ELSEVIER

Comput. Methods Appl. Mech. Engrg. 190 (2000) 411–430

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Finite element stabilization parameters computed from element matrices and vectors

Tayfun E. Tezduyar *, Yasuo Osawa

Mechanical Engineering and Materials Science, Team for Advanced Flow Simulation and Modeling, Rice University MS 321, 6100 Main Street, Houston, TX 77005, USA

Received 24 May 1999; received in revised form 7 June 1999

Abstract

We propose new ways of computing the stabilization parameters used in the stabilized finite element methods such as the streamline-upwind/Petrov–Galerkin (SUPG) and pressure-stabilizing/Petrov–Galerkin (PSPG) formulations. The parameters are computed based on the element-level matrices and vectors, which automatically take into account the local length scales, advection field and the Reynolds number. We describe how we compute these parameters in the context of first a time-dependent advection–diffusion equation and then the Navier–Stokes equations of unsteady incompressible flows.

1. Introduction

Stabilized formulations played a major role in the past two decades in making the finite element method a reliable and powerful approach in flow simulation and modeling. Among the most notable stabilized formulations are the streamline-upwind/Petrov–Galerkin (SUPG) formulation for incompressible flows [1], SUPG formulation for compressible flows [2], Galerkin/least-squares (GLS) formulation [3], and pressure-stabilizing/Petrov–Galerkin (PSPG) formulation for incompressible flows [4]. These stabilization techniques prevent numerical oscillations and other instabilities in solving problems with high Reynolds and/or Mach numbers and shocks and strong boundary layers, as well as when using equal-order interpolation functions for velocity and pressure and other unknowns. The SUPG, GLS and PSPG formulations stabilize the method without introducing excessive numerical dissipation. Because its symptoms are not necessarily qualitative, excessive numerical dissipation is not always easy to detect. This concern makes it desirable to seek and employ stabilized formulations developed with objectives that include keeping numerical dissipation to a minimum.

When the implementation of the SUPG, GLS or PSPG formulations is based on a sound understanding of these methods, they perform quite well. Furthermore, this class of stabilized formulations substantially improve the convergence rate in iterative solution of the large, coupled nonlinear equation system that needs to be solved at every time step of a flow computation. Such nonlinear systems are typically solved with the Newton–Raphson method, which involves, at its every iteration step, solution of a large, coupled linear equation system. It is in iterative solution of such linear equation systems that using a good stabilized method makes substantial difference in convergence, and this was pointed out in [5].

* Corresponding author. Tel.: +1 713 348 6051; fax: +1 713 348 5423; http://www.mems.rice.edu/TAFSM/.
E-mail address: tezduyar@rice.edu (T.E. Tezduyar).

0045-7825/00/$ - see front matter
PII: S0045-7825(00)00211-5

The SUPG formulation for incompressible flows was introduced in [6], with detailed description of the formulation and numerical examples given in [1]. The SUPG formulation for compressible flows was first introduced, in the context of conservation variables, in [2]. After that, several SUPG-like methods for compressible flows were developed. Taylor–Galerkin method [7], for example, is very similar, and under certain conditions is identical, to one of the stabilization methods introduced in [2]. Another example of the subsequent SUPG-like methods for compressible flows in conservation variables is the streamline-diffusion method described in [8]. Later, following [2], the SUPG formulation for compressible flows was recast in entropy variables and supplemented with a shock-capturing term [9]. It was shown in [10,11] that, the SUPG formulation introduced in [2], when supplemented with a similar shock-capturing term, is very comparable in accuracy to the one that was recast in entropy variables. It was shown in [12] for inviscid flows and in [11] for viscous flows that for 2D test problems computed, the SUPG formulation in conservation and entropy variables yield indistinguishable results. A recently introduced SUPG formulation for compressible flows in augmented conservation variables [13] leads to a proper incompressible flow formulation in the limit as the Mach number is taken to zero.

In the SUPG, GLS and PSPG methods, selection of the stabilization parameter, which is almost universally known as τ, has attracted a significant amount of attention and research. This stabilization parameter involves a measure of the local length scale (also known as "element length") and other parameters such as the local Reynolds and Courant numbers. Selection of the element length also attracted attention. Various element lengths and τ's were proposed starting with those in [1,2] followed by the one introduced in [14], and those proposed in the subsequently reported SUPG, GLS and PSPG methods. A number of τ's, dependent upon spatial and temporal discretizations, were introduced and tested in [15]. More recently, τ's which are applicable to higher-order elements were proposed by Franca et al. [16].

In this paper we introduce new ways of computing the stabilization parameter τ. The parameters we propose are computed from the element-level matrices and vectors, and these automatically take into account the local length scales as well as the advection field and the element-level Reynolds number. In Section 2, we describe these new ways of computing the stabilization parameter for a time-dependent advection–diffusion equation, and in Section 3 for the Navier–Stokes equations of unsteady incompressible flows. Numerical examples and concluding remarks are given in Sections 4 and 5.

2. Advection–diffusion equation

Let us consider over a domain Ω with boundary Γ the following time-dependent advection–diffusion equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \nabla \cdot (\nu \nabla \phi) = 0 \quad \text{on } \Omega, \tag{1}$$

where ϕ represents the quantity being transported (e.g., temperature, concentration), $\mathbf{u}$ is a divergence-free advection field, and ν is the diffusivity. The essential and natural boundary conditions associated with Eq. (1) are represented as

$$\phi = g \quad \text{on } \Gamma_g, \tag{2}$$

$$\mathbf{n} \cdot \nu \nabla \phi = h \quad \text{on } \Gamma_h, \tag{3}$$

where g and h are given functions, $\mathbf{n}$ is the unit normal vector at the boundary, and Γ_g and Γ_h are the complementary subsets of Γ. The initial condition consists of the form

$$\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) \text{ on } \Omega. \tag{4}$$

Let us assume that we have constructed some suitably defined finite-dimensional trial solution and test function spaces $\mathscr{S}_\phi^h$ and $\mathscr{V}_\phi^h$. The stabilized finite element formulation of Eq. (1) can then be written as follows: find $\phi^h \in \mathscr{S}_\phi^h$ such that $\forall w^h \in \mathscr{V}_\phi^h$:

$$\int_\Omega w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h \right) \mathrm{d}\Omega + \int_\Omega \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tau_{\mathrm{SUPG}} \mathbf{u}^h \cdot \nabla w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h - \nabla \cdot \left(\nu \nabla \phi^h \right) \right) \mathrm{d}\Omega = \int_{\Gamma_h} w^h h \, \mathrm{d}\Gamma. \tag{5}$$

Here n_{el} is the number of elements and Ω_e is the element domain corresponding to element e. τ_{SUPG} is the SUPG stabilization parameter.

Let us use the notation $\mathbf{b}$: $\int_{\Omega_e}(\ldots)\mathrm{d}\Omega$: $\mathbf{b}_{\mathrm{V}}$ to denote the element-level matrix $\mathbf{b}$ and element-level vector $\mathbf{b}_{\mathrm{V}}$ corresponding to the element-level integration term $\int_{\Omega_e}(\ldots)\mathrm{d}\Omega$. We now define the following element-level matrices and vectors:

$$\mathbf{m} : \int_{\Omega_e} w^h \frac{\partial \phi^h}{\partial t} \, \mathrm{d}\Omega \qquad : \mathbf{m}_{\mathrm{V}}, \tag{6}$$

$$\mathbf{c} : \int_{\Omega_e} w^h \mathbf{u}^h \cdot \nabla \phi^h \, \mathrm{d}\Omega \qquad : \mathbf{c}_{\mathrm{V}}, \tag{7}$$

$$\mathbf{k} : \int_{\Omega_e} \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega \qquad : \mathbf{k}_{\mathrm{V}}, \tag{8}$$

$$\tilde{\mathbf{k}} : \int_{\Omega_e} \mathbf{u}^h \cdot \nabla w^h \; \mathbf{u}^h \cdot \nabla \phi^h \, \mathrm{d}\Omega \qquad : \tilde{\mathbf{k}}_{\mathrm{V}}, \tag{9}$$

$$\tilde{\mathbf{c}} : \int_{\Omega_e} \mathbf{u}^h \cdot \nabla w^h \frac{\partial \phi^h}{\partial t} \, \mathrm{d}\Omega \qquad : \tilde{\mathbf{c}}_{\mathrm{V}}. \tag{10}$$

We define the element-level Reynolds and Courant numbers as follows:

$$Re = \frac{\|\mathbf{u}^h\|^2}{\nu} \frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{k}}\|}, \tag{11}$$

$$Cr_u = \frac{\Delta t}{2} \frac{\|\mathbf{c}\|}{\|\mathbf{m}\|}, \tag{12}$$

$$Cr_\nu = \frac{\Delta t}{2} \frac{\|\mathbf{k}\|}{\|\mathbf{m}\|}, \tag{13}$$

$$Cr_{\tilde{\nu}} = \frac{\Delta t}{2} \tau_{\mathrm{SUPG}} \frac{\|\tilde{\mathbf{k}}\|}{\|\mathbf{m}\|}, \tag{14}$$

where $\|\mathbf{b}\|$ is the norm of matrix $\mathbf{b}$.

Remark 1. The Courant numbers defined above can be used for determining the time step size of the computation.

The components of the element-matrix-based τ_{SUPG} are defined as follows:

$$\tau_{\mathrm{S1}} = \frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{k}}\|}, \tag{15}$$

$$\tau_{\mathrm{S2}} = \frac{\Delta t}{2} \frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{c}}\|}, \tag{16}$$

$$\tau_{\mathrm{S3}} = \tau_{\mathrm{S1}} Re = \left(\frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{k}}\|} \right) Re. \tag{17}$$

Remark 2. Because $\tilde{\mathbf{c}} = \mathbf{c}^{\mathrm{T}}$, for some definitions of the matrix norm, $\|\tilde{\mathbf{c}}\| = \|\mathbf{c}\|$, and therefore $\tau_{\mathrm{S2}} = (\Delta t/2)$.

