ADVANCES IN APPLIED MECHANICS, VOLUME 28

(1992) p. 1-44

Stabilized Finite Element Formulations for Incompressible Flow Computations†

T. E. TEZDUYAR

Department of Aerospace Engineering and Mechanics
and
Minnesota Supercomputer Institute
University of Minnesota
Minneapolis, Minnesota

I. Introduction

Finite element computation of incompressible flows involves two main sources of potential numerical instabilities associated with the Galerkin formulation of a problem. One source is due to the presence of advection terms in the governing equations, and can result in spurious node-to-node oscillations primarily in the velocity field. Such oscillations become more

† This research was sponsored by NASA-Johnson Space Center (under grant NAG 9-449), NSF (under grant MSM-8796352), U.S. Army (under contract DAAL03-89-C-0038), and the University of Paris VI.

apparent for advection-dominated (i.e., high Reynolds number) flows and flows with sharp layers in the solution. The other source of instability is due to using inappropriate combinations of interpolation functions to represent the velocity and pressure fields. These instabilities usually appear as oscillations primarily in the pressure field. In fact, there is not much about either of these numerical instabilities that could be considered to be inherent to the finite element formulation. Such instabilities appear also in the standard versions of other discretization techniques such as finite difference and finite volume methods.

This chapter consists of a review of the stabilized finite element formulations designed to prevent the potential numerical instabilities just described. The stabilization techniques that are reviewed more extensively than others are the Galerkin/least-squares (GLS), streamline-upwind/Petrov-Galerkin (SUPG), and pressure-stabilizing/Petrov-Galerkin (PSPG) formulations. All these formulations are consistent in the sense that, for reasons to be explained soon, an exact solution still satisfies the stabilized formulation. The descriptions of the stabilized formulations emphasized in this chapter, and the numerical examples presented, have all been extracted from recent papers by Tezduyar *et al.* (1990c, d, e) and Liou and Tezduyar (1990).

The SUPG stabilization for incompressible flows is achieved by adding to the Galerkin formulation a series of terms, each in the form of an integral over a different element. These integrals involve the product of the residual of the momentum equation and the advective operator acting on the test function. This formulation was introduced by Hughes and Brooks (1979). A comprehensive description of the formulation, together with various numerical examples, can be found in Brooks and Hughes (1982). The implementation of the SUPG formulation in Brooks and Hughes (1982) was based on Q1P0 (bilinear velocity/constant pressure) elements and one-step time-integration of the semi-discrete equations obtained by using such elements. The SUPG stabilization for the vorticity–stream function formulation of incompressible flow problems, including those with multiply-connected domains, was introduced by Tezduyar *et al.* (1988).

It is relevant to mention that the SUPG stabilization has been successfully applied to not only incompressible flows, but also compressible flows. In fact, there has been always some exchange of technology between these two application areas. The SUPG stabilization for hyperbolic systems in general and compressible Euler equations in particular was first introduced in a NASA report by Tezduyar and Hughes (1982). This report includes a

detailed stability and accuracy analysis and several one- and two-dimensional examples. The method was also presented in an AIAA paper by Tezduyar and Hughes (1983). The journal version of the NASA report was published with some additional numerical examples (Hughes and Tezduyar, 1984). The stabilization techniques introduced in Tezduyar and Hughes (1982) constituted a pilot work for compressible flows. For example, the Taylor–Galerkin stabilization method, which appeared in an article by Donea (1984) is very similar (under certain conditions identical) to one of the stabilization methods introduced in Tezduyar and Hughes (1982). Another example is the SUPG stabilization for compressible flows in the entropy variables formulation (Hughes *et al.*, 1987). Among others with interest in SUPG stabilization, Johnson and his group (see, e.g., Johnson and Saranen, 1986) has been perhaps one of the most involved ones.

Because in the SUPG stabilization the stabilizing terms added involve the residual of the momentum equation as a factor, when an exact solution is substituted into the stabilized formulation, these added terms vanish, and as a result the stabilized formulation is satisfied by the exact solution in the same way as the Galerkin formulation is satisfied. It is because of this property of the SUPG stabilization (and the other stabilization approaches emphasized in this chapter) that numerical oscillations are prevented without introducing excessive numerical diffusion (i.e., without "over-stabilizing"), and therefore without compromising the accuracy of the solution.

Two other stabilization techniques that became quite known in the past several years should be mentioned here. One is the selective mass lumping method of Kawahara *et al.* (1982). This method has been successfully used particularly in solving flow problems governed by the shallow water equations. It can be shown that there is a close relationship between this method and adding isotropic numerical diffusion to the governing equations. In fact, although currently the selective lumping parameter used in this method is determined empirically, some theoretical guidelines in determining this parameter can be provided based on this relationship. The other stabilization technique is the balanced tensor diffusivity (BTD) method of Gresho *et al.* (1984). In this method, a streamline diffusion term is added to the differential equations to compensate for the time truncation error corresponding to the forward Euler time-integration. It was shown by Gresho (1990) and Gresho and Chan (1990) that the BTD method exhibits the symptoms of excessive diffusion for certain test problems, particularly

for the problem involving an inviscid vortex for which the initial condition is also an exact solution that a reliable algorithm is expected to maintain as accurately as possible. Gresho predicted that the SUPG stabilization would exhibit similar symptoms. It was essentially this prediction that motivated the work leading to the article by Tezduyar *et al.* (1990a).

Based on numerical experiments with the inviscid vortex and the unsteady flow past a cylinder at Reynolds number 100, it was shown in Tezduyar *et al.* (1990a) that:

(a) the SUPG stabilization for the vorticity–stream function formulation exhibits no symptoms of excessive diffusion;
(b) the SUPG stabilization, as implemented in Brooks and Hughes (1982) with the Q1P0 element and the one-step time-integration, does exhibit symptoms of excessive diffusion;
(c) this situation can be improved significantly if the one-step time-integration scheme is replaced by the multi-step T6 scheme proposed in Tezduyar *et al.* (1990a), and in which the SUPG stabilization is applied only to the advection step; this scheme shows virtually no symptoms of excessive diffusion.

