that the Fourier coefficients of the velocity are

$$\hat{\tilde{\boldsymbol{u}}}(\boldsymbol{k}) = \left(\nu \frac{|\boldsymbol{k}|^2}{h^2} + \mathrm{i} \frac{\boldsymbol{a} \cdot \boldsymbol{k}}{h} \right)^{-1} \hat{\boldsymbol{r}}_1(\boldsymbol{k}) - \mathrm{i} \frac{h}{|\boldsymbol{k}|^2} \boldsymbol{k} \hat{r}_2(\boldsymbol{k}).$$

It is seen that the second term only affects the component of $\hat{\tilde{\boldsymbol{u}}}(\boldsymbol{k})$ in the direction of $\boldsymbol{k}$. We will neglect the contribution of r_2 in $\tilde{\boldsymbol{u}}$. Apart from the fact that this will allow us to formulate a simpler method, this approximation implicitly assumes that the subscales are driven by the residual of the momentum equations $\boldsymbol{r}_1$ rather than by the error in satisfying the incompressibility constraint by the finite element solution. Moreover, from the analytical point of view we will need matrix $\boldsymbol{\tau}_K$ in (18) symmetric and positive-definite (see below) and since $\boldsymbol{r}_1$ does not appear in the expression for $\tilde{p}$, r_2 cannot appear in the expression for $\tilde{\boldsymbol{u}}$. Therefore, the approximation we suggest for $\tilde{\boldsymbol{u}}$ is

$$\tilde{\boldsymbol{u}} \approx \tau_1 \boldsymbol{r}_1, \quad \tau_1 = \left[\left(c_1 \frac{\nu}{h^2} \right)^2 + \left(c_2 \frac{|\boldsymbol{a}|}{h} \right)^2 \right]^{-1/2}. \tag{34}$$

Remark 1. The stabilization parameter in Eq. (27) for the convection–diffusion-reaction equation behaves asymptotically in h, ν, $|\boldsymbol{a}|$ and σ as

$$\tau \sim \left[c_1 \frac{\nu}{h^2} + \sigma + c_2 \frac{|\boldsymbol{a}|}{h} \right]^{-1}.$$

This expression was proposed in [34] using a completely different reasoning. Likewise, from Eqs. (33) and (34) we see that

$$\tau_2 = \frac{h^2}{c_1 \tau_1}.$$

This relationship between τ_1 and τ_2 was also found in [35] *based only on the convergence analysis* of the Oseen problem and using a stabilization technique similar to the Galerkin/least-squares (GLS) method, found by dropping $\boldsymbol{V}_{h,\mathrm{ort}}$ in (17). It is simpler than what is proposed for example in [29,36,32] and, moreover, can be justified from the previous heuristic reasoning.

2.4. Orthogonal subscales

The starting point of our developments have been the decompositions $\mathscr{W} = \mathscr{W}_h \oplus \tilde{\mathscr{W}}$ and $\mathscr{W}_0 = \mathscr{W}_{h,0} \oplus \tilde{\mathscr{W}}_0$. If $\cong$ denotes an isomorphism between two vector spaces, we have that $\tilde{\mathscr{W}} \cong \mathscr{W}_h^{\perp} \cap \mathscr{W}$ and $\mathscr{W}_0 \cong \mathscr{W}_{h,0}^{\perp} \cap \mathscr{W}_0$. Nevertheless, there are many possibilities to choose $\tilde{\mathscr{W}}$ and $\tilde{\mathscr{W}}_0$. *The particular one adopted in this work is to take precisely*

$$\tilde{\mathscr{W}} = \mathscr{W}_h^{\perp} \cap \mathscr{W}. \tag{35}$$

Note that $\mathcal{W}_h^{\perp}$ is not closed in $\mathcal{W}$, but it will be a closed subspace of the final approximating space.

To obtain a feasible numerical method we need to introduce some approximations. The first concerns the choice for $\tilde{\mathcal{W}}_0$. First, we assume that functions in $\tilde{\mathcal{W}}$ already vanish on $\partial\Omega$, and thus $\tilde{\mathcal{W}}_0 \approx \tilde{\mathcal{W}}$. Additionally we assume that $\mathcal{W}_h^{\perp} \cap \mathcal{W} \approx \mathcal{W}_h^{\perp}$, which can be thought of as a nonconforming approximation for the subscales. Altogether, this amounts to saying that

$$\tilde{\mathcal{W}}_0 \approx \tilde{\mathcal{W}} \approx \mathcal{W}_h^{\perp}. \tag{36}$$

With this approximation, it follows from (17) that

$$\boldsymbol{V}_{h,\mathrm{ort}} \in \tilde{\mathcal{W}}_0^{\perp} \approx \mathcal{W}_h, \tag{37}$$

$$\tilde{\boldsymbol{U}} \in \tilde{\mathcal{W}}_0 \approx \mathcal{W}_h^{\perp}, \tag{38}$$

which means that $\boldsymbol{V}_{h,\mathrm{ort}}$ is a finite element function and therefore *numerically computable*. We refer to this particular choice for the space of $\tilde{\boldsymbol{U}}$, motivated by the election (35) and the approximation (36), as the *space of orthogonal subscales*.

Imposing condition (38) in expression (18) for $\tilde{\boldsymbol{U}}$ we have that

$$(\tilde{\boldsymbol{U}}, \boldsymbol{V}_h) = \sum_K (\boldsymbol{\tau}_K[\boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h)], \boldsymbol{V}_h) + \sum_K (\boldsymbol{\tau}_K \boldsymbol{V}_{h,\mathrm{ort}}, \boldsymbol{V}_h) = 0 \quad \forall \boldsymbol{V}_h \in \mathcal{W}_h. \tag{39}$$

Let us assume that matrices $\boldsymbol{\tau}_K$ are all symmetric and positive-definite. From (39) it follows that $\boldsymbol{V}_{h,\mathrm{ort}}$ is the projection of the residual $\mathscr{L}(\boldsymbol{U}_h) - \boldsymbol{F}$ onto the finite element space with respect to the L^2 inner product weighted element by element by the matrices of algorithmic parameters $\boldsymbol{\tau}_K$. We denote this weighted inner product and its associated norm by

$$(\boldsymbol{X}, \boldsymbol{Y})_{\boldsymbol{\tau}} := \sum_K (\boldsymbol{\tau}_K, \boldsymbol{X}, \boldsymbol{Y})_K = \sum_K (\boldsymbol{X}, \boldsymbol{\tau}_K \boldsymbol{Y})_K, \tag{40}$$

$$\|\boldsymbol{Y}\|_{\boldsymbol{\tau}} := \sqrt{(\boldsymbol{Y}, \boldsymbol{Y})_{\boldsymbol{\tau}}}. \tag{41}$$

In these expressions, the functions $\boldsymbol{X}$ are $\boldsymbol{Y}$ need not being continuous for the local L^2 products to make sense.

Equation (39) now becomes

$$(\boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h), \boldsymbol{V}_h)_{\boldsymbol{\tau}} + (\boldsymbol{V}_{h,\mathrm{ort}}, \boldsymbol{V}_h)_{\boldsymbol{\tau}} = 0, \quad \forall \boldsymbol{V}_h \in \mathcal{W}_h. \tag{42}$$

If we call Π_{τ} the projection onto $\mathcal{W}_h$, associated to the inner product (40), hereafter referred to as τ-*projection*, we see that

$$\boldsymbol{V}_{h,\mathrm{ort}} = -\Pi_{\tau}[\boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h)], \tag{43}$$

Likewise, we will denote by $\Pi_{\tau,0}$ the τ-*projection* onto $\mathcal{W}_{h,0}$ and $\Pi_{\tau}^{\perp} := I - \Pi_{\tau}$, where I is the identity in $\mathcal{W}_h$.

From (18) and (43) it follows that

$$\tilde{\boldsymbol{U}} = \boldsymbol{\tau}_K \Pi_r^{\perp}[\boldsymbol{F} - \mathscr{L}(\boldsymbol{U}_h)] \quad \text{in } K \in \mathscr{P}_h. \tag{44}$$

If this expression is now introduced in (14) and, as already mentioned, the integrals over the interelement boundaries are neglected, we finally obtain the modified discrete problem: find $\boldsymbol{U}_h \in \mathcal{W}_{h,0}$ such that

$$B_h(\boldsymbol{U}_h, \boldsymbol{V}_h) = \langle \boldsymbol{F}, \boldsymbol{V}_h \rangle - (\Pi_{\tau}^{\perp}(\boldsymbol{F}), \mathscr{L}^*(\boldsymbol{V}_h))_{\tau}, \quad \forall \boldsymbol{V}_h \in \mathcal{W}_{h,0}. \tag{45}$$

where the stabilized bilinear form B_h is

$$B_h(\boldsymbol{U}_h, \boldsymbol{V}_h) = B(\boldsymbol{U}_h, \boldsymbol{V}_h) - (\Pi_{\tau}^{\perp}[(\mathscr{L}(\boldsymbol{U}_h))], \mathscr{L}^*(\boldsymbol{V}_h))_{\tau}. \tag{46}$$

The hope is that the stability properties of (45) are much better than those of the original discrete problem (10).

Remark 2. Eq. (44), together with (37) and (38), indirectly determine the approximation to the space $\mathscr{W}$ in which the discrete solution is sought. This space is $\mathscr{W}_h$ enlarged with piecewise discontinuous functions generated by functions in $\mathscr{W}_h$ as indicated by (44). We could have started the developments by identifying $\mathscr{W}$ with this finite dimensional vector space, which in this case would be an approximation to the space of the continuous problem.

