International Journal of Computational Fluid Dynamics
Vol. 22, No. 3, March 2008, 135–152

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

A variational subgrid scale model for transient incompressible flows

G. Houzeaux[a]* and J. Principe[b]

[a]*Barcelona Supercomputing Center, Campus Nord UPC, Edificio C6 – E201, Jordi Girona 1-3, 08034 Barcelona, Spain*
[b]*Universitat Politècnica de Catalunya Campus Nord UPC, Edificio C1, Jordi Girona 1-3, 08034 Barcelona, Spain*

(*Received 4 July 2007; final version received 26 October 2007*)

We introduce in this paper a variational subgrid scale model for the solution of the incompressible Navier–Stokes equations. With respect to classical multiscale-based stabilisation techniques, we retain the subgrid scale effects in the convective term and integrate the subgrid scale equation in time. The method is applied to the Navier–Stokes equations in an accelerating frame of reference and with Dirichlet (essential), Neumann (natural) and mixed boundary conditions. The concrete objective of the paper is to test a numerical algorithm for solving the non-linear subgrid scale equation and the introduction of the subgrid scale into the grid scale equation. The performance of the technique is demonstrated through the solution of two numerical examples: one to test the tracking of the subgrid scale in the convection term and the other to investigate the effects of considering the subgrid scale transient.

Keywords: Navier–Stokes equations; incompressible flow; stabilisation technique; finite element; multiscale method

1. Introduction

Variational subgrid scale techniques are widely used today for the solution of the incompressible Navier–Stokes equations. They are also being considered as a possible way of modelling turbulence phenomena at the numerical level in a similar way as large eddy simulation techniques do (Calo 2004, Koobus and Farhat 2004, Codina *et al.* 2007). A step in this direction is to consider the subgrid scale time dependent and to consider its effect on the non-linear convective term. Without considering its potential use as a way of modelling turbulence, this idea leads to important improvements on the discrete formulation of the problem. From a theoretical point of view, the use of transient subgrid scales explains how the stabilisation parameter should depend on the time step size and makes space and time discretisations commutative and the tracking of the subscales along the non-linear process provides global momentum conservation (Codina 2002, Codina *et al.* 2007). The paper does not intend to devise a new subgrid scale model, but rather to develop a numerical algorithm and to test its convergence and accuracy. The present subgrid scale model is based on Codina and Blasco (2002) which was developed for the advection–diffusion–reaction equation and applied here to the solution of the stationary and transient Navier–Stokes equations. It is shown that the method does not only provide the necessary stabilisation of the formulation but also enables to obtain more accurate solutions than the classical GLS approach for an equivalent mesh. The authors first focus on reviewing how this method fits into the global picture of stabilisation techniques, which is done in the next section. After presenting the governing equations of the problem in the third section, the Galerkin formulation as well as the time discretisation are presented in the fourth section. In the fifth section, the multiscale concept is applied to our problem. The system of equations for the resolved and subgrid scales are derived. Then, two possible ways of approximating the subgrid scale are proposed; the first one coincides with the usual GLS method for linear elements, while the second one tracks the effects of the subgrid scale in all terms of the equations. We also focus on the numerical implementation of the method which is done in the sixth section. In the last section, we solve two numerical examples showing the main characteristics of the method.

2. Stabilisation techniques

2.1 *Classical formulations*

The classical stabilised finite element formulations can be understood in the context of the advection–diffusion–reaction (ADR) equation:

$$\mathcal{L}(u) := -\varepsilon\Delta u + \mathbf{a}\cdot\nabla u + su = f \quad \text{in } \Omega, \qquad (1)$$

where $\varepsilon > 0$ and $s \geq 0$ are constant, $\mathbf{a}$ is divergence free, and Ω is a two or three dimensional domain. This equation should be provided with boundary conditions.

*Corresponding author. Email: guillaume.houzeaux@bsc.es

ISSN 1061-8562 print/ISSN 1029-0257 online
© 2008 Taylor & Francis
DOI: 10.1080/10618560701816387
http://www.informaworld.com

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Let $\{K\}$ be a regular finite element partition $\mathcal{P}_h$ of the domain Ω, with index K ranging from 1 to the number of elements. The diameter of $\{K\}$ will be denoted by h. And let us construct the functional linear spaces from the previous partition so that the resulting finite element approximation is said to be *conforming*. Given two functions u and v we define (u, v) as the L^2-inner product. Also, $|| \ ||_\infty$, $|| \ ||_0$ and $|| \ ||_1$ are the L^∞, L^2 and H^1 norms in Ω, respectively.

Assuming homogeneous Dirichlet and Neumann boundary conditions, the discrete Galerkin formulation of the problem Quarteroni and Valli (1994) consists of finding u_h in the appropriate space U_h such that

$$a(u_h, v_h) = (f, v_h) \quad v_h \in V_h, \tag{2}$$

where V_h is an appropriate test function space and the bilinear form a is defined as

$$a(w, v) := \varepsilon(\nabla w, \nabla v) + (\mathbf{a}\cdot\nabla w, v) + (sw, v). \tag{3}$$

It is well known that the Galerkin method lacks stability (Johnson 1987, Quarteroni and Valli 1994, Roos *et al.* 2007). The effects of the stabilisation techniques consist of the addition of a stabilisation term $S(u_h, v_h)$ to the original equation such that the stabilised system reads:

$$a(u_h, v_h) + S(u_h, v_h) = (f, v_h). \tag{4}$$

We define the residual of the equation as

$$R(u_h) := f - L(u_h). \tag{5}$$

Table 1 shows the expression of the stabilisation term $S(u_h, v_h)$ for the main stabilisation methods used in the literature, where τ is a stabilisation parameter that can depend on the element size h and the equation coefficients, and where

$$\int_{\Omega'} (\cdot) \mathrm{d}\Omega := \sum_K \int_K (\cdot) \mathrm{d}\Omega$$

is the integral over all the elements.

Table 1. Stabilisation term $S(u_h, v_h)$ of common stabilisation methods.

Method	Stabilisation term $S(u_h, v_h)$
Non-consistent	
Artificial viscosity (AV)	$-\int_{\Omega'} \nabla v_h \cdot \tau \nabla u_h \mathrm{d}\Omega$
Streamline upwind (SU)	$-\int_{\Omega'} (a\cdot\nabla v_h)\tau(a\cdot\nabla u_h)\mathrm{d}\Omega$
Consistent	
SU Petrov-Galerkin (SUPG)	$-\int_{\Omega'} (a\cdot\nabla v_h)\tau R(u_h)\mathrm{d}\Omega$
Galerkin Least-Square (GLS)	$-\int_{\Omega'} L(v_h)\tau R(u_h)\mathrm{d}\Omega$
Douglas–Wang (DW)	$-\int_{\Omega'} L^*(v_h)\tau R(u_h)\mathrm{d}\Omega$
Variational Multiscale	
Algebraic Models (ASGS)	$-\int_{\Omega'} L^*(v_h)\tilde{u}\mathrm{d}\Omega$ with $\tilde{u} = \tau R(u_h)$

2.2 Multiscale concept

Conceptually, the subgrid scale (SGS) method is based on enlarging some finite element space by adding information about the part of the solution of a variational problem that cannot be resolved by the computational grid. The first motivation for considering such an enrichment is to take into account phenomena that take place at scales smaller than that of the discretisation but, nevertheless, relevant to the overall response of the physical system under study. This is the case of problems like turbulence (Calo 2004) in fluid dynamics, and strain localisation (Garikipati and Hughes 1998) or the homogenisation of composite materials (How and Wu 1997) in solid mechanics. A second motivation has to do with spurious numerical effects due to the poor performance of the discrete model when dealing with the smallest scales that the computational grid is not able to capture. Perhaps the most egregious example is the sub-diffusivity introduced by the discretisation of the convective-dominated diffusion equation that has been analysed in the previous section. As a consequence, some stabilisation techniques must be used in order to obtain meaningful solutions. As mentioned above, it can be shown that, in essence, many of these techniques can be explained within the framework of the subgrid scale method.

The way to recover the stabilised formulation using the subgrid scale approach is the following. First we decompose additively the exact solution u into the resolved scale u_h, the one associated with the computational grid, and a subgrid scale term $\tilde{u}$ so that the exact solution u is their sum:

$$u = u_h + \tilde{u}. \tag{6}$$

As the SGS method explicitly splits the exact solution into two components, it is usually referred to as a *two-level multiscale* method. In the literature, these two scales have been referred to using different names; let us mention the following:

$$\begin{aligned} u_h &: \text{grid scale, coarse scale, resolved scale,} \\ \tilde{u} &: \text{subgrid scale, fine scale, unresolved scale.} \end{aligned} \tag{7}$$

A very interesting feature of the SGS method is that, on the one hand, a high level of generality and abstraction of the formulation can be attained. On the

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

other hand, its main specific instances, for example that based on residual free bubbles, can also be quite convenient in practice from a computational point of view. The fact that the subscale degrees of freedom can be condensed at the element level is appealing not only in terms of computational cost but also in terms of respecting the architecture of standard finite element codes.

Let us consider the abstract linear problem: find $u \in U$ such that

$$\mathcal{L}(u) = f, \tag{8}$$

where $\mathcal{L}$ is some linear differential operator and U is some suitable function space, which can be decomposed into a resolved and unresolved component such that $U = U_h \oplus \tilde{U}$. Consider the additive decomposition of the test function space $V = V_h \oplus \tilde{V}$, where V_h and $\tilde{V}$ are the test function spaces of the resolved and unresolved scales, respectively. The weak form of the abstract problem can be written as a system of equations: find $(u_h, \tilde{u}) \in U_h \times \tilde{U}$ such that

$$(\mathcal{L}(u_h + \tilde{u}), v_h + \tilde{v}) = (f, v_h + \tilde{v}) \quad \forall (v_h, \tilde{v}) \in V_h \times \tilde{V}. \tag{9}$$

By taking successively $v_h = 0$ and $\tilde{v} = 0$ we obtain the following system:

$$\begin{aligned} (\mathcal{L}(u_h), v_h) + (\mathcal{L}(\tilde{u}), v_h) &= (f, v_h) \quad \forall v_h \in V_h, \\ (\mathcal{L}(u_h), \tilde{v}) + (\mathcal{L}(\tilde{u}), \tilde{v}) &= (f, \tilde{v}) \quad \forall \tilde{v} \in \tilde{V}. \end{aligned} \tag{10}$$

The first equation is formally equivalent to

$$(\mathcal{L}(u_h), v_h) + (\tilde{u}, \mathcal{L}^*(v_h)) = (f, v_h) \quad \forall v_h \in V_h, \tag{11}$$

where $\mathcal{L}^*$ is the adjoint of $\mathcal{L}$.

Operating formally again and assuming $\mathcal{L}$ to be invertible, we can solve the second equation for $\tilde{u}$:

$$\tilde{u} = \mathcal{L}^{-1} R(u_h). \tag{12}$$

We can now substitute this expression into the previous one and obtain the equation for the resolved scale:

$$(\mathcal{L}(u_h), v_h) + (\mathcal{L}^{-1} R(u_h), \mathcal{L}^*(v_h)) = (f, v_h) \quad \forall v_h \in V_h. \tag{13}$$

Figure 1 illustrates how the subgrid scale effect is taken into account at the resolved scale level (that is at the nodes) in the latter equation.