It is the belief of this author that the symptoms of excessive diffusion is not due to the SUPG stabilization in general, but the combination of the SUPG stabilization, the Q1P0 element, and the one-step time-integration. The pressure function space is too poor for the discrete formulation to benefit from the consistency property of the SUPG stabilization. The situation improves significantly in the T6 formulation because the SUPG stabilization is applied only to the advection step, and that step does not involve any pressure terms. In fact, there is more evidence to support this belief. For higher-order elements such as Q2P1 (biquadratic velocity/linear pressure) and pQ2P1 (pseudo-quadratic version of Q2P1), for which the pressure function space is richer, it was shown by Tezduyar *et al.* (1990b) that, for the same set of test problems used in Tezduyar *et al.* (1990a), the excess diffusion exhibited by the one-step and T6 schemes are quite comparable and very small. Furthermore, it was shown in Tezduyar *et al.* (1990c) that for the stabilized Q1Q1 (bilinear velocity and pressure) element, which has just a richer pressure function space than the Q1P0 element, the excess diffusion exhibited by the one-step and T6 schemes are, again, quite comparable and very small.

It is quite well-known that, without any kind of stabilization, for reliable computations, appropriate combinations of interpolation functions must be

used to represent the velocity and pressure. Elements that do not satisfy the Brezzi condition (Brezzi, 1973), yet look attractive for some reason, should be handled with care. For example, the Q1P0 element is one that does not satisfy this condition, yet it has always been a very popular element. Nevertheless it is an element that can potentially yield unstable computations. The Q2P1 and pQ2P1 elements, on the other hand, are known to be among the quadrilateral elements satisfying the Brezzi condition, and have been successfully implemented with the SUPG stabilization (Tezduyar *et al.*, 1990b) to be used for high Reynolds number flows. Recently, Pironneau and Rappaz (1989) and Bristeau *et al.* (1990) showed that inappropriate combinations of interpolation functions can lead to numerical oscillations also in some compressible flow problems. Furthermore, they showed that combinations similar to those known to be stable for incompressible flows can be successfully used for compressible flows.

It was shown that (see Brezzi and Pitkaranta, 1984, and Hughes *et al.*, 1986), with proper stabilization, elements that do not satisfy the Brezzi condition can be used for Stokes flow problems. The Petrov–Galerkin stabilization proposed in Hughes *et al.* (1986) is achieved, just like in the SUPG stabilization, by adding to the Galerkin formulation a series of integrals over element domains. Again, these terms involve the residual of the momentum equation as a factor, and therefore the stabilized formulation is consistent. Several researchers have been actively involved with stabilization techniques for Stokes flows, and many articles on this subject appeared in the recent literature (or about to appear soon); to give a few examples: Hughes and Franca (1987); Franca and Hughes (1988); Franca and Dutra do Carmo (1989); Douglas and Wang (1989); Franca and Stenberg (1990); and Silvester and Kechkar (1990).

The PSPG stabilization term proposed in Tezduyar *et al.* (1990c) is a generalization, to finite Reynolds number flows, of the Petrov–Galerkin stabilization term proposed in Hughes *et al.* (1986) for Stokes flows. The coefficients in the PSPG stabilization terms vary with the Reynolds number (based on a global scaling velocity) very much as the coefficients in the SUPG stabilization terms do. In the zero Reynolds number limit, the PSPG stabilization term reduces to the one proposed in Hughes *et al.* (1986). In Tezduyar *et al.* (1990c), the SUPG and PSPG stabilizations are used together with both one-step (T1) and multi-step (T6) time-integration schemes. With the T1 scheme, the SUPG and PSPG stabilizations are applied simultaneously. As will be explained soon, another way to arrive at this combined SUPG/PSPG stabilization is by considering the GLS

stabilization for the steady-state equations of incompressible flows. With the T6 scheme, on the other hand, the SUPG stabilization is applied only to the steps involving the pressure terms. Both schemes were implemented in Tezduyar *et al.* (1990c) based on the Q1Q1 and P1P1 (linear velocity and pressure) elements, and were successfully applied to a set of nearly standard test problems. Also, recently, Lundgren and Mansour (1990) applied this type of stabilization techniques to Lagrangian finite element computation of viscous free-surface flows.

The GLS stabilization is a more general stabilization approach that includes the essence of the SUPG and PSPG type stabilizations. This approach has been successfully applied to Stokes flows (Hughes and Franca, 1987), compressible flows (Hughes *et al.* 1989, and Shakib, 1988), and incompressible flows at finite Reynolds numbers (Hansbo and Szepessy, 1990, Tezduyar *et al.*, 1990d, e, and Liou and Tezduyar, 1990). In the GLS stabilization of incompressible flows, the stabilizing terms added are obtained by minimizing the sum of the squared residual of the momentum equation integrated over each element domain. Consequently, just like in the SUPG and PSPG stabilizations, because the stabilizing terms involve the residual of the momentum equation as a factor, the stabilized formulation is consistent.

For time-dependent problems, a strict implementation of the GLS stabilization technique necessitates finite element discretization in both space and time, and therefore leads to a space–time finite element formulation of the problem. The space–time finite element formulation has recently been successfully used, in conjunction with the GLS stabilization, for various problems with fixed spatial domains. This author is most familiar with references Hughes *et al.* (1987), Hughes and Hulbert (1988), Shakib (1988), and Hansbo and Szepessy (1989). The basics of the space–time formulation, its implementation, and the associated stability and accuracy analysis can be found in these references. It is important to realize that the finite element interpolation functions are discontinuous in time, so that the fully discrete equations are solved one space–time slab at a time, and this makes the computations feasible. Still, the computational cost associated with the space–time finite element formulations using piecewise linear functions in time is quite heavy. For large-scale problems, it becomes imperative to employ efficient iteration methods to reduce the cost involved. This was achieved in Liou and Tezduyar (1990) by using the generalized minimal residual (GMRES) iteration algorithm (Saad and Schultz, 1983) with the clustered element-by-element (CEBE) preconditioners (Liou and Tezduyar, 1990).

With a slightly more liberal implementation of the GLS stabilization, computation of time-dependent incompressible flow problems can be achieved by using the finite element discretization in space only, rather than in both space and time. To do this, we first consider the GLS stabilization for the steady-state equations of incompressible flows. Then, in the definition of the stabilizing terms, we replace the residual of the steady-state equations with the time-dependent one. These stabilizing terms are added to the Galerkin formulation of the time-dependent equations. The stabilized formulation obtained this way is, of course, still consistent. Furthermore, this stabilization is very close to the combined SUPG/PSPG stabilization mentioned previously.

