Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 193 (2004) 1403–1419
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Approximation of the incompressible Navier–Stokes equations using orthogonal subscale stabilization and pressure segregation on anisotropic finite element meshes

Ramon Codina [a,*], Orlando Soto [b]

[a] International Center for Numerical Methods in Engineering (CIMNE), Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain
[b] Laboratory for Computational Fluid Dynamics, George Mason University, MS 4C7, 4400 University Drive. Fairfax, VA 22030-4444, USA

Received 12 February 2003; received in revised form 24 October 2003; accepted 9 December 2003

Abstract

This paper describes a finite element model to solve the incompressible Navier–Stokes equations based on the stabilization with orthogonal subscales and a pressure segregation. The former consists of adding a least-square form of the component orthogonal to the finite element space of the convective and pressure gradient terms; this allows to deal with convection-dominated flows and to use equal velocity–pressure interpolation. The pressure segregation is inspired in fractional step schemes, although the converged solution corresponds to that of a monolithic time integration. Likewise, we put special emphasis on the use of anisotropic grids. In particular, we describe some possible choices for the calculation of the element length that appears in the stabilization parameters and check their behavior in two classical numerical examples. The preconditioning strategy used to solve the resulting algebraic system for very anisotropic meshes is also briefly described.

Keywords: Incompressible Navier-Stokes equations; Orthogonal subscale stabilization; Pressure segregation; Anisotropic finite element meshes

1. Introduction

The treatment of the pressure in numerical approximations of incompressible flow problems is still an active subject of research basically for two reasons. First, its approximation needs to be different from that of the velocity field in order to have a stable numerical scheme. Secondly, its coupling with the velocity

* Corresponding author.
E-mail addresses: ramon.codina@upc.es (R. Codina), soto@scs.gmu.edu (O. Soto).
URLs: http://www.rmee.upc.es/homes/codina (R. Codina), http://www.scs.gmu.edu/~soto (O. Soto).

0045-7825/$ - see front matter
doi:10.1016/j.cma.2003.12.030

components makes the solution of the linear system arising from the discretization of the equations highly demanding from the computational point of view.

Referring to the pressure approximation, when finite element methods are used this leads to the well known inf–sup stability condition for the velocity and pressure finite element spaces if the standard Galerkin formulation is used. To satisfy it is possible, and several velocity pressure pairs are known to fulfill the inf–sup condition. However, there is also the possibility of modifying the discrete variational formulation of the problem so as to circumvent it. Finite element formulations of this kind may fall basically into two categories, namely, methods that allow the use of equal interpolations (and therefore continuous pressures) and techniques to stabilize simple elements, such as the Q_1/P_0 pair (multilinear velocity, piecewise constant pressure). Examples of the first group are the methods in [5,17], the Galerkin/least-squares (GLS) technique [21,22,25] and least-squares methods for first-order systems as those in [4], whereas examples of the second are those in [20,30], among others.

Concerning the velocity–pressure coupling, fractional step methods for the incompressible Navier–Stokes equations have enjoyed widespread popularity since the original works of Chorin [7] and Temam [34]. The reason for this relies on the computational efficiency of these methods (see e.g. [26,29,35]), basically due to the uncoupling of the pressure from the velocity components. However, several issues related to these methods still deserve further analysis, and perhaps the most salient of these are the behavior of the computed pressure near boundaries and the stability of the pressure itself.

In this paper we address these two aspects of the pressure treatment. On the one hand, we describe a finite element method able to deal with *equal* velocity pressure interpolations. On the other hand, we present an iterative algorithm that allows to uncouple the calculation of the pressure from that of the velocity. It is motivated by what is commonly done in fractional step methods, although the converged solution of the iterative procedure is that of the *monolithic* (coupled velocity–pressure) time discretization. In this sense, our approach can be viewed as a *predictor–multicorrector* method.

Apart from the pressure treatment, another important issue to be considered in the numerical approximation of incompressible flows are the (numerical) instability problems found when the viscous term is small compared to the convective one. Both the inf–sup condition and the convection instabilities can be overcome by resorting from the standard Galerkin method 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 [23,24] for the scalar convection–diffusion equation (see also [3,6] for related methods). The basic idea is to approximate the *effect* of the component of the continuous solution which can not 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 [8] as an extension of a stabilization method originally introduced for the Stokes problem in [13] and fully analyzed for the stationary Navier–Stokes equations in [14]. It is further elaborated in [11]. A summary of the method proposed in this last reference is presented here.

In this paper we put special emphasis on the use of anisotropic finite element meshes, particularly in the way to compute the element length when the aspect ratio is very high, and in the preconditioning of the algebraic system resulting from these anisotropic meshes. Concerning the calculation of the element length, we propose four options and test them in two numerical examples. These options are the maximum length of the element edges, the minimum, the element length in the direction of the flow and the length proposed in [19], for which some error estimates are available for scalar elliptic problems. Referring to the preconditioning method, we recall very briefly the linelet strategy proposed in [32].

We have organized the paper as follows. In the following section we present the problem to approximate, some notation and the time discretization we will use. In Section 3 we summarize the stabilized finite element formulation and in Section 4 we describe some possible ways to compute the element length appearing in the stabilization parameters. In Section 5 we describe the pressure segregation strategy we

propose, together with some computational aspects of the algorithm. Some numerical results are presented in Section 6 and finally some conclusions are drawn in Section 7.

2. Problem statement

2.1. Continuous problem

Let Ω be the domain of $\mathbb{R}^{n_{\rm sd}}$ occupied by the fluid, where $n_{\rm sd} = 2$ 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

$$\partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nu\nabla^2\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega,\ t \in (0,T), \tag{1}$$

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

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \Gamma,\ t \in (0,T), \tag{3}$$

$$\boldsymbol{u} = \boldsymbol{u}^0 \quad \text{in } \Omega,\ t = 0, \tag{4}$$

where ν is the kinematic viscosity, $\boldsymbol{f}$ is the force vector and $\boldsymbol{u}^0$ is the velocity initial condition. We have considered the homogeneous Dirichlet boundary condition (3) for simplicity.

To write the weak form of problem (1)–(4) we need to introduce some notation. We denote by $H^1(\Omega)$ the Sobolev space of functions whose first derivatives belong to $L^2(\Omega)$, and by $H^1_0(\Omega)$ the subspace of $H^1(\Omega)$ of functions with zero trace on Γ. A bold character is used for the vector counterpart of these spaces. The L^2 scalar product in a set ω is denoted by $(\cdot,\cdot)_\omega$, and the L^2 norm by $\|\cdot\|_\omega$. The subscript ω is omitted when it coincides with Ω. To pose the problem, we also need the functional spaces $\boldsymbol{V}_{\rm st} = \boldsymbol{H}^1_0(\Omega)^{n_{\rm sd}}$, and $Q_{\rm st} = \{q \in L^2(\Omega) | \int_\Omega q = 0\}$, as well as $\boldsymbol{V} = \boldsymbol{L}^2(0,T;\boldsymbol{V}_{\rm st})$ and $Q = L^2(0,T;Q_{\rm st})$ for the transient problem.

Assuming for simplicity the force vector to be square integrable, the weak form of problem (1)–(4) consists in finding $(\boldsymbol{u}, p) \in \boldsymbol{V}\times Q$ such that

$$(\partial_t\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}) + \nu(\nabla\boldsymbol{u}, \nabla\boldsymbol{v}) - (p, \nabla\cdot\boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}), \tag{5}$$

$$(q, \nabla\cdot\boldsymbol{u}) = 0, \tag{6}$$

for all $(\boldsymbol{v}, q) \in \boldsymbol{V}_{\rm st}\times Q_{\rm st}$, and satisfying the initial condition in a weak sense.