Equation (12) is the solution for the subgrid scale. Obviously, $\mathcal{L}^{-1}$ is unknown. If it were not, then we would know the exact solution. Thus, an approximation to the inverse is to be found. The classical (historical)

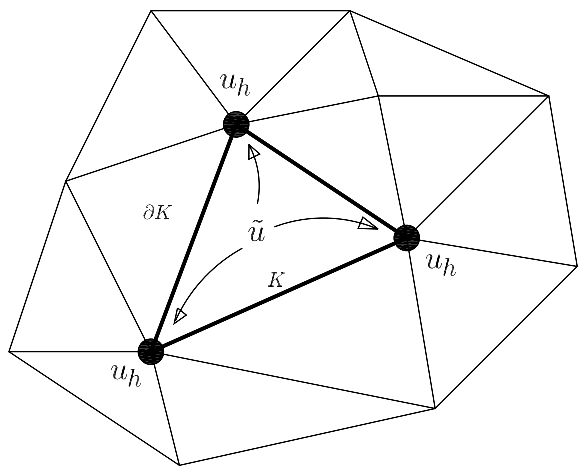


Figure 1. Schematic concept of the subgrid scale stabilisation (SGS).

way to approximate it is to substitute the differential operator by an algebraic one:

$$\mathcal{L}^{-1} \approx \tau_K, \tag{14}$$

where τ_K is evaluated in the element K. Such methods are referred to as *Algebraic Subgrid Scale method* (ASGS). For example, Codina and Blasco (2002) justified an expression for τ_K using a Fourier analysis and the associated ASGS stabilisation method models $\tilde{u}$ like

$$\tilde{u} \approx \left(c_1 \frac{\varepsilon}{h^2} + c_2 \frac{|\mathbf{a}|_\infty}{h} + s \right)^{-1} R(u_h), \tag{15}$$

with $c_1 = 4$ and $c_2 = 2$. The expression used in this work is based on the same arguments but applied to the Navier–Stokes equations (Codina 2002).

2.3 Beyond the classical approach

The SGS method as presented so far does not consider certain issues that are important for the realistic modelling of physical systems. We are especially, but not exclusively, interested in those that are relevant when dealing with the Navier–Stokes equations.

The SGS concept can be naturally extended to consider both time dependency and non-linearities (this was not the case of the GLS method, although it is recovered using the SGS concept for the ADR equation using linear interpolation). In this case, it is crucial to consider the subgrid scales for the evolution and consistent linearisation of the unknown. When used for stabilisation, this has an effect in the quality of the numerical solution. For the sake of clarity, let us consider the transient and non-linear cases separately. Another important issue is the choice of the space where the

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

subgrid scales are sought. The transient case, non-linear case, and the choice of the subgrid space are now briefly discussed. The method presented in this work for the Navier–Stokes equations retains the transient and non-linear tracking of the subgrid scale, but does not seek it in the orthogonal space.

2.3.1 *Transient case*

For the transient case, our starting problems consists of solving in a given time interval:

$$(\partial_t u, v) + (\mathcal{L}(u), v) = (f, v). \tag{16}$$

The corresponding system of equations for the resolved and subgrid scales is

$$\begin{aligned}(\partial_t u_h + \partial_t \tilde{u}, v_h) + (\mathcal{L}(u_h), v_h) + (\tilde{u}, \mathcal{L}^*(v_h)) &= (f, v_h),\\ (\partial_t u_h + \partial_t \tilde{u}, \tilde{v}) + (\mathcal{L}(u_h), \tilde{v}) + (\mathcal{L}(\tilde{u}), \tilde{v}) &= (f, \tilde{v}).\end{aligned} \tag{17}$$

The classical ASGS method does not retain the time derivative of the subgrid scale in none of the terms of latter equations, and yields 17. One possibility to take it into account is to solve the subgrid scale equation by approximating the inverse of $\mathcal{L}$ like in the stationary case:

$$\partial_t \tilde{u} + \tau_K^{-1} \tilde{u} = f - \partial_t u_h - \mathcal{L}(u_h), \tag{18}$$

and maintain the time derivatives of $\tilde{u}$ in the systems of equations.

2.3.2 *Non-linear case*

Consider a non-linear differential equation of the form

$$(\mathcal{L}(u, u), v) = (f, v). \tag{19}$$

To solve it numerically, it is customary to consider an iteration counter i and solve the following linearised equation for u^i:

$$(\mathcal{L}(u^{i-1}, u^i), v) = (f, v). \tag{20}$$

The system of equations for the resolved and subgrid scales reads:

$$\begin{aligned}(\mathcal{L}(u^{i-1}, u_h^i), v_h) + (\mathcal{L}(u^{i-1}, \tilde{u}^i), v_h) &= (f, v_h),\\ (\mathcal{L}(u^{i-1}, u_h^i), \tilde{v}) + (\mathcal{L}(u^{i-1}, \tilde{u}^i), \tilde{v}) &= (f, \tilde{v}).\end{aligned} \tag{21}$$

The classical approach simply substitutes u^{i-1} by u_h^{i-1}. The idea is now to consider the whole unknown, that is $u_h^{i-1} + \tilde{u}^{i-1}$ in both equations.

In the context of the Navier–Stokes equations, this means that the classical approach approximates the term $u^{i-1}\cdot\nabla u^i$ by $u_h^{i-1}\cdot\nabla(u_h^i + \tilde{u}^i)$. The correct approach consists of considering rather the following convection term: $(u_h^{i-1} + \tilde{u}^{i-1})\cdot\nabla(u_h^i + \tilde{u}^i)$. This may be a *very* important issue when trying to simulate turbulent flows using the multiscale approach (Codina *et al.* 2007), as the convective term is responsible for the existence of turbulent flow.

2.3.3 *Space of the subgrid scale*

The accurate modelling of the subscales is another crucial point for the overall performance of the method. The idea is to optimise the information furnished by the subgrid scales. In order to minimise redundancy, a natural requirement is to model the subscales as belonging to some function space that is orthogonal, in some sense, to the finite element part of the solution u_h. This idea has been exploited by Codina in (Codina and Blasco 2002, Codina 2002). First, the subgrid scale is written as

$$\tilde{u} = \tau_K R(u_h) + \tau_K u_{h,\mathrm{ort}}. \tag{22}$$

Then it is required to be orthogonal to all $v_h \in V_h$, that is

$$(\tilde{u}, v_h) = 0 \Rightarrow (\tau_K u_{h,\mathrm{ort}}, v_h) = -(\tau_K R(u_h), v_h). \tag{23}$$

The concept of the *Orthogonal Subgrid Scale* (OSS) stabilisation is illustrated in Figure 2.

3. Physical problem

We consider the incompressible Oseen and Navier–Stokes equations in an accelerated frame of reference in

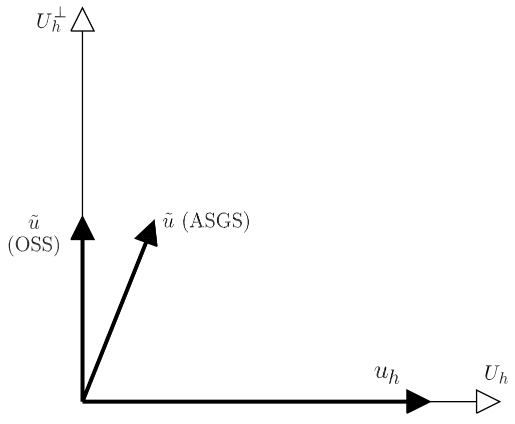


Figure 2. Schematic concept of the orthogonal subgrid scale (OSS) stabilisation.

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

a domain Ω of R^d with $d = 2$ or 3 (Batchelor 1970). Let μ be the viscosity of the fluid which is not necessarily constant, and ρ its density, assumed to be constant.

Let u be the velocity and p the mechanical pressure. The velocity strain rate $\varepsilon(\mathbf{u})$ and the stress tensor σ are

$$\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^t), \quad \sigma = -p\mathrm{I} + 2\mu\varepsilon(\mathbf{u}),$$

where I is the d-dimensional identity matrix, that is $\mathbf{I}_{ij} = \delta_{ij}, i, j = 1, \ldots, d$. The traction $\sigma{\cdot}n$ is the force acting on a unit fluid surface element with unit outwards normal n.

3.1 Basic flow equations

The problem consists of finding u and p such that they satisfy the following equations:

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho(\mathbf{u}_c{\cdot}\nabla)\mathbf{u} - \nabla{\cdot}[2\mu\varepsilon(\mathbf{u})] + \nabla p = \rho f \quad \text{in } \Omega, \qquad \nabla{\cdot}\mathbf{u} = 0 \quad \text{in } \Omega, \tag{24}$$

where f is the force term. The convection velocity $\mathbf{u}_c$ is a given divergence-free vector field when the Oseen equations are considered and $\mathbf{u}_c = \mathbf{u}$ when the Navier–Stokes equations are considered.

Let $\mathbf{U} := [\mathbf{u}, p]^t$ and define the differential operator $\mathcal{L}(\mathbf{U})$ and force term F as

$$\mathcal{L}(\mathbf{U}) := \begin{bmatrix} \rho(\mathbf{u}_c{\cdot}\nabla)\mathbf{u} - \nabla{\cdot}[2\mu\varepsilon(\mathbf{u})] + \nabla p \\ \nabla{\cdot}\mathbf{u} \end{bmatrix}, \tag{25}$$

$$F := \begin{bmatrix} \rho f \\ 0 \end{bmatrix}. \tag{26}$$

Introduce also the matrix $\mathbf{M}$ such that

$$\mathbf{M} = \mathrm{diag}(\rho I, 0). \tag{27}$$

The compact form of the governing equations reads:

$$\mathbf{M}\partial_t\mathbf{U} + \mathcal{L}(\mathbf{U}) = F. \tag{28}$$

3.2 Boundary conditions

In order to close the Navier–Stokes system of equations, initial and boundary conditions must be provided. Let us denote the boundary of Ω as $\partial\Omega$ which we partition as follows:

$$\partial\Omega = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}} \cup \Gamma_{\mathrm{M}}, \tag{29}$$

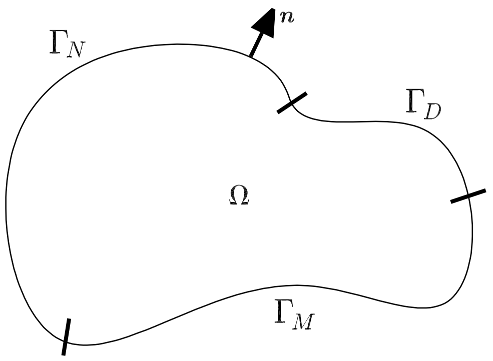


Figure 3. Boundary with Dirichlet, Neumann and mixed conditions.

where D, N and M stand for Dirichlet, Neumann and mixed, respectively. The boundary conditions are:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}}, \\ \sigma{\cdot}n &= t_{\mathrm{N}} \quad \text{on } \Gamma_{\mathrm{N}}, \\ \mathbf{u}{\cdot}n &= 0, \\ \sigma{\cdot}n - (n{\cdot}\sigma{\cdot}n)n &= t_{\mathrm{M}} \quad \text{on } \Gamma_{\mathrm{M}}, \end{aligned} \tag{30}$$

where the vector n is the outward unit normal to Ω. See Figure 3. Boundary values of $\mathbf{u}_{\mathrm{D}}$, t_{N}, $\mathbf{u}_{\mathrm{M}}$, t_{M} are assumed to be known. Some simple examples of boundary conditions are:

- Equation (30_1): velocity known at infinity $u_{\mathrm{D}} = u_\infty$ or $u_{\mathrm{D}} = 0$ on walls;
- Equation (30_2): for uniform flows (or approximately for high Reynolds number flows),

$$\sigma{\cdot}n \approx -pn. \tag{31}$$

Then, if we know that the pressure of the fluid is p_∞ on Γ_{N}, we have the following equivalence:

$$t_{\mathrm{N}} = -p_\infty n \Leftrightarrow p = p_\infty. \tag{32}$$

- Equation (30_3): the mixed boundary condition is usually used in turbulent flows with t_{M} given by the so-called wall function (Bradshaw and Huang 1995, Codina and Soto 1999).

4. Galerkin formulation

4.1 Weak form

Let v and q be the velocity and pressure test functions, respectively. v vanishes on Γ_{D} and $v{\cdot}n = 0$ on Γ_{M}.