Perhaps one of the most striking applications of the stabilized space–time finite element formulation is, as it was first pointed out by Tezduyar *et al.* (1990d, e), in computing moving boundaries and interfaces. The DSD/ST (Deforming-Spatial-Domain/Space–Time) procedure introduced by Tezduyar *et al.* (1990d, e) serves this purpose, and was successfully applied to several unsteady incompressible flow problems involving moving boundaries and interfaces, such as free-surface flows, liquid drops, two-liquid flows, and flows with drifting cylinders. In the DSD/ST procedure, the finite element formulation of a problem is written over its space–time domain, and therefore the deformation of the spatial domain with respect to time is taken into account automatically. Furthermore, in the DSD/ST procedure the frequency of remeshing is minimized. Here, we define remeshing as the process of generating a new mesh, and projecting the solution from the old mesh to the new one. Since remeshing, in general, involves projection errors, minimizing the frequency of remeshing results in minimizing the projection errors.

The outline of the rest of this chapter is as follows. In Section II, the governing equations of the unsteady incompressible flows are reviewed. The review of the space–time and GLS formulations is presented in Section III. The SUPG and PSPG stabilizations are reviewed in Section IV. In Section V, as an application to moving boundaries and interfaces, the DSD/ST procedure is reviewed. Sections III, IV, and V include numerical examples for the methods reviewed in those sections. Concluding remarks are given in Section VI.

II. The Governing Equations

Let $\Omega_t \subset R^{n_{sd}}$ be the spatial domain at time $t \in (0, T)$, where n_{sd} is the number of space dimensions. Let Γ_t denote the boundary of Ω_t. We

consider the following velocity–pressure formulation of the Navier–Stokes equations governing unsteady incompressible flows:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \qquad \text{on } \Omega_t \ \forall t \in (0, T), \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{on } \Omega_t \ \forall t \in (0, T), \tag{2.2}$$

where ρ and $\mathbf{u}$ are the density and velocity, and $\boldsymbol{\sigma}$ is the stress tensor given as

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}), \tag{2.3}$$

with

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \tfrac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}] \tag{2.4}$$

Here, p and μ are the pressure and the dynamic viscosity, and $\mathbf{I}$ is the identity tensor. The part of the boundary at which the velocity is assumed to be specified is denoted by $(\Gamma_t)_g$:

$$\mathbf{u} = \boldsymbol{g} \qquad \text{on } (\Gamma_t)_g \ \forall t \in (0, T). \tag{2.5}$$

The "natural" boundary conditions associated with (2.1) are the conditions on the stress components, and these are the conditions assumed to be imposed at the remaining part of the boundary:

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \boldsymbol{h} \qquad \text{on } (\Gamma_t)_h \ \forall t \in (0, T). \tag{2.6}$$

The homogeneous version of (2.6), which corresponds to the "traction-free" (i.e., zero normal and shear stress) conditions, is often imposed at the out flow boundaries. As initial condition, a divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified over the domain Ω_t at $t = 0$:

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{on } \Omega_0. \tag{2.7}$$

Let us now consider two immiscible fluids, A and B, occupying the domain Ω_t. Let $(\Omega_t)_{\mathrm{A}}$ denote the subdomain occupied by fluid A, and $(\Gamma_t)_{\mathrm{A}}$ denote the boundary of this subdomain. Similarly, let $(\Omega_t)_{\mathrm{B}}$ and $(\Gamma_t)_{\mathrm{B}}$ be the subdomain and boundary associated with fluid B. Furthermore, let $(\Gamma_t)_{\mathrm{AB}}$ be the intersection of $(\Gamma_t)_{\mathrm{A}}$ and $(\Gamma_t)_{\mathrm{B}}$, i.e., the interface between fluids A and B.

The kinematical conditions at the interface $(\Gamma_t)_{\mathrm{AB}}$ are based on the continuity of the velocity field. The dynamical conditions at the interface, for two-dimensional problems, can be expressed by the following equation:

$$\mathbf{n}_{\mathrm{A}} \cdot \boldsymbol{\sigma}_{\mathrm{A}} + \mathbf{n}_{\mathrm{B}} \cdot \boldsymbol{\sigma}_{\mathrm{B}} = \mathbf{n}_{\mathrm{A}}\, \gamma / R_{\mathrm{A}} \qquad \text{on } (\Gamma_t)_{\mathrm{AB}} \ \forall t \in (0, T), \tag{2.8}$$

where $\mathbf{n}_A$ and $\mathbf{n}_B$ are the unit outward normal vectors at the interface, $\boldsymbol{\sigma}_A$ and $\boldsymbol{\sigma}_B$ are the stress tensors, γ is the surface tension coefficient, and R_A is the radius of curvature defined to be positive when $\mathbf{n}_A$ points towards the center of curvature. The condition (2.8) is formally applicable also for free-surface flows (i.e., when the second fluid does not exist), provided that subdomain $(\Omega_t)_A$ is the one assigned to be occupied by the fluid.

III. The Space-Time Formulation and the Galerkin/ Least-Squares Stabilization

A. The Method

Let us first assume that the spatial domain is fixed in time. Under this assumption, the subscript t is dropped from the symbols Ω_t and Γ_t. In the space-time finite element formulation, the time interval $(0, T)$ is partitioned into subintervals $I_n = (t_n, t_{n+1})$, where t_n and t_{n+1} belong to an ordered series of time levels $0 = t_0 < t_1 < \cdots < t_N = T$. The space-time slab Q_n is defined as the space-time domain $\Omega \times I_n$. The lateral surface of Q_n is denoted by P_n; this is the surface described by the boundary Γ, as t traverses I_n. Similar to the way it was represented by Eqs. (2.5) and (2.6), P_n is decomposed into $(P_n)_g$ and $(P_n)_h$ with respect to the type of boundary condition being imposed.