The previous developments are applicable to any linear system of convection–diffusion-reaction equations. Let us apply these ideas to the particular case of the Oseen equations. The adjoint of $\mathscr{L}$ is now given by (16) and the matrix of stabilization parameters by

$$\boldsymbol{\tau}_K = \mathrm{diag}(\boldsymbol{\tau}_{1,K}, \tau_{2,K}), \qquad \boldsymbol{\tau}_{1,K} = \tau_{1,K}\boldsymbol{I}_d, \tag{47}$$

where $\boldsymbol{I}_d$ is the $d \times d$ identity matrix and with $\tau_{1,K}$ and $\tau_{2,K}$ computed elementwise as indicated by (34) and (33), respectively, and replacing the Euclidian norm of $\boldsymbol{a}$ by $|\boldsymbol{a}|_{\infty,K}$, the maximum of the Euclidian norm of $\boldsymbol{a}$ in the element domain K. Matrix $\boldsymbol{\tau}_K$ defined in (47) is symmetric and positive-definite, a requirement needed for (40) to be an inner product.

We will introduce further simplifying assumptions that will lead to a method easy to implement and with good stability properties. These are:

- The weighted projection Π_τ associated to the inner product defined in (40) will be approximated by the L^2 projection, denoted by Π. Likewise, $\Pi_\tau^\perp$ will be approximated by $\Pi^\perp = I - \Pi$. The difference between Π_r and Π depends on the variation of the stabilization parameters from element to element. From the computational point of view, it is very convenient to use Π, since L^2 projections can be computed very efficiently.
- $\tau_{1,K}\Pi^\perp(\boldsymbol{f}\,) = 0$, which means that the force vector belongs to the finite element space $\mathscr{W}_h$ or it is approximated by an element of this space. In any case, the term $\tau_{1,K}\Pi^\perp(\boldsymbol{f}\,)$ is of the same order as the optimal error that can be expected. Taking for example $\boldsymbol{a} = \boldsymbol{0}$, for $\boldsymbol{f}\, \in \boldsymbol{H}^m(\Omega)$, $m = -1, 0, \ldots$, we may expect $\boldsymbol{u} \in \boldsymbol{H}^{m+2}(\Omega)$, $p \in H^{m+1}(\Omega)$ and an L^2 velocity error of order $\mathscr{O}(h^r)$, with $r = \min\{m+2, p+1\}$ and p the order of the finite element interpolation, and this is precisely the order of $\tau_{1,K}\Pi^\perp(\boldsymbol{f}\,)$.
- Second order derivatives of finite element functions within element interiors will be neglected. They are exactly zero for linear elements and for higher order interpolations disregarding them leads to a method which is still consistent (in a sense explained later; cf. Remark 4).

Under these conditions, the second term in the RHS of (45) vanishes and the stabilized bilinear form (46) reduces to

$$B_I(\boldsymbol{U}_h, \boldsymbol{V}_h) = B(\boldsymbol{U}_h, \boldsymbol{V}_h) + (\Pi^\perp(\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h + \nabla p_h), \boldsymbol{a}\cdot\nabla\boldsymbol{v}_h + \nabla q_h)_{\tau_1} + (\Pi^\perp(\nabla\cdot\boldsymbol{u}_h), \nabla\cdot\boldsymbol{v}_h)_{\tau_2}, \tag{48}$$

where B is defined in (8).

Once arrived to (48) it is observed that what the present method provides with respect to the standard Galerkin method is a least-squares control on the component of the terms $\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h + \nabla p_h$ and $\nabla\cdot\boldsymbol{u}_h$ orthogonal to the corresponding finite element spaces.

There is a simple modification of the bilinear form (48) which leads to another stabilized method with slightly better stability properties. The idea is to control separately the components of $\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h$ and ∇p_h orthogonal to $\mathscr{V}_h$. The bilinear form associated to this method is

$$B_{II}(\boldsymbol{U}_h, \boldsymbol{V}_h) = B(\boldsymbol{U}_h, \boldsymbol{V}_h) + (\Pi^{\perp}(\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h), \boldsymbol{a}\cdot\nabla\boldsymbol{v}_h)_{\tau_1} + (\Pi^{\perp}(\nabla p_h), \nabla q_h)_{\tau_1} + (\Pi^{\perp}(\nabla\cdot\boldsymbol{u}_h), \nabla\cdot\boldsymbol{v}_h)_{\tau_2}. \quad (49)$$

Dropping the orthogonal projection $\Pi^{\perp}$ the method reduces to a general version of that analyzed in [27], which has a consistency error that makes it only applicable with P_1 elements.

Remark 3. Both methods I and II could be slightly modified by projecting onto $\mathscr{W}_{h,0}$ in (42) instead of projecting onto $\mathscr{W}_h$. However, even though the global convergence is optimal, projecting onto $\mathscr{W}_{h,0}$ leads to spurious numerical boundary layers, similar to those found for the pressure in classical fractional step schemes for the transient problem (see for example [37]). Further discussion about this point can be found in [10].

Remark 4. There is a way to formulate the present method in a manner that it can be viewed as *consistent*. Indeed, if we introduce

$$B_I^*([\boldsymbol{u}_h, p_h, \boldsymbol{\xi}_h, \delta_h], [\boldsymbol{v}_h, q_h, \boldsymbol{\eta}_h, \gamma_h]) := B([\boldsymbol{u}_h, p_h], [\boldsymbol{v}_h, q_h]) + (\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h + \nabla p_h - \boldsymbol{\xi}_h, \boldsymbol{a}\cdot\nabla\boldsymbol{v}_h + \nabla q_h - \boldsymbol{\eta}_h)_{\tau_1} + (\nabla\cdot\boldsymbol{u}_h - \delta_h, \nabla\cdot\boldsymbol{v}_h - \gamma_h)_{\tau_{r_2}}$$

the discrete problem is equivalent to find $[\boldsymbol{u}_h, p_h, \boldsymbol{\xi}_h, \delta_h] \in \mathscr{V}_{h,0}\times\mathscr{Q}_{h,0}\times\mathscr{V}_h\times\mathscr{Q}_h$ such that $B_I^*([\boldsymbol{u}_h, p_h, \boldsymbol{\xi}_h, \delta_h], [\boldsymbol{v}_h, q_h, \boldsymbol{\eta}_h, \gamma_h]) = \langle \boldsymbol{f}, \boldsymbol{v}_h\rangle$ for all $[\boldsymbol{v}_h, q_h, \eta_h, \gamma_h] \in \mathscr{V}_{h,0}\times\mathscr{Q}_{h,0}\times\mathscr{V}_h\times\mathscr{Q}_h$. This problem is consistent in the sense that, for smooth enough solutions $[\boldsymbol{u}, p]$ of the continuous problem, $B_I^*([\boldsymbol{u}, p, \boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p, \nabla\cdot\boldsymbol{u}], [\boldsymbol{v}_h, q_h, \boldsymbol{\eta}_h, \gamma_h]) = \langle \boldsymbol{f}, \boldsymbol{v}_h\rangle$.

3. Transient Oseen equations

3.1. Discretization in time

Let us consider now the transient Oseen problem, that is,

$$\partial_t\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \boldsymbol{a}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \quad t\in]0,T[, \quad (50)$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \quad t\in]0,T[, \quad (51)$$

supplied with an initial condition and the homogeneous Dirichlet condition for the velocity. In order to be able to use the same notation as in the previous section, we need to introduce the matrix

$$\boldsymbol{M} = \mathrm{diag}(\boldsymbol{I}_d, 0),$$

which allows us to write Eqs. (50) and (51) as

$$\boldsymbol{M}\partial_t\boldsymbol{U} + \mathscr{L}(\boldsymbol{U}) = \boldsymbol{F}, \quad (52)$$

where the notation involved is the same as before. For the sake of simplicity, we will consider throughout that $\boldsymbol{F}$ is time-independent.

Problem (52) needs to be approximated both in space and in time. For the time discretization we will consider here the simple trapezoidal rule, although the ideas to be developed can be equally applied to any other finite difference time integration scheme. If time is also discretized using finite elements, the methodology would be the straightforward extension of what has been developed for the stationary problem. However, the goal is to analyze how does the time discretization affect the stabilization method when using finite differences.

Let us consider a uniform partition of the time interval of analysis $[0,T]$ with time step size δt. We will denote by a superscript the time step level at which the algorithmic solution is computed. If $\theta \in [0,1]$ and $\boldsymbol{U}^n$ is known, the trapezoidal rule applied to the variational form of Eq. (52) consists of finding $\boldsymbol{U}^{n+1}$ as the solution of the problem

$$(\boldsymbol{M}\delta_t \boldsymbol{U}^n, \boldsymbol{V}) + B(\boldsymbol{U}^{n+\theta}, \boldsymbol{V}) = L(\boldsymbol{V}) \quad \forall \boldsymbol{V} \in \mathscr{W}_0. \tag{53}$$

Here and in what follows, we use the notation

$$f^{n+\theta} := \theta f^{n+1} + (1-\theta) f^n, \quad \delta_t f^n := \frac{1}{\delta t}(f^{n+1} - f^n)$$

for any function f.