We introduce the following functional spaces:

$$V_u = \{v \in H^1(\Omega)^d | v_{|\Gamma_D} = 0, (v\cdot n)_{|\Gamma_M} = 0\}, \quad (33)$$

$$V_p = L^2(\Omega), \quad (34)$$

$$U_u = \{v \in H^1(\Omega)^d | v_{|\Gamma_D} = \mathbf{u}_g, (v\cdot n)_{|\Gamma_M} = 0, t \in (0,T)\}, \quad (35)$$

$$U_p = \{p \in L^2(\Omega) | \int_\Omega p \mathrm{d}\Omega = 0 \text{ if } \Gamma_N = \emptyset, t \in (0,T)\}. \quad (36)$$

The last space is the functional space for the unknown pressure. If the normal component of the traction is not prescribed anywhere on the boundary, then the pressure is only defined up to any additive constant. This is why we explicitly require its average over Ω to be zero.

In order to obtain the weak form, the governing equations are multiplied by the test functions v and q, and the viscous and pressure terms of the momentum equation is integrated by parts. As a result, the natural boundary condition is the prescription of the traction on Γ_N. Let us define $V := [v, q]^t$ as well as $U = U_u \times U_p$ and $V = V_u \times V_p$. We introduce the bilinear form $B(\mathbf{U}, \mathbf{V})$ such that

$$\begin{aligned} B(\mathbf{U}, \mathbf{V}) :=& (\rho(\mathbf{u}_c\cdot\nabla)\mathbf{u}, v) + (2\mu\varepsilon(\mathbf{u}), \varepsilon(v)) \\ & - (p, \nabla\cdot v) + (q, \nabla\cdot\mathbf{u}), \end{aligned} \quad (37)$$

and the linear form $L(V)$ such that

$$L(\mathbf{V}) := (\rho f, v) + (t_N, v)_{\Gamma_N} + (t_M, v)_{\Gamma_M}. \quad (38)$$

The weak form can be written in a compact form as follows: find $\mathbf{U} \in U$ such that

$$(\mathbf{M}\partial_t\mathbf{U}, \mathbf{V}) + B(\mathbf{U}, \mathbf{V}) = L(\mathbf{V}) \qquad \forall \mathbf{V} \in V. \quad (39)$$

The weak form of the momentum equation can be recovered by simply taking $q = 0$ while that of the continuity equation can be found by taking $v = 0$.

4.2 Space discretisation

The discrete weak form consists of finding $\mathbf{U}_h \in U_h$ such that

$$(\mathbf{M}\partial_t\mathbf{U}_h, \mathbf{V}_h) + B(\mathbf{U}_h, \mathbf{V}_h) = L(\mathbf{V}_h) \qquad \forall \mathbf{V}_h \in V_h, \quad (40)$$

together with the initial and boundary conditions.

4.3 Time discretisation

The time discretisation is carried out using the generalised trapezoidal rule, i.e. a finite difference scheme. Let us introduce a uniform partition of the time interval $[0, T]$ and define

$$\mathbf{u}^{n+\theta} := \theta\mathbf{u}^{n+1} + (1-\theta)\mathbf{u}^n, \quad (41)$$

$$\delta t := t^n - t^{n-1}, \quad (42)$$

$$\delta_t\mathbf{u}^{n+\theta} := \frac{\mathbf{u}^{n+\theta} - \mathbf{u}^n}{\theta\delta t}, \quad (43)$$

where δt is the time step size and superscript n denotes the approximated solution at time $n\delta t$. The parameter $\theta \in [0,1]$ determines the order of the scheme. A first order scheme is obtained by choosing $\theta = 1$ (Euler) and a second order method is obtained with $\theta = 0.5$ (Crank–Nicolson). According to this integration rule, we have to solve the following equation for the unknown $U^{n+\theta}$:

$$(\rho\delta_t\mathbf{u}^{n+\theta}, v_h) + B^{n+\theta}(\mathbf{U}_h^{n+\theta}, \mathbf{V}_h) = \mathbf{L}^{n+\theta}(\mathbf{V}_h) \quad \forall \mathbf{V}_h \in V_h, \quad (44)$$

from which we compute the velocity at time step $n+1$ as

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \frac{\mathbf{u}_h^{n+\theta} - \mathbf{u}_h^n}{\theta}. \quad (45)$$

For the sake of clarity, we drop the superscript $n+\theta$ and consider always the unknowns at this time. The temporal term is therefore approximated by:

$$\mathbf{M}\partial_t U_h \approx \frac{1}{\theta\delta t}\mathbf{M}(U_h - U_h^n). \quad (46)$$

5. Stabilised formulation

5.1 Resolved/subgrid scale decomposition

Let us decompose the exact solution $\mathbf{U}$ into a resolved scale $\mathbf{U}_h$ and a subgrid scale $\tilde{\mathbf{U}}$ such that

$$\mathbf{U} = \mathbf{U}_h + \tilde{\mathbf{U}}, \quad (47)$$

where $\tilde{\mathbf{U}}$ belongs to a space $\tilde{\mathbf{U}}$ that completes $\mathbf{U}_h$ in $\mathbf{U}$. That is $\mathbf{U} = \mathbf{U}_h \oplus \tilde{\mathbf{U}}$. The same sum is performed for the test function space $V = V_h \oplus \tilde{V}$.

By substituting Equation (47) into Equation (39), we obtain

$$\begin{aligned} &(\mathbf{M}\partial_t\mathbf{U}_h, \mathbf{V}_h) + B(\mathbf{U}_h, \mathbf{V}_h) + (\mathbf{M}\partial_t\tilde{\mathbf{U}}, \mathbf{V}_h) + B(\tilde{\mathbf{U}}, \mathbf{V}_h) \\ &\quad = L(\mathbf{V}_h) \quad \forall \mathbf{V}_h \in V_h, \\ &(\mathbf{M}\partial_t\mathbf{U}_h, \tilde{\mathbf{V}}) + B(\mathbf{U}_h, \tilde{\mathbf{V}}) + (\mathbf{M}\partial_t\tilde{\mathbf{U}}, \tilde{\mathbf{V}}) + B(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}) \\ &\quad = L(\tilde{\mathbf{V}}) \quad \forall \tilde{\mathbf{V}} \in \tilde{V}. \end{aligned} \quad (48)$$

Note that the traction $\sigma\cdot n$ present in the right-hand side includes the resolved as well as the subgrid scale

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

components of the velocity and pressure:

$$\sigma = -(p_h + \tilde{p})I + 2\mu\varepsilon(u_h + \tilde{u}). \tag{49}$$

The idea now is the following:

(1) Take out the unresolved scale $\tilde{\mathbf{U}}$ from the differential operator involved in the bilinear form B so that

$$\text{formally, } \; B(\tilde{\mathbf{U}}, V_h) = (\tilde{\mathbf{U}}, \mathcal{D}(\mathbf{V}_h)),$$

where $\mathcal{D}$ is a differential operator: this will be done next;

(2) Solve the unresolved scale equation for $\tilde{U}$: done in Section 5.2;

(3) Substitute the result into the modified resolved scale equation: done in Section 5.3.

The first task is carried out by substituting the integral over Ω by the sum of elemental integrals, and by further integrating by parts over each element K. This explains why the differential operator $\mathcal{D}$ is simply the adjoint of $\mathcal{L}$. In addition to Equation (6), let us define

$$\int_{\partial\Omega'} (\cdot) \mathrm{d}\Gamma := \sum_K \int_{\partial K} (\cdot) \; \mathrm{d}\Gamma, \tag{50}$$

$$\int_{\partial\Omega''} (\cdot) \mathrm{d}\Gamma := \sum_K \int_{\partial K} (\cdot) \mathrm{d}\Gamma - \int_{\Gamma_\mathrm{N}\cup\Gamma_\mathrm{M}} (\cdot) \; \mathrm{d}\Gamma. \tag{51}$$

We can obtain the following system of equations equivalent to the continuous problem (39):

$$\begin{aligned}(\mathbf{M}\partial_t\mathbf{U}_h, \mathbf{V}_h) &+ B(\mathbf{U}_h, \mathbf{V}_h) + (\mathbf{M}\partial_t\tilde{\mathbf{U}}, \mathbf{V}_h) \\ &+ \int_{\Omega'} \tilde{\mathbf{U}}\cdot\mathcal{L}^*(\mathbf{V}_h)\mathrm{d}\Omega \\ &+ \int_{\partial\Omega'} \tilde{\mathbf{u}}\cdot[(\mathbf{u}_c\cdot n)v_h + 2\mu\varepsilon(v_h)\cdot n \\ &+ q_h n]\mathrm{d}\Gamma = L(\mathbf{V}_h)\end{aligned} \tag{52}$$

$$\begin{aligned}\int_{\partial\Omega''} (\sigma\cdot n)\cdot\tilde{v}\mathrm{d}\Gamma &+ \int_{\Omega'} \tilde{\mathbf{V}}\cdot[\mathbf{M}\partial_t\tilde{\mathbf{U}} + \mathcal{L}(\tilde{\mathbf{U}})]\mathrm{d}\Omega \\ &= \int_{\Omega'} \tilde{\mathbf{V}}\cdot[\mathbf{F} - \mathbf{M}\partial_t\mathbf{U}_h - \mathcal{L}(\mathbf{U}_h)]\mathrm{d}\Omega\end{aligned} \tag{53}$$

where the adjoint operator $\mathcal{L}^*(V_h)$ is given by

$$\mathcal{L}^*(\mathbf{V}_h) := \begin{bmatrix} -\rho(\mathbf{u}_c\cdot\nabla)v_h - \nabla\cdot[2\mu\varepsilon(v_h)] - \nabla q_h \\ -\nabla\cdot v_h \end{bmatrix}. \tag{54}$$

The boundary integral in Equation (53) excludes the outer boundary as the integration by parts over all the element boundary generates the integral of the total traction which cancels with the one already present on the right-hand side.

5.2 *Solution of the subgrid scale equation*

The boundary integral in Equation (53) vanishes as the exact traction is continuous across element boundaries and we obtain:

$$\begin{aligned}M\partial_t\tilde{\mathbf{U}} + \mathcal{L}(\tilde{\mathbf{U}}) = R(\mathbf{U}_h) := F - \mathbf{M}\partial_t\mathbf{U}_h - \mathcal{L}(\mathbf{U}_h) \\ \forall K \in \mathcal{P}_h,\end{aligned} \tag{55}$$

together with boundary conditions for $\tilde{\mathbf{U}}$ on ∂K, which of course are unknown.

We are now going to solve this equation for $\mathbf{U}$. We first assume the differential operator $\mathcal{L}$ can be approximated in such a way that

$$\tau_K^{-1} \approx \mathcal{L}, \tag{56}$$

where τ_K is a square matrix such that

$$\tau_K = \begin{bmatrix} \tau_1\mathbf{I} & 0 \\ 0 & \tau_2 \end{bmatrix} \tag{57}$$

depending on the element K and on the coefficient of operator $\mathcal{L}$. That is, the *differential operator* is substituted by an *algebraic operator*. This explains why the approximations that we present in the following are referred to as algebraic subgrid scale models. Note that this approximation has been justified in Codina and Blasco (2002) using Fourier analysis. Therefore, the subgrid scale equation becomes:

$$\mathbf{M}\partial_t\tilde{\mathbf{U}} + \boldsymbol{\tau}_K^{-1}\tilde{\mathbf{U}} = R(\mathbf{U}_h). \tag{58}$$

Applying the trapezoidal rule to discretise the time derivative of the subgrid scale, last equation yields:

$$\frac{1}{\theta\delta t}\mathbf{M}(\tilde{\mathbf{U}} - \tilde{\mathbf{U}}^n) + \boldsymbol{\tau}_K^{-1}\tilde{\mathbf{U}} = \mathbf{R}(\mathbf{U}_h), \tag{59}$$

which leads to the subgrid scale expression:

$$\tilde{\mathbf{U}} = \left(\frac{1}{\theta\delta t}\mathbf{M} + \tau_K^{-1}\right)^{-1}\left(\mathbf{R}(\mathbf{U}_h) + \frac{1}{\theta\delta t}\mathbf{M}\tilde{\mathbf{U}}^n\right), \tag{60}$$

where, according to Codina (2001), the coefficients of matrix τ_K are given by

$$\tau_1 = \left(c_1\frac{\mu}{\rho h^2} + c_2\rho\frac{|\mathbf{u}_c|}{h}\right)^{-1}, \tag{61}$$

$$\tau_2 = c_1\mu + c_2\rho|\mathbf{u}_c|h, \tag{62}$$

with $c_1 = 4$, $c_2 = 2$.