Finite element discretization of a space-time slab Q_n is achieved by dividing it into elements Q_n^e, $e = 1, 2, \ldots, (n_{el})_n$, where $(n_{el})_n$ is the number of elements in the space-time slab Q_n. Associated with this discretization, for each space-time slab we define the following finite element interpolation function spaces for the velocity and pressure:

$$(S_{\mathbf{u}}^h)_n = \{\mathbf{u}^h \mid \mathbf{u}^h \in [H^{1h}(Q_n)]^{n_{sd}},\ \mathbf{u}^h \doteq \boldsymbol{g}^h \text{ on } (P_n)_g\}, \tag{3.1}$$

$$(V_{\mathbf{u}}^h)_n = \{\mathbf{w}^h \mid \mathbf{w}^h \in [H^{1h}(Q_n)]^{n_{sd}},\ \mathbf{w}^h \doteq \mathbf{0} \text{ on } (P_n)_g\}, \tag{3.2}$$

$$(S_p^h)_n = (V_p^h)_n = [q^h \mid q^h \in H^{1h}(Q_n)]. \tag{3.3}$$

Here $H^{1h}(Q_n)$ represents the finite-dimensional function space over the space-time slab Q_n. This space is formed by using, over the parent (element) domains, first-order polynomials in space and time. It is also possible to use zeroth-order polynomials in time. In either case, globally, the interpolation functions are continuous in space but discontinuous in time.

The space-time formulation of (2.1)–(2.7) can be written as follows: Start with

$$(\mathbf{u}^h)_0^- = (\mathbf{u}_0)^h; \tag{3.4}$$

sequentially for $Q_1, Q_2, \ldots, Q_{N-1}$, given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in (S_{\mathbf{u}}^h)_n$ and $p^h \in (S_p^h)_n$, such that $\forall \mathbf{w}^h \in (V_{\mathbf{u}}^h)_n$ and $\forall q^h \in (V_p^h)_n$,

$$\int_{Q_n} \mathbf{w}^h \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h\right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\, dQ - \int_{(P_n)_h} \mathbf{w}^h \cdot \boldsymbol{h}\, dP$$

$$+ \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h\, dQ + \int_{\Omega} (\mathbf{w}^h)_n^+ \cdot ((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-)\, d\Omega$$

$$+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau \left[\rho\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right) - \nabla \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h)\right]$$

$$\cdot \left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h\right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\right] dQ = 0. \tag{3.5}$$

In the variational formulation given by (3.5), the following notation is being used:

$$(\mathbf{u}^h)_n^{\pm} = \lim_{\delta \to 0} \mathbf{u}^h(t_n \pm \delta), \tag{3.6}$$

$$\int_{Q_n} (\cdots)\, dQ = \int_{I_n} \int_{\Omega} (\cdots)\, d\Omega\, dt, \tag{3.7}$$

$$\int_{P_n} (\cdots)\, dP = \int_{I_n} \int_{\Gamma} (\cdots)\, d\Gamma\, dt. \tag{3.8}$$

Remarks

1. If we were in a standard finite element formulation, rather than a space–time one, the Galerkin formulation of (2.1)–(2.7) would have consisted of the first four integrals (their spatial versions of course) appearing in Eq. (3.5). In the space-time formulation, because the interpolation functions are discontinuous in time, the fifth integral in Eq. (3.5) enforces, weakly, the continuity of the velocity in time. The remaining series of integrals in Eq. (3.5) are the least-squares terms added to the Galerkin variational formulation to assure the numerical stability of the computations. The coefficient τ determines the weight of such added terms.

2. This kind of stabilization of the Galerkin formulation is referred to as the Galerkin/least-squares (GLS) procedure, and can be considered as a generalization of the stabilization based on the streamline-upwind/Petrov-Galerkin (SUPG) procedure employed for incompressible flows. It is with such stabilization procedures that it is possible to use elements that have equal-order interpolation functions for velocity and pressure, and that are otherwise unstable.

3. It is important to realize that the stabilizing terms added involve the momentum equation as a factor. Therefore, despite these additional terms, an exact solution is still admissible to the variational formulation given by Eq. (3.5)

The coefficient τ used in this formulation is obtained by a simple multi-dimensional generalization of the optimal τ given in Shakib (1988) for one-dimensional space-time formulation. The expression for the τ used in this formulation is

$$\tau = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\|\mathbf{u}^h\|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-1/2}, \tag{3.9}$$

where ν is the kinematic viscosity, and Δt and h are the temporal and spatial "element lengths." For steady-state computations, a different definition for τ is used:

$$\tau = \left[\left(\frac{2\|\mathbf{u}^h\|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right]^{-1/2}. \tag{3.10}$$

For derivation of τ for higher-order elements, see Franca *et al.* (1990).

Remark

4. Because the finite element interpolation functions are discontinuous in time, the fully discrete equations can be solved one space-time slab at a time. Still, the memory needed for the global matrices involved in this method is quite substantial. For example, in two dimensions, the memory needed for space-time formulation (with interpolation functions that are piecewise linear in time) of a problem is approximately four times greater compared with using the finite element method only for spatial discretization. However, iteration methods can be employed to substantially reduce the cost involved in solving the linear equation systems arising from the space-time finite element discretization. It was shown in Liou and Tezduyar (1990) that

the clustered element-by-element (CEBE) preconditioners (Liou and Tezduyar, 1990), together with the generalized minimal residual (GMRES) method (Saad and Schultz, 1983) can be effectively used for this purpose.

B. Numerical Examples

In this section, numerical examples are presented from the space–time finite element computations based on the CEBE/GMRES iteration method. The interpolation functions used for velocity and pressure are piecewise bilinear in space and piecewise linear in time. These computations involve no global coefficient matrices, and therefore need substantially less computer memory and time compared to non-iterative solution of the fully discrete equations (see Remark 4). By using very large time step sizes (e.g., 100,000) the steady-state solutions are obtained in a few time steps. For the description of the iteration method, its performance characteristics, and the details of the numerical examples, see Liou and Tezduyar (1990).

The lid-driven cavity flow at Reynolds number 1000

In this problem, the cavity has a square shape, and the Reynolds number is based on the size of the cavity and the velocity of the lid. A uniform mesh with 64×64 elements and 4225 nodes is employed. Every time step, approximately 25,000 equations are solved simultaneously. Figure 1 shows, for the steady-state solution, velocity components along the vertical and horizontal centerlines, pressure, vorticity, and stream function.