Remark 3. In the special case of a 1D problem, $\tau_{S1} = (h/2|u|)$, $\tau_{S2} = (\Delta t/2)$ and $\tau_{S3} = (h^2/4\nu)$, which are the popular limits for τ_{SUPG} for the advection-dominated, transient-dominated and diffusion-dominated cases, respectively.

Several different but similar ways have been used to construct τ_{SUPG} from its components. We propose the form

$$\tau_{SUPG} = \left(\frac{1}{\tau_{S1}^r} + \frac{1}{\tau_{S2}^r} + \frac{1}{\tau_{S3}^r} \right)^{-1/r}, \tag{18}$$

which is based on the inverse of τ_{SUPG} being defined as the r-norm of the vector with components $1/\tau_{S1}$, $1/\tau_{S2}$ and $1/\tau_{S3}$. We note that the higher the integer r is, the sharper the switching between τ_{S1}, τ_{S2} and τ_{S3} becomes.

Remark 4. It is conceivable that we calculate a separate τ for each element node, or degree of freedom, or element equation. In that case, each component of τ would be calculated separately for each element node, or degree of freedom, or element equation. For this, we first represent an element matrix $\mathbf{b}$ in terms of its row vectors or row matrices: $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_{ex}}$. If we want a separate τ for each element node, then $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_{ex}}$, would be the row matrices corresponding to each element node, with $n_{ex} = n_{en}$, where n_{en} is the number of element nodes. If we want a separate τ for each degree of freedom, then $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_{ex}}$, would be the row matrices corresponding to each degree of freedom, with $n_{ex} = n_{dof}$, where n_{dof} is the number of degrees of freedom. If we want a separate τ for each element equation, then $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_{ex}}$ would be the row vectors corresponding to each element equation, with $n_{ex} = n_{ee}$, where n_{ee} is the number of element equations. Based on this, the components of τ would be calculated using the norms of these row matrices or vectors, instead of the element matrices. For example, a separate τ_{S1} for each element node would be calculated by using the expression $(\tau_{S1})_a = \|\mathbf{c}_a\|/\|\tilde{\mathbf{k}}_a\|$, $a = 1, 2, \ldots, n_{en}$. We should also note that in some special cases some of these alternative ways of computing τ might give the same result.

The components of the element-vector-based τ_{SUPG} are defined as follows:

$$\tau_{SV1} = \frac{\|\mathbf{c}_V\|}{\|\tilde{\mathbf{k}}_V\|}, \tag{19}$$

$$\tau_{SV2} = \frac{\|\mathbf{c}_V\|}{\|\tilde{\mathbf{c}}_V\|}, \tag{20}$$

$$\tau_{SV3} = \tau_{SV1} Re = \left(\frac{\|\mathbf{c}_V\|}{\|\tilde{\mathbf{k}}_V\|} \right) Re. \tag{21}$$

With these three components,

$$(\tau_{SUPG})_V = \left(\frac{1}{\tau_{SV1}^r} + \frac{1}{\tau_{SV2}^r} + \frac{1}{\tau_{SV3}^r} \right)^{-1/r}. \tag{22}$$

Remark 5. The definition of τ_{SUPG} given by Eq. (22) can be seen as a nonlinear definition because it depends on the solution. However, in marching from time level n to $n+1$ the element vectors can be evaluated at level n. This might be preferable in some cases.

Remark 6. In some cases it might be desirable to have a dynamic switching between τ_{SUPG} and $(\tau_{SUPG})_V$ during the computation.

Remark 7. Both definitions of τ_{SUPG} are applicable to higher-order elements.

Remark 8. It was pointed out in [5], and it is now well-known that the stabilization substantially improves the convergence in iterative solution of the linear equation system that needs to be solved at each Newton–Raphson step of the solution of the nonlinear equation system encountered at each time step.

Remark 9. It is also well-known that using higher-order elements degrades the convergence in iterative solution of such linear equation systems. It has been observed that [17] using interpolation functions which are spatially discontinuous across element boundaries improves the convergence for higher-order elements. Leaving aside the fact that such discontinuous methods come with substantial increases in computational cost which might render the approach impractical in large-scale 3D computations, it is our opinion that this convergence improvement is due to breaking the global, unintended approximation of the lower-order functions by the higher-order functions. As one uses higher- and higher-order functions, the global approximation of the lower-order functions by the higher-order functions gets better, and the approximate and unintended linear dependence of the lower-order functions on the higher-order functions increases. The spatial discontinuity breaks this global approximate linear dependence.

Remark 10. In Eq. (5), the SUPG stabilization term involves the residual of the governing equation, which includes the second-order term $\nabla \cdot (\nu \nabla \phi^h)$. For linear (triangular and tetrahedral) elements this term vanishes. For bilinear (quadrilateral) and trilinear (hexahedral) elements, this term vanishes for certain special geometries (e.g. rectangles and bricks), and is largely under-represented for more general geometries. This steals away from the consistency of the stabilized finite element formulation as defined by Eq. (5). If one desires to remedy the situation, it is our opinion that this could be accomplished by modifying the stabilized formulation given by Eq. (5) in such a way that the terms representing the discontinuity of the flux $\nu \nabla \phi^h$ across element boundaries (i.e. flux "jump" terms) are included in the SUPG stabilization terms. It is also our opinion that the best way to derive effective stabilized formulations is to start with the Galerkin formulation of the problem, reverse integrate-by-parts to derive the finite-dimensional Euler–Lagrange form of the equations, and consider including as a factor in the stabilization terms any of the terms appearing in the Euler–Lagrange form (such as the governing partial differential equations and the flux jump terms as well as the difference between the natural boundary condition and the flux at that boundary). The procedure is described in more detail in Appendix A.

3. Navier–Stokes equations of incompressible flows

We write the Navier–Stokes equations of incompressible flows as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}\right) - \nabla \cdot \boldsymbol{\sigma} = 0 \quad \text{on } \Omega, \tag{23}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega, \tag{24}$$

where ρ is density (constant in this case), $\mathbf{u}$ is the velocity vector, $\mathbf{f}$ is the external force and $\boldsymbol{\sigma}$ is the stress tensor

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + \mathbf{T}. \tag{25}$$

Here p is the pressure, $\mathbf{I}$ is the identity tensor and

$$\mathbf{T} = 2\mu\boldsymbol{\varepsilon}(\mathbf{u}), \tag{26}$$

where $\mu = \rho\nu$ is the viscosity, ν is the kinematic viscosity, and $\boldsymbol{\varepsilon}$ is the strain-rate tensor

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}((\nabla \mathbf{u}) + (\nabla \mathbf{u})^{\mathrm{T}}). \tag{27}$$

The essential and natural boundary conditions associated with Eq. (23) are represented as

$$\mathbf{u} = \mathbf{g} \text{ on } \Gamma_g, \tag{28}$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \text{ on } \Gamma_h. \tag{29}$$

The initial condition on $\mathbf{u}$ is given as

$$\mathbf{u}(\mathbf{x},0)=\mathbf{u}_0(\mathbf{x}), \tag{30}$$

where $\nabla\cdot\mathbf{u}_0=0$.

Let us again assume that we have some suitably defined finite-dimensional trial solution and test function spaces for velocity and pressure: $\mathscr{S}_{\mathbf{u}}^h$, $\mathscr{V}_{\mathbf{u}}^h$, $\mathscr{S}_p^h$ and $\mathscr{V}_p^h=\mathscr{S}_p^h$. The stabilized finite element formulation of Eqs. (23) and (24) can then be written as follows: find $\mathbf{u}^h\in\mathscr{S}_{\mathbf{u}}^h$ and $p^h\in\mathscr{S}_p^h$ such that $\forall\mathbf{w}^h\in\mathscr{V}_{\mathbf{u}}^h$ and $q^h\in\mathscr{V}_p^h$:

$$\begin{aligned}
&\int_\Omega \mathbf{w}^h\cdot\rho\left(\frac{\partial\mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h-\mathbf{f}\right)\mathrm{d}\Omega+\int_\Omega\boldsymbol{\varepsilon}(\mathbf{w}^h):\boldsymbol{\sigma}(p^h,\mathbf{u}^h)\,\mathrm{d}\Omega+\int_\Omega q^h\nabla\cdot\mathbf{u}^h\,\mathrm{d}\Omega\\
&\quad+\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\frac{1}{\rho}[\tau_{\mathrm{SUPG}}\rho\mathbf{u}^h\cdot\nabla\mathbf{w}^h+\tau_{\mathrm{PSPG}}\nabla q^h]\cdot\left[\rho\left(\frac{\partial\mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h\right)-\nabla\cdot\boldsymbol{\sigma}(p^h,\mathbf{u}^h)-\rho\mathbf{f}\right]\mathrm{d}\Omega\\
&\quad+\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\tau_{\mathrm{LSIC}}\nabla\cdot\mathbf{w}^h\rho\nabla\cdot\mathbf{u}^h\,\mathrm{d}\Omega=\int_{\Gamma_h}\mathbf{w}^h\cdot\mathbf{h}^h\,\mathrm{d}\Gamma.
\end{aligned}\tag{31}$$

Here τ_{PSPG} is the PSPG (pressure-stabilizing/Petrov–Galerkin) stabilization parameter and τ_{LSIC} is the LSIC (least-squares on incompressibility constant) stabilization parameter.