We introduce the following definitions:

$$\tau'_K := \left(\frac{1}{\theta\delta t}\mathbf{M} + \tau_K^{-1}\right)^{-1}, \tag{63}$$

$$d := \frac{1}{\theta\delta t}\mathbf{M}\tilde{\mathbf{U}}^n (= d(\tilde{U}^n)). \tag{64}$$

Note that the matrix τ'_K is also diagonal and is expressed as:

$$\tau'_K = \begin{bmatrix} \tau'_1\mathbf{I} & 0 \\ 0 & \tau'_2 \end{bmatrix}, \tag{65}$$

where we can easily check that $\tau'_2 = \tau_2$. According to these definitions, the subgrid scale can be rewritten in a compact form as:

$$\tilde{\mathbf{U}} = \tau'_K(\mathbf{R}(\mathbf{U}_h) + \mathbf{d}). \tag{66}$$

We observe that d can be written

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_{\tilde{\mathbf{u}}} \\ 0 \end{bmatrix}, \quad \text{with} \quad \mathbf{d}_{\tilde{\mathbf{u}}} = \frac{\rho}{\theta\delta t}\tilde{\mathbf{u}}^n, \tag{67}$$

where $d_{\tilde{u}}$ represents the effects of the velocity subgrid scale of the previous time step. We also observe that if the time derivative is neglected in the subgrid scale equation we find that $\tau'_K = \tau_K$ and $d = 0$ and we obtain the subgrid scale without convection tracking. We distinguish two possibilities:
Subgrid scale without time tracking:

$$\tilde{\mathbf{U}} = \tau_K\mathbf{R}(\mathbf{U}_h). \tag{68}$$

Subgrid scale with time tracking:

$$\tilde{\mathbf{U}} = \tau'_K(\mathbf{R}(\mathbf{U}_h) + \mathbf{d}). \tag{69}$$

5.3 Stabilised resolved scale equation

The boundary integral in Equation (52) is neglected (Codina 2001). Let us decompose the differential operator $\mathcal{L}$ into two components $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$ with

$$\mathcal{L}_1(\mathbf{U}) := \begin{bmatrix} \rho(\mathbf{u_c}\cdot\nabla)\mathbf{u} \\ \nabla\cdot\mathbf{u} \end{bmatrix}, \quad \mathcal{L}_2(\mathbf{U}) := \begin{bmatrix} -\nabla\cdot[2\mu\varepsilon(\mathbf{u})] + \nabla p \\ 0 \end{bmatrix}, \tag{70}$$

where $\mathcal{L}_2$ represents the part of the operator that is integrated by parts. For the sake of clarity we substitute the sum of the integrals over the elements K by the integral over the whole domain Ω. Therefore, we have that

$$\big(\mathbf{M}\partial_t\mathbf{U}_h + \mathcal{L}_1(\mathbf{U}_h), \mathbf{V}_h\big) + \big(2\mu\varepsilon(\mathbf{u}_h), \varepsilon(v_h)\big) - (p_h, \nabla\cdot v_h) \tag{71}$$

$$+\big(\mathbf{M}\partial_t\tilde{\mathbf{U}}, \mathbf{V}_h\big) + \big(\tilde{\mathbf{U}}, \mathcal{L}^*(\mathbf{V}_h)\big) = L(\mathbf{V}_h). \tag{72}$$

Substituting the expressions (59) and (66) in the time derivative term of the subgrid scale in this equation we obtain the compact form of the resolved scale equation:

$$\begin{aligned}(\mathbf{M}\partial_t\mathbf{U}_h &+ \mathcal{L}_1(\mathbf{U}_h), \mathbf{V}_h) + (2\mu\varepsilon(\mathbf{u}_h), \varepsilon(v_h)) \\ &- (p_h, \nabla\cdot v_h) + \Big(\mathbf{R}(\mathbf{U}_h) - \tau'^{-1}_K\tau'_K(\mathbf{R}(\mathbf{U}_h) + \mathbf{d}), \mathbf{V}_h\Big) \\ &+ (\tau'_K(\mathbf{R}(\mathbf{U}_h) + \mathbf{d}), \mathcal{L}^*(\mathbf{V}_h)) = L(\mathbf{V}_h).\end{aligned} \tag{73}$$

Let us introduce the following definitions:

$$\begin{aligned} r_{\mathrm{m}}(\mathbf{u}_h, p_h) = \rho f - \rho/(\theta\delta t)(\mathbf{u}_h - \mathbf{u}_h^n) - \rho(\mathbf{u}_c\cdot\nabla)\mathbf{u}_h \\ + \nabla\cdot[2\mu\varepsilon(u_h)] - \nabla p_h,\end{aligned} \tag{74}$$

$$r_{\mathrm{c}}(\mathbf{u}_h) = -\nabla\cdot\mathbf{u}_h, \tag{75}$$

$$p_{\mathrm{m}}(v_h) = \Big(\tau'^{-1}_1\tau'_1\Big)v_h - \tau'_1(-\rho(\mathbf{u}_c\cdot\nabla)v_h - \nabla\cdot[2\mu\varepsilon(v_h)]), \tag{76}$$

$$p_{\mathrm{c}}(q_h) = \tau'_1\nabla q_h, \tag{77}$$

$$p_{\mathrm{m}}(v_h) = \tau'_2\nabla\cdot v_h, \tag{78}$$

$$p_{\mathrm{c}}(q_h) = q_h, \tag{79}$$

where 'm' stands for momentum and 'c' for continuity. We can obtain the expanded form of the stabilised resolved scale equation:

$$\begin{aligned}&(-r_{\mathrm{m}}(\mathbf{u}_h, p_h), p_{\mathrm{m}}(v_h)) + (2\mu\varepsilon(\mathbf{u}_h), \varepsilon(v_h)) \\ &\quad - (p_h, \nabla\cdot v_h) - (\nabla\cdot[-2\mu\varepsilon(\mathbf{u}_h)] + \nabla p_h, v_h) \\ &\quad - (r_{\mathrm{c}}(\mathbf{u}_h), p_{\mathrm{m}}(v_h)) = \big(\rho/(\theta\delta t)\tilde{\mathbf{u}}_h^n, p_{\mathrm{m}}(v_h)\big) \\ &\quad + (t_{\mathrm{N}}, v_h)_{\Gamma_{\mathrm{N}}} + (t_{\mathrm{M}}, v_h)_{\Gamma_{\mathrm{M}}}, \\ &\qquad (-r_{\mathrm{m}}(\mathbf{u}_h, p_h), p_{\mathrm{c}}(q_h)) - (r_{\mathrm{c}}(\mathbf{u}_h), p_{\mathrm{c}}(q_h)) \\ &\quad = (\rho/(\theta\delta t)\tilde{\mathbf{u}}_h^n, p_{\mathrm{c}}(q_h)).\end{aligned} \tag{80}$$

This formulation corresponds to the case of the Oseen equations in which the convective velocity u_c is known. In the case of the Navier–Stokes equations, the convective velocity is an unknown of the problem $u_c = u$. We distinguish two possibilities:

$$\text{Stabilisation without convection tracking}: \quad \mathbf{u}_c = \mathbf{u}_h. \tag{81}$$

$$\text{Stabilisation with convection tracking}: \mathbf{u}_c = \mathbf{u}_h + \tilde{\mathbf{u}}. \tag{82}$$

The first choice corresponds to that used in classical stabilisation schemes and misses the subgrid scale

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

contribution to the convection. The second choice which appears as the natural one retains this contribution. The subgrid scale is then computed from Equation (60).

At this stage we observe that we have one source of non-linearity: the convective term which appears both in the resolved scale and subgrid scale equations. Therefore the problem must be linearised. This aspect is now treated in the following section.

Let us close this section mentioning the results of the numerical analysis of this method which can be found in Codina *et al.* (2007). As it has been mentioned in the introduction, the use of transient subgrid scales provides the correct dependence of the stabilisation parameter with respect to the time step size (but also introduces a term on the right-hand side that makes the results of a steady calculation independent of the time step size) making the space and time discretisations commutative. If the linearised problem (with a given advection velocity) is considered, stability of $\tau_1^{1/2}(\rho u_c \cdot \nabla u_h + \nabla p_h)$ can be probed for any h and δt (usually the restriction $\delta t \geq C\tau_1$ is needed). This result is proved using a rather weak norm but if this restriction is imposed a stronger result can be derived. The other important aspect of the formulation is that conservation of momentum is achieved the subscales are tracked along the non-linear process. We refer to Codina *et al.* (2007) for further details.

6. Numerical implementation

The numerical implementation presented here has been designed to consider the different possibilities for the approximation of the subgrid scale already mentioned. The linearisation of the problem is considered next followed by a description of the proposed algorithm.

6.1 Linearisation

The complete problem defined by Equations (80) and (60) is highly non-linear. When convection tracking is considered (Equation (82)), the original quadratic convective term gives rise to four terms in each equation when the scale splitting is considered. Although our linearisation will be a simple one, let us note where these terms are in the final formulation. The scale splitting of the convective term leads to

$$(v, \rho \mathbf{u}_c \cdot \nabla \mathbf{u}) = (\rho \mathbf{u}_c \cdot \nabla \mathbf{u}_h, v_h) + (\rho \mathbf{u}_c \cdot \nabla \tilde{\mathbf{u}}, v_h) + (\rho \mathbf{u}_c \cdot \nabla \mathbf{u}_h, \tilde{v}) + (\rho u_c \cdot \nabla \tilde{\mathbf{u}}, \tilde{v}). \tag{83}$$

The first term of the right-hand side is the classical Galerkin term. The second one has been integrated by parts and the tracking consists in evaluating the test function (which is the adjoint operator) using u_c. The third term has been moved to the right-hand side of the subgrid scale equation and is part of the residual. The last one was considered in the algebraic approximation of the subgrid scale equation and it can be seen in the dependence of the stabilisation parameter τ_1 on u_c. In turn, the last two terms are used to compute $\tilde{u}$ and therefore affect u_c (which is a manifestation of the non-linearity of the subgrid scale equation). Let us introduce a linearisation iteration counter i. The equations are considered at iteration $i+1$ and if no particular mention is made, the variables are considered at this iteration without specific notation. We consider two possibilities. The first possibility is the linearisation using the Picard method and is defined by:

$$\mathbf{u}_c = \mathbf{u}_h^i + \tilde{\mathbf{u}}^i.$$

As will be explained in the next subsection, as the subgrid scale $\tilde{\mathbf{u}}^i$ depends on the residual of the finite element component $\mathbf{u}_h^i$ which is computed first and then used to evaluate $\tilde{\mathbf{u}}^i$ during the formation of the finite element matrix for the next iteration. The second possibility consists of a Newton type linearisation of the resolved part of the convective term in the residual as follows:

$$\mathbf{u}_c = \mathbf{u}_h^i + \tilde{\mathbf{u}}^i, \tag{84}$$

$$r_{\mathrm{m}}(\mathbf{u}_h, p_h) \Leftarrow r_{\mathrm{m}}(\mathbf{u}_h, p_h) + \rho\left(\mathbf{u}_h^{i+1} \cdot \nabla\right)\mathbf{u}_h^i - \rho\left(\mathbf{u}_h^i \cdot \nabla\right)\mathbf{u}_h^i. \tag{85}$$

We will refer to it as the Newton–Raphson method but we note that it corresponds to a Newton linearisation only when convection tracking is not considered.