3.2. Subgrid scale decomposition and modeling of the subscales

From the semidiscrete problem (53) we can now obtain the fully discrete formulation applying the same ideas as for the stationary case. We start by considering the same decompositions $\mathscr{W} = \mathscr{W}_h \oplus \tilde{\mathscr{W}}$ and $\mathscr{W}_0 = \mathscr{W}_{h,0} \oplus \tilde{\mathscr{W}}_0$, which allow us to split (53) into two equations, the first of which is

$$(\boldsymbol{M}\delta_t \boldsymbol{U}_h^n, \boldsymbol{V}_h) + (\boldsymbol{M}\delta_t \tilde{\boldsymbol{U}}^n, \boldsymbol{V}_h) + B(\boldsymbol{U}_h^{n+\theta}, \boldsymbol{V}_h) + \sum_K \int_K \tilde{\boldsymbol{U}}^{n+\theta} \cdot \mathscr{L}^*(\boldsymbol{V}_h)\, \mathrm{d}\Omega = L(\boldsymbol{V}_h) \quad \forall \boldsymbol{V}_h \in \mathscr{W}_{h,0}, \tag{54}$$

which corresponds to (14) of the stationary problem. Observe that we have already neglected the contribution from the integrals over the element boundaries. Note also that there is a contribution from the transient evolution of the subscales. These are solution of

$$\begin{aligned} &\boldsymbol{M}\delta_t \tilde{\boldsymbol{U}}^n + \mathscr{L}(\tilde{\boldsymbol{U}}^{n+\theta}) = \boldsymbol{R}_t^{n+\theta} \quad \text{in } K \in \mathscr{P}_h, \\ &\boldsymbol{R}_t^{n+\theta} := \boldsymbol{F} - [\boldsymbol{M}\delta_t \boldsymbol{U}_h^n + \mathscr{L}(\boldsymbol{U}_h^{n+\theta})] + \boldsymbol{V}_{h,\mathrm{ort}}, \end{aligned} \tag{55}$$

which is the counterpart of (17) for the stationary case. Again, $\boldsymbol{V}_{h,\mathrm{ort}}$ is an element in $\tilde{\mathscr{W}}_0^{\perp}$. In what follows, it is understood that the subscales are computed within each element K of the finite element partition $\mathscr{P}_h$.

We can equivalently write (55) as

$$\left(\boldsymbol{M}\frac{1}{\theta\delta t} + \mathscr{L}\right)\tilde{\boldsymbol{U}}^{n+\theta} = \boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\boldsymbol{U}}^n + \boldsymbol{R}_t^{n+\theta}, \tag{56}$$

from where a closed-form expression for $\tilde{\boldsymbol{U}}$ has to be proposed. It is the same *modeling* step as for the stationary case. There, the operator $\mathscr{L}$ was replaced by matrix $\boldsymbol{\tau}^{-1}$, which was designed on the grounds that both the exact and the modeled subscales had approximately the same L^2 norm over each element. To be consistent with the approximations made in the previous section, let us introduce the matrix

$$\boldsymbol{\tau}_t := \left(\boldsymbol{M}\frac{1}{\theta\delta t} + \boldsymbol{\tau}^{-1}\right)^{-1} = \mathrm{diag}(\tau_{1,t}\boldsymbol{I}_d, \tau_2), \tag{57}$$

$$\tau_{1,t} := \left(\frac{1}{\theta\delta t} + \frac{1}{\tau_1}\right)^{-1}.$$

The modeling of (56) proposed is

$$\tilde{\boldsymbol{U}}^{n+\theta} = \boldsymbol{\tau}_t \boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\boldsymbol{U}}^n + \boldsymbol{\tau}_t \boldsymbol{R}_t^{n+\theta}. \tag{58}$$

Once $\tilde{\boldsymbol{U}}^{n+\theta}$ is computed we can obtain $\tilde{\boldsymbol{U}}^{n+1}$.

At this point we can impose that the subscales be orthogonal to the finite element space. This determines the function $\boldsymbol{V}_{h,\mathrm{ort}}$ in $\tilde{\mathscr{W}}_0^{\perp}$. The counterparts of Eqs. (43) and (44) are

$$\boldsymbol{V}_{h,\mathrm{ort}} = -\Pi\big[\boldsymbol{F} - \big(\boldsymbol{M}\delta_t \boldsymbol{U}_h^n + \mathscr{L}\big(\boldsymbol{U}_h^{n+\theta}\big)\big)\big], \tag{59}$$

$$\tilde{\boldsymbol{U}}^{n+\theta} = \boldsymbol{\tau}_t \boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\boldsymbol{U}}^n + \boldsymbol{\tau}_t \Pi^{\perp}\big[\boldsymbol{F} - \big(\boldsymbol{M}\delta_t \boldsymbol{U}_h^n + \mathscr{L}\big(\boldsymbol{U}_h^{n+\theta}\big)\big)\big]. \tag{60}$$

Expression (60) is what we were looking for.

3.3. Stabilized finite element problem

The previous development is general and applicable to any system of convection–diffusion-reaction equations. Let us specialize it to the transient Oseen problem using the same approximations as for the stationary case. First, observe that

$$\Pi^{\perp}(\boldsymbol{F}) = \boldsymbol{0}, \quad \text{same approximation as for the stationary case,} \tag{61}$$

$$\Pi^{\perp}(\boldsymbol{M}\delta_t \boldsymbol{U}_h^n) = \boldsymbol{0}, \quad \text{since } \boldsymbol{M}\delta_t \boldsymbol{U}_h^n \text{ is a finite element function,} \tag{62}$$

$$(\boldsymbol{M}\delta_t \tilde{\boldsymbol{U}}^n, \boldsymbol{V}_h) = 0, \quad \text{since } \boldsymbol{M}\delta_t \tilde{\boldsymbol{U}}^n \text{ is orthogonal to } \mathscr{W}_{h,0}. \tag{63}$$

Using (61) and (62) in (60), taking into account expression (57) for $\boldsymbol{\tau}_t$ and neglecting the orthogonal projection of second derivatives as we did for the stationary case, we obtain the expression for the velocity and pressure subscales

$$\tilde{\boldsymbol{u}}^{n+\theta} = \tau_{1,t}\frac{1}{\theta\delta t}\tilde{\boldsymbol{u}}^n - \tau_{1,t}\Pi^{\perp}(\boldsymbol{a}\cdot\nabla \boldsymbol{u}_h^{n+\theta} + \nabla p_h^{n+\theta}), \tag{64}$$

$$\tilde{p}^{n+\theta} = -\tau_2 \Pi^{\perp}(\nabla\cdot \boldsymbol{u}_h^{n+\theta}), \tag{65}$$

which inserted into the equation for the finite element solution (54) and noting (63) yields

$$\begin{aligned}&(\delta_t \boldsymbol{u}_h^n, \boldsymbol{v}_h) + B\big(\boldsymbol{U}_h^{n+\theta}, \boldsymbol{V}_h\big) + \big(\Pi^{\perp}\big(\boldsymbol{a}\cdot\nabla \boldsymbol{u}_h^{n+\theta} + \nabla p_h^{n+\theta}\big), \boldsymbol{a}\cdot\nabla \boldsymbol{v}_h + \nabla q_h\big)_{\tau_{1,t}} + \big(\Pi^{\perp}\big(\nabla\cdot \boldsymbol{u}_h^{n+\theta}\big), \nabla\cdot \boldsymbol{v}_h\big)_{\tau_2}\\ &\quad = L(\boldsymbol{V}_h) + \frac{1}{\theta\delta t}(\tilde{\boldsymbol{u}}^n, \boldsymbol{a}\cdot\nabla \boldsymbol{v}_h + \nabla q_h)_{\tau_{1,t}}.\end{aligned} \tag{66}$$

This is the transient version of problem defined by the bilinear form B_I defined in (48). The main differences of the stabilizing terms of this transient problem with respect to the stationary one are

- The stabilization parameter τ_1 is replaced by $\tau_{1,t}$ (see (57)).
- There is a RHS contribution that comes from the fact that subscales need to be tracked in time.

This fact gives an answer to the question of how does the finite difference time integration affect the stabilization. First, we see that the stability parameter $\tau_{1,t}$ is certainly affected by δt. Expressions similar to (57) have been proposed for example in [38]. However, if no RHS modification is introduced, the steady-state solution would depend on the magnitude of δt and the method would lack stability for $\delta t \to 0$. On the other hand, if the present approach is used, it is clear that the stabilization terms tend to those of the stationary problem as the steady solution is reached. This can be seen for example from (58): If $\tilde{\boldsymbol{U}}^{n+\theta} = \tilde{\boldsymbol{U}}^n$ it is easily checked that $\tilde{\boldsymbol{U}}^{n+\theta} = \boldsymbol{\tau}\boldsymbol{R}_t^{n+\theta}$. In fact, using this assumption is a possible alternative. We could assume that the subscales do not change in time, and thus that $\tilde{\boldsymbol{U}}^{n+\theta} = \tilde{\boldsymbol{U}}^n$ always. This would lead to the

same stabilization terms as for the stationary problem. Since the basic assumption is that the temporal variation of the subscales is negligible, we call them *quasi-static subscales*.

It is interesting to note that in [20] the authors analyze a nonlinear Galerkin approximation of the Navier–Stokes equations in which the temporal derivatives of the subgrid scales are also neglected. Furthermore, the space of velocity subgrid scales used in this reference is also orthogonal to the space of large scales for the velocity (pressures are not split), although in this case both are finite element spaces.

3.4. Tracking of subscales

For quasi-static subscales the second term in the RHS of (66) disappears and there is no need to store $\tilde{\boldsymbol{u}}^n$. However, when this approximation is not used, subscales need to be tracked. Let us describe how this can be done.

Let us consider a standard Lagrangian finite element interpolation and expand $\boldsymbol{U}_h^n$ as

$$\boldsymbol{U}_h^n(\boldsymbol{x}) = \sum_{a=1}^{n_{\mathrm{p}}} \mathsf{N}_a(\boldsymbol{x})\mathsf{U}_a^n, \tag{67}$$

where n_{p} is the total number of nodes of the finite element mesh, U_a^n is the nodal value of $\boldsymbol{U}_h^n(\boldsymbol{x})$ at node a and $\mathsf{N}_a(\boldsymbol{x}) = N_a(\boldsymbol{x})\boldsymbol{I}_{d+1}$, with $N_a(\boldsymbol{x})$ the shape (basis) function of node a and $\boldsymbol{I}_{d+1}$ the $(d+1)\times(d+1)$ identity matrix.