2.2. Time discretization

Any time integration of (5) and (6) is in principle possible. However, we shall concentrate on the monolithic trapezoidal rule (solving for the velocity and the pressure at the same time). The time discretized version of (5) and (6) in this case consists in solving the following problem: from known $\boldsymbol{u}^n$, find $\boldsymbol{u}^{n+1} \in \boldsymbol{V}_{\rm st}$ and $p^{n+1} \in Q_{\rm st}$ such that

$$(\delta^n_t\boldsymbol{u} + \boldsymbol{u}^{n+\theta}\cdot\nabla\boldsymbol{u}^{n+\theta}, \boldsymbol{v}) + \nu(\nabla\boldsymbol{u}^{n+\theta}, \nabla\boldsymbol{v}) - (p^{n+\theta}, \nabla\cdot\boldsymbol{v}) = (\bar{\boldsymbol{f}}^{n+\theta}, \boldsymbol{v}), \tag{7}$$

$$(q, \nabla\cdot\boldsymbol{u}^{n+\theta}) = 0, \tag{8}$$

for all $(\boldsymbol{v}, q) \in \boldsymbol{V}_{\rm st}\times Q_{\rm st}$, where δt is the time step size, superscript m refers to the time step level $t^m = m\delta t$, $\theta \in (0, 1]$ and we use the notation

$$\boldsymbol{u}^{n+\theta} := \theta\boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{u}^n, \quad \delta\boldsymbol{u}^n := \boldsymbol{u}^{n+1} - \boldsymbol{u}^n \quad \text{and} \quad \delta^n_t\boldsymbol{u} := \frac{\delta\boldsymbol{u}^n}{\delta t}.$$

The force term $\bar{\boldsymbol{f}}^{n+\theta}$ in (7) and below has to be understood as the time average of the force in the interval $[t^n, t^{n+1}]$, even though we use a superscript $n+\theta$ to characterize it. The pressure value computed here has been identified as the pressure evaluated at $t^{n+\theta}$, although this is irrelevant for the velocity approximation. The values of interest of θ are $\theta = 1/2$, corresponding to the second order Crank–Nicolson scheme, and $\theta = 1$, which corresponds to the backward Euler method. In this case, the convective term in (7) can be replaced by $(\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^{n+1}, \boldsymbol{v})$, since it also leads to a first order unconditionally stable scheme, well suited for the long term time integration.

3. Stabilization with orthogonal subscales

The purpose of this section is to summarize the steps that lead to the finite element formulation proposed in [11]. Let $\mathscr{T}_h$ denote a finite element partition of the domain Ω of diameter h, from which we construct the finite element spaces Q_h, $\boldsymbol{V}_h$ and $\boldsymbol{V}_{h,0}$, approximations to Q_{st}, $\boldsymbol{H}^1(\Omega)^{n_{\mathrm{sd}}}$ and $\boldsymbol{V}_{\mathrm{st}}$, respectively. The former is made up with continuous functions of degree k_q and the other two with continuous vector functions of degree k_v, the latter verifying the homogeneous Dirichlet boundary conditions. In the following, finite element functions will be identified with a subscript h.

The discrete problem is obtained by approximating $\boldsymbol{u}$ and p. If $\boldsymbol{u}_h$ and p_h are the finite element unknowns, we approximate $\boldsymbol{u} \approx \boldsymbol{u}_h + \tilde{\boldsymbol{u}}$ and $p \approx p_h$, that is, the velocity is approximated by its finite element component plus an additional term that we call *subgrid scale* or *subscale* (for the sake of simplicity, the pressure subscale will be taken as zero). We call $\boldsymbol{u}^n \approx \boldsymbol{u}_*^n := \boldsymbol{u}_h^n + \tilde{\boldsymbol{u}}^n$ and $p^n \approx p_h^n$ the velocity and the pressure for time level n. Considering the spatial interpolation, we assume that $\boldsymbol{u}_h^n$ and p_h^n are constructed using the standard finite element interpolation. In particular, equal velocity–pressure interpolation is possible with the formulation to be presented.

The important point is the behavior assumed for $\tilde{\boldsymbol{u}}^n$. To simplify the discussion, we assume that it vanishes on the interelement boundaries, that is, it is a bubble-like function [3,6]. However, contrary to what is commonly done, we do not assume any particular behavior of $\tilde{\boldsymbol{u}}^n$ within the element domains. We will indicate later on how to approximate it.

If in (7) $\boldsymbol{u}^n$ is replaced by $\boldsymbol{u}_*^n := \boldsymbol{u}_h^n + \tilde{\boldsymbol{u}}^n$, p^n is replaced by p_h^n, the terms involving $\tilde{\boldsymbol{u}}^n$ are integrated by parts, and the test functions are taken in the finite element space, one gets

$$(\delta_t^n \boldsymbol{u}_h + \boldsymbol{u}_*^{n+\theta} \cdot \nabla \boldsymbol{u}_h^{n+\theta}, \boldsymbol{v}_h) + \nu(\nabla \boldsymbol{u}_h^{n+\theta}, \nabla \boldsymbol{v}_h) - (p_h^{n+\theta}, \nabla \cdot \boldsymbol{v}_h) + (\delta_t^n \tilde{\boldsymbol{u}}, \boldsymbol{v}_h) - (\tilde{\boldsymbol{u}}^{n+\theta}, \nu \nabla_h^2 \boldsymbol{v}_h + \boldsymbol{u}_*^{n+\theta} \cdot \nabla \boldsymbol{v}_h) = (\bar{\boldsymbol{f}}^{n+\theta}, \boldsymbol{v}_h), \tag{9}$$

$$(q_h, \nabla \cdot \boldsymbol{u}_h^{n+\theta}) - (\tilde{\boldsymbol{u}}^{n+\theta}, \nabla q_h) = 0, \tag{10}$$

which must hold for all $\boldsymbol{v}_h \in \boldsymbol{V}_{h,0}$ and $q_h \in Q_h$. The notation ∇_h^2 is used to indicate that the Laplacian needs to be evaluated element by element. It is important to note that the advection velocity in (9) is $\boldsymbol{u}_*^{n+\theta}$.

The equation for the subscales $\tilde{\boldsymbol{u}}^{n+1}$ is obtained by taking the velocity test function in (7) in its space. The result is that, within each element [10]:

$$\delta_t \tilde{\boldsymbol{u}}^n + \boldsymbol{u}_*^{n+\theta} \cdot \nabla \tilde{\boldsymbol{u}}^{n+\theta} - \nu \nabla^2 \tilde{\boldsymbol{u}}^{n+\theta} = \boldsymbol{r}^{n+\theta} + \boldsymbol{v}_{h,\mathrm{ort}}^{n+\theta}, \tag{11}$$

$$\boldsymbol{r}^{n+\theta} := \boldsymbol{f}^{n+\theta} - (-\nu \nabla^2 \boldsymbol{u}_h^{n+\theta} + \boldsymbol{u}_*^{n+\theta} \cdot \nabla \boldsymbol{u}_h^{n+\theta} + \nabla p_h^{n+\theta}), \tag{12}$$

where $\boldsymbol{v}_{h,\mathrm{ort}}^{n+\theta}$ is a function L^2-orthogonal to the space of subscales.

The next step is *to model* the solution $\tilde{\boldsymbol{u}}^{n+1}$ of (11). This means to give a closed-form expression for it that approximates the exact solution in some sense. It is shown in [11] that if we replace (11) by the algebraic equation

$$\left(\frac{1}{\theta\delta t}+\frac{1}{\tau}\right)\tilde{\boldsymbol{u}}^{n+\theta}=\frac{1}{\theta\delta t}\tilde{\boldsymbol{u}}^{n}+\boldsymbol{r}^{n+\theta}+\boldsymbol{v}_{h,\mathrm{ort}}^{n+\theta}, \tag{13}$$

with

$$\tau:=\left(c_1\frac{\nu}{h^2}+c_2\frac{|\boldsymbol{u}_*^{n+\theta}|}{h}\right)^{-1}, \tag{14}$$

then there are values of the constants c_1 and c_2 (which do not depend neither on the discretization nor on the equation coefficients ν and $|\boldsymbol{u}_*^{n+\theta}|$) for which the solutions of (11) and (13) have approximately the same L^2-norm over the elements. Note that in (13) we do not require the subscales to vanish on the element boundaries.