6.2 Final algorithm

The final algorithm is developed to take into account the different levels of approximation. The simplest one corresponds to the classical approach in which neither time nor convection tracking is considered. The second one, the time tracking of the subscales, consists of retaining the time derivative of the subgrid scale. The third one, the convection tracking, consists of adding the subgrid scale velocity $\tilde{u}$ to the resolved scale velocity when the convection velocity is defined. The last one is the full scheme 'without approximation'.

In any of these cases there are some calculations that need to be performed at each Gauss point. They are: the calculation of the residuals, the calculation of the test functions and the evaluation of the stabilisation parameters. In the first two cases the convection velocity is simply taken as

$$\mathbf{u}_c = \mathbf{u}_h^i. \tag{86}$$

Only in the last two cases the equation for the subgrid scale equation (60) needs also to be solved. As mentioned in the previous subsection this is done while the the matrix of the system at the next iteration i is being computed which permits us to reuse the Gauss point calculations already mentioned. The resulting algorithm can be described as follows.

At each time step n and linearisation iteration i we introduce a linearisation index j and a relaxation factor $\tilde{\alpha}$. The iterative scheme for the subgrid scale is the following: let the initial guess be $\tilde{\mathbf{u}}^{i,0} = \tilde{\mathbf{u}}^{i-1}$, then solve the following equation for $j = 1, 2, \cdots$ until convergence is achieved:

$$\tilde{\mathbf{u}}^* = \left(\frac{\rho}{\theta\delta t} + \tau_1^{-1}\right)^{-1}\left(r_{\rm m}\left(\mathbf{u}_h^i, p_h^i\right) + \frac{\rho\tilde{\mathbf{u}}^n}{\theta\delta t}\right), \qquad (87)$$

$$\tilde{\mathbf{u}}^{i,j+1} = \tilde{\alpha}\tilde{\mathbf{u}}^* + (1 - \tilde{\alpha})\tilde{\mathbf{u}}^{i,j}, \qquad (88)$$

with

$$\tau_1 = \left(c_1\frac{\mu}{\rho h^2} + c_2\rho\frac{|\mathbf{u}_c|}{h}\right)^{-1}, \qquad (89)$$

$$\mathbf{u}_c = \mathbf{u}_h^i + \tilde{\mathbf{u}}^{i,j}. \qquad (90)$$

The complete general algorithm for the algebraic subgrid scale model with transient subgrid scales and convection tracking is shown in Algorithm 1. It is composed of three main iteration loops. The time loop over index n; the resolved velocity linearisation loop over index i; and the velocity subgrid scale linearisation loop over index j. If convection tracking is not considered, the inner loop is skipped. In Algorithm 114 it is understood that the momentum residual is evaluated using the shape functions and that $r_{\rm m}(u_h^i, p_h^i)$ is computed simply multiplying it by the unknowns at the previous iteration. These residuals are then reused when the system is assembled.

Algorithm 1 Stabilised Navier–Stokes equations

```
Set time initial values $u_h^0$ and $\tilde{u}^0$.
for time steps $n = 1, 2, \ldots$ do
  Set linearisation initial values $u_h^0 = u_h^n$ and $\tilde{u}^0 = \tilde{u}^n$.
  $i = 0$.
  while convergence not achieved do
    for elements do
      for Gauss points do
        Set $u_c$ using 86 or 90.
        Compute residual $r_{\rm m}(u_h^i, p_h^i)$.
        Compute $\tau_1$ using 89.
        if convection tracking then
          Set linearisation initial values $\tilde{u}^{i,0} = \tilde{u}^{i-1}$.
          $j = 0$.
          while Convergence not achieved do
            Compute $\tilde{u}^{i,j+1}$ using 87 and 88.
            Update $u_c$ using 90.
            Update residual $r_{\rm m}(u_h^i, p_h^i)$.
            Update $\tau_1$ using 89.
            $j = j + 1$.
          end while
        else if Time tracking then
          Update $\tilde{u}^i$ using 87.
        end if
        Compute continuity residual $r_{\rm c}(u_h)$.
        Compute test functions: $p_{\rm m}(v_h)$, $p_{\rm c}(q_h)$, $p_{\rm m}(v_h)$,
          $p_{\rm c}(q_h)$.
        Compute terms in 80.
        Assemble element matrix and right-hand side in
          global system.
      end for
    end for
    Solve linear system for $u_h$ and $p_h$.
    $i = i + 1$.
  end while
  Update $u_h^{n+1} = 1/\theta u_h + (1 - 1/\theta)u_h^{n-1}$
  if Time tracking then
    Update $\tilde{u}^{n+1} = 1/\theta\tilde{u} + (1 - 1/\theta)\tilde{u}^{n-1}$
  end if
end for
```

7. Numerical examples

We present the solution of three numerical examples. The first example aims at illustrating the subgrid scale concept with respect to the Galerkin method. Through the next two examples we propose to study the characteristics of the stabilisation method with convection and/or time tracking of the subgrid scale. The purpose is to check the stabilisation property of the method, its accuracy for stationary and transient flows, as well as the convergence of Algorithm 1. In the following we define:

$$L^2 \text{ residual of an unknown vector } x \text{ at iteration } i = 100 \times \sqrt{\frac{(x^i - x^{i-1})^2}{x^{i^2}}}.$$

In order to concentrate ourselves on the analysis of the effects of the convection and time tracking of the subgrid scales, the characteristic length h appearing in the expressions for the stabilisation parameters τ_1 and τ_2 (Equations (61) and (62)), is taken as the minimum element length for each element (it introduces less numerical diffusion than if we take the maximum length). In all examples the $Q1/Q1$ element is used and the numerical integration is performed using four Gauss points.

7.1 Illustrative example

Let us solve a simple example to illustrate the multiscale concept, and in particular to identify the stabilising effects and the location of the subgrid scale. We consider the advection–diffusion–reaction Equation (3) with a curved advection field. The geometry as well as the boundary conditions are illustrated in Figure 4.

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

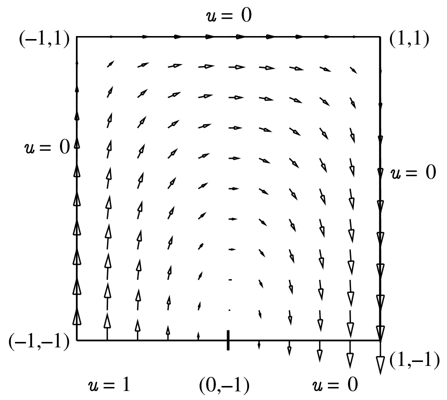


Figure 4. Illustrative example: geometry and boundary conditions.

The data of the problem are:

$$\varepsilon = 10^{-2}, \tag{91}$$

$$s = 10^{-4}, \tag{92}$$

$$\mathrm{a} = \frac{1}{2}[(1 - x^2)(1 + y), -x(4 - (1 + y)^2)]^t. \tag{93}$$

The computational domain is a square domain $\Omega = (-1, 1) \times (-1, 1)$. The Dirichlet boundary conditions for u are

$$u = u_0 = 1 \quad \text{at } y = -1, \quad -1 < x < 0, \tag{94}$$

$$u = 0 \quad \text{elsewhere.} \tag{95}$$

According to the data, we expect to have parabolic boundary layers on the left, top and right sides of the cavity and an exponential boundary layer on the bottom right part. This is precisely the zone where the subgrid scale modelling acts.

Figure 5 shows the results obtained on a regular mesh composed of 10 $Q1$ elements in each direction. The top part of the figure shows the results obtained without stabilisation, that is the Galerkin solution. The top left figure is the unknown, which exhibits strong oscillation due to the poor resolution of the exponential boundary layer. The top right figure is the subgrid scale in percentage, that is $100 \times |\tilde{u}|/|u_0|$ obtained on the Gauss points of the elements, where $\tilde{u}$ has been computed using Equation (15). The value on the nodes was set to zero for visualisation purposes. The bottom left figure shows the stabilised solution, using the subgrid scale concept. Finally, the bottom right figure shows the subgrid scale.

In the case of the Galerkin solution, we observe that $\tilde{u}$ is a good measure of the error of the solution. In the case of the stabilised solution, we can observe that most of the subgrid scale is concentrated in the first layer of elements in the boundary layer where the unknown goes from almost one to the prescribed value of zero. There, the percentage of the subgrid scale is around 45% of the prescribed value of 1.

7.2 *Cavity flow*

We consider the cavity flow at different Reynolds numbers and for different mesh sizes. We compare the results and convergences obtained with the classical stabilisation method and the subgrid scale approach with convection tracking. In order to study the convergence properties of the stabilisation method, all the cases are solved using the stationary equations, except one case for which convergence cannot be achieved this way. The computational domain is $\Omega = (0, 0) \times (1, 1)$ and the boundary conditions are $u = 0$ at $x = 0, x = 1$ and $y = 0$, and $u = (1, 0)$ on the rest of the boundary. The density of the fluid is set to unity and the Reynolds number is defined as $\mathrm{Re} = 1/\mu$.

We consider five different regular and structured meshes with 5×5, 10×10, 20×20, 40×40 and 80×80 Q_1/Q_1 elements. As the flow is confined ($\Gamma_\mathrm{N} = \emptyset$), the pressure was imposed on the bottom left corner to zero. When not explicitly mentioned, the subgrid scale equation (87) is solved using a maximum number of 20 iterations and a convergence tolerance of the L^2-residual of 10^{-8}. We are going to study the following points:

- Effects of the subgrid scales;
- Accuracy;
- Global convergence: with respect to mesh size and Reynolds number;
- Convergence of the subgrid scale equation;
- Linearisation techniques: Picard *versus* Newton–Raphson.

7.2.1 *Effects of the subgrid scales*

To have a first glance at the subgrid scale approach with convection tracking, we start with the solution of the flow at Re = 100 on the 10×10 and 40×40 meshes. The resolved velocity at the nodes and the subgrid velocity at the Gauss points is shown in Figure 6. We can outline two expected results. Firstly, the subgrid scale level decreases with the mesh size. Remember that when convection dominates,

$$\tilde{\mathbf{u}} \rightarrow \frac{h}{2\rho|u_c|}\mathbf{R}(\mathbf{U}_h).$$

Secondly, most of the subgrid scale is located in the zones of the cavity with the highest gradients. We will see that this yields a global effect on the solution in the cavity, as already shown in Section 7.1. We observe that the subgrid scale points outside of the cavity in the top left corner, and inside of the cavity in the top right corner.