Using expression (60), (61) and (62), Eq. (67) allows us to expand the subscales as

$$\begin{aligned}\tilde{\boldsymbol{U}}^{n+\theta} &= \boldsymbol{\tau}_t\boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\boldsymbol{U}}^n + \boldsymbol{\tau}_t\Pi^{\perp}[\boldsymbol{F} - (\boldsymbol{M}\delta_t\boldsymbol{U}_h^n + \mathscr{L}(\boldsymbol{U}_h^{n+\theta}))] = \boldsymbol{\tau}_t\boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\boldsymbol{U}}^n - \boldsymbol{\tau}_t\Pi^{\perp}[\mathscr{L}(\boldsymbol{U}_h^{n+\theta})]\\ &= \boldsymbol{\tau}_t\boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\boldsymbol{U}}^n - \sum_{a=1}^{n_{\mathrm{p}}}[\boldsymbol{\tau}_t\Pi^{\perp}(\mathscr{L}(\mathsf{N}_a))]\mathsf{U}_a^{n+\theta}\end{aligned} \tag{68}$$

within each element K. Note, however, that $\mathsf{U}_a^{n+\theta}$ does not depend on K. We thus conclude that the subscales can be written in terms of a set of "nodal values" $\tilde{\mathsf{U}}_{a,K}$, $a = 1,\ldots,n_p$, $K \in \mathscr{P}_h$, as

$$\tilde{\boldsymbol{U}}^{n+\theta} = \sum_{a=1}^{n_p}\left[\boldsymbol{\tau}_t\Pi^{\perp}(\mathscr{L}(\mathsf{N}_a))\right]\tilde{\mathsf{U}}_{a,K}^{n+\theta}, \quad \text{in } K \in \mathscr{P}_h, \tag{69}$$

where, according to (68),

$$\tilde{\mathsf{U}}_{a,K}^{n+\theta} = \boldsymbol{\tau}_t\boldsymbol{M}\frac{1}{\theta\delta t}\tilde{\mathsf{U}}_{a,K}^n - \mathsf{U}_a^{n+\theta}. \tag{70}$$

Note that, in general, the set

$$\{\boldsymbol{\tau}_t\Pi^{\perp}(\mathscr{L}(\mathsf{N}_a))|_K, \quad a = 1,\ldots,n_p,\ K \in \mathscr{P}_h\}$$

is not a basis for the space of subscales, but, according to (69), it spans this space. Obviously, it is understood that the coefficient $\tilde{\mathsf{U}}_{a,K}$ is zero when node a does not belong to element K. It is also interesting to remark that if $\boldsymbol{\tau}_t$ were *the same* for all the elements, $\tilde{\mathsf{U}}_{a,K}$ would be the same for all the elements K sharing node a, and thus the number of "nodal values" needed to track $\tilde{\boldsymbol{U}}$ would be n_{p}.

For the particular case of the Oseen problem that we are considering, the equation to update the subscales (68) can be explicitly written as

$$\begin{bmatrix}\tilde{\boldsymbol{u}}^{n+\theta}\\ \tilde{p}^{n+\theta}\end{bmatrix} = \begin{bmatrix}\frac{\tau_{1,t}}{\theta\delta t}\tilde{\boldsymbol{u}}^n\\ 0\end{bmatrix} - \sum_{a=1}^{n_{\mathrm{p}}}\begin{bmatrix}\tau_{1,t}\Pi^{\perp}(\boldsymbol{a}\cdot\nabla\mathsf{N}_a) & \tau_{1,t}\Pi^{\perp}(\nabla N_a)\\ \tau_2\Pi^{\perp}(\nabla\cdot\mathsf{N}_a) & 0\end{bmatrix}\begin{bmatrix}\mathsf{u}_a^{n+\theta}\\ \mathsf{p}_a^{n+\theta}\end{bmatrix},$$

where u_a and p_a, $a = 1,\ldots,n_{\mathrm{p}}$, are the velocity and pressure nodal values, respectively, and now $\mathsf{N}_a = N_a\boldsymbol{I}_d$.

This approach to track the subscales in completely general and can be used in any finite element implementation. However, in the most common element based implementation another approach is possible. In this case, the contributions to the global discrete variational equation are computed element by element using numerical integration, and thus the subscales are only needed at the integration points. Instead of using the "nodal values" given by (70), one can simply evaluate (68) at the integration points and store the results for use in the next time step.

3.5. Explicit approximation of the subscales

Even if the finite element unknown $\boldsymbol{u}_h$ is treated implicitly in time, one could consider the possibility of solving explicitly the subscale $\tilde{\boldsymbol{u}}$. Of course, the pressure subscale $\tilde{p}$ needs to be treated implicitly. The problem for $\tilde{\boldsymbol{U}}^{n+1} = [\tilde{\boldsymbol{u}}^{n+1}, \tilde{p}^{n+1}]$ to be solved instead of (55) is

$$\delta_t \tilde{\boldsymbol{u}}^n - \nu \Delta \tilde{\boldsymbol{u}}^n + \boldsymbol{a} \cdot \nabla \tilde{\boldsymbol{u}}^n + \nabla \tilde{p}^{n+1} = \boldsymbol{r}_{1,t}^{n+\theta},$$
$$\nabla \cdot \tilde{\boldsymbol{u}}^{n+1} = r_2^{n+1},$$

where $\boldsymbol{r}_{1,t}$ and r_2 are the components of $\boldsymbol{R}_t$ in (55). Following the same procedure as before, the expressions found for $\tilde{\boldsymbol{u}}^{n+1}$ and $\tilde{p}^{n+1}$ are

$$\tilde{\boldsymbol{u}}^{n+1} = \left(1 - \frac{\delta t}{\tau_1}\right)\tilde{\boldsymbol{u}}^n - \delta t \Pi^{\perp}(\boldsymbol{a} \cdot \nabla \boldsymbol{u}_h^{n+\theta} + \nabla p_h^{n+\theta}), \tag{71}$$

$$\tilde{p}^{n+1} = -\tau_2 \Pi^{\perp}(\nabla \cdot \boldsymbol{u}^{n+1}).$$

For a fixed τ_1 we have that

$$\tau_{1,t} = \delta t + \mathcal{O}(\delta t^2), \quad \frac{\tau_{1,t}}{\delta t} = 1 - \frac{\delta t}{\tau_1} + \mathcal{O}(\delta t^2),$$

and thus (64) and (71) differ from a term of order $\mathcal{O}(\delta t^2)$. The expression for the pressure subscale has not changed from (65).

Clearly, not much is gained by treating explicitly $\tilde{\boldsymbol{u}}$. However, an important computational gain is obtained if the orthogonal projection in (71) is computed as $\Pi^{\perp} = I - \Pi$, with the identity applied in time step $n + \theta$ and Π in time step n, that is, (64) is replaced by

$$\tilde{\boldsymbol{u}}^{n+1} = \frac{\tau_{1,t}}{\theta \delta t}\tilde{\boldsymbol{u}}^n - \tau_{1,t}[(\boldsymbol{a} \cdot \nabla \boldsymbol{u}_h^{n+\theta} + \nabla p_h^{n+\theta}) - \Pi(\boldsymbol{a} \cdot \nabla \boldsymbol{u}_h^n + \nabla p_h^n)].$$

This and other implementation aspects of the formulation presented here are similar to those of the pressure stabilization technique described in [39].

4. Extension to the Navier–Stokes problem

4.1. Temporal discretization and linearization

Before discretizing in space the incompressible Navier–Stokes equations (1) and (2), let us consider the time discretization using the generalized trapezoidal rule, as for the Oseen equations in Section 3. Using the same notation as above, at each time step the problem to be solved is

$$\delta_t \boldsymbol{u}^n + \boldsymbol{u}^{n+\theta} \cdot \nabla \boldsymbol{u}^{n+\theta} - \nu \Delta \boldsymbol{u}^{n+\theta} + \nabla p^{n+1} = \boldsymbol{f}, \tag{72}$$

$$\nabla \cdot \boldsymbol{u}^{n+\theta} = 0. \tag{73}$$

Observe that the pressure computed in (72) has been considered evaluated at time level $n + 1$ (this simplifies the final algorithm of the following section).

This problem is nonlinear. Before going to the finite element discretization we can linearize it. Again, several options are possible, but now we will restrict ourselves to the simple fixed point (or Picard) algorithm, which leads to an Oseen problem within each iteration step. Denoting by $f^{n,i}$ the ith iteration of the unknown f at time level n, the linearized form of problem (72) and (73) is

$$\delta_t \boldsymbol{u}^{n,i} + \boldsymbol{u}^{n+\theta,i-1} \cdot \nabla \boldsymbol{u}^{n+\theta,i} - \nu \Delta \boldsymbol{u}^{n+\theta,i} + \nabla p^{n+1,i} = \boldsymbol{f}, \tag{74}$$

$$\nabla \cdot \boldsymbol{u}^{n+\theta,i} = 0. \tag{75}$$

This is the linear system to which we apply the previous developments.