It still remains to define the space of the subscales. A main feature of our formulation is that we take it (approximately) orthogonal to the finite element space. Imposing this in (13) allows to compute $\boldsymbol{v}_{h,\mathrm{ort}}^{n+\theta}$, which turns out to be minus the projection of $\boldsymbol{r}^{n+\theta}$ onto the finite element space $\boldsymbol{V}_h$. Therefore,

$$\left(\frac{1}{\theta\delta t}+\frac{1}{\tau}\right)\tilde{\boldsymbol{u}}^{n+\theta}=\frac{1}{\theta\delta t}\tilde{\boldsymbol{u}}^{n}+P_h^{\perp}(\boldsymbol{r}^{n+\theta}), \tag{15}$$

where $P_h^{\perp}=I-P_h$, P_h being the L^2-projection onto $\boldsymbol{V}_h$.

To complete the description of the method, we will make two further approximations. First, we will consider that $P_h^{\perp}(\boldsymbol{f})=\boldsymbol{0}$, that is, $\boldsymbol{f}$ is a finite element function. This does not alter the accuracy of the final formulation. Secondly, we will neglect the orthogonal projection of viscous term in $\boldsymbol{r}^{n+\theta}$ and $\nu\nabla_h^2\boldsymbol{v}_h$ in (9). This is exact for linear elements and leads to a consistent formulation for higher order elements. Thus, the final system of equations we are left with is

$$\begin{aligned}&(\delta_t^n\boldsymbol{u}_h+\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta},\boldsymbol{v}_h)+\nu(\nabla\boldsymbol{u}_h^{n+\theta},\nabla\boldsymbol{v}_h)-(p_h^{n+\theta},\nabla\cdot\boldsymbol{v}_h)\\&\quad+(\tau_tP_h^{\perp}(\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta}+\nabla p_h^{n+\theta}),\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{v}_h)=(\bar{\boldsymbol{f}}^{n+\theta},\boldsymbol{v}_h)+\frac{1}{\theta\delta t}(\tau_t\tilde{\boldsymbol{u}}^{n},\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{v}_h),\end{aligned} \tag{16}$$

$$(q_h,\nabla\cdot\boldsymbol{u}_h^{n+\theta})+(\tau_tP_h^{\perp}(\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta}+\nabla p_h^{n+\theta}),\nabla q_h)=\frac{1}{\theta\delta t}(\tau_t\tilde{\boldsymbol{u}}^{n},\nabla q_h), \tag{17}$$

where

$$\tau_t:=\left(\frac{1}{\theta\delta t}+\frac{1}{\tau}\right)^{-1}.$$

Note that the term $(\delta_t^n\tilde{\boldsymbol{u}},\boldsymbol{v}_h)$ in (9) vanishes because of the orthogonality of $\delta_t^n\tilde{\boldsymbol{u}}$ and $\boldsymbol{v}_h$. Note also that the parameter τ_t has been included within the inner product, since in principle it changes from point to point. The terms multiplied by this parameter must be responsible for the enhancement of stability with respect to the standard Galerkin method; we will call them *stabilization terms*.

The formulation we propose is finally given by system (16) and (17), together with (15) to update the subscales. Here, we have only sketched how to obtain it; the reader is referred to [11] for further details. Likewise, we have some results of convergence analyses for simplified problems. In particular, the stationary and linear case (with $\boldsymbol{u}_*$ replaced by a given advection velocity) is fully analyzed in [12], and the *transient* advection–diffusion equation (also linear) in [15]. The results of these papers show that the goal of being able to deal with convection dominated flows and using equal velocity–pressure interpolation is certainly achieved.

Except for the fact that we have deleted the viscous contribution in the stabilization terms, all the steps we have followed have a heuristic motivation. Once arrived to the final problem (16) and (17), we can make some further modifications *provided the consistency of the method is maintained*, that it to say, these modifications do not alter the fact that the exact solution is still a solution of the discrete problem. For the discussion of the next section and the numerical results, we will make the following modifications:

- We will consider the subscales *quasi-static*. As explained in [11], this leads to $\tilde{\boldsymbol{u}}^{n+\theta} = \tau P_h^{\perp}(\boldsymbol{r}^{n+\theta})$ as the solution of (15).
- The advection velocity $\boldsymbol{u}_*$ will be replaced $\boldsymbol{u}_h$. This means that we neglect the influence of the subscales in the transport of momentum. However, this point needs to be further explored, since this influence is the key of turbulence modeling.
- We will remove $P_h^{\perp}(\nabla p_h^{n+\theta})$ from (16) and $P_h^{\perp}(\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta})$ from (17). This implies that instead of adding a least-square form of $P_h^{\perp}(\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta}+\nabla p_h^{n+\theta})$ in the variational system we add a least-square form of $P^{\perp}(\nabla p_h^{n+\theta})$ *plus* a least-square form of $P_h^{\perp}(\boldsymbol{u}_*^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta})$. This possibility is also analyzed in [12] and shown to have even (slightly) improved stability.

With all these modifications, we arrive at the discrete problem

$$(\delta_t^n\boldsymbol{u}_h+\boldsymbol{u}_h^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta},\boldsymbol{v}_h)+\nu(\nabla\boldsymbol{u}_h^{n+\theta},\nabla\boldsymbol{v}_h)-(p_h^{n+\theta},\nabla\cdot\boldsymbol{v}_h)+(\tau P_h^{\perp}(\boldsymbol{u}_h^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta}),\boldsymbol{u}_h^{n+\theta}\cdot\nabla\boldsymbol{v}_h)$$
$$=(\bar{\boldsymbol{f}}^{n+\theta},\boldsymbol{v}_h),\quad\forall\boldsymbol{v}_h\in\boldsymbol{V}_{h,0}, \tag{18}$$

$$(q_h,\nabla\cdot\boldsymbol{u}^{n+\theta})+(\tau P_h^{\perp}(\nabla p_h^{n+\theta}),\nabla q_h)=0,\quad\forall q_h\in Q_h, \tag{19}$$

which will be further analyzed in the Section 5 from a more computational standpoint.

4. Characteristic element length

In this section we discuss some possible ways to compute the characteristic element length h appearing in the expression of the stabilization parameter given by (14).

The first remark to be made is that if the mesh is *isotropic*, that is to say, both the diameter of the circle (sphere in 3D) inscribed to the element and the external one go to zero at the same rate, the way to compute the element length will not affect *the convergence rate* of the method. In this case, if h_1 and h_2 are two possible ways to measure h, we will have that $ch_1\leqslant h_2\leqslant c'h_1$ for appropriate constants c and c'. The error estimates in terms of h_1 and h_2 will be the same up to a constant depending on c and c'.

Even though the error estimates are the same, the error for a given h may be different depending on the way to compute it. Likewise, the convergence behavior of the iterative scheme may depend on the way to compute h. This is what we want to check from numerical experiments, two of which will be presented later on.

The situation is different when the mesh is *anisotropic*. In this case, the element size in different directions go to zero at different rates. The interpolation theory and, obviously, the error analysis, is not so mature in this case as in the case of isotropic meshes. Some results applied to stabilized finite element methods can be found in [1,2,18,19]. We will describe here the way to compute h proposed in this last reference [19]. However, our intention is not to discuss its convergence properties, but only to check its behavior for a given mesh compared to other possibilities.

We have implemented four different ways to compute the characteristic element length: the maximum length of the element edges, the minimum, the element size in the direction of the flow and the expression proposed in [18]. The first two ways to compute h are obvious, so we describe now the other two. To simplify the notation we will consider the two-dimensional case, although what follows is completely general.

4.1. Element length in the flow direction

In this section we briefly describe the way we compute the element length in the flow direction, which was already proposed in [16].

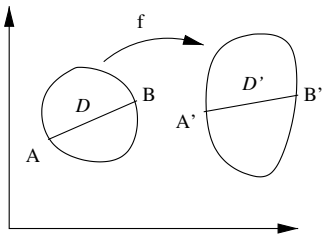

Fig. 1. Transformation of a domain in $\mathbb{R}^2$ by an affine mapping.