"Steady-state" solution for flow past a cylinder at Reynolds number 100

In this test problem, the dimensions of the computational domain, normalized by the cylinder diameter, are 30.5 and 16.0 in the flow and cross-flow directions, respectively. The free-stream velocity is 0.125. Reynolds number is based on the free-stream velocity and the diameter of the cylinder. Symmetry conditions are imposed at the upper and lower computational boundaries, and the traction-free condition is imposed at the outflow boundary. A mesh with 5400 elements and 5510 nodes is employed. Every time step approximately 33,000 equations are solved simultaneously. Figure 2 shows, for the "steady-state" solution, pressure, vorticity, stream function and stationary stream function.

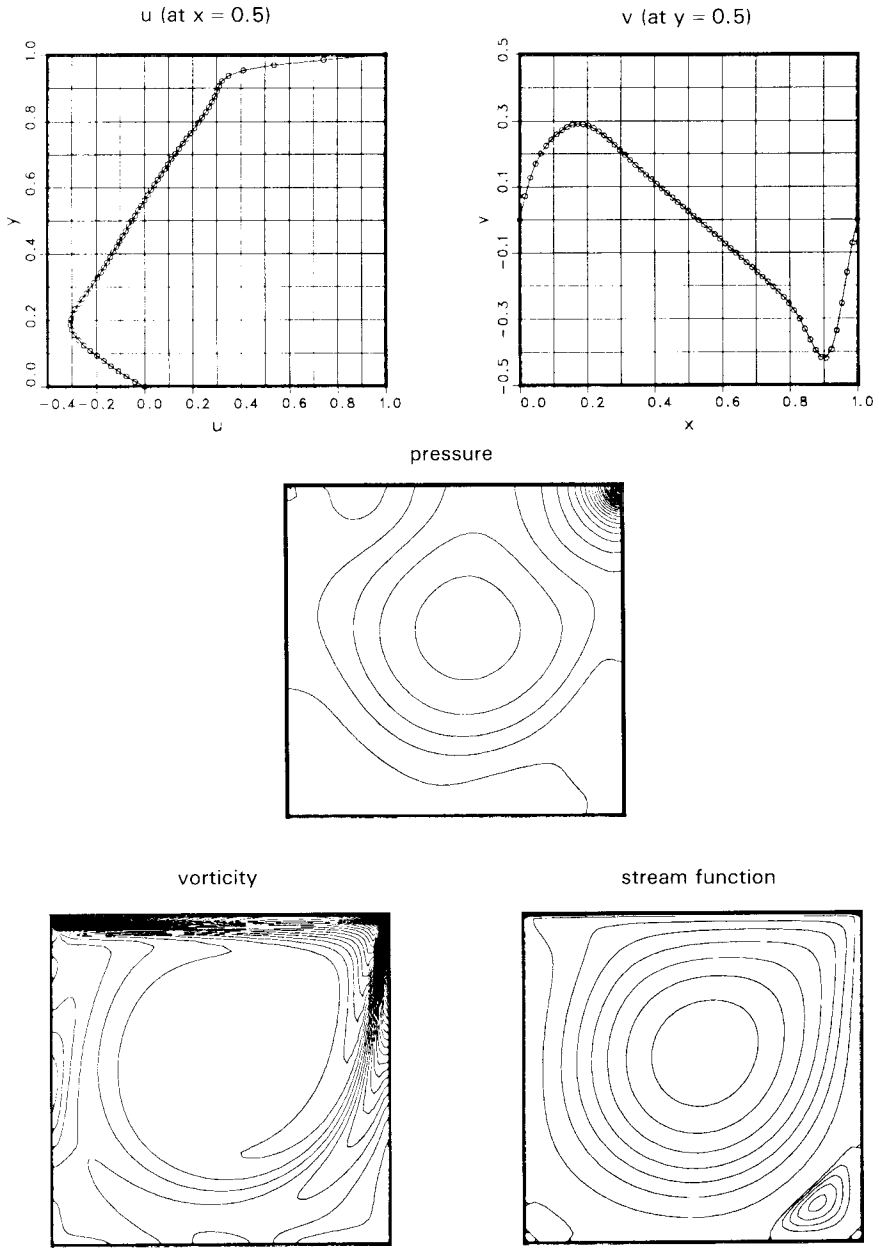

FIG. 1. Steady-state solution for the lid-driven cavity flow at Reynolds number 1000: velocity components along the vertical and horizontal center lines, pressure, vorticity, and stream function (Liou and Tezduyar, 1990).