4.2. Final algorithm

Problem (74) and (75) is clearly an Oseen problem for the velocity $\boldsymbol{u}^{n+\theta,i}$, being the advection velocity given by $\boldsymbol{a} \equiv \boldsymbol{u}^{n+\theta,i-1}$. Hence, we can apply the formulation developed in the previous section in a straightforward manner. However, there is an important remark to be made. When the unknown velocity is split into its finite element component and the subscale, this decomposition also affects the advection velocity $\boldsymbol{a}$, that is to say, we will have

$$\boldsymbol{a} \equiv \boldsymbol{u}_h^{n+\theta,i-1} + \tilde{\boldsymbol{u}}^{n+\theta,i-1}.$$

This implies that the velocity subscale not only need to be tracked in time, but also along the iterative process. Of course, expression (64) is still valid, and it will allow us to carry out this tracking.

The final algorithm is written in Box 1. There are three points that we would like to emphasize of this formulation, two of which are common with that of the transient Oseen equations:

- The velocity subscales appear also in the advection velocity $\boldsymbol{a}$. It is not $\boldsymbol{u}_h^{n+\theta,i-1}$, as for the standard Galerkin method, but $\boldsymbol{u}_h^{n+\theta,i-1} + \tilde{\boldsymbol{u}}^{n+\theta,i-1}$.
- The stabilization parameter $\tau_{1,t}$ depends on the time step size δt.
- The velocity subscales appear both evaluated in the previous time step (the term coming from the approximation to their temporal derivative) and in the previous iteration (in $\boldsymbol{a}$). Thus, they have to be stored at each iteration and at each time step.

Remark 5. Apart from the particular time integration scheme and linearization method adopted in Box 1, the main feature of this algorithm is that the projections onto the finite element space have been treated iteratively using the same iterative loop as for the fixed point linearization of the convective term. As mentioned in the previous section, it is also possible to treat these projections explicitly (that is, computed with values of the previous time step), or even with a nested iteration within each linearization step. In any case, we have observed from numerical experiments that it is important to treat the advective velocity implicitly. If $\theta = 1$, one could perform a single iteration within each time step while keeping formal first order accuracy. However, we have noticed that this may lead to instability problems. For example, when stationary solutions are searched, the convergence towards the steady state can be nonmonotone.

Remark 6. In [13] the authors present also a multiscale decomposition of the incompressible Navier–Stokes equations, but with a different goal than ours. Rather than obtaining a *stabilized numerical method*, they intend to develop a *physical model*. However, the underlying concepts are similar. The main difference is that instead of the closed-form expression for the subscales (64) (in fact its counterpart for the Navier–Stokes equations) they model these subscales by adding a Smagorinsky-type eddy viscosity. Nevertheless, our approach has an inherent modeling of the classical terms for which a closure equation has to be given in turbulence, such as the Reynolds stress and the Cross stress [13]. How does our *numerical method behave as physical model* is a point that needs further research.

5. Numerical examples

In this section we present three simple numerical examples. The first is an example of the behavior of the formulation for the stationary Oseen equations, whereas the other two are transient incompressible flows computed with the formulation of Box 1. These examples are the classical cavity flow problem and the flow over a cylinder. For much more complex applications (using quasi-static subscales) the reader is referred to [40].

5.1. Stabilized formulation for comparison

The formulation presented in this paper will be compared with the algebraic subgrid scale (ASGS) method, as presented in [35], which is similar to the GLS method [29,32,36]. This formulation leads to

Box 1. Algorithm for solving the Navier–Stokes equations

- Read (or compute) $\boldsymbol{u}_h^0$ and set $p_h^0 = 0$, $\tilde{\boldsymbol{u}}^0 = \boldsymbol{0}$.

FOR $n = 0, \ldots, N-1$ DO:

- Set $i = 0$
- Set $\boldsymbol{u}_h^{n+\theta,0} = \boldsymbol{u}_h^n$, $p_h^{n+1,0} = p_h^n$, $\tilde{\boldsymbol{u}}^{n+\theta,0} = \tilde{\boldsymbol{u}}^n$.

WHILE (not converged) DO:

- $i \leftarrow i + 1$
- Set $\boldsymbol{a} = \boldsymbol{u}_h^{n+\theta,i-1} + \tilde{\boldsymbol{u}}^{n+\theta,i-1}$
- Compute $\tau_{1,t}$ and τ_2 from (34), (33) and (57)
- Compute the projections

$$\boldsymbol{\xi}_h = \Pi\left(\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h^{n+\theta,i-1} + \nabla p_h^{n+1,i-1}\right)$$

$$\delta_h = \Pi\left(\nabla\cdot\boldsymbol{u}_h^{n+\theta,i-1}\right)$$

- Compute $\boldsymbol{u}_h^{n+\theta,i}$ and $p_h^{n+1,i}$ by solving

$$\begin{aligned}&\left(\delta_t\boldsymbol{u}_h^{n,i},\boldsymbol{v}_h\right) + \left(\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h^{n+\theta,i},\boldsymbol{v}_h\right) + \nu\left(\nabla\boldsymbol{u}_h^{n+\theta,i},\nabla\boldsymbol{v}_h\right) - \left(p_h^{n+1,i},\nabla\cdot\boldsymbol{v}_h\right) + \left(q_h,\nabla\cdot\boldsymbol{u}_h^{n+\theta,i}\right)\\ &\quad + \left(\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h^{n+\theta,i} + \nabla p_h^{n+1,i},\boldsymbol{a}\cdot\nabla\boldsymbol{v}_h + \nabla q_h\right)_{\tau_{1,t}} + \left(\nabla\cdot\boldsymbol{u}_h^{n+\theta},\nabla\cdot\boldsymbol{v}_h\right)_{\tau_2}\\ &= \langle\boldsymbol{f},\boldsymbol{v}_h\rangle + \frac{1}{\theta\delta t}\left(\tilde{\boldsymbol{u}}^n,\boldsymbol{a}\cdot\nabla\boldsymbol{v}_h + \nabla q_h\right)_{\tau_{1,t}} + \left(\boldsymbol{\xi}_h,\boldsymbol{a}\cdot\nabla\boldsymbol{v}_h + \nabla q_h\right)_{\tau_{1,t}} + \left(\delta_h,\nabla\cdot\boldsymbol{v}_h\right)_{\tau_2}\end{aligned}$$

- Update the subscales

$$\tilde{\boldsymbol{u}}^{n+\theta,i} = \tau_{1,t}\frac{1}{\theta\delta t}\tilde{\boldsymbol{u}}^n - \tau_{1,t}\Pi^{\perp}\left(\boldsymbol{a}\cdot\nabla\boldsymbol{u}_h^{n+\theta,i} + \nabla p_h^{n+1,i}\right)$$

- Check convergence

END

- Set up converged values

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

END

the discrete variational problem

$$
\begin{aligned}
&(\delta_t \boldsymbol{u}_l^{n,i}, \boldsymbol{v}_h) + (\boldsymbol{u}_h^{n+\theta,i-1} \cdot \nabla \boldsymbol{u}_h^{n+\theta,i}, \boldsymbol{v}_h) + \nu(\nabla \boldsymbol{u}_h^{n+\theta,i}, \nabla \boldsymbol{v}_h) - (p_h^{n+1,i}, \nabla \cdot \boldsymbol{v}_h) + (q_h, \nabla \cdot \boldsymbol{u}_h^{n+\theta,i}) \\
&\quad + (\delta_t \boldsymbol{u}_h^{n,i} - \nu \Delta \boldsymbol{u}_h^{n+\theta,i} + \boldsymbol{u}_h^{n+\theta,i-1} \cdot \nabla \boldsymbol{u}_h^{n+\theta,i} + \nabla p_h^{n+1,i}, \nu \Delta \boldsymbol{v}_h + \boldsymbol{u}_h^{n+\theta,i-1} \cdot \nabla \boldsymbol{v}_h + \nabla_{qh})_{\tau_1} \\
&\quad + (\nabla \cdot \boldsymbol{u}_h^{n+\theta}, \nabla \cdot \boldsymbol{v}_h)_{\tau 2} = \langle \boldsymbol{f}, \boldsymbol{v}_h \rangle + (\boldsymbol{f}, \nu \Delta \boldsymbol{v}_h + \boldsymbol{u}_h^{n+\theta,i-1} \cdot \nabla \boldsymbol{v}_h + \nabla q_h)_{\tau_1}
\end{aligned} \tag{76}
$$

instead of the problem in Box 1. Note that the main differences between both formulations are

- The advective velocity, which in (76) is $\boldsymbol{u}_h^{n+\theta,i-1}$. However, in the numerical examples we will use also this velocity instead of $\boldsymbol{u}_h^{n+\theta,i-1} + \tilde{\boldsymbol{u}}^{n+\theta,i-1}$ in the algorithm of Box 1.
- The use of τ_1 in (76) instead of $\tau_{1,t}$ in Box 1, related to the presence of $\tilde{\boldsymbol{u}}^n$ in the second case, which is absent in (76). However, as it has been mentioned at the end of Section 3.3, this difference disappears using quasi-static subscales, a possibility that we will also consider in the examples.
- The term $(\boldsymbol{f}, \nu \Delta \boldsymbol{v}_h + \boldsymbol{u}_h^{n+\theta,i-1} \cdot \nabla \boldsymbol{v}_h + \nabla q_h)_{\tau_1}$ in (76), which replaces the projection terms in the RHS of the problem in Box 1.
- A very important fact from the computational point of view is that in (76) the Laplacian of both the velocity and the velocity test functions within each element has to be added to the convective and pressure terms to maintain consistency. The evaluation of second derivatives is a costly and cumbersome process in finite element implementations which can be avoided by using the formulation of Box 1. Using a standard finite element implementation with an isoparametric transformation from the integration domain to the real one, the calculation of second derivatives implies evaluating the Hessian at the integration points using isoparametric coordinates and transforming it to the Cartesian coordinates. However, there is also the possibility of reconstructing the second derivatives from a least square smoothing of the first in an iterative way as proposed in [41]. In any case, there is an additional difficulty in the evaluation of the viscous term if the viscosity is variable (either due to a non-Newtonian behavior or to the use of a turbulence model), since in this case its gradients would be required using the ASGS formulation, and not with the method of Box 1.
- We have included the term $\delta_t \boldsymbol{u}_h^{n,i}$ in the expression of the element residual in (76). If time is discretized using the discontinuous Galerkin method with a piecewise constant time interpolation, one arrives to (76) with $\theta = 1$ and without this term. However, much better results are obtained if term is included. This has also been observed in [41,42].