Let D be a convex domain in $\mathbb{R}^2$ transformed into $D' \subset \mathbb{R}^2$ by an affine mapping $\boldsymbol{f} = (f_1, f_2)$. Using the notation of Fig. 1, let $\ell = |B - A|$, $\ell' = |B' - A'|$, where $\boldsymbol{v}$ the vector from A to B and $\boldsymbol{v}' = (\boldsymbol{Df})\boldsymbol{v}$, where $\boldsymbol{Df}$ is the Jacobian matrix of $\boldsymbol{f}$. Since

$$\boldsymbol{f}(B) = \boldsymbol{f}(A) + \ell' \frac{\boldsymbol{v}'}{|\boldsymbol{v}'|} = \boldsymbol{f}(A) + \boldsymbol{Df}(B - A), \tag{20}$$

we have that

$$\ell' (\boldsymbol{Df})^{-1} \boldsymbol{v}' = |\boldsymbol{v}'|(B - A). \tag{21}$$

Taking the Euclidian norm on both sides of (21) and considering that $\boldsymbol{Df}^{-1}\boldsymbol{v}' = \boldsymbol{v}$ we get

$$\ell' = \frac{|\boldsymbol{v}'|}{|\boldsymbol{v}|} \ell. \tag{22}$$

Formula (22) allows us to compute the characteristic length in the flow direction as

$$h = \frac{|\boldsymbol{u}|}{|\boldsymbol{u}_0|} h_0, \tag{23}$$

where subscript naught indicates that the value corresponds to the parent domain of the element. Eq. (23) reduces the computation of h to that of h_0, which can be easily estimated since the geometry is now very simple. In our computations we have taken $h_0 = 2$ for quadrilateral elements with parent domain $[-1, 1] \times [-1, 1]$ and $h_0 = 0.7$ for triangular elements using as parent domain the triangle of vertices $(0, 0)$, $(0, 1)$, $(1, 0)$.

Observe that the length h defined by (23) depends on the point of the element domain. Thus, it will be numerically different at each integration point. Also, the exact value of h_0 depends on each point, although the assumption of a constant value seems reasonable. Observe also that from (20) it can be seen that (23) will be exact whenever the mapping $\boldsymbol{f}$ can be considered affine. This will always be the case with straightsided triangles and parallelograms in two dimensions.

4.2. Element length for anisotropic meshes

In this section we describe the way to compute h for each element proposed in [18] to account for the anisotropy of the mesh.

As before, let $\boldsymbol{Df}$ be the Jacobian of the isoparametric mapping from the parent domain to the element under consideration, and let $\boldsymbol{Df} = \boldsymbol{BZ}$ be its polar decomposition, with $\boldsymbol{B}$ symmetric and positive-definite and $\boldsymbol{Z}$ orthogonal. If λ_1, λ_2 are the two eigenvalues of $\boldsymbol{B}$, with $\lambda_1 > \lambda_2 > 0$, the proposal of the above mentioned reference is to take

$$h = \lambda_2 \tag{24}$$

as element length to use in the expression of the stabilization parameters.

5. Pressure segregation and iterative procedure

In this section we will consider two possibilities of uncoupling the calculation of the pressure from the velocity. The first is a classical fractional step scheme, which we will describe in detail when it is applied to our stabilized formulation. The second is an iterative scheme for the original *monolithic* problem but which takes as starting point the fractional step method. This will lead to an iterative scheme in which the pressure in the momentum equation is lagged one iteration with respect to the velocity. After computing this velocity, the pressure can be updated. In this sense, the algorithm we propose can be viewed as a *predictor–multicorrector* method.

5.1. Matrix version of the problem

For the purposes of this section, it is convenient to write the matrix version of problem (18) and (19). Let us note first that the orthogonal projections of these equations can be written as

$$P_h^{\perp}(\boldsymbol{u}_h^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta}) = \boldsymbol{u}_h^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta} - \boldsymbol{y}_h^{n+\theta},$$

$$P_h^{\perp}(\nabla p_h^{n+\theta}) = \nabla p_h^{n+\theta} - \boldsymbol{z}_h^{n+\theta},$$

where $\boldsymbol{y}_h^{n+\theta}$ and $\boldsymbol{z}_h^{n+\theta}$ are the solution of

$$(\boldsymbol{y}_h^{n+\theta}, \boldsymbol{v}_h) = (\boldsymbol{u}_h^{n+\theta}\cdot\nabla\boldsymbol{u}_h^{n+\theta}, \boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{25}$$

$$(\boldsymbol{z}_h^{n+\theta}, \boldsymbol{v}_h) = (\nabla p_h^{n+\theta}, \boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h. \tag{26}$$

From these expressions it is easily checked that the discrete variational problem (18) and (19), together with the projection equations (25) and (26), leads to the nonlinear algebraic system

$$\mathsf{M}\delta_t^n\mathsf{U} + \mathsf{K}(\mathsf{U}^{n+\theta})\mathsf{U}^{n+\theta} + \mathsf{G}\mathsf{P}^{n+\theta} + \mathsf{S}_u(\tau;\mathsf{U}^{n+\theta})\mathsf{U}^{n+\theta} - \mathsf{S}_y(\tau;\mathsf{U}^{n+\theta})\mathsf{Y}^{n+\theta} = \mathsf{F}^{n+\theta}, \tag{27}$$

$$\mathsf{D}\mathsf{U}^{n+\theta} + \mathsf{S}_p(\tau)\mathsf{P}^{n+\theta} - \mathsf{S}_z(\tau)\mathsf{Z}^{n+\theta} = 0, \tag{28}$$

$$\mathsf{M}\mathsf{Y}^{n+\theta} - \mathsf{C}(\mathsf{U}^{n+\theta})\mathsf{U}^{n+\theta} = 0, \tag{29}$$

$$\mathsf{M}\mathsf{Z}^{n+\theta} - \mathsf{G}\mathsf{P}^{n+\theta} = 0, \tag{30}$$

where U, P, Y and Z are the arrays of nodal unknowns for $\boldsymbol{u}$, $\boldsymbol{p}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, respectively. If we denote the node indexes with superscripts a, b, the space indexes with subscripts i, j, and the standard shape function of node a by N^a, the components of the arrays involved in these equations are:

$$\mathsf{M}_{ij}^{ab} = (N^a, N^b)\delta_{ij} \quad (\delta_{ij} \text{ is the Kronecker } \delta),$$

$$\mathsf{K}(\mathsf{U}^{n+\theta})_{ij}^{ab} = (N^a, \boldsymbol{u}_h^{n+\theta}\cdot\nabla N^b)\delta_{ij} + \nu(\nabla N^a, \nabla N^b)\delta_{ij},$$

$$\mathsf{G}_i^{ab} = (N^a, \partial_i N^b),$$

$$\mathsf{S}_u(\tau;\mathsf{U}^{n+\theta})_{ij}^{ab} = (\tau\boldsymbol{u}_h^{n+\theta}\cdot\nabla N^a, \boldsymbol{u}_h^{n+\theta}\cdot\nabla N^b)\delta_{ij},$$

$$\mathsf{S}_y(\tau;\mathsf{U}^{n+\theta})_{ij}^{ab} = (\tau\boldsymbol{u}_h^{n+\theta}\cdot\nabla N^a, N^b)\delta_{ij},$$

$$\mathsf{D}_j^{ab} = (N^a, \partial_j N^b),$$

$$\mathsf{S}_p(\tau)^{ab} = (\tau \nabla N^a, \nabla N^b),$$

$$\mathsf{S}_z(\tau)_j^{ab} = (\tau \partial_j N^a, N^b),$$

$$\mathsf{C}(\mathsf{U}^{n+\theta})_{ij}^{ab} = (N^a, \boldsymbol{u}_h^{n+\theta} \cdot \nabla N^b)\delta_{ij},$$

$$\mathsf{F}_i^a = (N^a, f_i).$$

It is understood that all the arrays are matrices (except F, which is a vector) whose components are obtained by grouping together the left indexes in the previous expressions (a and possibly i) and the right indexes (b and possibly j). Likewise, (27) needs to be modified to account for the Dirichlet boundary conditions (matrix G can be replaced by $-\mathsf{D}^t$ when this is done).