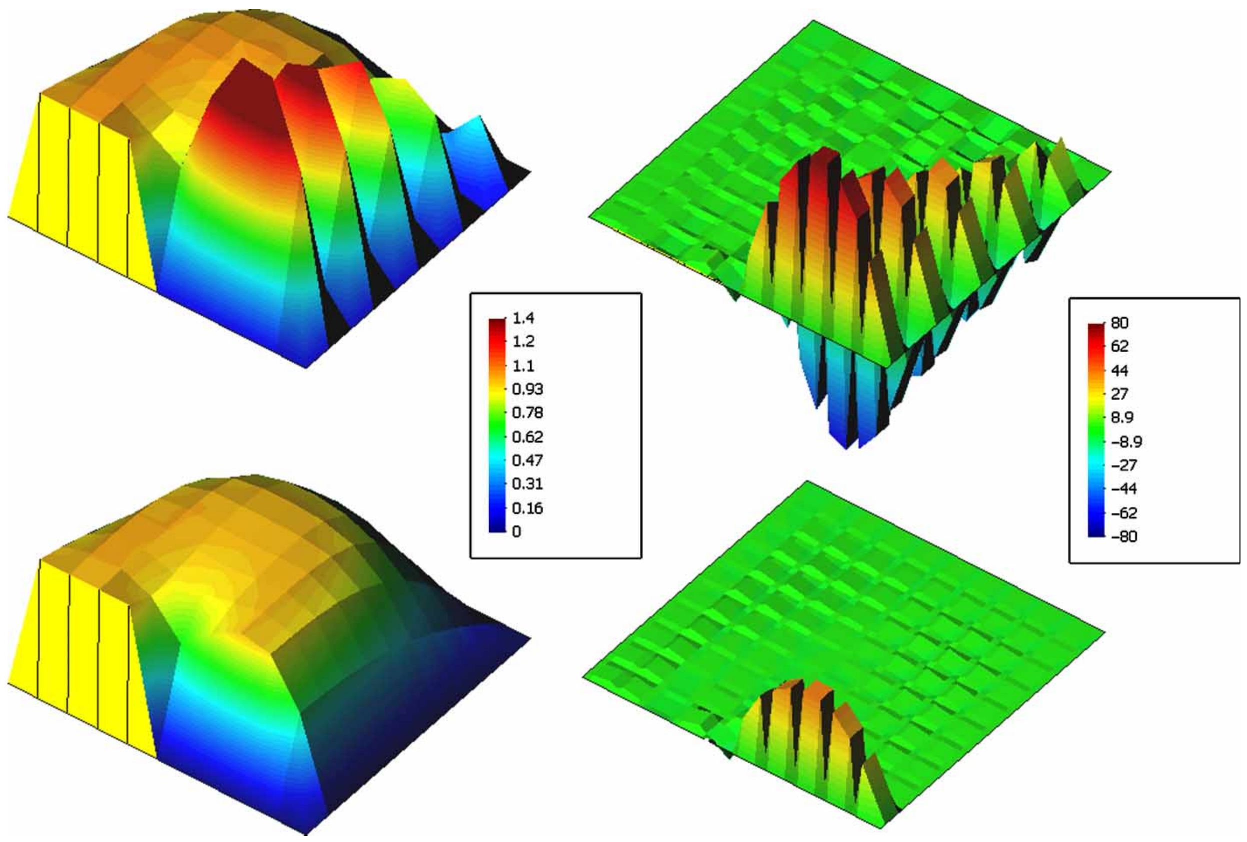


Figure 5. Illustrative example: effects of the SGS stabilisation. (top) (left) Galerkin solution. (top) (right) subgrid scale $100 \times |\tilde{u}|/|u_0|$ at the Gauss points calculated from the Galerkin solution. (bottom) (left) Stabilised solution using the subgrid scale method. (bottom) (right) subgrid scale $100 \times |\tilde{u}|/|u_0|$ at the Gauss points.

7.2.2 *Accuracy*

Now let us look at the results of the simulations in more detail. We consider the case $\mathrm{Re} = 100$. They exhibit two clear tendencies of the subgrid scale model with convection tracking. On the one hand, it is much less diffusive: the solution converges faster to the finest grid solution when the grid size increases than the classical method. Figure 7 shows the solution obtained on the

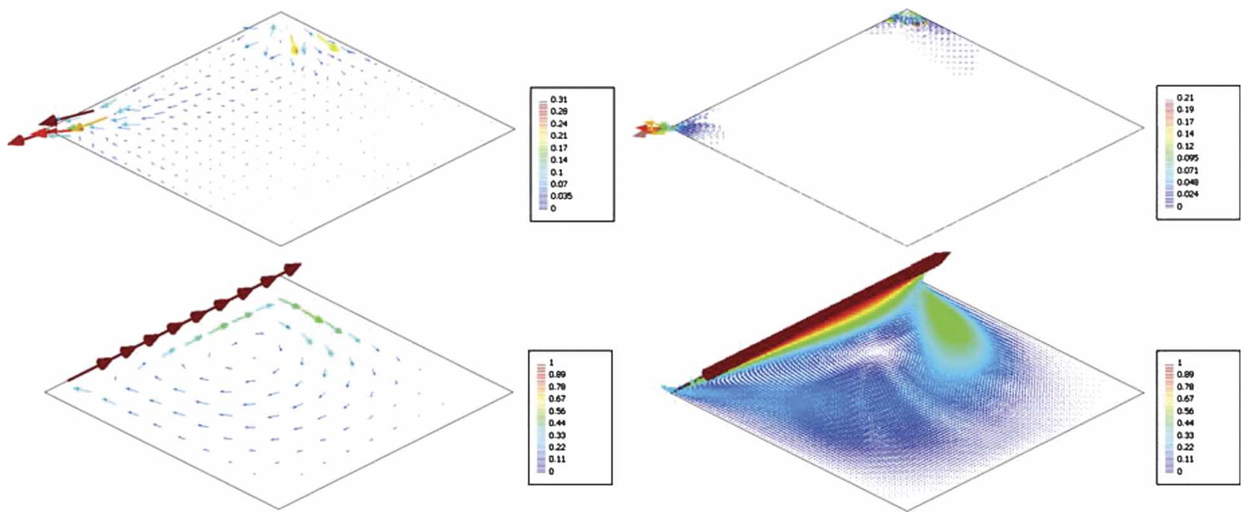


Figure 6. Cavity: solution at $\mathrm{Re} = 100$. (left) 10×10 mesh. (right) 40×40 mesh. (bottom) Velocity u. (top) velocity subgrid scale $\tilde{u}$.

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

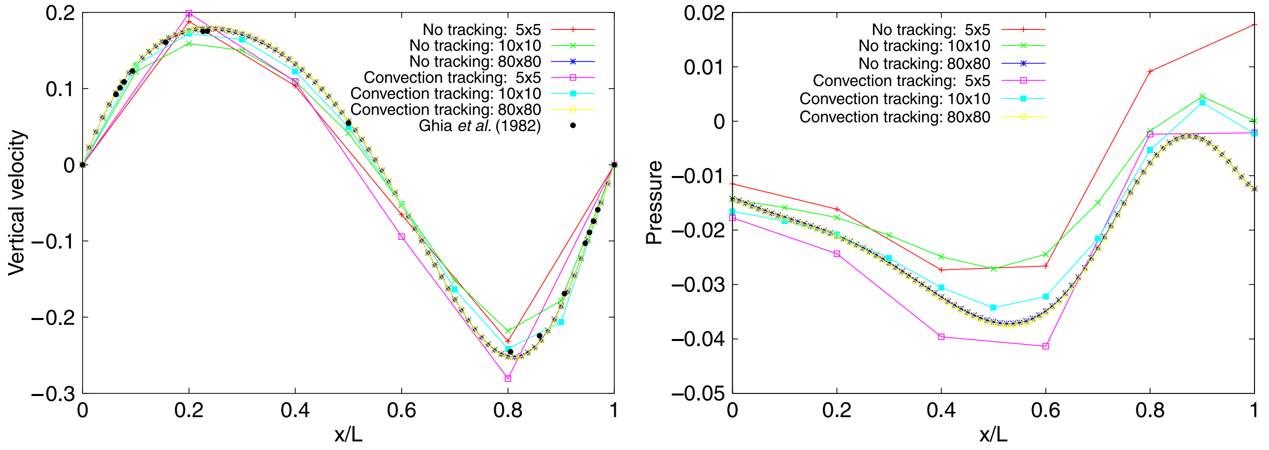


Figure 7. Cavity: solution on the horizontal centreline for the 5×5, 10×10 and 80×80 meshes at Re $= 100$. (left) vertical velocity. (right) pressure.

horizontal centreline for the 5×5, 10×10 and 80×80 meshes. We observe that both the pressure and the vertical velocity on the coarse meshes are better captured when convection tracking is used.

7.2.3 Global convergence

On the other hand, convergence of the stabilisation method with convection tracking cannot be achieved for two coarse meshes and high Reynolds numbers. In fact, Table 2 shows the value of the relaxation factor used to solve the subgrid scale equation for which the convection tracking stabilisation converges. The relaxation factor was varied from 0.1 to 1.0 with steps of 0.1. However, convergence can be achieved passing through a transient state, for example using one iteration per time step. In this case, the accuracy achieved by the convection tracking is even more noticeable, as shown by Figure 8. It shows the solution obtained on the 10×10, where convergence has been achieved using a time step of $\delta t = 1$ and one iteration per time step. The accuracy of the solution obtained with convection tracking is quite impressive when compared to results of Ghia *et al.* (1982).

Table 2. Maximum relaxation factor used in subgrid scale equation for which convergence is achieved.

Reynolds number	10	100	200	400	600	800	1000
5×5 mesh	1.0	0.7	X	X	X	X	X
10×10 mesh	1.0	1.0	0.7	X	X	X	X
20×20 mesh	1.0	1.0	0.9	0.1	X	X	X
40×40 mesh	1.0	1.0	1.0	1.0	1.0	0.7	X
80×80 mesh	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Note: X = convergence not achieved.

The convergence histories are compared for two Reynolds number and different meshes in Figure 9. We observe better convergence of the stabilisation without convection tracking. In fact, the convection tracking introduces an additional non-linearity.

7.2.4 Convergence of the subgrid scale equation

The subgrid scale equation is a non-linear equation to be solved at each Gauss point (see Equations (87) and (88)). The iterative algorithm is controlled by a relaxation factor, a maximum number of iterations as well as a convergence tolerance. From the user's point of view, in order to limit the number of parameters to adjust, we would like to perform only one iteration without relaxing, that is to couple the iterative loop of the resolved and subgrid scales. However, when the effects of the subgrid scale become important (for

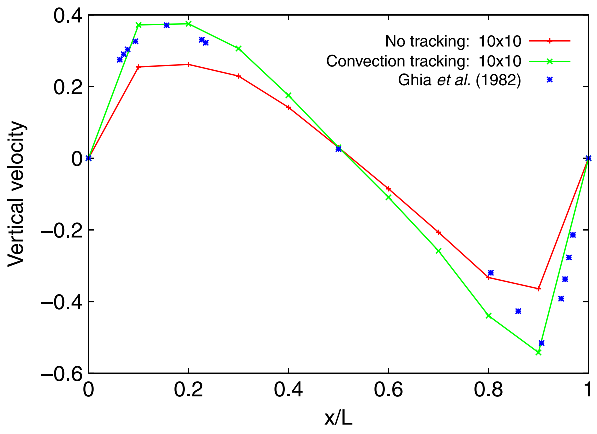
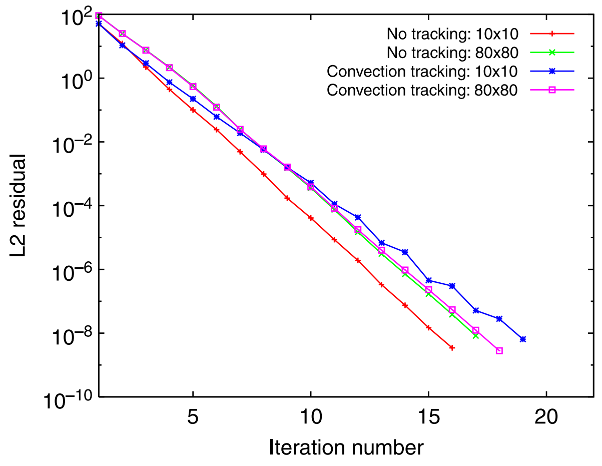


Figure 8. Cavity: vertical velocity on the horizontal centreline for the 10×10 mesh at Re $= 1000$.



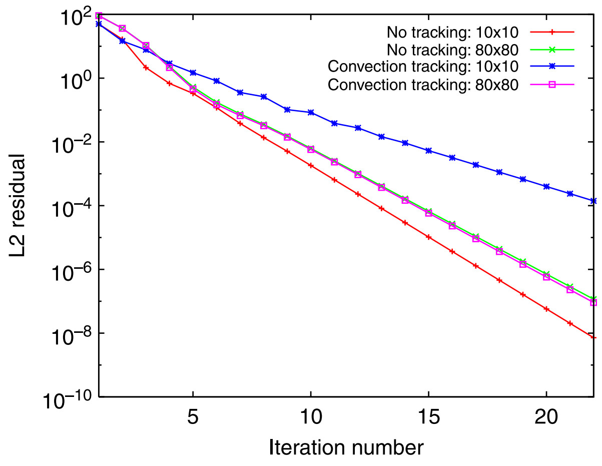


Figure 9. Cavity: convergence history. (left) Re = 100. (right) Re = 200.

coarse meshes or high Reynolds numbers), this is not possible. Figure 10 shows the convergence histories of the subgrid scale velocity at Re = 100 for the 10 × 10 and 80 × 80 meshes for a tolerance of 10^{-8}, a relaxation factor as indicated in Table 2 and for varying maximum number of iterations. We observe that it is important to have the subgrid scale well converged in order to obtain a good global convergence.