We now define the following element-level matrices and vectors:

$$\mathbf{m}:\int_{\Omega_e}\mathbf{w}^h\cdot\rho\frac{\partial\mathbf{u}^h}{\partial t}\,\mathrm{d}\Omega \qquad :\mathbf{m}_{\mathrm{V}}, \tag{32}$$

$$\mathbf{c}:\int_{\Omega_e}\mathbf{w}^h\cdot\rho(\mathbf{u}^h\cdot\nabla\mathbf{u}^h)\,\mathrm{d}\Omega \qquad :\mathbf{c}_{\mathrm{V}}, \tag{33}$$

$$\mathbf{k}:\int_{\Omega_e}\boldsymbol{\varepsilon}(\mathbf{w}^h):2\mu\boldsymbol{\varepsilon}(\mathbf{u}^h)\,\mathrm{d}\Omega \qquad :\mathbf{k}_{\mathrm{V}}, \tag{34}$$

$$\mathbf{g}:\int_{\Omega_e}(\nabla\cdot\mathbf{w}^h)p^h\,\mathrm{d}\Omega \qquad :\mathbf{g}_{\mathrm{V}}, \tag{35}$$

$$\mathbf{g}^{\mathrm{T}}:\int_{\Omega_e}q^h(\nabla\cdot\mathbf{u}^h)\,\mathrm{d}\Omega \qquad :\mathbf{g}_{\mathrm{V}}^{\mathrm{T}}, \tag{36}$$

$$\tilde{\mathbf{k}}:\int_{\Omega_e}(\mathbf{u}^h\cdot\nabla\mathbf{w}^h)\cdot\rho(\mathbf{u}^h\cdot\nabla\mathbf{u}^h)\,\mathrm{d}\Omega \qquad :\tilde{\mathbf{k}}_{\mathrm{V}}, \tag{37}$$

$$\tilde{\mathbf{c}}:\int_{\Omega_e}(\mathbf{u}^h\cdot\nabla\mathbf{w}^h)\cdot\rho\frac{\partial\mathbf{u}^h}{\partial t}\,\mathrm{d}\Omega \qquad :\tilde{\mathbf{c}}_{\mathrm{V}}, \tag{38}$$

$$\tilde{\gamma}:\int_{\Omega_e}(\mathbf{u}^h\cdot\nabla\mathbf{w}^h)\cdot\nabla p^h\,\mathrm{d}\Omega \qquad :\tilde{\gamma}_{\mathrm{V}}, \tag{39}$$

$$\beta:\int_{\Omega_e}\nabla q^h\cdot\frac{\partial\mathbf{u}^h}{\partial t}\,\mathrm{d}\Omega \qquad :\beta_{\mathrm{V}}, \tag{40}$$

$$\gamma:\int_{\Omega_e}\nabla q^h\cdot(\mathbf{u}^h\cdot\nabla\mathbf{u}^h)\,\mathrm{d}\Omega \qquad :\gamma_{\mathrm{V}}, \tag{41}$$

$$\theta:\int_{\Omega_e}\nabla q^h\cdot\nabla p^h\,\mathrm{d}\Omega \qquad :\theta_{\mathrm{V}}, \tag{42}$$

$$\mathbf{e}:\int_{\Omega_e}(\nabla\cdot\mathbf{w}^h)\rho(\nabla\cdot\mathbf{u}^h)\,\mathrm{d}\Omega \qquad :\mathbf{e}_{\mathrm{V}}. \tag{43}$$

Remark 11. In the definition of the element-level matrices listed above, we assume that $\mathbf{u}^h$ appearing in the advective operator (i.e. in $\mathbf{u}^h\cdot\nabla\mathbf{u}^h$ and $\mathbf{u}^h\cdot\nabla\mathbf{w}^h$) is evaluated at time level n rather than $n+1$. The definition would essentially be the same if we, alternatively, assumed that it is evaluated at time

level $n+1$ but nonlinear iteration level i rather than $i+1$. Except, in the first option, in the advective operator we use $(\mathbf{u}^h)_n$, whereas in the second option we use $(\mathbf{u}^h)^i_{n+1}$. The second option can be seen as a nonlinear definition. The first option might be preferable in some cases. In the definition of the element-level-vectors, we face the same choices in terms of the evaluation of $\mathbf{u}^h$ in the advective operator.

Remark 12. We note that $\tilde{\mathbf{c}} = \mathbf{c}^{\mathrm{T}}$ and $\tilde{\gamma} = \gamma^{\mathrm{T}}$.

The element-level Reynolds and Courant numbers are defined the same way as they were defined before, as given by Eqs. (11)–(14).

Remark 13. Remark 1 applies also in this case.

The components of the element-matrix-based τ_{SUPG} are defined the same way as they were defined before, as given by Eqs. (15)–(17).

Remark 14. Remarks 2 and 3 also apply in this case.

τ_{SUPG} is constructed from its components the same way as it was constructed before, as given by Eq. (18).

Remark 15. Remark 4 applies also in this case.

The components of the element-vector-based τ_{SUPG} are defined the same way as they were defined before, as given by Eqs. (19)–(21). The construction of $(\tau_{\mathrm{SUPG}})_{\mathrm{V}}$ is also the same as it was before, given by Eq. (22).

Remark 16. Remarks 5–7 apply also in this case.

The components of the element-matrix-based τ_{PSPG} are defined as follows:

$$\tau_{\mathrm{P1}} = \frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\gamma\|}, \tag{44}$$

$$\tau_{\mathrm{P2}} = \frac{\Delta t}{2}\frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\beta\|}, \tag{45}$$

$$\tau_{\mathrm{P3}} = \tau_{\mathrm{P1}} Re = \left(\frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\gamma\|}\right) Re. \tag{46}$$

Remark 17. Remark 3 applies also in this case.

τ_{PSPG} is constructed from its components as follows:

$$\tau_{\mathrm{PSPG}} = \left(\frac{1}{\tau_{\mathrm{P1}}^r} + \frac{1}{\tau_{\mathrm{P2}}^r} + \frac{1}{\tau_{\mathrm{P3}}^r}\right)^{-1/r}. \tag{47}$$

Remark 18. Remark 4 applies also in this case.

The components of the element-vector-based τ_{PSPG} are defined as follows:

$$\tau_{\mathrm{PV1}} = \tau_{\mathrm{P1}}, \tag{48}$$

$$\tau_{\mathrm{PV2}} = \tau_{\mathrm{PV1}} \frac{\|\gamma_{\mathrm{V}}\|}{\|\beta_{\mathrm{V}}\|}, \tag{49}$$

$$\tau_{\mathrm{PV3}} = \tau_{\mathrm{PV1}} Re. \tag{50}$$

With these components,

$$(\tau_{\mathrm{PSPG}})_{\mathrm{V}} = \left(\frac{1}{\tau_{\mathrm{PV1}}^r} + \frac{1}{\tau_{\mathrm{PV2}}^r} + \frac{1}{\tau_{\mathrm{PV3}}^r} \right)^{-1/r}. \tag{51}$$

Remark 19. Remarks 5–7 apply also in this case.

The element-matrix-based τ_{LSIC} is defined as follows:

$$\tau_{\mathrm{LSIC}} = \frac{\|\mathbf{c}\|}{\|\mathbf{e}\|}. \tag{52}$$

Remark 20. In the special case of a 1D problem, $\tau_{\mathrm{LSIC}} = |u|h/2$.

Remark 21. Remark 4 applies also in this case.

We define the element-vector-based τ_{LSIC} to be identical to the element-matrix-based τ_{LSIC}

$$(\tau_{\mathrm{LSIC}})_{\mathrm{V}} = \tau_{\mathrm{LSIC}}. \tag{53}$$

Remark 22. Remark 7 applies also in this case.

Remark 23. Remarks 8 and 9 apply also to the Navier–Stokes equations of incompressible flows.

Remark 24. Remark 10 applies also to the stabilized formulation of the Navier–Stokes equations of incompressible flows. However it needs to be pointed out that, for both the advection–diffusion and Navier–Stokes equations, under-representation of the second-order terms $\nabla \cdot (\nu \nabla \phi^h)$ and $\nabla \cdot (2\mu\boldsymbol{\varepsilon}(\mathbf{u}^h))$ does not steal much from the consistency of the stabilized finite element formulation at Reynolds numbers high enough (i.e. $Re \gg 1$) to render these second-order terms negligible compared to the advective term.

4. Numerical tests

For the purpose of numerical tests and comparison with stabilization parameters we used earlier, we define here those stabilization parameters which are based on an earlier definition of the length scale h [14]:

$$h_{\mathrm{UGN}} = 2 \, \|\mathbf{u}^h\| \left(\sum_{a=1}^{n_{\mathrm{en}}} |\mathbf{u}^h \cdot \nabla N_a| \right)^{-1}, \tag{54}$$

where N_a is the interpolation function associated with node a. The stabilization parameters are defined as follows:

$$\tau_{\mathrm{SUGN1}} = \frac{h_{\mathrm{UGN}}}{2\|\mathbf{u}\|}, \tag{55}$$

$$\tau_{\mathrm{SUGN2}} = \frac{\Delta t}{2}, \tag{56}$$

$$\tau_{\mathrm{SUGN3}} = \frac{h_{\mathrm{UGN}}^2}{4\nu}, \tag{57}$$

$$(\tau_{\mathrm{SUPG}})_{\mathrm{UGN}} = \left(\frac{1}{\tau_{\mathrm{SUGN1}}^2} + \frac{1}{\tau_{\mathrm{SUGN2}}^2} + \frac{1}{\tau_{\mathrm{SUGN3}}^2} \right)^{-1/2}, \tag{58}$$

$$(\tau_{\mathrm{PSPG}})_{\mathrm{UGN}} = (\tau_{\mathrm{SUPG}})_{\mathrm{UGN}}, \tag{59}$$

$$(\tau_{\mathrm{LSIC}})_{\mathrm{UGN}} = \frac{h_{\mathrm{UGN}}}{2} \|\mathbf{u}^h\| z. \tag{60}$$

Here z is given as follows:

$$z = \begin{cases} \left(\dfrac{Re_{\mathrm{UGN}}}{3} \right) & Re_{\mathrm{UGN}} \leqslant 3, \\ 1 & Re_{\mathrm{UGN}} > 3, \end{cases} \tag{61}$$

where $Re_{\mathrm{UGN}} = \dfrac{\|\mathbf{u}^h\| h_{\mathrm{UGN}}}{2\nu}$.