5.2. Fractional step scheme

Even though fractional step schemes are often introduced at the continuous level, the splitting can also be done for the algebraic nonlinear system arising after the spatial discretization, in our case (27)–(30). This is the approach advocated in [27,28] and that we will follow here. It allows in particular to obviate the controversial issue of boundary conditions for the intermediate velocity to be introduced.

Eq. (27) is exactly equivalent to the system

$$\mathsf{M}\frac{1}{\delta t}(\tilde{\mathsf{U}}^{n+1} - \mathsf{U}^n) + \mathsf{K}(\mathsf{U}^{n+\theta})\mathsf{U}^{n+\theta} + \mathsf{G}\mathsf{P}^n + \mathsf{S}_u(\tau; \mathsf{U}^{n+\theta})\mathsf{U}^{n+\theta} - \mathsf{S}_y(\tau; \mathsf{U}^{n+\theta})\mathsf{Y}^{n+\theta} = \mathsf{F}^{n+\theta}, \tag{31}$$

$$\mathsf{M}\frac{1}{\delta t}(\mathsf{U}^{n+1} - \tilde{\mathsf{U}}^{n+1}) + \mathsf{G}(\mathsf{P}^{n+\theta} - \mathsf{P}^n) = 0, \tag{32}$$

where $\tilde{\mathsf{U}}^{n+1}$ is an auxiliary variable. If the solution of the discrete problem behaves as we could expect, form (32) we see that the difference between U^{n+1} and $\tilde{\mathsf{U}}^{n+1}$ will be of order $\mathcal{O}(\delta t^2)$. At this point we can make the essential approximation of replacing $\mathsf{U}^{n+\theta}$ by $\tilde{\mathsf{U}}^{n+\theta} := \theta\tilde{\mathsf{U}}^{n+1} + (1-\theta)\mathsf{U}^n$ in (31) and also in (29). This should not alter the accuracy of the time integration scheme, which is at most $\mathcal{O}(\delta t^2)$. Likewise, we can express $\mathsf{U}^{n+\theta}$ in terms of $\tilde{\mathsf{U}}^{n+\theta}$ using (32) and insert the result in (28), which yields

$$\delta t \mathsf{D}\mathsf{M}^{-1}\mathsf{G}(\mathsf{P}^{n+\theta} - \mathsf{P}^n) - \mathsf{S}_p(\tau)\mathsf{P}^{n+\theta} + \mathsf{S}_z(\tau)\mathsf{Z}^{n+\theta} - \mathsf{D}\tilde{\mathsf{U}}^{n+\theta} = 0. \tag{33}$$

At this point, it is very convenient to make a further approximation. Observe that $\mathsf{D}\mathsf{M}^{-1}\mathsf{G}$ represents an approximation to the Laplacian operator. In order to avoid dealing with this matrix (which is computationally feasible only if M is approximated by a diagonal matrix), we can approximate

$$\mathsf{D}\mathsf{M}^{-1}\mathsf{G} \approx \mathsf{L}, \quad \text{with components} \quad \mathsf{L}^{ab} = -(\nabla N^a, \nabla N^b). \tag{34}$$

Matrix L is the standard approximation to the Laplacian operator. Clearly, this approximation is only possible when continuous pressure interpolations are employed.

After using approximation (34) in (33) the problem to be solved is:

$$\mathsf{M}\frac{1}{\delta t}(\tilde{\mathsf{U}}^{n+1} - \mathsf{U}^n) + \mathsf{K}(\tilde{\mathsf{U}}^{n+\theta})\tilde{\mathsf{U}}^{n+\theta} + \mathsf{G}\mathsf{P}^n + \mathsf{S}_u(\tilde{\tau}^{n+\theta}; \tilde{\mathsf{U}}^{n+\theta})\tilde{\mathsf{U}}^{n+\theta} - \mathsf{S}_y(\tilde{\tau}^{n+\theta}; \tilde{\mathsf{U}}^{n+\theta})\mathsf{Y}^{n+\theta} = \mathsf{F}^{n+\theta}, \tag{35}$$

$$\mathsf{M}\mathsf{Y}^{n+\theta} - \mathsf{C}(\tilde{\mathsf{U}}^{n+\theta})\tilde{\mathsf{U}}^{n+\theta} = 0, \tag{36}$$

$$\delta t \mathsf{L}(\mathsf{P}^{n+\theta} - \mathsf{P}^n) - \mathsf{S}_p(\tilde{\tau}^{n+\theta})\mathsf{P}^{n+\theta} + \mathsf{S}_z(\tilde{\tau}^{n+\theta})\mathsf{Z}^{n+\theta} - \mathsf{D}\tilde{\mathsf{U}}^{n+\theta} = 0, \tag{37}$$

$$\mathsf{M}\mathsf{Z}^{n+\theta} - \mathsf{G}\mathsf{P}^{n+\theta} = 0, \tag{38}$$

$$\mathsf{M}\frac{1}{\delta t}(\mathsf{U}^{n+1} - \tilde{\mathsf{U}}^{n+1}) + \mathsf{G}(\mathsf{P}^{n+\theta} - \mathsf{P}^n) = 0, \tag{39}$$

where $\tilde{\tau}^{n+\theta}$ indicates that the parameter τ is computed with the intermediate velocity $\tilde{\mathsf{U}}^{n+\theta}$. These equations have been written in the order they can be solved: first, one can solve system (35) and (36) to obtain $\tilde{\mathsf{U}}^{n+1}$ (and also Y^{n+1}), then system (37) and (38) allows us to obtain P^{n+1} (and Z^{n+1}) and finally (39) yields the end-of-step velocity U^{n+1}.

Problem (35)–(39) is the fractional step version of the stabilized finite element method we propose. The stability properties when Y is neglected (that is to say, when convection is not stabilized) are fully discussed in [9].

5.3. Monolithic iterative scheme

System (35)–(39) is nonlinear, and therefore the first step to solve it is to linearize it. Likewise, variables Y and Z are coupled with $\tilde{\mathsf{U}}$ and P, respectively, and this makes the formulation expensive from the computational point of view. There is the possibility of uncoupling these variables by using a block-iterative scheme. In the same iterative loop we can deal with the linearization of the convective term in the momentum equation (35) and the stabilization terms (those multiplied by the parameter τ), although there is of course the possibility to use nested loops.

Denoting by a superscript the iteration counter, the linearized form of system (35)–(39) we propose is:

$$\mathsf{M}\frac{1}{\delta t}(\tilde{\mathsf{U}}^{n+1,i+1} - \mathsf{U}^n) + \mathsf{K}(\tilde{\mathsf{U}}^{n+\theta,i})\tilde{\mathsf{U}}^{n+\theta,i+1} + \mathsf{G}\mathsf{P}^n + \mathsf{S}_u(\tilde{\tau}^{n+\theta,i};\tilde{\mathsf{U}}^{n+\theta,i})\tilde{\mathsf{U}}^{n+\theta,i+1} - \mathsf{S}_y(\tilde{\tau}^{n+\theta,i};\tilde{\mathsf{U}}^{n+\theta,i})\mathsf{Y}^{n+\theta,i} = \mathsf{F}^{n+\theta}, \tag{40}$$

$$\mathsf{M}\mathsf{Y}^{n+\theta,i+1} - \mathsf{C}(\tilde{\mathsf{U}}^{n+\theta,i+1})\tilde{\mathsf{U}}^{n+\theta,i+1} = 0, \tag{41}$$

$$\delta t \mathsf{L}(\mathsf{P}^{n+\theta,i+1} - \mathsf{P}^n) - \mathsf{S}_p(\tilde{\tau}^{n+\theta,i+1})\mathsf{P}^{n+\theta,i+1} + \mathsf{S}_z(\tilde{\tau}^{n+\theta,i+1})\mathsf{Z}^{n+\theta,i} - \mathsf{D}\tilde{\mathsf{U}}^{n+\theta,i+1} = 0, \tag{42}$$

$$\mathsf{M}\mathsf{Z}^{n+\theta,i+1} - \mathsf{G}\mathsf{P}^{n+\theta,i+1} = 0, \tag{43}$$

$$\mathsf{M}\frac{1}{\delta t}(\mathsf{U}^{n+1,i+1} - \tilde{\mathsf{U}}^{n+1,i+1}) + \mathsf{G}(\mathsf{P}^{n+\theta,i+1} - \mathsf{P}^n) = 0. \tag{44}$$

These equations are all linear and uncoupled, that is to say, they can be solved successively.