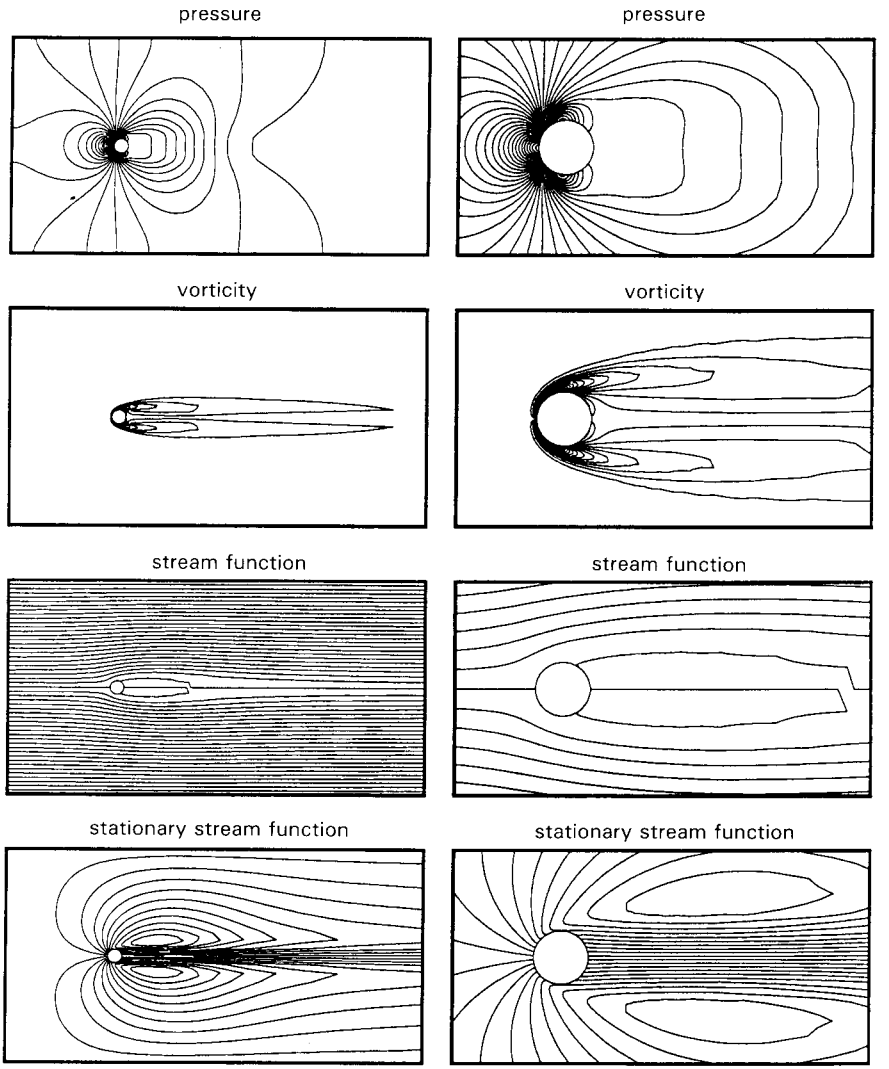

FIG. 2. "Steady-state" solution for flow past a cylinder at Reynolds number 100: pressure, vorticity, stream function, and stationary stream function (Liou and Tezduyar, 1990).

IV. The Formulations with the SUPG and PSPG Stabilizations

A. The Method

In this section, the variational formulations with the SUPG and PSPG stabilization terms are described. These formulations are based on finite element discretization in space only, rather than in both space and time.

Let us discretize the domain Ω by subdividing it into elements Ω^e, $e = 1, 2, \ldots, n_{\text{el}}$, where n_{el} is the number of elements. Associated with this discretization, we define the following finite element interpolation function spaces for the velocity and pressure:

$$S_{\mathbf{u}}^{\text{h}} = \{\mathbf{u}^{\text{h}} \mid \mathbf{u}^{\text{h}} \in [H^{1\text{h}}(\Omega)]^{n_{\text{sd}}},\ \mathbf{u}^{\text{h}} \doteq \boldsymbol{g}^{\text{h}} \text{ on } \Gamma_g\}, \tag{4.1}$$

$$V_{\mathbf{u}}^{\text{h}} = \{\mathbf{w}^{\text{h}} \mid \mathbf{w}^{\text{h}} \in [H^{1\text{h}}(\Omega)]^{n_{\text{sd}}},\ \mathbf{w}^{\text{h}} \doteq \mathbf{0} \text{ on } \Gamma_g\}, \tag{4.2}$$

$$S_p^{\text{h}} = V_p^{\text{h}} = \{q^{\text{h}} \mid q^{\text{h}} \in H^{1\text{h}}(\Omega)\}, \tag{4.3}$$

where $H^{1\text{h}}(\Omega)$ represents the finite-dimensional function space over the spatial domain Ω. This space is formed by using, over the element domains, first-order polynomials in space. The stabilized Galerkin formulation of (2.1)–(2.7) can be written as follows: Find $\mathbf{u}^{\text{h}} \in S_{\mathbf{u}}^{\text{h}}$ and $p^{\text{h}} \in S_p^{\text{h}}$ such that, $\forall \mathbf{w}^{\text{h}} \in V_{\mathbf{u}}^{\text{h}}$ and $\forall q^{\text{h}} \in V_p^{\text{h}}$,

$$\begin{aligned}
&\int_\Omega \mathbf{w}^{\text{h}} \cdot \rho\left(\frac{\partial \mathbf{u}^{\text{h}}}{\partial t} + \mathbf{u}^{\text{h}} \cdot \nabla \mathbf{u}^{\text{h}}\right) d\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^{\text{h}}) : \boldsymbol{\sigma}(p^{\text{h}}, \mathbf{u}^{\text{h}})\, d\Omega \\
&\quad - \int_{\Gamma_h} \mathbf{w}^{\text{h}} \cdot \boldsymbol{h}\, d\Gamma + \int_\Omega q^{\text{h}}\, \nabla \cdot \mathbf{u}^{\text{h}}\, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} (\boldsymbol{\delta}^{\text{h}} + \boldsymbol{\epsilon}^{\text{h}}) \\
&\quad \cdot \left[\rho\left(\frac{\partial \mathbf{u}^{\text{h}}}{\partial t} + \mathbf{u}^{\text{h}} \cdot \nabla \mathbf{u}^{\text{h}}\right) - \nabla \cdot \boldsymbol{\sigma}(p^{\text{h}}, \mathbf{u}^{\text{h}})\right] d\Omega = 0.
\end{aligned} \tag{4.4}$$

As it can be seen from Eq. (4.4), two stabilizing terms have been added to the standard Galerkin formulation of (2.1)–(2.7); the one with $\boldsymbol{\delta}^{\text{h}}$ is the SUPG term, and the one with $\boldsymbol{\epsilon}^{\text{h}}$ is the PSPG (pressure-stabilizing/Petrov–Galerkin) term. The Petrov–Galerkin functions $\boldsymbol{\delta}^{\text{h}}$ and $\boldsymbol{\epsilon}^{\text{h}}$ are defined as

$$\boldsymbol{\delta}^{\text{h}} = \tau_{\text{SUPG}}\, \mathbf{u}^{\text{h}} \cdot \nabla \mathbf{w}^{\text{h}}. \tag{4.5}$$

$$\boldsymbol{\epsilon}^{\text{h}} = \tau_{\text{PSPG}} \frac{1}{\rho} \nabla q^{\text{h}}, \tag{4.6}$$

where

$$\tau_{\mathrm{SUPG}} = \frac{\mathrm{h}}{2\|\mathbf{u}^{\mathrm{h}}\|} z(\mathrm{Re}_{\mathbf{u}}), \tag{4.7}$$

$$\tau_{\mathrm{PSPG}} = \frac{\mathrm{h}^{\#}}{2\|\mathbf{U}\|} z(\mathrm{Re}_{\mathbf{U}}^{\#}). \tag{4.8}$$