4.1. Elements with simple shapes

In this section, for some 2D simple element shapes, we compare the stabilization parameters calculated with the approaches described in this paper. We use four different quadrilateral elements and three different triangles. The quadrilateral elements are: a square, a rectangle with aspect ratio 2, a parallelogram, and a trapezoid (see Fig. 1). The triangular elements are: a right isosceles triangle, a right triangle, and an

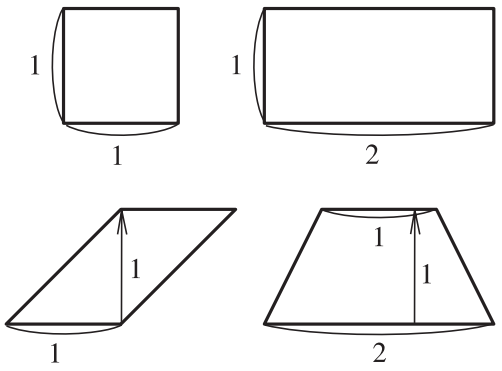

Fig. 1. Quadrilateral elements used in numerical tests with 2D simple element shapes. A square (upper-left), a rectangle with aspect ratio 2 (upper-right), a parallelogram (lower-left), and a trapezoid (lower-right).

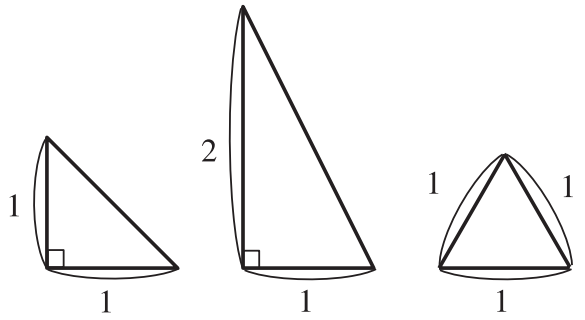

Fig. 2. Triangular elements used in numerical tests with 2D simple element shapes. A right isosceles triangle (left), a right triangle (middle), and an equilateral triangle (right).

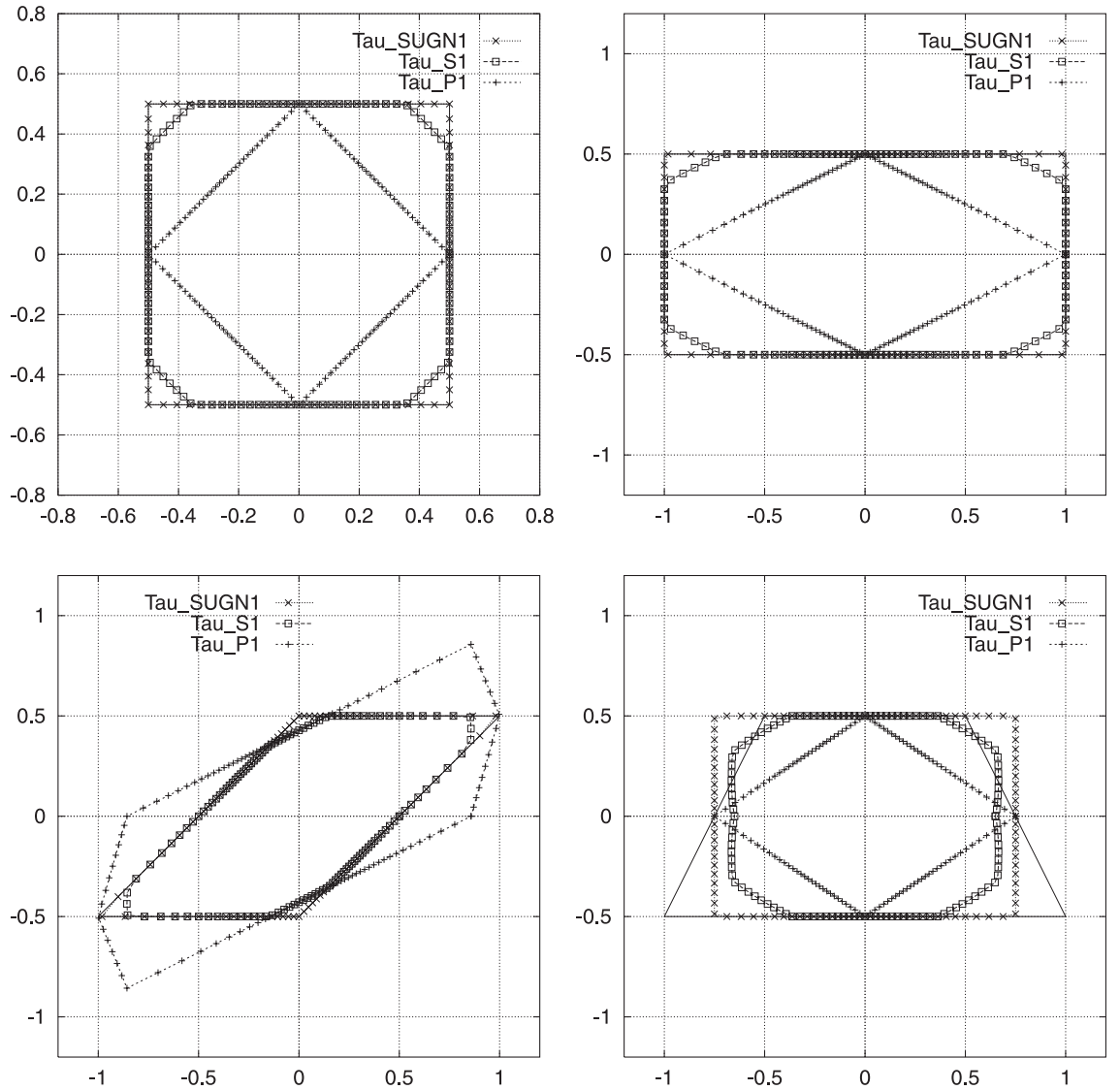

Fig. 3. τ_{SUGN1}, τ_{S1}, and τ_{P1} for quadrilateral elements. A square (upper-left), a rectangle with aspect ratio 2 (upper-right), a parallelogram (lower-left), and a trapezoid (lower-right).

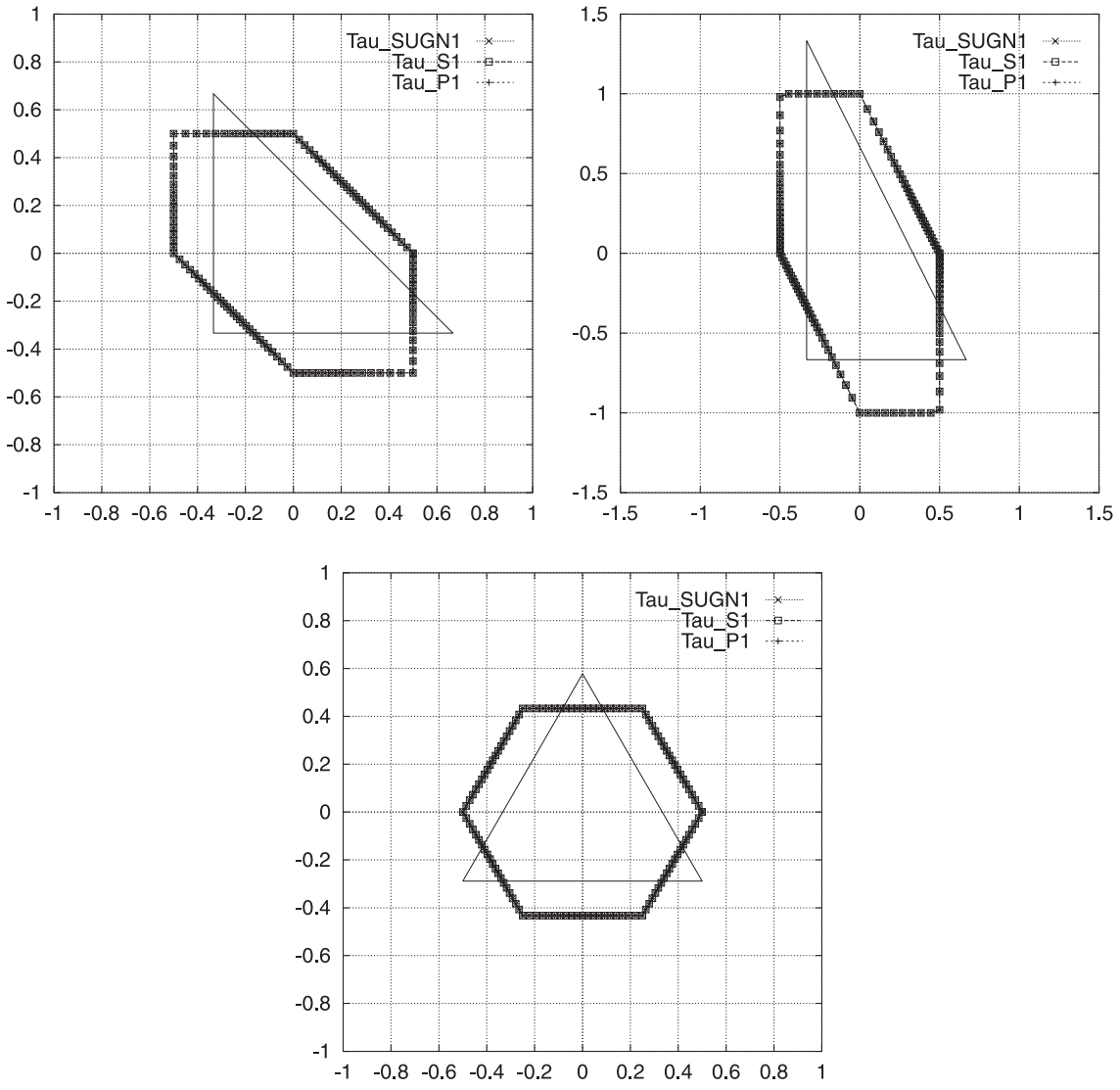

Fig. 4. τ_{SUGN1}, τ_{S1}, and τ_{P1} for triangular elements. A right isosceles triangle (upper-left), a right triangle (upper-right), and an equilateral triangle (lower).

equilateral triangle (see Fig. 2). We set $\|\mathbf{u}^h\| = 1$ and $\Delta t = 1$, and vary the flow direction from 0° to 360° at equal intervals.