All the arguments that led us to the fractional step scheme (35)–(39) are valid if instead of using the pressure P^n we replace it by the any other pressure. In particular, in the iterative scheme (40)–(44) we can replace it by $\mathsf{P}^{n+\theta,i}$. If the resulting iterative scheme converges, the second term in the left-hand-side of (44) will disappear, and therefore *the intermediate velocity will converge to the end-of-step one*. Thus, we do not need to distinguish between $\tilde{\mathsf{U}}$ and U and (44) can be simply ignored. The final iterative scheme is

$$\begin{aligned}\mathsf{M}\frac{1}{\delta t}(\mathsf{U}^{n+1,i+1} - \mathsf{U}^n) + \mathsf{K}(\mathsf{U}^{n+\theta,i})\mathsf{U}^{n+\theta,i+1} + \mathsf{G}\mathsf{P}^{n+\theta,i} + \mathsf{S}_u(\tau^{n+\theta,i};\mathsf{U}^{n+\theta,i})\mathsf{U}^{n+\theta,i+1} \\ - \mathsf{S}_y(\tau^{n+\theta,i};\mathsf{U}^{n+\theta,i})\mathsf{Y}^{n+\theta,i} = \mathsf{F}^{n+\theta},\end{aligned} \tag{45}$$

$$\mathsf{M}\mathsf{Y}^{n+\theta,i+1} - \mathsf{C}(\mathsf{U}^{n+\theta,i+1})\mathsf{U}^{n+\theta,i+1} = 0, \tag{46}$$

$$\delta t\mathsf{L}(\mathsf{P}^{n+\theta,i+1} - \mathsf{P}^{n+\theta,i}) - \mathsf{S}_p(\tau^{n+\theta,i+1})\mathsf{P}^{n+\theta,i+1} + \mathsf{S}_z(\tau^{n+\theta,i+1})\mathsf{Z}^{n+\theta,i} - \mathsf{D}\mathsf{U}^{n+\theta,i+1} = 0, \tag{47}$$

$$\mathsf{M}\mathsf{Z}^{n+\theta,i+1} - \mathsf{G}\mathsf{P}^{n+\theta,i+1} = 0. \tag{48}$$

Apparently, this is a straightforward iteration procedure for solving the original *monolithic* problem (27)–(30) freezing the pressure gradient in the momentum equation. However, there is a term whose presence would be hardly motivated by looking only at this system, namely, the term $\delta t\mathsf{L}(\mathsf{P}^{n+\theta,i+1} - \mathsf{P}^{n+\theta,i})$. The motivation to introduce it comes from the inspection of what happens in the fractional step scheme we have described, even though now we aim to converge to the original nonsplit problem (27)–(30).

The convective term in (40) and (45) could be easily linearized using the second order Newton–Raphson scheme instead of the fixed point method we have used. However, for the stabilization terms the latter is the simplest strategy, unless numerical differentiation is employed. Moreover, the block-iterative coupling proposed for Y and Z would lead anyway to at most linear convergence of the iterative scheme.

We have found the iterative procedure (45)–(48) very efficient (only scalar equations need to be solved) *and robust*. Without the term $\delta t\mathsf{L}(\mathsf{P}^{n+\theta,i+1} - \mathsf{P}^{n+\theta,i})$ in (47) convergence turns out to be much harder. We first used this scheme in [31] (see also [33]). Some more numerical results are presented in the following section.

5.4. Preconditioning for very anisotropic meshes

Apart from the need of properly defining the element size to be used in the expression of the stabilization parameters, another implication of the use of (very) anisotropic finite element meshes is the poor conditioning of the matrix of the final algebraic system to be solved at each iteration of each time step. This may cause serious problems to the convergence of iterative schemes to solve this system, often making them fail to converge.

In this section, we describe the basic idea of the preconditioner proposed in [32] that we have used in the numerical experiments to check the behavior of the stabilized finite element method presented here. In particular, we have used it for pressure equation (47), whose matrix is symmetric and positive definite. We have used the Conjugate Gradient algorithm to solve it.

Let $\boldsymbol{A}$ be the matrix of the system to be solved and let $\boldsymbol{P}$ be the preconditioner we are looking for. The way to construct $\boldsymbol{P}$ is as follows. First, the nodal points of the finite element mesh are grouped to form "lines" *following the direction normal to the grid stretching*. Then, the preconditioner is built by assembling the diagonal entries of the system matrix A_{ii} and also the nondiagonal entries A_{ij}, $i \neq j$, corresponding to nodes of edges belonging to the linelets. An important condition for the final structure of the preconditioner is that a nodal point can only belong to one linelet, i.e., a linelet does not cross any other one. Thus, if the nodal points are renumbered following the linelets, the preconditioning matrix associated to the degrees of freedom belonging to a line is tri-diagonal. In addition, it can be shown that diagonal preconditioning automatically holds for the degrees of freedom associated to the points of the mesh that do not belong to any linelet. For details and references, see [32].

Concerning the way to solve the other equations in system (45)–(48), we solve the projection equations (46) and (48) either by using a diagonal mass matrix obtained from a closed quadrature rule or by using this matrix as a preconditioner of a Jacobi iteration to converge to the solution with the consistent mass matrix (very few iterations are required to converge with very stringent tolerances). The first option has been used in the numerical examples presented next. Finally, system (45) is solved using a standard GMRES solver either with diagonal scaling or with ILU preconditioner. Again, the option used in the following examples is the former.

6. Numerical examples

In this section we present the results of two classical numerical experiments we have performed using the stabilized formulation described in this paper. The goal is to compare the effect of different ways to compute the element length appearing in the expression of the stabilization parameters when the mesh is anisotropic. We can anticipate that the quality of the solution in all the cases is very good, in the sense that no pressure and velocity oscillations occur. The stabilization method as such, regardless of the way to compute the characteristic element length, performs well in allowing equal velocity–pressure interpolation and dealing with high element Reynolds numbers.

6.1. Flow over a cylinder

This example involves the flow past a cylinder, a widely solved benchmark problem. The computational domain is $\bar{\Omega} = [0, 15] \times [0, 10] \setminus D$, 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 $y = 10$ 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 $x = 15$. The Reynolds number is 100, based on the cylinder diameter and the prescribed inflow velocity. The finite element mesh employed consists of 3636 linear triangles, with 1888 nodal points. The stretching in the boundary layer is 1:2400.

To integrate the flow equations in time we have used the Crank–Nicholson scheme with a time step $\delta t = 0.05$ (the first 10 time steps have been solved with the backward Euler method).

The results shown in Figs. 2–5 correspond to $t = 3000$, when the vortex shedding is fully developed. Figs. 2 and 3 show the pressure contours and the streamlines, respectively.

Fig. 4 shows the evolution of the pressure vertical forces on the cylinder for different expressions of h. It can be seen that the most dissipative results are obtained when the characteristic element length is computed as the maximum length of the element edges. The other three possibilities yield similar results. The highest amplitude and frequency, and thus the minimum dissipation, is obtained when h is computed in the flow direction. A little more dissipative are the solutions obtained with h computed as indicated in [18] and taking the minimum length of the edges.

The linelets used in the construction of the preconditioner to solve the pressure equation in this particular example are shown in Fig. 5. It can be seen that they grow from the cylinder boundary, where the most stretched elements are located.

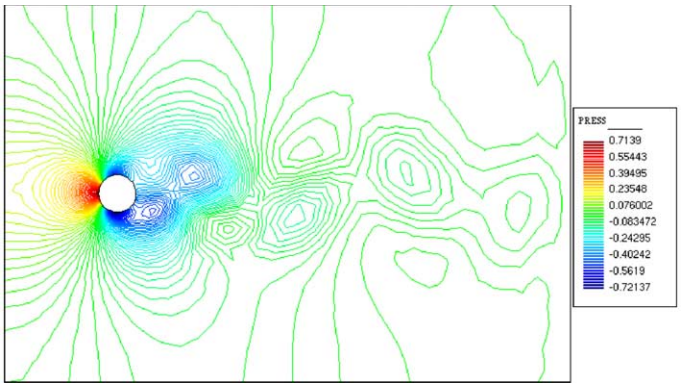

Fig. 2. Pressure.