Here, $\mathrm{Re}_{\mathbf{u}}$ and $\mathrm{Re}_{\mathbf{U}}^{\#}$ are the element Reynolds numbers, which are based, respectively, on the local velocity $\mathbf{u}^{\mathrm{h}}$ and a global scaling velocity $\mathbf{U}$. That is,

$$\mathrm{Re}_{\mathbf{u}} = \frac{\|\mathbf{u}^{\mathrm{h}}\|\,\mathrm{h}}{2\nu}, \tag{4.9}$$

$$\mathrm{Re}_{\mathbf{U}}^{\#} = \frac{\|\mathbf{U}\|\,\mathrm{h}^{\#}}{2\nu}, \tag{4.10}$$

The "element length" h is computed by using the expression

$$h = 2\left(\sum_{a=1}^{n_{\mathrm{en}}} |\mathbf{s}\cdot\nabla N_a|\right)^{-1}, \tag{4.11}$$

where n_{en} is the number of nodes in the element, N_a is the basis function associated with node a, and $\mathbf{s}$ is the unit vector in the direction of the local velocity. The "element length" $h^{\#}$, on the other hand, is defined to be equal to the diameter of the circle which is area-equivalent to the element. The function $z(\mathrm{Re})$ used in Eqs. (4.7) and (4.8) is defined as

$$z(\mathrm{Re}) = \begin{cases} \mathrm{Re}/3, & 0 \le \mathrm{Re} \le 3, \\ 1, & 3 \le \mathrm{Re}. \end{cases} \tag{4.12}$$

The spatial discretization of Eq. (4.4) leads to the following set of nonlinear ordinary differential equations.

$$(\mathbf{M} + \mathbf{M}_{\delta})\,\mathbf{a} + \mathbf{N}(\mathbf{v}) + \mathbf{N}_{\delta}(\mathbf{v}) + (\mathbf{K} + \mathbf{K}_{\delta})\mathbf{v} - (\mathbf{G} + \mathbf{G}_{\delta})\mathbf{p} = \mathbf{F} + \mathbf{F}_{\delta}), \tag{4.13}$$

$$\mathbf{G}^{\mathrm{T}}\mathbf{v} + \mathbf{M}_{\epsilon}\mathbf{a} + \mathbf{N}_{\epsilon}(\mathbf{v}) + \mathbf{K}_{\epsilon}\mathbf{v} + \mathbf{G}_{\epsilon}\mathbf{p} = \mathbf{E} + \mathbf{E}_{\epsilon}, \tag{4.14}$$

where $\mathbf{v}$ is the vector of unknown nodal values of $\mathbf{u}^{\mathrm{h}}$, $\mathbf{a}$ is the time derivative of $\mathbf{v}$, and $\mathbf{p}$ is the vector of nodal values of p^{h}. The matrices $\mathbf{M}$, $\mathbf{N}$, $\mathbf{K}$ and $\mathbf{G}$ are derived, respectively, from the time-dependent, advective, viscous, and pressure terms. The vector $\mathbf{F}$ is due to the boundary conditions (2.5) and (2.6) (i.e., the $\boldsymbol{g}$ and $\boldsymbol{h}$ terms), whereas the vector $\mathbf{E}$ is due to the boundary condition (2.5). The subscripts δ and ϵ identify the SUPG and PSPG contributions, respectively.

First, consider the time-integration of Eqs. (4.13) and (4.14) by a one-step generalized trapezoidal rule; i.e., given $(\mathbf{u}^h)_n$, find $(\mathbf{u}^h)_{n+1}$ and $(p^h)_{n+1}$ (this will be referred to as T1 formulation). When written in an incremental form, the T1 formulation leads to

$$\mathbf{M}^* \Delta\mathbf{a} - \mathbf{G}^* \Delta\mathbf{p} = \mathbf{R}, \tag{4.15}$$

$$(\mathbf{G}^T)^* \Delta\mathbf{a} + \mathbf{G}_\epsilon \Delta\mathbf{p} = \mathbf{Q}, \tag{4.16}$$

where

$$\mathbf{R} = \mathbf{F} + \mathbf{F}_\delta - [(\mathbf{M} + \mathbf{M}_\delta)\mathbf{a} + \mathbf{N}(\mathbf{v}) + \mathbf{N}_\delta(\mathbf{v}) + (\mathbf{K} + \mathbf{K}_\delta)\mathbf{v} - (\mathbf{G} + \mathbf{G}_\delta)\mathbf{p}], \tag{4.17}$$

$$\mathbf{Q} = \mathbf{E} + \mathbf{E}_\epsilon - [\mathbf{G}^T\mathbf{v} + \mathbf{M}_\epsilon\mathbf{a} + \mathbf{N}_\epsilon(\mathbf{v}) + \mathbf{K}_\epsilon\mathbf{v} + \mathbf{G}_\epsilon\mathbf{p}], \tag{4.18}$$

$$\mathbf{M}^* = \mathbf{M} + \mathbf{M}_\delta + \alpha\,\Delta t\left(\frac{\partial \mathbf{N}}{\partial \mathbf{v}} + \frac{\partial \mathbf{N}_\delta}{\partial \mathbf{v}} + \mathbf{K} + \mathbf{K}_\delta\right), \tag{4.19}$$

$$\mathbf{G}^* = \mathbf{G} + \mathbf{G}_\delta, \tag{4.20}$$

$$(\mathbf{G}^T)^* = \mathbf{M}_\epsilon + \alpha\,\Delta t\left(\frac{\partial \mathbf{N}_\epsilon}{\partial \mathbf{v}} + \mathbf{K}_\epsilon + \mathbf{G}^T\right). \tag{4.21}$$

The parameter α controls the stability and accuracy of the time integration algorithm.