The formulation presented in this paper and summarized in Box 1 will be referred to as orthogonal subscale stabilization (OSS), following the definition introduced in [9]. The particular implementation we have used in the nodal-based one described in [43].

The numerical examples presented aim to demonstrate that the OSS *introduces less numerical diffusion than the ASGS method while being equally stable*. In particular, peaks are better captured. Likewise, in spite of the smaller amount of numerical diffusion, the evolution to the steady state is similar using the OSS and the ASGS method. Thus, the OSS can be considered as an alternative to reach steady states in a flow calculation. An additional conclusion will be that if δt is much larger than τ_1 it is not necessary to track the subscales in time, since considering them as quasi-static leads to very similar results.

5.2. Oseen flow in an L-shaped domain

The purpose of this example is to check the performance of the OSS method in a simple stationary Oseen problem but showing three features of practical interest: the presence of internal layers, of boundary layers

and high pressure variations. These features appear in problems of physical interest, even though this one is only introduced as a numerical test, without physical motivation.

The computational domain is taken as the interior of $[0,3]\times[0,3]\backslash[0,2]\times[1,3]$. The inlet is taken at $x=0$, where a discontinuous inflow velocity $\boldsymbol{u}=(1,0)$ for $0\leqslant y\leqslant 1/2$ and $\boldsymbol{u}=(0,0)$ for $1/2<y\leqslant 1$ is prescribed. A zero pressure is prescribed at the outlet $y=3$ and on the rest of the boundary $\boldsymbol{u}$ is fixed to $(0,0)$. The Oseen equations (7) are solved, taking $\boldsymbol{a}=(1,0)$ and $\nu=10^{-4}$. For such a small viscosity, the inflow discontinuous profile propagates inwards with little smearing and a velocity boundary layer is created at $x=3$. We will refer to the *given* velocity $\boldsymbol{a}$ as the *advection velocity*, not to be confused with the velocity $\boldsymbol{u}$ unknown of the problem.

The domain is discretized using 2000 biquadratic elements of equal size, yielding 8241 nodal points. For these elements, second order derivatives cannot be neglected in the ASGS method (76).

Pressure contours and velocity vectors are shown in Fig. 1. These results have been obtained using the OSS method, and are very similar to those obtained using the ASGS formulation. The differences are observed in Fig. 2. Three main conclusions can be drawn from these. First, internal layers are approximated similarly, with the same overshoots and undershoots in both methods. This could be expected, since both the ASGS and OSS introduce, among other terms, streamline diffusion, but no cross-wind numerical dissipation. From the y-velocity section at $y=2$ it is seen that the OSS yields more oscillations near the boundary layer, which are due to the fact that it introduces less numerical diffusion (observe that it only affects the component of the gradient of the unknown orthogonal to the finite element space, not the whole gradient). This is also the reason why the pressure variation is much better captured using OSS that ASGS, as it is seen from the pressure section at the wall $x=3$. A similar behavior was found in the numerical examples presented in [9] for advection–diffusion and Stokes problems.

Referring to the cost of the calculation, it obviously depends on the particular implementation adopted. We have dealt with the projections onto the finite element space iteratively. Giving the reference 100 time units (t.u.) to the solution of the linear system in the first iteration, which is obviously the same for the OSS and the ASGS methods, the construction of the system matrix has taken 13.57 t.u. for OSS and 24.59 t.u. for ASGS. This additional cost of ASGS is due to the need of evaluating the whole element residual and, in particular, second derivatives. Each further iteration of the OSS method, in which only the right-hand side has to be updated and the system matrix is already factored, takes 6.31 t.u. In this example, eight iterations have been needed to reach convergence with a tolerance of 10^{-5} in the relative Euclidian norm of the array of velocity nodal values.

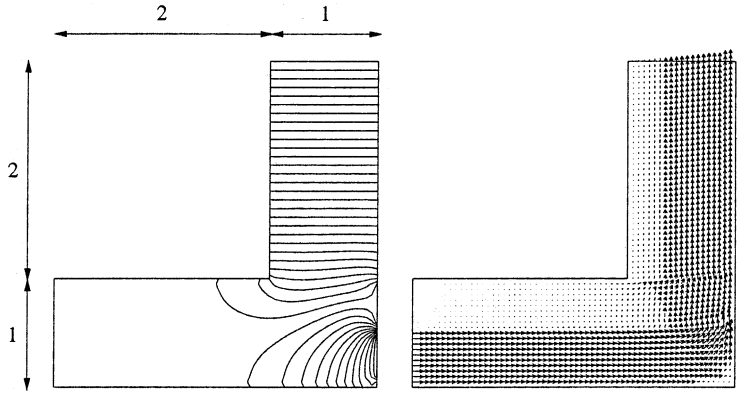

Fig. 1. Pressure contours (left) and vectors of the velocity unknown $\boldsymbol{u}$ (right) for the Oseen flow in an L-shaped domain.

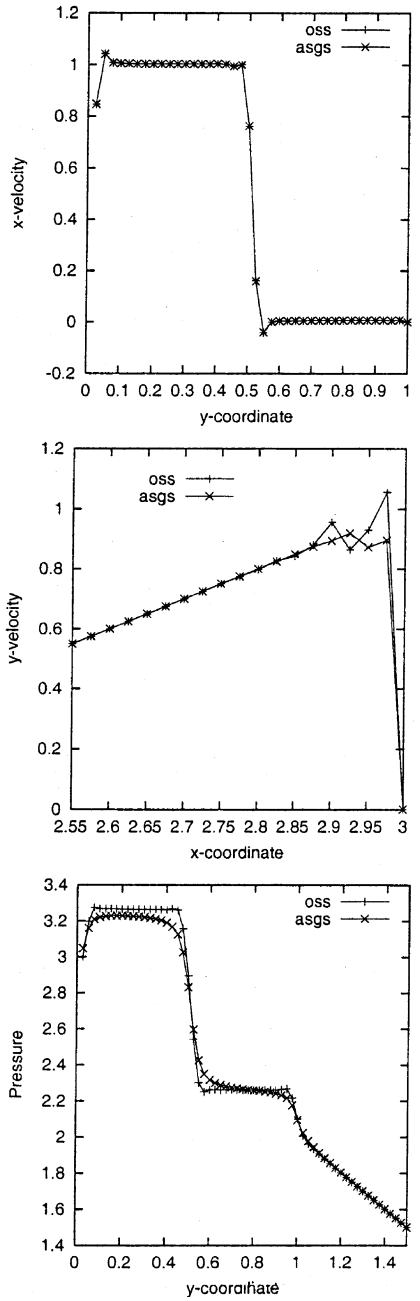

Fig. 2. x-Velocity at $x = 1$ (top), y-velocity at $y = 2$ (middle) and pressure at $x = 3$ (bottom) for the Oseen flow in an L-shaped domain.

5.3. Cavity flow problem

This benchmark test case consists in the prediction of various vortices inside the two-dimensional cavity $\Omega =]0,1[\times]0,1[$ when a velocity $u_x = 1$, $u_y = 0$ is prescribed along the lid $y = 1$. The results presented here

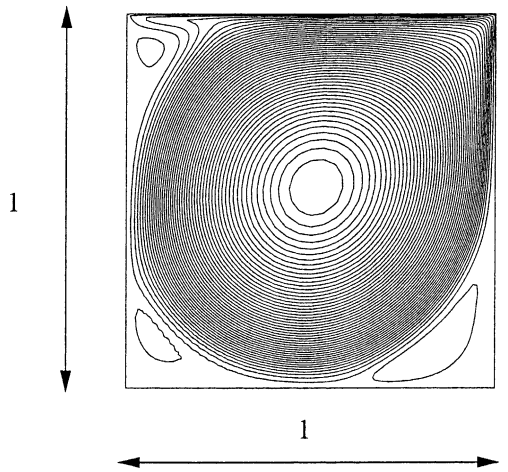

Fig. 3. Streamline pattern for the cavity flow problem. Results obtained using the OSS method and the fine mesh.

correspond to a Reynolds number of 5000 and using the so called "ramp condition" ($\boldsymbol{u} = \boldsymbol{0}$ at the corner nodes).

The computational domain has been discretized using two meshes. The first one, referred to as *coarse* in the following, consists of 40×40 uniform bilinear elements (1681 nodal points), whereas the second, which we will call *fine*, consists of 5408 linear triangles and 2809 nodal points, and it is refined near the boundaries. The general streamline pattern obtained with this mesh is shown in Fig. 3. These results have been obtained using the OSS method and the fine mesh, although they are very similar to those computed using the ASGS formulation.

To determine the accuracy of the numerical results we have compared them with those presented in [44], which were obtained using a very fine grid and have become a standard reference. The comparison of the x- and y-velocity profiles along the cavity mid-sections $x = 0.5$ and $y = 0.5$ is shown in Fig. 4.

As it has been mentioned (cf. Remark 5), we have observed from several numerical experiments that the OSS formulation is sensitive to the convergence tolerance within each time step. In this case we have reached the steady state by stepping in time with $\delta t = 10$ and using three iterations per time step. This leads to a monotone convergence to the steady state that is shown in Fig. 5. For this particular case, it is slightly better using the OSS than the ASGS method.