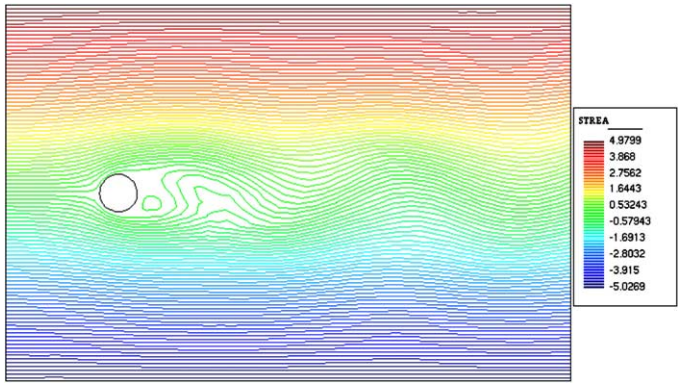

Fig. 3. Streamlines.

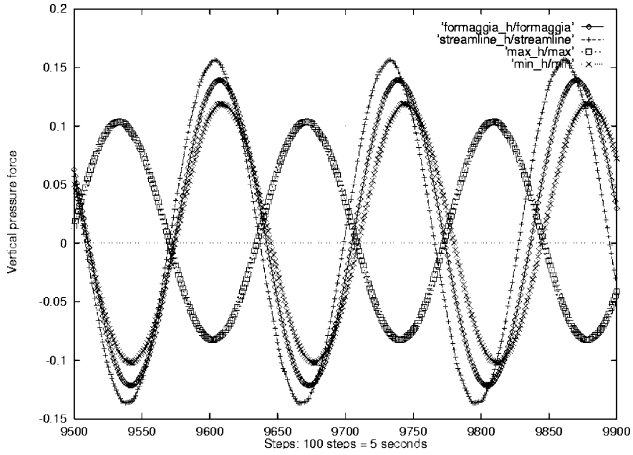

Fig. 4. Evolution of the pressure vertical forces on the cylinder.

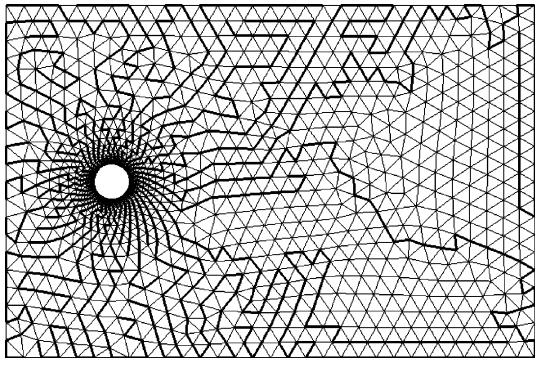

Fig. 5. Linelets.

The mean number of iterations of the Conjugate Gradient algorithm to solve Eq. (47) within each iteration of each time step has been 174 to reach a tolerance of 10^{-8}, whereas with no preconditioning other than a diagonal scaling 10 000 iterations were required to reach a residual of the order of 10^{-3}.

6.2. Flow over a backward facing step

This second example is the classical benchmark of a flow over a backward facing step. The length of the inflow channel is 4 and its width 1, the total length of the computational domain 40 and the width of the channel 2. A parabolic velocity profile with maximum value 1 is prescribed on the inflow, whereas the no-slip condition is prescribed on the rest of the walls except the outflow, where a zero traction condition is fixed. The Reynolds number, computed with a velocity 2/3 and the step height 1 is 1000.

The finite element mesh employed consists of 16 472 linear triangles and 8542 nodal points. The stretching on the boundary limit is about 1:40. The flow equations where advanced in time with the backward Euler method and a Courant number of 100 to obtain the steady state solution, which was assumed to be reached when $\|\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}\| \leqslant 10^{-3}\|\boldsymbol{u}^{n+1}\|\delta t$.

Figs. 6 and 7 show the pressure contours and the streamlines for this example using different ways to compute h. It can be observed that the results are all very similar. Looking at the maximum pressure difference, the highest value in this case is obtained with h computed as indicated in [18], followed by h

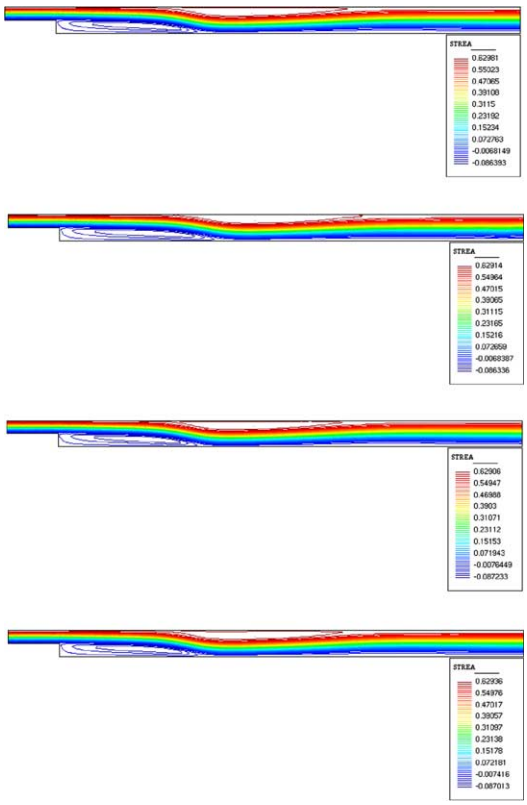

Fig. 6. Streamlines. From the top to the bottom: h in the direction of the flow, h proposed by Formaggia and Perotto [19], maximum length of the element edges and minimum length of the element edges.

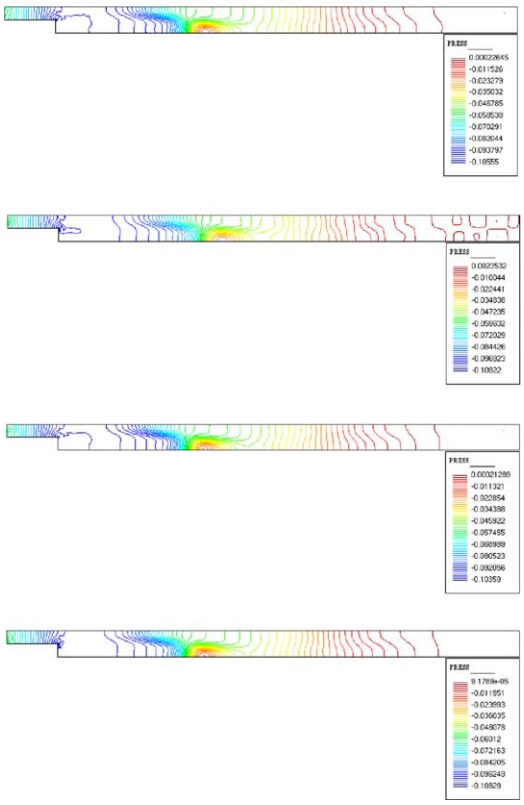

Fig. 7. Pressure. From the top to the bottom: h in the direction of the flow, h proposed by Formaggia and Perotto [19], maximum length of the element edges and minimum length of the element edges.