Figs. 3 and 4 show, for the quadrilateral and triangular elements respectively, τ_{SUGN1}, τ_{S1}, and τ_{P1}. We note that for the triangular elements all three are equivalent.

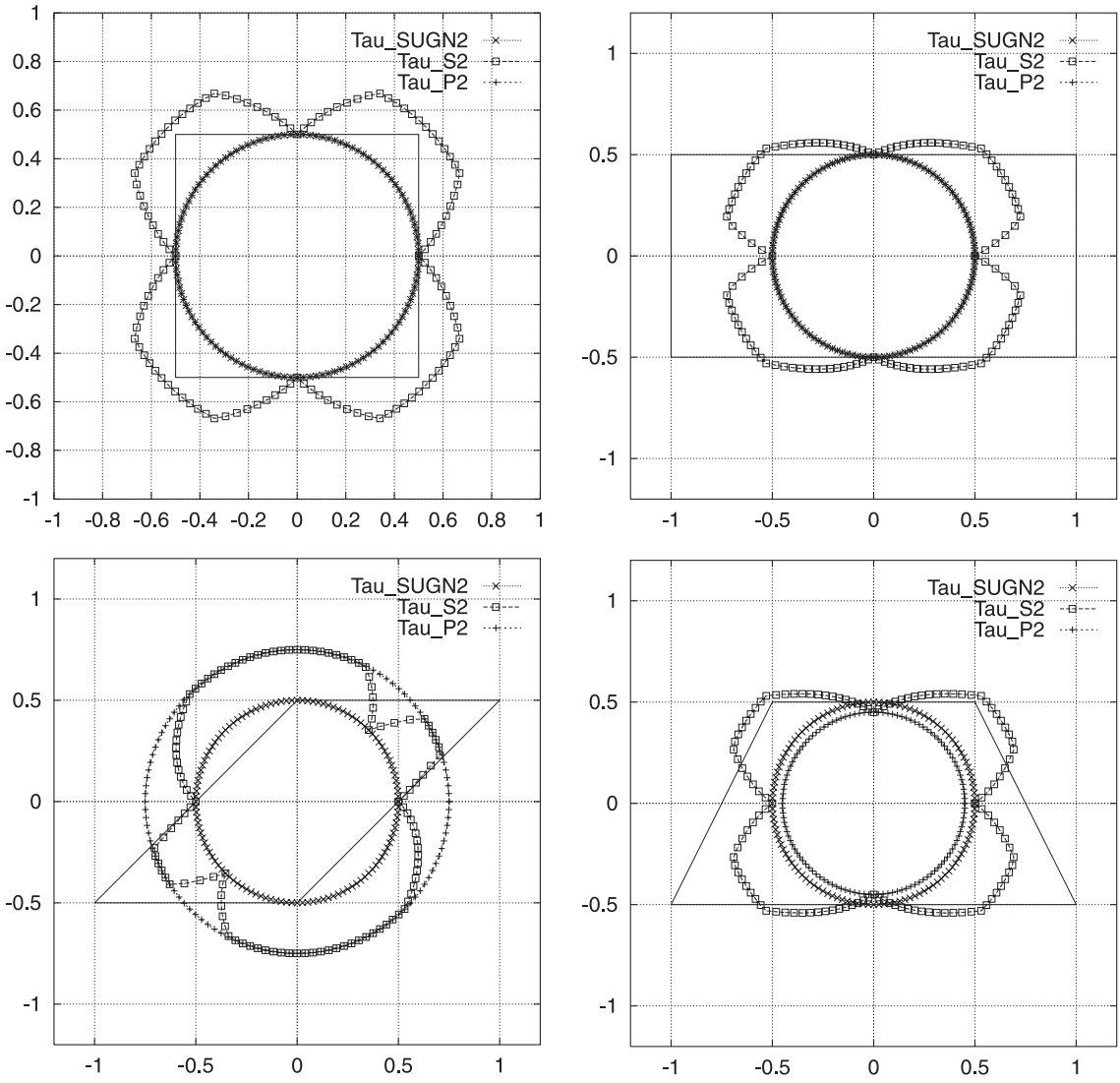

Fig. 5. τ_{SUGN2}, τ_{S2}, and τ_{P2} for quadrilateral elements. A square (upper-left), a rectangle with aspect ratio 2 (upper-right), a parallelogram (lower-left), and a trapezoid (lower-right).

Figs. 5 and 6 show, for the quadrilateral and triangular elements respectively, τ_{SUGN2}, τ_{S2}, and τ_{P2}. For the quadrilateral elements all three are different from each other, except for the special shapes of square and rectangle, for which τ_{SUGN2} and τ_{P2} are identical. For the triangular elements for all three shapes τ_{SUGN2} is different than than the other two.

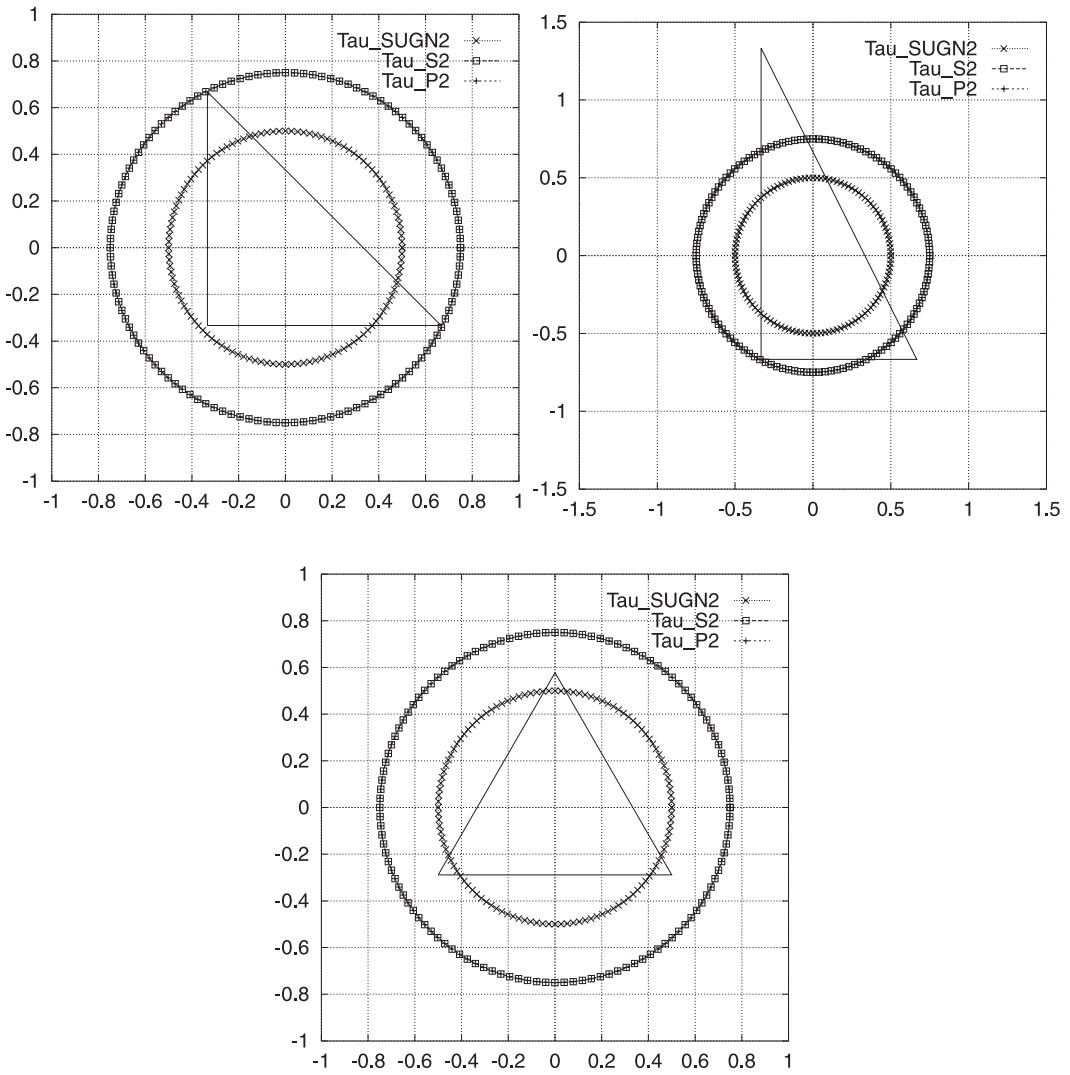

Fig. 6. τ_{SUGN2}, τ_{S2}, and τ_{P2} for triangular elements. A right isosceles triangle (upper-left), a right triangle (upper-right), and an equilateral triangle (lower).