It should be pointed out that the subgrid scale Equation (87) contains the norm of the convection velocity and this can seriously damage the convergence. In fact, let us consider the extreme case of a very high subgrid scale. This equation reduces to

$$\tilde{\mathbf{u}} = -c_2 \rho^2 \frac{|\tilde{\mathbf{u}}|}{h} (\tilde{\mathbf{u}} \cdot \nabla) \mathbf{u}_h^i, \tag{96}$$

which admits two solutions with opposite signs. This has been observed in practice.

7.2.5 *Linearisation strategy*

Figure 11 compares the convergence histories obtained at Re = 100 using the Newton and Picard linearisation strategies. We remark that for the coarse mesh, quadratic

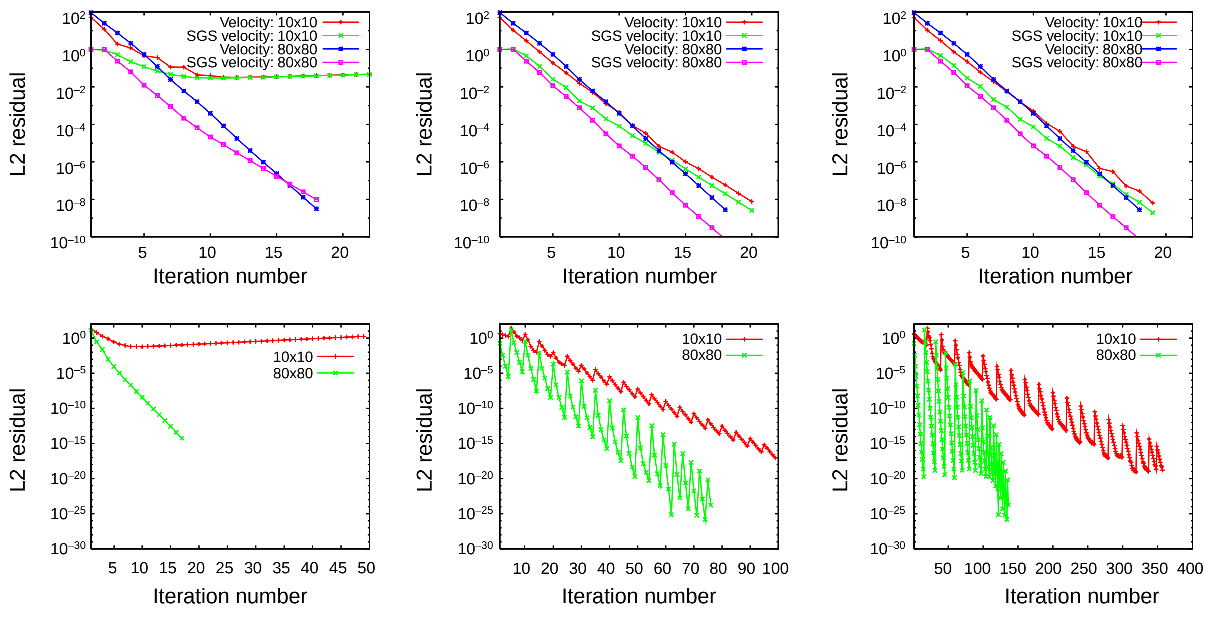


Figure 10. Cavity: convergence of the subgrid scale equations for different numbers of maximum iterations in the subgrid scale equation (1), (5), (20) from left to right. (top) convergence of the velocity and velocity subgrid scale. (bottom) convergence of the velocity subgrid scale in subgrid scale equation.

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

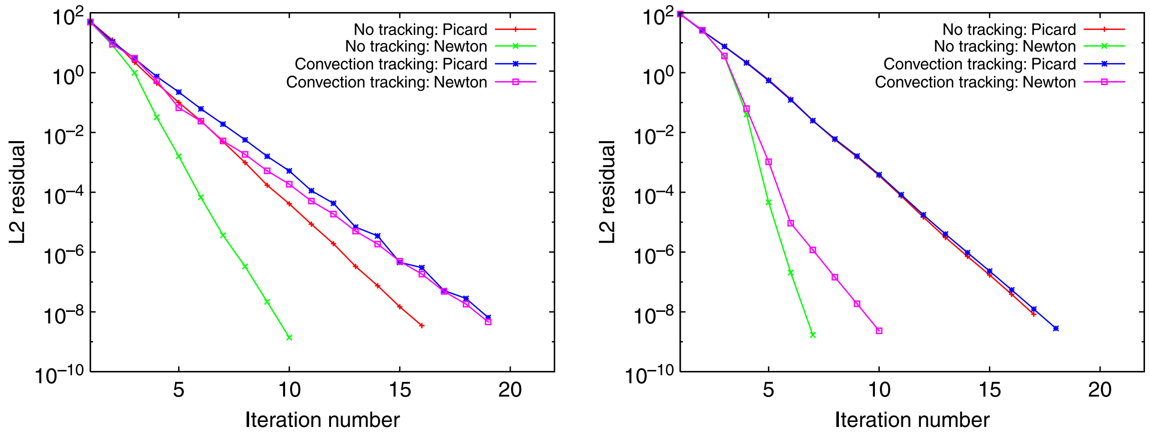


Figure 11. Cavity: Convergence history at Re = 100. Comparisons of Picard and Newton–Raphson strategies. (left) 10 × 10 mesh. (right) 80 × 80 mesh.

convergence is not achieved with the convection tracking. This is due to the fact the subgrid scale was not retained in the additional terms involved in the Newton–Raphson method (see Equations (84) and (85)) and in this case the subgrid scale is of the same order as the velocity (see Figure 6). If the mesh is refined, see Figure 11 (right), this is almost no longer true as the subgrid scale looses weight with respect to the resolved scale.

We conclude this numerical example by an obvious statement: the gain in accuracy is obtained at the expense of convergence deterioration. Also, the convergence of the subgrid scale equation is a crucial point in the convergence of the whole algorithm, when the subgrid scale is of the same order as the velocity, that is for coarse meshes and high Reynolds numbers. This convergence can hardly be obtained in some cases, and, surely, a better strategy is to be found to solve the subgrid scale equation (e.g. passing through a false transient state as done for the solution obtained in Figure 8).

7.3 *Flow over a cylinder*

Through this example we want to check the behaviour of the stabilisation method with time and/or convection tracking in a transient simulation. To this end, we solve the two-dimensional flow around a circular cylinder at a Reynolds number Re = 100. The cylinder is located at position $(0, 0)$ and is located in a rectangle of length 18 and height 24, as shown in Figure 12. The size of the computational domain was chosen according to observations of (Behr *et al.* 1995) concerning the blocking of the solution when the horizontal walls are too close to the cylinder. The diameter of the cylinder is $d = 1$, the inflow velocity $u_\infty = 1$ in the horizontal direction. The density of the fluid is set to unity so that the Reynolds number is defined as Re $= 1/\mu$. The boundary conditions are zero velocity on the cylinder, symmetry condition on the top and bottom walls (i.e. zero tangential traction and normal velocity), and zero traction at the outflow.

Four meshes are considered. They are respectively composed of 320, 1280, 5120 and 20,480 Q_1/Q_1 elements. All meshes are structured and refined near the cylinder. Two time steps are considered, $\delta = 0.1$ and $\delta t = 0.2$ with the second order Crank–Nicolson scheme. It was not necessary to relax the subgrid scale to achieve convergence. The maximum number of iterations and tolerance to solve this equation are 20 and 10^{-8}, respectively. The extra cost to track the subgrid scale in time and convection is 4.4% of the total element

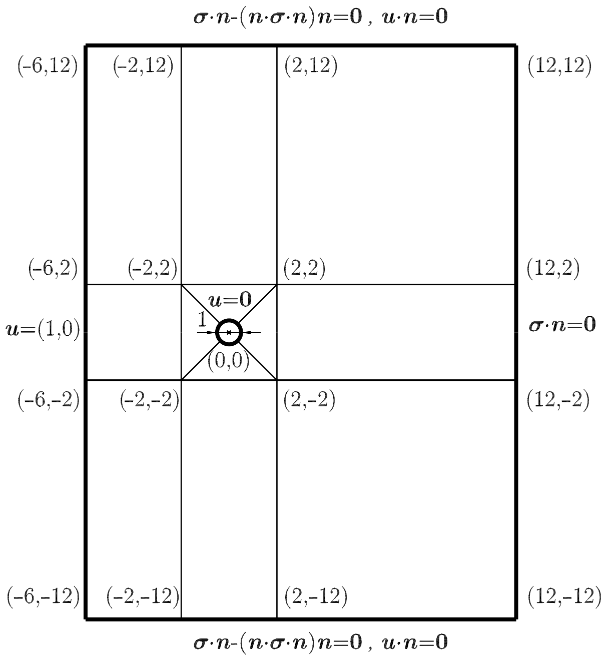


Figure 12. Cylinder: geometry and boundary conditions.

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

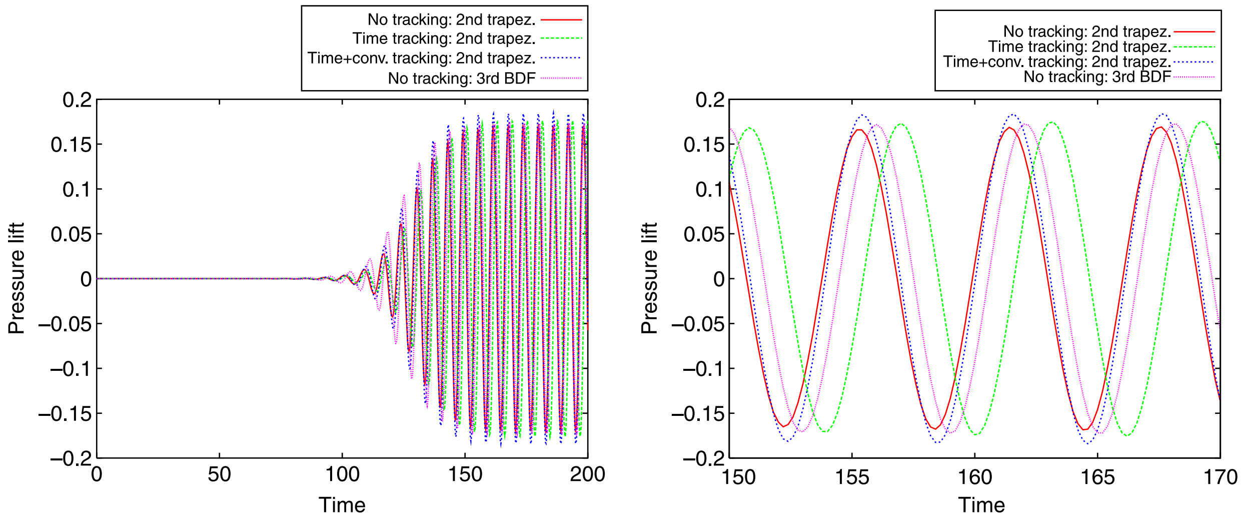


Figure 13. Cylinder: pressure lift on the cylinder. 1280-element mesh, $\delta t = 0.2$. (left) complete time evolution. (right) zoom.

Table 3. Amplitude of the pressure lift[a] for different meshes and $\delta t = 0.2$.