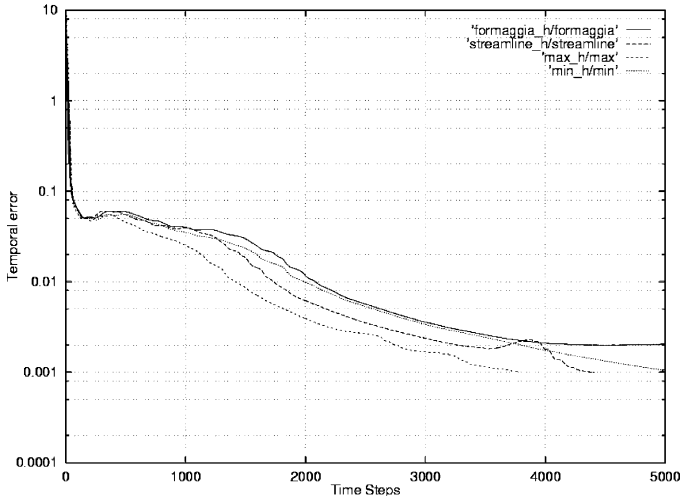

Fig. 8. Convergence to the steady-state for the flow over a backward facing step.

computed following the streamlines. This gives an indication that these two are the less dissipative alternatives for h, although in this case the conclusion is not so clear as in the previous example.

Fig. 8 shows the convergence towards the steady state. Again, the behavior obtained for the different ways to compute h is very similar. The slowest in the case corresponds to (24), whereas the fastest is obtained taking the maximum length of the element edges. This again is representative of how dissipative is the scheme.

In this example, the mean number of iterations of the Conjugate Gradient algorithm to solve Eq. (47) within each iteration of each time step has been 212 to reach a tolerance of 10^{-8}, whereas no convergence to the prescribed tolerance was obtained without preconditioning.

7. Conclusions

We have presented a finite element scheme for the incompressible Navier–Stokes equations with two main features: it incorporates a stabilization procedure and an iterative scheme to deal with the pressure. We have found this iterative procedure, given by (45)–(48), very efficient (only scalar equations need to be solved) *and robust*. Without the term $\delta t \mathsf{L}(\mathsf{P}^{n+\theta,i+1} - \mathsf{P}^{n+\theta,i})$ in (47) convergence turns out to be much harder. In fact, for the two numerical examples presented it was impossible to obtain convergence within each time step without this term, even though this point should be further analyzed (particularly in terms of the time step size and the viscosity, which are the two parameters most relevant to nonlinear convergence). We first used this scheme in [31] (see also [33]), where other numerical simulations using it can be found.

We have also discussed different ways to compute the element length that appears in the stabilization parameters. As it could be expected, it is directly related with the numerical dissipation of the scheme. It is important to remark, though, that the two choices that yield less dissipation, namely, (23) and (24), still succeed in stabilizing the pressure and allow to deal with convection dominated flows.

References

[1] T. Apel, G. Lube, Anisotropic mesh refinement for a singularly perturbed reaction diffusion model problem, Appl. Numer. Math. 26 (1998) 415–433.
[2] Th. Apel, G. Lube, Anisotropic mesh refinement in stabilized Galerkin methods, Numer. Math. 74 (1996) 261–282.
[3] C. Baiocchi, F. Brezzi, L.P. Franca, Virtual bubbles and Galerkin/least-squares type methods (Ga.L.S), Comput. Methods Appl. Mech. Engrg. 105 (1993) 125–141.
[4] P. Bochev, Z. Cai, T.A. Manteuffel, S.F. McCormick, Analysis of velocity-flux first-order system least-squares principles for the Navier–Stokes equations: Part I, SIAM J. Numer. Anal. 35 (1998) 990–1009.
[5] F. Brezzi, J. Douglas, Stabilized mixed methods for the Stokes problem, Numer. Math. 53 (1988) 225–235.
[6] F. Brezzi, L.P. Franca, T.J.R. Hughes, A. Russo, $b = \int g$, Comput. Methods Appl. Mech. Engrg. 145 (1997) 329–339.
[7] A.J. Chorin, A numerical method for solving incompressible viscous problems, J. Computat. Phys. 2 (1967) 12–26.
[8] R. Codina, Stabilization of incompressibility and convection through orthogonal subscales in finite element methods, Comput. Methods Appl. Mech. Engrg. 190 (2000) 1579–1599.
[9] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Computat. Phys. 170 (2001) 112–140.
[10] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Comput. Methods Appl. Mech. Engrg. 190 (2001) 2681–2706.
[11] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Engrg. 191 (2002) 4295–4321.
[12] R. Codina, Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales, submitted for publication.
[13] R. Codina, J. Blasco, A finite element formulation for the Stokes problem allowing equal velocity–pressure interpolation, Comput. Methods Appl. Mech. Engrg. 143 (1997) 373–391.

[14] R. Codina, J. Blasco, Analysis of a pressure-stabilized finite element approximation of the stationary Navier–Stokes equations, Numer. Math. 87 (2000) 59–81.
[15] R. Codina, J. Blasco, Analysis of a stabilized finite element approximation of the transient convection–diffusion–reaction equation using orthogonal subscales, Comput. Visualization Sci. 4 (2002) 167–174.
[16] R. Codina, E. Oñate, M. Cervera, The intrinsic time for the streamline upwind/Petrov–Galerkin formulation using quadratic elements, Comput. Methods Appl. Mech. Engrg. 94 (1992) 239–262.
[17] J. Douglas, J. Wang, An absolutely stabilized finite element method for the Stokes problem, Math. Computat. 52 (1989) 495–508.
[18] L. Formaggia, S. Micheleti, S. Perotto, Anisotropic mesh adaption with applications to CFD problems, in: H.A. Mang, F.G. Rammerstorfer, J. Eberhardsteiner (Eds.), Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V), July 7–12, 2002, Vienna, Austria. Vienna University of Technology, Austria, ISBN 3-9501554-0-6, Available from <http://wccm.tuwien.ac.at>.
[19] L. Formaggia, S. Perotto, Anisotropic error estimates for elliptic problems, Numer. Math. 94 (2003) 67–92.
[20] M. Fortin, S. Boivin, Iterative stabilization of the bilinear velocity–constant pressure element, Int. J. Numer. Methods Fluids 10 (1990) 125–140.
[21] L. Franca, R. Stenberg, Error analysis of some Galerkin least-squares methods for the elasticity equations, SIAM J. Numer. Anal. 28 (1991) 1680–1697.
[22] 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, Comput. Methods Appl. Mech. Engrg. 105 (1993) 285–298.
[23] T.J.R. Hughes, Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations, Comput. Methods Appl. Mech. Engrg. 127 (1995) 387–401.
[24] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1998) 3–24.
[25] T.J.R. Hughes, L.P. Franca, M. Balestra, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: a stable Petrov–Galerkin formulation for the Stokes problem accommodating equal-order interpolations, Comput. Methods Appl. Mech. Engrg. 59 (1986) 85–99.
[26] R. Natarajan, A numerical method for incompressible viscous flow simulation, J. Computat. Phys. 100 (1992) 384–395.
[27] J.B. Perot, An analysis of the fractional step method, J. Computat. Phys. 108 (1993) 51–58.
[28] A. Quarteroni, F. Saleri, A. Veneziani, Factorization methods for the numerical approximation of Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 188 (2000) 505–526.
[29] J. Shen, Hopf bifurcation of the unsteady regularized driven cavity-flow, J. Computat. Phys. 95 (1991) 228–245.
[30] P.J. Silvester, N. Kechkar, Stabilised bilinear-constant velocity–pressure finite elements for the conjugate gradient solution of the Stokes problem, Comput. Methods Appl. Mech. Engrg. 79 (1990) 71–86.
[31] O. Soto, R. Codina, A numerical model for mould filling using a stabilized finite element method and the VOF technique, Int. J. Numer. Methods Fluids, submitted for publication.
[32] O. Soto, R. Löhner, F. Camelli, A linelet preconditioner for incompressible flow solvers, Int. J. Numer. Methods Heat Fluid Flow 13 (2003) 133–147.
[33] O. Soto, R. Löhner, J. Cebral, R. Codina. A time-accurate implicit-monolithic finite element scheme for incompressible flow problems. in: Eccomas CFD 2001, CD Proceedings, 2001.
[34] R. Temam, Sur l'approximation de la solution deséquations de Navier–Stokes par la méthode des pas fractionaires (I), Arch. Rational Mech. Anal. 32 (1969) 135–153.
[35] S. Turek, A comparative study of time-stepping techniques for the incompressible Navier–Stokes equations: from fully implicit nonlinear schemes to semi-implicit projection methods, Int. J. Numer. Methods Fluids 22 (1996) 987–1011.