Table 1
RMS of the error at the nodes

	$(\tau_{SUPG})_{UGN}$	τ_{SUPG}	$(\tau_{SUPG})_{V}$
Along $y = 0.5$	7.67E−2	7.73E−2	7.59E−2
Along $x = 0.5$	4.97E−2	5.01E−2	4.91E−2

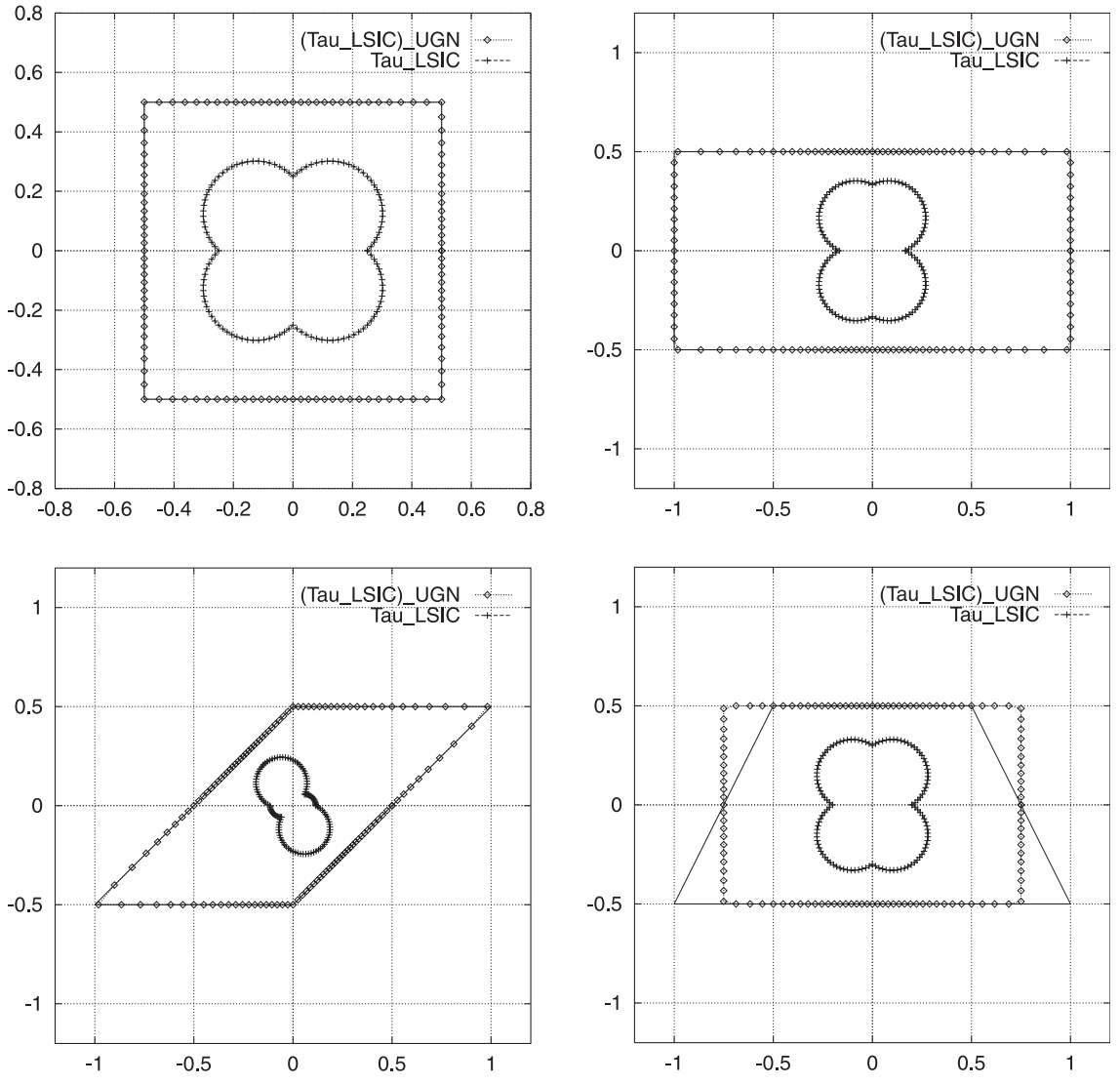

Fig. 7. $(\tau_{\mathrm{LSIC}})_{\mathrm{UGN}}$ and τ_{LSIC} for quadrilateral elements. A square (upper-left), a rectangle with aspect ratio 2 (upper-right), a parallelogram (lower-left), and a trapezoid (lower-right).

Figs. 7 and 8 show, for the quadrilateral and triangular elements respectively, $(\tau_{\mathrm{LSIC}})_{\mathrm{UGN}}$ and τ_{LSIC}. We note that in all cases τ_{LSIC} smaller than $(\tau_{\mathrm{LSIC}})_{\mathrm{UGN}}$.

4.2. 2D advection–diffusion problem

Here we compare the performance of the stabilization parameters in a 2D advection–diffusion problem, where the advection is skew to the mesh and the diffusivity is negligible. Fig. 9 shows the problem set up. The diffusivity $\nu = 1 \times 10^{-6}$ and $\|\mathbf{u}^h\| = 1$. The flow direction is 30° from the x-axis. The domain is square and is discretized by using 20×20 square elements. The time step size is 0.1. The stabilization parameters tested are: $(\tau_{\mathrm{SUPG}})_{\mathrm{UGN}}$, τ_{SUPG}, and $(\tau_{\mathrm{SUPG}})_{\mathrm{V}}$.

Fig. 10 shows the solution along $y = 0.5$ and $x = 0.5$. The exact solution is calculated by assuming zero diffusion. Table 1 shows the root-mean-square (RMS) of the error at the nodes. We observe that the solutions obtained with the stabilization parameters tested are almost identical.

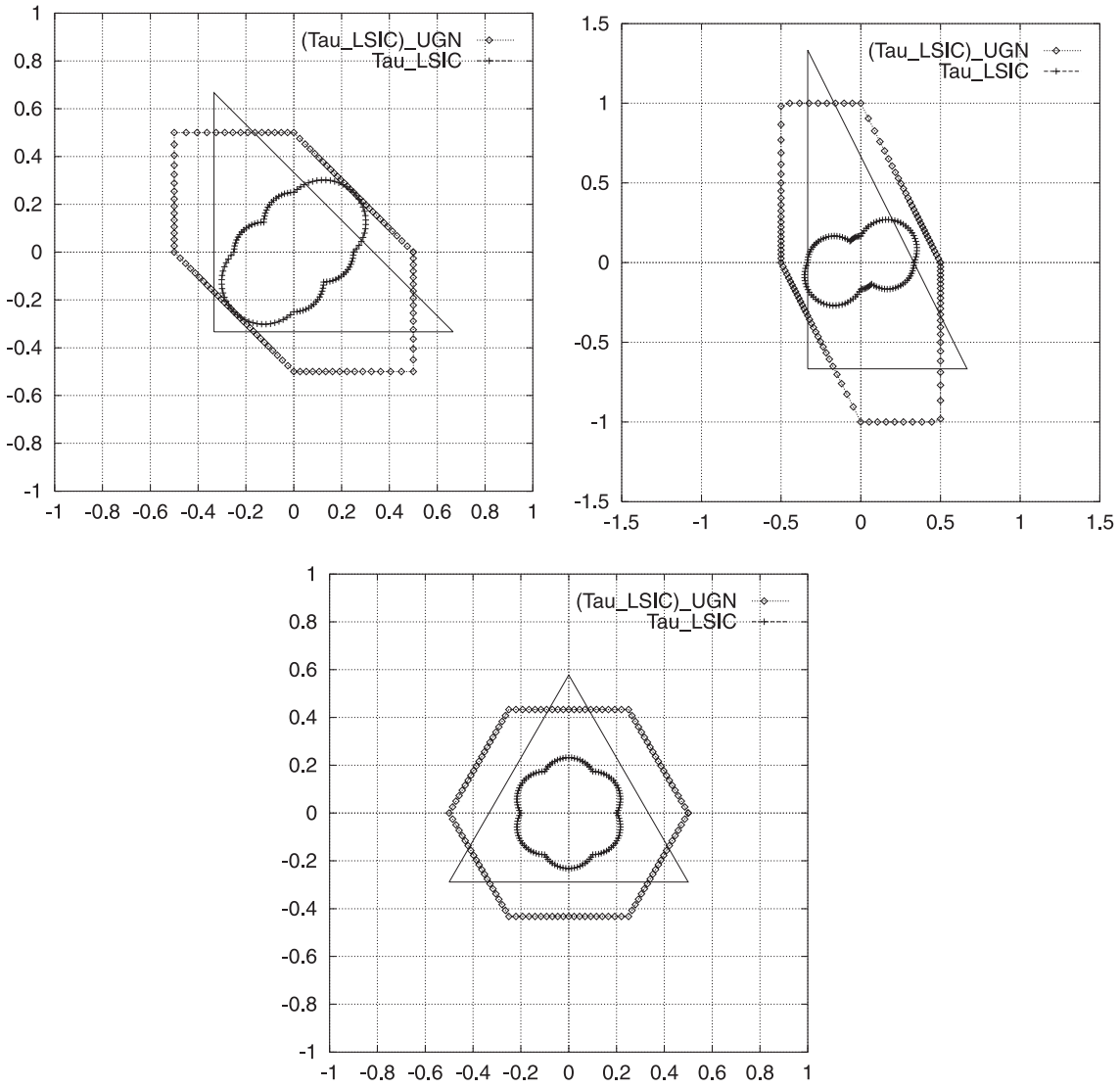

Fig. 8. $(\tau_{LSIC})_{UGN}$ and τ_{LSIC} for triangular elements. A right isosceles triangle (upper-left), a right triangle (upper-right), and an equilateral triangle (lower).

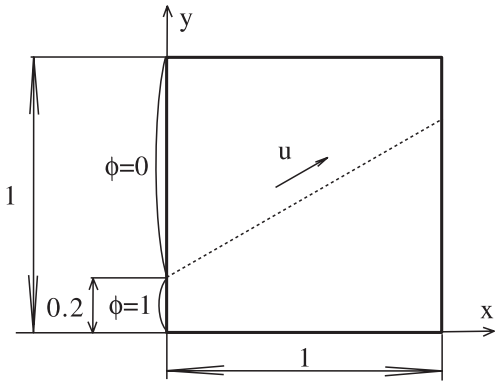

Fig. 9. 2D advection–diffusion problem. Problem set up.