Method	No tracking	No tracking (BFD3)	Time tracking	Time + conv. tracking
5×5	0	0	0	0
10×10	0.2608	0.2294	0.2926	0.3288
20×20	0.3398	0.3460	0.3518	0.3686
30×30	0.3243	0.3328	0.3326	0.3435

Note: 0 = Stationary solution. [a](Behr *et al.* 1995): 0.3706, 0.3659

calculations (including all operations at Gauss points and assembly).

We first examine the pressure lift for the 1280-element mesh and $\delta t = 0.2$ to compare the results of the classical approach and the stabilisation method with time and time + convection tracking. The pressure lift is defined as:

$$\text{pressure lift} = \frac{\int_{\text{cylinder}} -pnd\Gamma}{\rho u_\infty^2},$$

where n is the outward normal to the cylinder. Its evolution is shown in Figure 13.

We observe that both the time and time + convection tracking are less diffusive than the classical method, in the sense that we obtain a higher amplitude. We also compared the results to those obtained with a third order Backward Finite Difference scheme in time (BDF). Both the time and convection trackings enable to obtain a better amplitude in the pressure lift.

In order to quantify the results, Tables 3, 4, 5 and 6, show the amplitude as well as the Strouhal number obtained for different numerical strategies (the Strouhal number is defined as fd/u_∞). The first two tables show the evolution of these values for different meshes with $\delta t = 0.2$, while the second table shows the evolution of these values for different time steps with the 20×20 mesh. First we note that nothing is gained in frequency when using the time and convection tracking. On the contrary we observe that the amplitude of the pressure lift

Table 4. Strouhal of the pressure lift[a] for different meshes and $\delta t = 0.2$.

Method	No tracking	No tracking (BFD3)	Time tracking	Time + conv. tracking
5×5	0	0	0	0
10×10	0.1372	0.1349	0.1352	0.1357
20×20	0.1635	0.1647	0.1630	0.1637
30×30	0.1688	0.1700	0.1687	0.1689

Note: 0 = Stationary solution. [a](Behr *et al.* 1995): 0.1624, 0.1661

Table 5. Amplitude of the pressure lift[a] for different δt and the 20×20 mesh.

Method	No tracking	No tracking (BFD3)	Time tracking	Time + conv. tracking
$\delta t = 0.2$	0.3398	0.3460	0.3518	0.3686
$\delta t = 0.4$	0.3388	0.3842	0.3522	0.3677
$\delta t = 0.8$	0.3386	0.4819	0.3358	0.3654

Note: 0 = Stationary solution. [a](Behr *et al.* 1995): 0.3706, 0.3659

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

Table 6. Strouhal of the pressure lift[a] for different δt and the 20×20 mesh.

Method	No tracking	No tracking (BFD3)	Time tracking	Time + conv. tracking
$\delta t = 0.2$	0.1635	0.1647	0.1630	0.1637
$\delta t = 0.4$	0.1619	0.1677	0.1613	0.1621
$\delta t = 0.8$	0.1555	0.1626	0.1550	0.1556

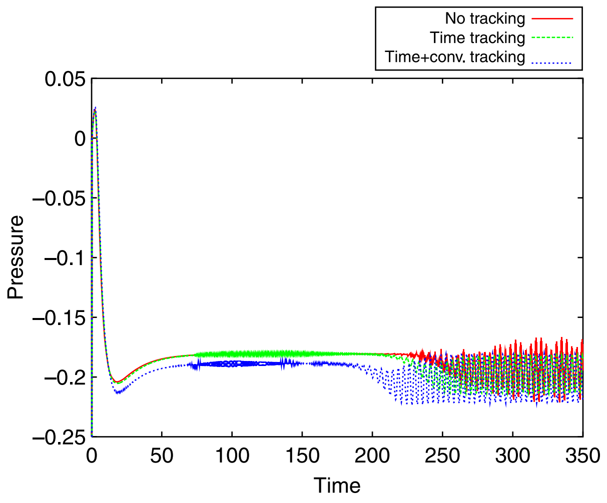
Note: 0 = Stationary solution. [a](Behr *et al.* 1995): 0.1624, 0.1661

converges much faster to its mesh converged value using the time tracking and even faster using the time + convection tracking. As for the time step tests, they do not exhibit any important differences.

Figures 14 and 15 show the pressure evolution behind the cylinder at position $(2.5, 0)$ and on the top of the cylinder at position $(0, 0.5)$ for the 10×10 mesh. The zoom at the solution shows that the classical method exhibits a time-step to time-step pressure oscillations. Both the time and time + convection tracking enable to correct it and to obtain a smooth pressure over the whole evolution. This point may be crucial when considering fluid structure interaction for which the nodal value of the pressure is required to pass the force to the structure solver. Note that this time decoupling of the pressure obtained with the classical method disappears when the mesh is refined.

8. Conclusion

We have revised the derivation of an Algebraic Subgrid Scale model used to stabilise the Navier–Stokes equations. This model is a two-level multiscale model,



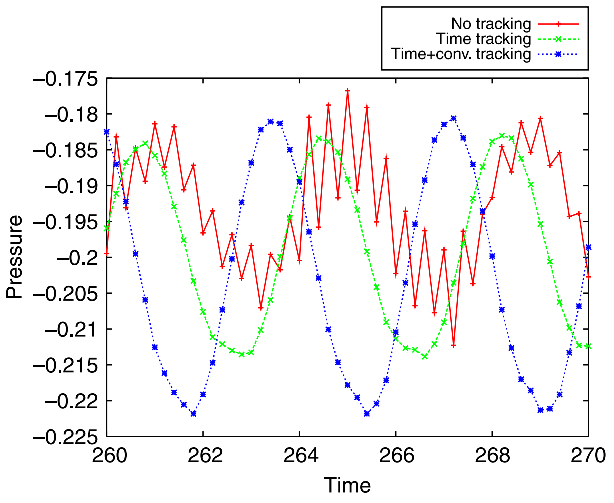
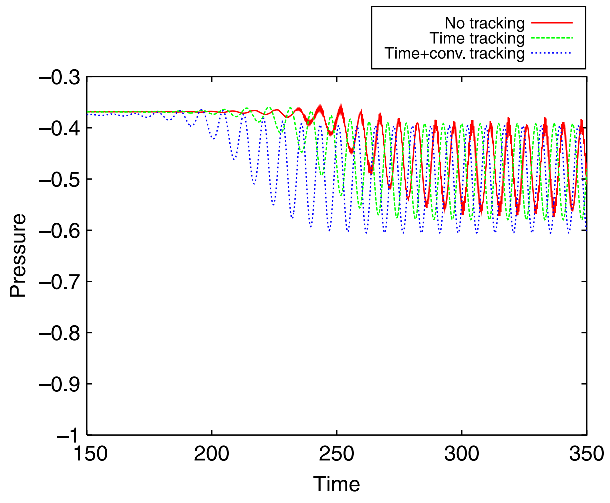


Figure 14. Cylinder: pressure evolution behind the cylinder, at $(2.5, 0)$. 10×10 mesh, $\delta t = 0.2$. (left) complete time evolution. (right) zoom.



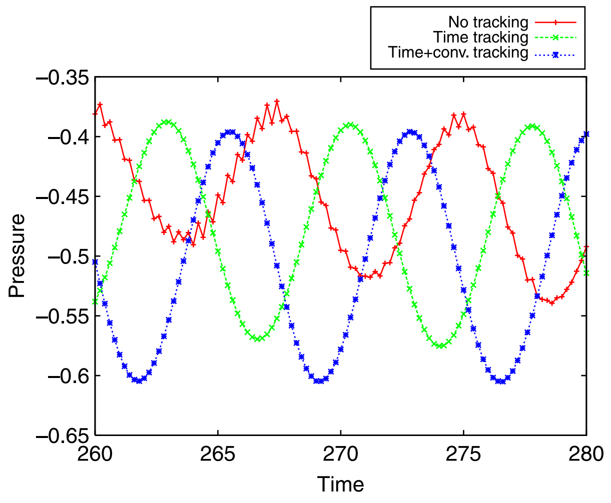


Figure 15. Cylinder: pressure evolution at the to of the cylinder, at $(0, 0.5)$. 10×10 mesh, $\delta t = 0.2$. (left) complete time evolution. (right) zoom.

Downloaded By: [Consorci de Biblioteques Universitaries de Catalunya] At: 16:16 13 February 2009

which splits the sought solution into a resolved scale and a subgrid scale. The subgrid scale is solved by approximating the inverse differential operator $\mathcal{L}^{-1}$ by an algebraic operator τ_K, which is evaluated element-wise. The method presented here is more general than the classical approach usually used in the literature, in the following sense:

- the time evolution $\partial_t \tilde{u}$ of the subgrid scale is taken into account;
- the subgrid scale is maintained in the convection velocity of the Navier–Stokes equations, i.e. $u_c = u_h + \tilde{u}$.

Through the solution of two numerical examples, we have shown the following points:

- the convection tracking could notably improve the solution in the case of the cavity flow. The convection tracking enables one to gain accuracy on coarse meshes and for high Reynolds numbers;
- the time and time + convection trackings improve the amplitude of the pressure lift in the case of the transient flow over a cylinder; in addition, the time tracking enables to smooth the pressure in time;
- relaxation was compulsory to solve the subgrid scale equation in the case of the cavity flow, when using the stationary equations. This was not the case for the other two flows.

The next points the authors are going to investigate are:

- the effects of the tracking with other elements (e.g. $P1/P1$);
- the use of more integration Gauss points to see if additional accuracy can be obtained.

Acknowledgements

This work has been carried out in the framework of the Spanish Project OPTIDIS (ENE2005-05274) and the research of Dr. Houzeaux has been partly done under a *Ramon y Cajal* contract with the Spanish *Ministerio de Educación y Ciencia*. J. Principe would like to acknowledge the support received from the Agencia de Gestió d'Ajuts Universitaris i de Recerca and the European Social Fund through a doctoral grant. The authors would like to thank Professor Ramon Codina for his structured imagination and for motivating this work, and also Professor Esteban Samaniego for his unstructured imagination and for participating in the writing of the introduction.

References

Batchelor, G., 1970. *An introduction to fluid dynamics*. Cambridge: Cambridge University Press.

Behr, M., Hastreiter, D., Mittal, S. and Tezduyar, T., 1995. 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, 309–316.

Bradshaw, P. and Huang, P., 1995. The law of the wall in turbulent flow. *Proceedings of the Royal Society of London, Osborne Reynolds centenary volume*, London (UK), 165–188.

Calo, V., 2004. *Residual based multiscale turbulence modeling: finite volume simulations of bypass transition*. Thesis (PhD). Department of Civil and Environmental Engineering, Stanford University.

Codina, R., 2001. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 190, 2681–2706.

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

Codina, R. and Blasco, J., 2002. Analysis of a stabilized finite element approximation of the transient convection-diffusion-reaction equation using orthogonal subscales. *Computing and Visualization in Science*, 4, 167–174.

Codina, R. and Soto, O., 1999. Finite element implementation of two-equation and algebraic stress turbulence models for steady incompressible flows. *International Journal for Numerical Methods in Fluids*, 30, 309–333.

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

Garikipati, K. and Hughes, T., 1998. A study of strain localization in a multiple scale framework: the one-dimensional problem. *Computer Methods in Applied Mechanics and Engineering*, 159, 193–222.

Ghia, U., Ghia, K. and Shin, C., 1982. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48, 387–411.

How, T. and Wu, X., 1997. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134, 169–189.

Johnson, C., 1987. *Numerical solution of partial differential equations by the finite element method*. Cambridge: Cambridge University Press.

Koobus, B. and Farhat, C., 2004. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes – application to vortex shedding. *Computer Methods in Applied Mechanics and Engineering*, 193 (15/16), 1367–1383.

Quarteroni, A. and Valli, A., 1994. *Numerical approximation of partial differential equations*. Springer-Verlag.

Roos, HG., Stynes, M. and Tobiska, L., 2007. *Numerical methods for singularly perturbed differential equations: convection-diffusion and flow problems*. Springer-Verlag.