5.4. Flow over a cylinder

This example involves the flow past a cylinder, another widely solved benchmark problem. The computational domain is $\overline{\Omega} = [0, 16] \times [0, 8] \setminus D$, with the cylinder D of diameter 1 and centered at $(4, 4)$. The velocity at $x = 0$ is prescribed to $(1, 0)$, whereas at $y = 0$ and 8 the y-velocity component is prescribed to 0 and the x-component is left free. The outflow (where both the x- and y-components are free) is $x = 16$. The Reynolds number is 100, based on the cylinder diameter and the prescribed inflow velocity. The finite element mesh employed consists of 4000 linear triangles, with 2100 nodal points, being refined near the cylinder.

In order to obtain the fully developed vortex shedding characteristic of this problem, 90 time steps have been performed with $\delta t = 1$ and $\theta = 0.5$ (Crank–Nicholson scheme), employing for that the ASGS formulation. The convergence tolerance within each time step has been taken as 10^{-2}, again in the relative Euclidian norm of the array of velocity nodal values (a single Picard iteration has been needed to converge).

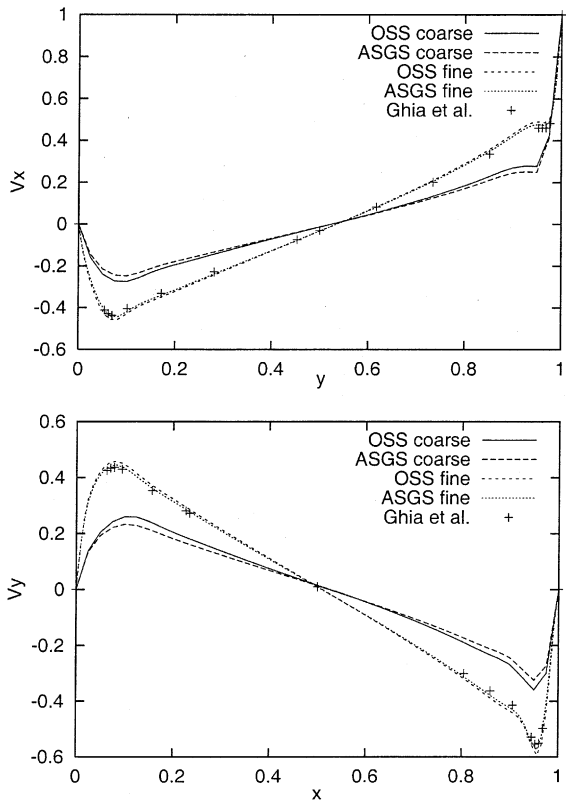

Fig. 4. x-Velocity profile along $x = 0.5$ (top) and y-velocity profile along $y = 0.5$ (bottom) for the flow inside a wall driven cavity.

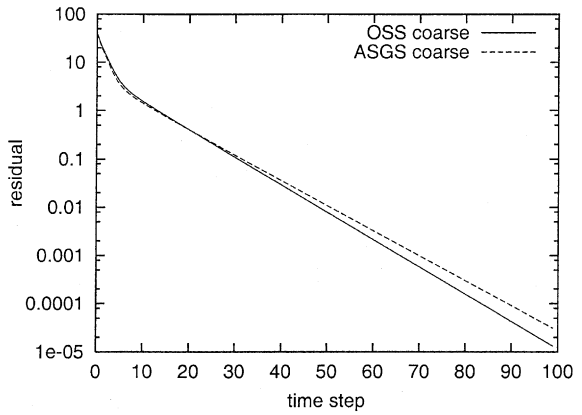

Fig. 5. Convergence to the steady state for the flow inside a wall driven cavity.

The solution thus obtained shows a fully developed periodic flow pattern. These results have been taken as the initial condition for a more accurate calculation, now computed with $\delta t = 0.1$ and requiring a con-

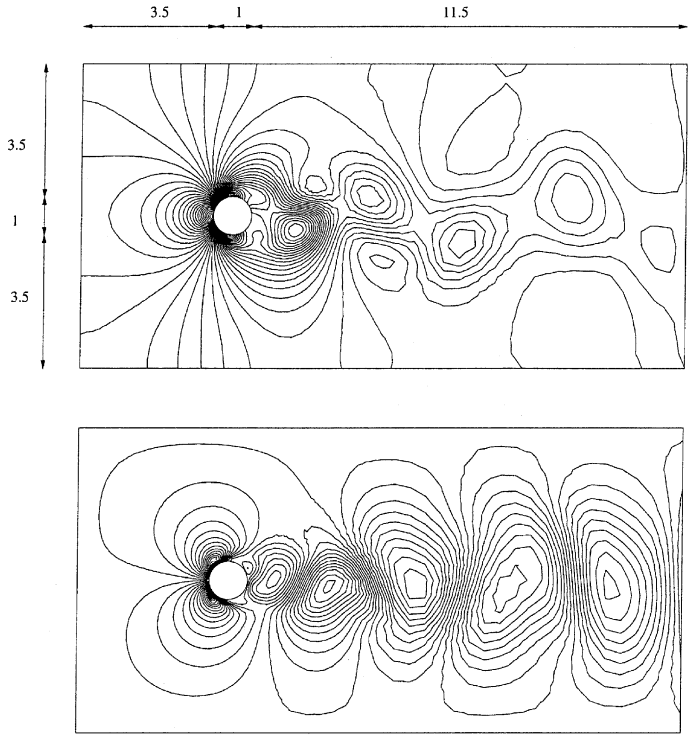

Fig. 6. Pressure and y-velocity contours at $t = 2.5$ for the flow over a cylinder. Results obtained with the formulation of Box 1.

vergence tolerance of 10^{-4}. Three Picard iterations have been performed for each time step, both for the ASGS and the OSS methods.

Pressure contours and contours of the y-velocity component at $t = 2.5$ (after the initial transient described) are shown in Fig. 6. These results have been obtained using the OSS method and tracking the subscales, although the corresponding pictures obtained with the ASGS formulation and the OSS method with quasi-static subscales are very similar.

The period of the oscillations has been found to be 5.70 time units with the OSS formulation and 5.76 using the ASGS method. The values given in Ref. [45] (using the classical SUPG formulation) and [46] are 6.0 and 5.6, respectively. In Ref. [47], the period obtained with a very fine mesh (3426 Q_2/P_1 elements, 14,000 nodal points) is 5.8 time units (see also [36] for results obtained using a similar stabilized formulation). As explained in [48], the period depends on the width on the computational domain, and domains wider than the one used here (which is very often used in the literature) are needed to guarantee that the boundaries do not affect this period.

The temporal evolution of the y-velocity component at point $(x, y) = (0.623, 0.4)$ in the time interval [40] is shown in Fig. 7. It is seen there that the ASGS method is slightly more diffusive than OSS. For this method, on the other hand, tracking the subscales or assuming that they are quasi-static leads to very similar results. The reason for this is that in this case $\delta t = 0.1$ is ten times larger than the average τ_1 (≈ 0.01). However, an interesting difference between both formulations can be observed from the pressure evolution in time (also at point $(0.623, 0.4)$), which is shown in Fig. 8. It is observed that both the ASGS method and the assumption of quasi-static subscales in the OSS formulation lead to a spurious time step-to-time step oscillation which is not present if the subscales are properly tracked. This effect is due to the

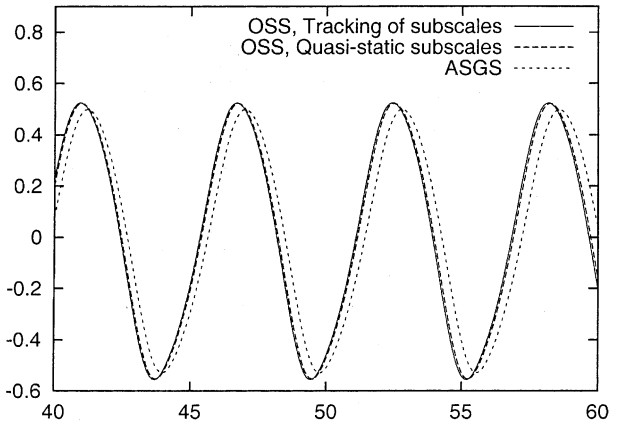

Fig. 7. Temporal evolution of the y-velocity at $(0.623, 0.4)$ for the flow over a cylinder ($\delta t = 0.1$).

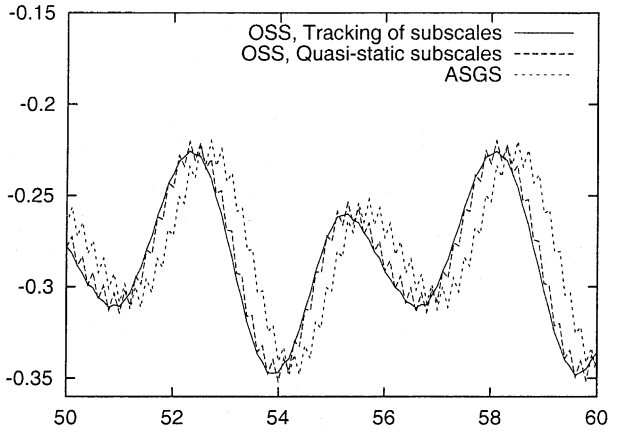

Fig. 8. Temporal evolution of the pressure at $(0.623, 0.4)$ for the flow over a cylinder ($\delta t = 0.1$).

variation of the element size from one element to another; it disappears if the mesh is refined or if only one mesh size is used to compute all the stabilization parameters.