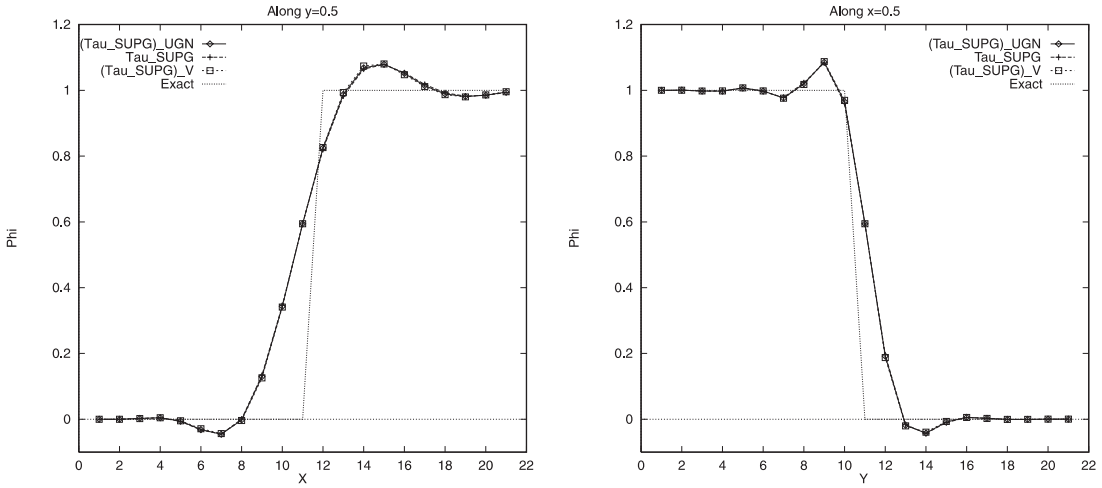

Fig. 10. 2D advection–diffusion problem. Solution obtained with three different stabilization parameters, compared to the exact solution. Along $y = 0.5$ (left) and $x = 0.5$ (right).

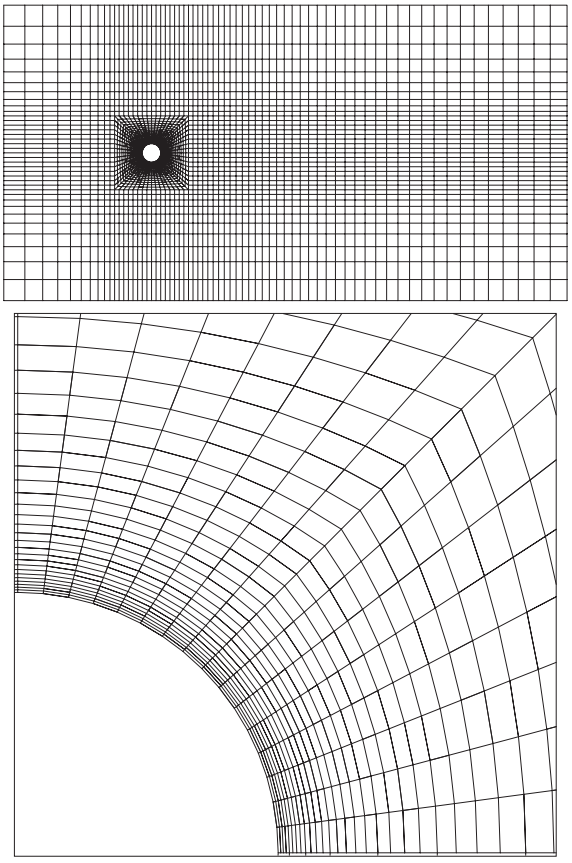

Fig. 11. Flow past a cylinder. Mesh.

4.3. 2D Navier–Stokes problem

In this section, we compare the stabilization parameters in computation of 2D flow past a cylinder at $Re = 100$. The cylinder, with the radius one unit, is located at the origin of the computational domain. The upstream, downstream and crossflow (i.e. top and bottom lateral) boundaries are located, respectively, at 16, 45 and 16 units from the origin (see Fig. 11). The mesh consists of 4558 nodes and 4424 quadrilateral elements. The boundary conditions consist of uniform inflow velocity, zero-shear stress and zero-normal velocity at the lateral boundaries, traction-free condition at the outflow boundary, and no-slip condition on the cylinder. The free-stream velocity is set to 1, and the time step size to 0.1. Three Newton–Raphson iterations are performed at every time step. At each Newton–Raphson step, the linear equation system encountered is solved iteratively with the GMRES update method with a Krylov space size of 20 and without restart.

Fig. 12 shows, at later stages of the computation, time history of the drag and lift coefficients. Table 2 shows the average drag coefficient and Strouhal number. Here $(\tau)_{\mathrm{UGN}}$ represents the group of stabilization parameters $(\tau_{\mathrm{SUPG}})_{\mathrm{UGN}}$, $(\tau_{\mathrm{PSPG}})_{\mathrm{UGN}}$, and $(\tau_{\mathrm{LSIC}})_{\mathrm{UGN}}$; τ represents τ_{SUPG}, τ_{PSPG}, and τ_{LSIC}; and $(\tau)_{\mathrm{V}}$ represents $(\tau_{\mathrm{SUPG}})_{\mathrm{V}}$, $(\tau_{\mathrm{PSPG}})_{\mathrm{V}}$, and $(\tau_{\mathrm{LSIC}})_{\mathrm{V}}$. Fig. 13 shows the values of the stabilization parameters along the vertical line passing through the origin and starting from the upper cylinder surface.

The solution obtained with τ shows slightly less dissipation than the other two solutions, as indicated by a slightly higher Strouhal number (in this Reynolds number range, the Strouhal number increases with increasing Reynolds number [18]).

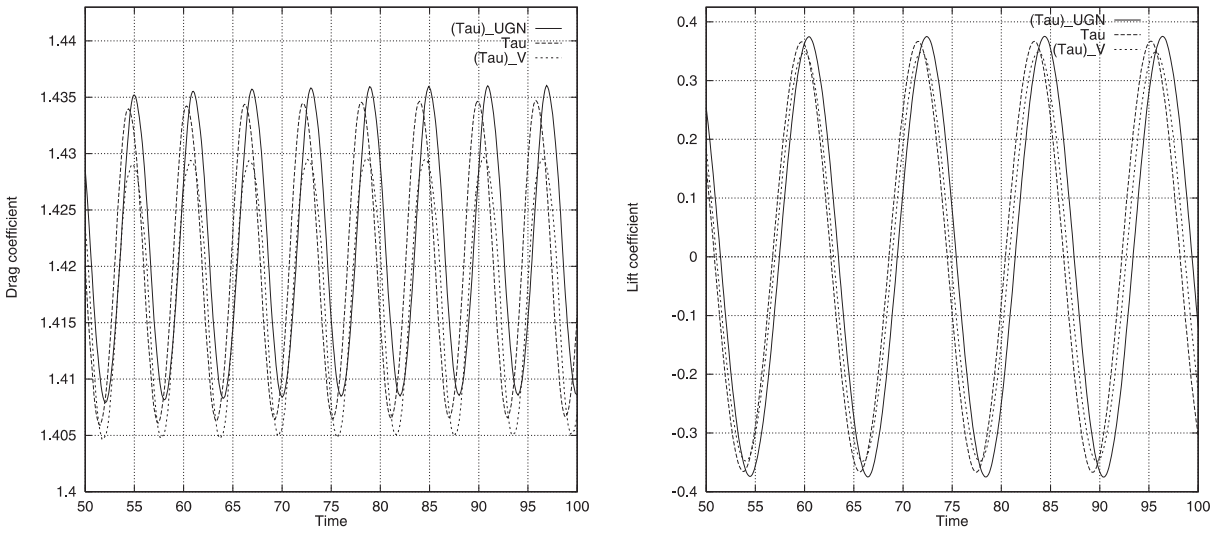

Fig. 12. Flow past a cylinder. Time history of the drag (left) and lift (right) coefficients.

Table 2
Flow past a cylinder[a]

	$(\tau)_{\mathrm{UGN}}$	τ	$(\tau)_{\mathrm{V}}$
Drag coefficient	1.422	1.421	1.417
Strouhal number	0.167	0.169	0.168

[a] Average drag coefficient and Strouhal number.

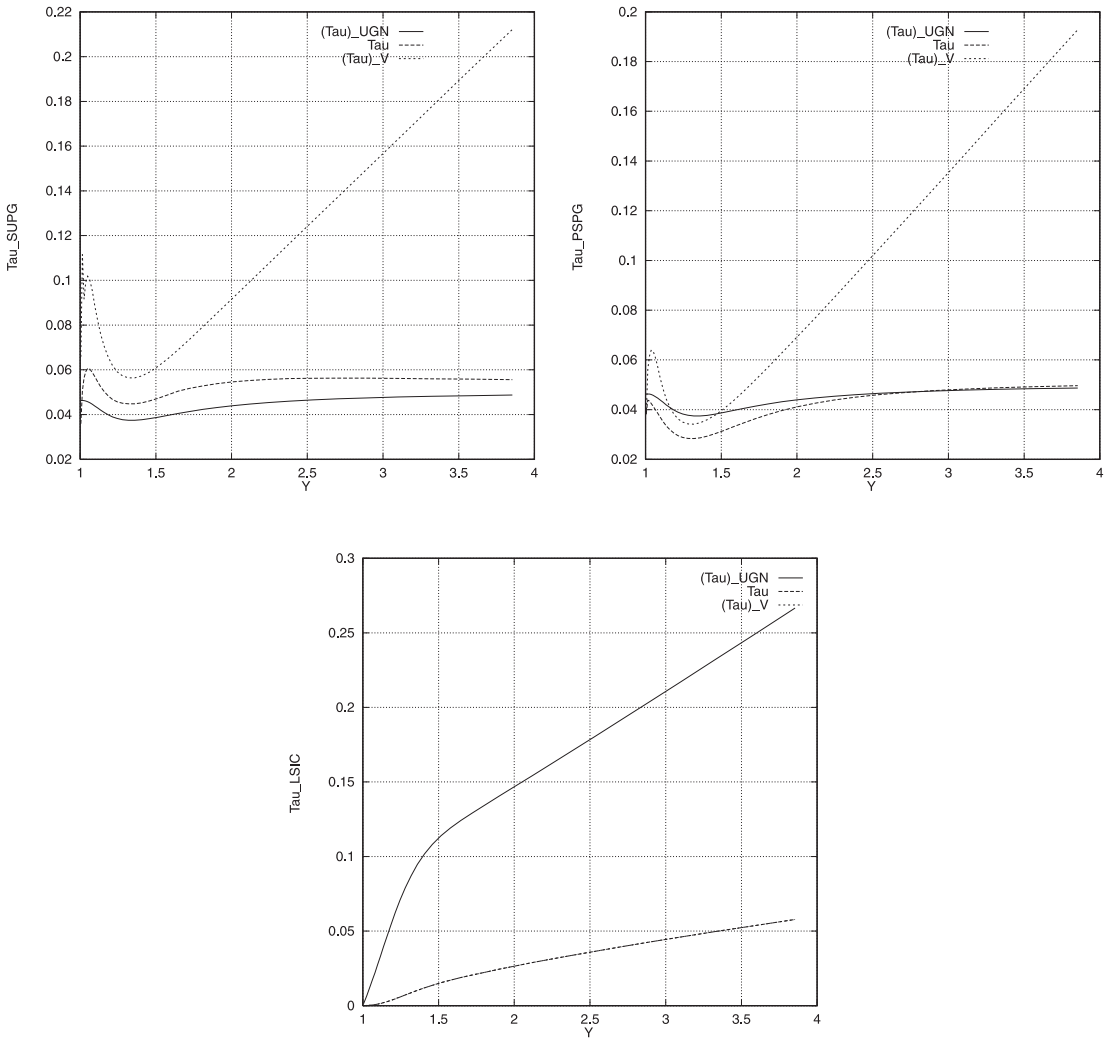

Fig. 13. Flow past a cylinder. Stabilization parameters: τ_{SUPG} (upper-left), τ_{PSPG} (upper-right), and τ_{LSIC} (lower).

5. Concluding remarks

In this paper, we introduced new methods for computing the stabilization parameters used in the stabilized finite element methods, particularly the SUPG and PSPG formulations. The parameters we proposed are computed from the element-level matrices and vectors, without separately computing local length scales. However, these parameters automatically take into account the local length scales, as well as the advection field and the element-level Reynolds number. We described these new methods of computing the stabilization parameters for a time-dependent advection–diffusion equation and the Navier–Stokes equations of unsteady incompressible flows. We carried out a number of numerical tests to demonstrate that these new methods produce stabilization parameters which are in some special cases comparable to those we designed earlier, and which are effective in flow problems governed by the advection–diffusion and Navier–Stokes equations.

Acknowledgements

The work reported in this paper was partially sponsored by NASA JSC (grant no. NAG9-1059), AFOSR (contract no. F49620-98-1-0214), NSF (grant no. CTS-9896278), and by the AHPCRC under the auspices of the Department of the Army, ARL cooperative agreement no. DAAH04-95-2-0003 and contract no. DAAH04-95-C-0008. The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. The second author has been supported by the Bridgestone Corp.

Appendix A. Inclusion of the flux jump terms in the stabilized formulation

We start with the Galerkin formulation of Eq. (1):

$$\int_{\Omega} w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h \right) \mathrm{d}\Omega + \int_{\Omega} \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega - \int_{\Gamma_h} w^h h \, \mathrm{d}\Gamma = 0. \tag{A.1}$$

Re-write the second term as follows:

$$\int_{\Omega} \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega. \tag{A.2}$$

Then integrate-by-parts:

$$\int_{\Omega^e} \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega = \int_{\Gamma^e} w^h \mathbf{n} \cdot \nu \nabla \phi^h \, \mathrm{d}\Gamma - \int_{\Omega^e} w^h \nabla \cdot \left(\nu \nabla \phi^h \right) \mathrm{d}\Omega. \tag{A.3}$$

Furthermore:

$$\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Gamma^e} w^h \mathbf{n} \cdot \nu \nabla \phi^h \, \mathrm{d}\Gamma = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Gamma^e_{\mathrm{if}}} w^h \mathbf{n} \cdot \nu \nabla \phi^h \, \mathrm{d}\Gamma + \int_{\Gamma_h} w^h \mathbf{n} \cdot \nu \nabla \phi^h \, \mathrm{d}\Gamma, \tag{A.4}$$

where Γ^e_{if} is the interior boundary (interior faces) of the element e. The first term on the right-hand side can also be written as a sum over interior faces:

$$\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Gamma^e_{\mathrm{if}}} w^h \mathbf{n} \cdot \nu \nabla \phi^h \, \mathrm{d}\Gamma = \sum_{k=1}^{n_{\mathrm{if}}} \int_{\Gamma_k} w^h \left(\mathbf{n}_1 \cdot \nu \nabla \phi_1^h + \mathbf{n}_2 \cdot \nu \nabla \phi_2^h \right) \mathrm{d}\Gamma, \tag{A.5}$$

where n_{if} is the number of interior faces, Γ_k is the kth interior face, and the subscripts 1 and 2 refer to elements sharing that face (see Fig. 14).

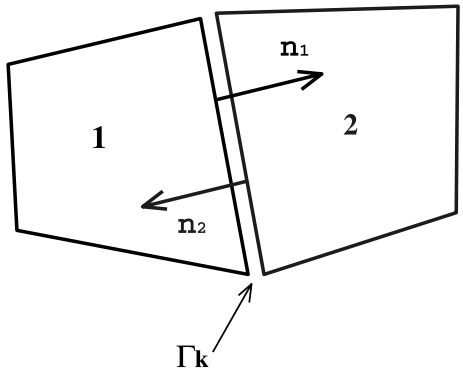

Fig. 14. The interior face.

Consequently, we can write:

$$\sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h - \nabla \cdot \left(\nu \nabla \phi^h \right) \right) \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Gamma_{\mathrm{if}}^e} w^h J \, \mathrm{d}\Gamma + \int_{\Gamma_h} w^h \left(\mathbf{n} \cdot \nu \nabla \phi^h - h \right) \mathrm{d}\Gamma, \tag{A.6}$$

where $J = \mathbf{n}_1 \cdot \nu \nabla \phi_1^h + \mathbf{n}_2 \cdot \nu \nabla \phi_2^h$ is the flux jump term.

Based on Eq. (A.6), we modify the stabilized formulation given by Eq. (5) as follows:

$$\begin{aligned} &\int_{\Omega} w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h \right) \mathrm{d}\Omega + \int_{\Omega} \nabla w^h \cdot \nu \nabla \phi^h \, \mathrm{d}\Omega \\ &\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tau_{\mathrm{SUPG}} \mathbf{u}^h \cdot \nabla w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h - \nabla \cdot \left(\nu \nabla \phi^h \right) \right) \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Gamma_{\mathrm{if}}^e} \tau_{\mathrm{SUPG}} \mathbf{u}^h \cdot \nabla w^h \, J \, \mathrm{d}\Gamma \\ &\quad + \int_{\Gamma_h} \tau_{\mathrm{SUPG}} \mathbf{u}^h \cdot \nabla w^h (\mathbf{n} \cdot \nu \nabla \phi^h - h) \, \mathrm{d}\Gamma = \int_{\Gamma_h} w^h h \, \mathrm{d}\Gamma. \end{aligned} \tag{A.7}$$

References

[1] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.

[2] T.E. Tezduyar, T.J.R. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations, in: Proceedings of the AIAA 21st Aerospace Sciences Meeting, AIAA Paper 83-0125, Reno, Nevada, 1983.

[3] T.J.R. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations, Comput. Methods Appl. Mech. Engrg. 73 (1989) 173–189 (1992)

[4] T.E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, Adv. Appl. Mech. 28 (1991) 1–44.

[5] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, S. Mittal, Parallel finite-element computation of 3D flows, IEEE Comput. 26 (1993) 27–36.

[6] T.J.R. Hughes, A.N. Brooks, A multi-dimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes (Ed.), Finite Element Methods for Convection Dominated Flows, AMD-vol. 34, ASME, New York, 1979, pp. 19–35.

[7] J. Donea, A Taylor–Galerkin method for convective transport problems, Int. J. Numer. Methods Engrg. 20 (1984) 101–120.

[8] C. Johnson, U. Navert, J. Pitkäranta, Finite element methods for linear hyperbolic problems, Comput. Methods Appl. Mech. Engrg. 45 (1984) 285–312.

[9] T.J.R. Hughes, L.P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multi-dimensional advective–diffusive systems, Comput. Methods Appl. Mech. Engrg. 63 (1987) 97–112.

[10] G.J. Le Beau, T.E. Tezduyar, Finite element computation of compressible flows with the SUPG formulation, in: M.N. Dhaubhadel, M.S. Engelman, J.N. Reddy (Eds.), Advances in Finite Element Analysis in Fluid Dynamics, FED-vol.123, ASME, New York, 1991, pp. 21–27.

[11] S. Aliabadi, S.E. Ray, T.E. Tezduyar, SUPG finite element computation of compressible flows with the entropy and conservation variables formulations, Comput. Mech. 11 (1993) 300–312.

[12] G.J. Le Beau, S.E. Ray, S.K. Aliabadi, T.E. Tezduyar, SUPG finite element computation of compressible flows with the entropy and conservation variables formulations, Comput. Methods Appl. Mech. Engrg. 104 (1993) 397–422.

[13] S. Mittal, T.E. Tezduyar, A unified finite element formulation for compressible and incompressible flows using augumented conservation variables, Comput. Methods Appl. Mech. Engrg. 161 (1998) 229–243.

[14] T.E. Tezduyar, Y.J. Park, Discontinuity capturing finite element formulations for nonlinear convection–diffusion-reaction problems, Comput. Methods Appl. Mech. Engrg. 59 (1986) 307–325.

[15] T.E. Tezduyar, D.K. Ganjoo, Petrov–Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: applications to transient convection–diffusuion problmes, Comput. Methods Appl. Mech. Engrg. 59 (1986) 49–71.

[16] L.P. Franca, S.L. Frey, T.J.R. Hughes, Stabilized finite element methods: I. Application to the advective–diffusive model, Comput. Methods Appl. Mech. Engrg. 95 (1992) 253–276.

[17] E. Süli, Private communication, 1999.

[18] A. Roshko, On the development of turbulent wakes from vortex street, NACA report 1191, NACA, 1954.