6. Conclusions

In this paper we have described a stabilized finite element method for the incompressible Navier–Stokes equations based on the decomposition of the unknowns into large scales and subgrid scales.

The way to deal with the subgrid scales proposed here is to give a closed-form expression for them based on a Fourier analysis of the problem of which they are solution. This can be considered a modeling strategy alternative to the addition of subgrid viscosity [12,13], the use of a coarse time integration in the nonlinear Galerkin method [23,25], or the approximation using bubble functions [7,17].

The bottom line of our approach is a *stabilized numerical method* which consists basically of adding a least-squares form of the component of the convective and pressure terms orthogonal to the finite element space into the discrete problem, as well as a RHS term that comes from the temporal derivative of the subscales which can be neglected if these are considered quasi-static.

Although not presented in this paper, the analysis of the linearized problem reveals that the method is stable and optimally convergent using equal velocity–pressure interpolations, even in the case of convection-dominated flows. These results are confirmed by the numerical experiments presented here, which show that the method has excellent accuracy. Nevertheless, localized oscillations near sharp layers are even stronger than with other stabilized methods, such as that given by (76).

Concerning the computational cost, for stationary linear or mildly nonlinear problems the formulation proposed here is certainly more expensive than (76), since the need to deal with projections onto the finite element space makes the algorithm necessarily iterative. However, for *transient calculations* it is very competitive, sometimes even cheaper, since less stabilizing terms appear. Moreover, these terms do not depend on the residual of the Navier–Stokes equations, which in some situations may be expensive or very difficult to evaluate. Examples of this are the presence of thermal or electromagnetic couplings, Coriolis forces and, above all, nonlinear viscosities, coming either from nonlinear constitutive models or from turbulence modeling.

References

[1] J.C. Simo, F. Armero, Unconditional stability and long term behavior of transient algorithms for the incompressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 111 (1994) 111–154.
[2] J. Blasco, R. Codina, Space and time error estimates for a first order, pressure stabilized finite element method for the incompressible Navier–Stokes equations, Applied Numerical Mathematics 38 (2001) 475–497.
[3] R. Temam, Navier–Stokes equations, North-Holland, Amsterdam, 1984.
[4] V. Girault, P.A. Raviart, Finite Element Approximation for Navier–Stokes Equations, Lecture Notes in Mathematics, vol. 749, Springer, Berlin, 1979.
[5] T.J.R. Hughes, Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations, Computer Methods in Applied Mechanics and Engineering 127 (1995) 387–401.
[6] 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.
[7] C. Baiocchi, F. Brezzi, L.P. Franca, Virtual bubbles and Galerkin/least-squares type methods (Ga.L.S), Computer Methods in Applied Mechanics and Engineering 105 (1993) 125–141.
[8] F. Brezzi, L.P. Franca, T.J.R. Hughes, A. Russo, $b = \int g$, Computer Methods in Applied Mechanics and Engineering 145 (1997) 329–339.
[9] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, Computer Methods in Applied Mechanics and Engineering 190 (2000) 1579–1599.
[10] R. Codina, J. Blasco, A finite element formulation for the Stokes problem allowing equal velocity–pressure interpolation, Computer Methods in Applied Mechanics and Engineering 143 (1997) 373–391.
[11] R. Codina, J. Blasco, Analysis of a pressure-stabilized finite element approximation of the stationary Navier–Stokes equations, Numerische Mathematik 87 (2000) 59–81.
[12] J.L. Guermond, Stabilization of Galerkin approximations of transport equations by subgrid modeling, Mathematical Modelling and Numerical Analysis 33 (1999) 1293–1316.
[13] 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.
[14] F. Brezzi, M.O. Bristeau, L. Franca, M. Mallet, G. Rogé, A relationship between stabilized finite element methods and the Galerkin method with bubble functions, Computer Methods in Applied Mechanics and Engineering 96 (1992) 117–129.
[15] F. Brezzi, D. Marini, E. Süli, Residual-free bubbles for advection–diffusion problems: the general error analysis, Numerische Mathematik 85 (2000) 31–47.
[16] C. Canuto, V. Van Kemenade, Bubble-stabilized spectral methods for the incompressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 135 (1996) 35–61.
[17] L.P. Franca, A. Nesliturk, M. Stynes, On the stability of residual free bubbles for convection–diffusion problems and their approximation by a two-level finite element method, Computer Methods in Applied Mechanics and Engineering 166 (1998) 35–49.

[18] A. Russo, Bubble stabilization of finite element methods for the linearized incompressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 132 (1996) 335–343.
[19] R. Pierre, Optimal selection of the bubble function in the stabilization of the Pl–Pl element for the Stokes problem, SIAM Journal on Numerical Analysis 32 (1995) 1210–1224.
[20] A.A.O. Ammi, M. Marion, Nonlinear Galerkin methods and mixed finite elements: two-grid algorithms for the Navier–Stokes equations, Numerische Mathematik 68 (1994) 189–213.
[21] R.M. Brown, P. Perry, Z. Shen, The additive turbulent decomposition for the two-dimensional incompressible Navier–Stokes equations: convergence theorems and error estimates, SIAM Journal on Numerical Analysis 59 (1998) 139–155.
[22] J.B. Burie, M. Marion, Multilevel methods in space and time for the Navier–Stokes equations, SIAM Journal on Numerical Analysis 34 (1997) 1574–1599.
[23] C. Foias, O. Manley, R. Temam, Modelling of the interaction of small and large eddies in two-dimensional turbulent flows, Mathematical Modelling and Numerical Analysis 22 (1988) 93–118.
[24] F. Jauberteau, C. Rosier, R. Temam, A nonlinear Galerkin method for the Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 80 (1990) 245–260.
[25] M. Marion, R. Temam, Nonlinear Galerkin methods, SIAM Journal on Numerical Analysis 26 (1989) 157–1139.
[26] R. Codina, Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales, Numerische Mathematik, submitted for publication.
[27] T. Chacón, A term by term stabilization algorithm for the finite element solution of incompressible flow problems, Numerische Mathematik 79 (1998) 283–319.
[28] L.P. Franca, T.J.R. Hughes, Convergence analyses of Galerkin least-squares methods for advective–diffusive forms of the Stokes and incompressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 105 (1993) 285–298.
[29] L.P. Franca, S.L. Frey, Stabilized finite element methods: II. The incompressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 99 (1992) 209–233.
[30] H.-G. Roos, M. Stynes, L. Tobiska, Numerical Methods for Singularly Perturbed Differential Equations—Convection–Diffusion and Flow Problems, Springer, Berlin, 1996.
[31] L. Tobiska, G. Lube, A modified streamline-diffusion method for solving the stationary Navier–Stokes equations, Numerische Mathematik 59 (1991) 13–29.
[32] L. Tobiska, R. Verfürth, Analysis of a streamline diffusion finite element method for the Stokes and Navier–Stokes equations, SIAM Journal on Numerical Analysis 33 (1996) 107–127.
[33] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer, Berlin, 1991.
[34] R. Codina, Comparison of some finite element methods for solving the diffusion–convection-reaction equation, Computer Methods in Applied Mechanics and Engineering 156 (1998) 185–210.
[35] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Computer Methods in Applied Mechanics and Engineering 190 (2001) 2681–2706.
[36] T.E. Tezduyar, S. Mittal, S.E. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity–pressure elements, Computer Methods in Applied Mechanics and Engineering 95 (1992) 221–242.
[37] P.M. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part I: Theory, International Journal for Numerical Methods in Fluids 11 (1990) 587–620.
[38] F. Shakib, T.J.R. Hughes, A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms, Computer Methods in Applied Mechanics and Engineering 87 (1991) 35–58.
[39] R. Codina, J. Blasco, G.C. Buscaglia, A. Huerta, Implementation of a stabilized finite element formulation for the incompressible Navier–Stokes equations based on a pressure gradient projection, International Journal for Numerical Methods in Fluids 37 (2001) 419–444.
[40] O. Soto, R. Codina, A numerical model for mould filling using a stabilized finite element method and the VOF technique, International Journal for Numerical Methods in Fluids, submitted for publication.
[41] K.E. Jansen, S.S. Collis, C. Whiting, F. Shakib, A better consistency for low-order stabilized finite element methods, Computer Methods in Applied Mechanics and Engineering 174 (1999) 153–170.
[42] G. Lube, D. Weiss, Stabilized finite element methods for singularly perturbed parabolic problems, Applied Numerical Mathematics 17 (1995) 431–459.
[43] R. Codina, A nodal-based implementation of a stabilized finite element method for incompressible flow problems, International Journal for Numerical Methods in Fluids 33 (2000) 737–766.
[44] U. Ghia, K.N. Ghia, C.T. Shin, High-resolutions for incompressible flow using the Navier–Stokes equations and a multi-grid method, Journal of Computational Physics 48 (1982) 387–441.
[45] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equation, Computer Methods in Applied Mechanics and Engineering 32 (1982) 199–259.

[46] P.M. Gresho, S.T. Chan, R.L. Lee, C.D. Upson, A modified finite element method for solving the time-dependent, incompressible Navier–Stokes equations. Part 2: Applications, International Journal for Numerical Methods in Fluids 4 (1984) 619–640.
[47] M.S. Engelman, M.A. Jamnia, Transient flow past a circular cylinder: a benchmark solution, International Journal for Numerical Methods in Fluids 11 (1990) 985–1000.
[48] M.A. Behr, D. Hastreiter, S. Mittal, T.E. Tezduyar, Incompressible flow past a circular cylinder: dependence of the computed flow field on the location of the lateral boundaries, Computer Methods in Applied Mechanics and Engineering 123 (1995) 309–316.