Remark

5. The systems (4.15) and (4.16) can be solved by treating the velocity explicitly in the momentum equation. Since the SUPG and PSPG supplements are applied to all terms in the momentum equation, in explicit computations the coefficient matrix of the pressure equation is generally not symmetric. All explicit T1 computations reported in this section are based on the symmetrization of the coefficient matrix of the pressure equation, and the results are obtained with two passes per time step. In such computations, $\mathbf{M}^*$, $\mathbf{G}^*$ and $(\mathbf{G}^T)^*$ are replaced with

$$\mathbf{M}^* = \mathbf{M}_L, \tag{4.22}$$

$$\mathbf{G}^* = \mathbf{G}, \tag{4.23}$$

$$(\mathbf{G}^T)^* = \alpha\,\Delta t\,\mathbf{G}^T, \tag{4.24}$$

where $\mathbf{M}_L$ is the lumped version of the mass matrix $\mathbf{M}$.

To write the T6 formulation (Tezduyar *et al.*, 1990a) of (2.1)–(2.7), we need to slightly modify the definition of the solution space for velocity:

$$(S_{\mathbf{u}}^{\mathrm{h}})_{n+\gamma} = \{\mathbf{u}^{\mathrm{h}} \mid \mathbf{u}^{\mathrm{h}} \in [H^{1\mathrm{h}}(\Omega)]^{n_{\mathrm{sd}}},\ \mathbf{u}^{\mathrm{h}} \doteq (\boldsymbol{g}^{\mathrm{h}})_{n+\gamma} \text{ on } \Gamma_g\}; \tag{4.25}$$

the definitions of the other function spaces remain as they were given by Eqs. (4.2) and (4.3). We now summarize the T6 formulation.

1. Find $(\mathbf{u}^{\mathrm{h}})_{n+\theta}^{\sim} \in (S_{\mathbf{u}}^{\mathrm{h}})_{n+\theta}$ such that, $\forall \mathbf{w}^{\mathrm{h}} \in V_{\mathbf{u}}^{\mathrm{h}}$,

$$\int_{\Omega} \mathbf{w}^{\mathrm{h}} \cdot \rho\left(\frac{(\mathbf{u}^{\mathrm{h}})_{n+\theta}^{\sim} - (\mathbf{u}^{\mathrm{h}})_n}{\theta\,\Delta t} + (\mathbf{u}^{\mathrm{h}})_n \cdot \nabla(\mathbf{u}^{\mathrm{h}})_n\right) d\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \delta^{\mathrm{h}} \cdot \rho\left(\frac{(\mathbf{u}^{\mathrm{h}})_{n+\theta}^{\sim} - (\mathbf{u}^{\mathrm{h}})_n}{\theta\,\Delta t} + (\mathbf{u}^{\mathrm{h}})_n \cdot \nabla(\mathbf{u}^{\mathrm{h}})_n\right) d\Omega = 0. \tag{4.26}$$

2. Find $(\mathbf{u}^{\mathrm{h}})_{n+\theta} \in (S_{\mathbf{u}}^{\mathrm{h}})_{n+\theta}$ and $(p^{\mathrm{h}})_{n+\theta} \in S_p^{\mathrm{h}}$ such that, $\forall \mathbf{w}^{\mathrm{h}} \in V_{\mathbf{u}}^{\mathrm{h}}$ and $\forall q^{\mathrm{h}} \in V_p^{\mathrm{h}}$,

$$\int_{\Omega} \mathbf{w}^{\mathrm{h}} \cdot \frac{\rho[(\mathbf{u}^{\mathrm{h}})_{n+\theta} - (\mathbf{u}^{\mathrm{h}})_{n+\theta}^{\sim}]}{\theta\,\Delta t}\, d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}^{\mathrm{h}}) : (\boldsymbol{\sigma}^{\mathrm{h}})_{n+\theta}\, d\Omega - \int_{\Gamma_h} \mathbf{w}^{\mathrm{h}} \cdot (\boldsymbol{h}^{\mathrm{h}})_{n+\theta}\, d\Gamma + \int_{\Omega} q^{\mathrm{h}}\, \nabla \cdot (\mathbf{u}^{\mathrm{h}})_{n+\theta}\, d\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\epsilon}^{\mathrm{h}} \cdot \left\{\frac{\rho[(\mathbf{u}^{\mathrm{h}})_{n+\theta} - (\mathbf{u}^{\mathrm{h}})_{n+\theta}^{\sim}]}{\theta\,\Delta t} - \nabla \cdot (\boldsymbol{\sigma}^{\mathrm{h}})_{n+\theta}\right\} d\Omega = 0. \tag{4.27}$$

3. Find $(\mathbf{u}^{\mathrm{h}})_{n+1-\theta}^{\sim} \in (S_{\mathbf{u}}^{\mathrm{h}})_{n+1-\theta}$ such that, $\forall \mathbf{w}^{\mathrm{h}} \in V_{\mathbf{u}}^{\mathrm{h}}$,

$$\int_{\Omega} \mathbf{w}^{\mathrm{h}} \cdot \frac{\rho[(\mathbf{u}^{\mathrm{h}})_{n+1-\theta}^{\sim} - (\mathbf{u}^{\mathrm{h}})_{n+\theta}]}{(1-2\theta)\,\Delta t}\, d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}^{\mathrm{h}}) : (\boldsymbol{\sigma}^{\mathrm{h}})_{n+\theta}\, d\Omega - \int_{\Gamma_h} \mathbf{w}^{\mathrm{h}} \cdot (\boldsymbol{h}^{\mathrm{h}})_{n+\theta}\, d\Gamma = 0. \tag{4.28}$$

4. Find $(\mathbf{u}^{\mathrm{h}})_{n+1-\theta} \in (S_{\mathbf{u}}^{\mathrm{h}})_{n+1-\theta}$ such that, $\forall \mathbf{w}^{\mathrm{h}} \in V_{\mathbf{u}}^{\mathrm{h}}$,

$$\int_{\Omega} \mathbf{w}^{\mathrm{h}} \cdot \rho\left(\frac{(\mathbf{u}^{\mathrm{h}})_{n+1-\theta} - (\mathbf{u}^{\mathrm{h}})_{n+1-\theta}^{\sim}}{(1-2\theta)\,\Delta t} + (\mathbf{u}^{\mathrm{h}})_{n+1-\theta} \cdot \nabla(\mathbf{u}^{\mathrm{h}})_{n+1-\theta}\right) d\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\delta}^{\mathrm{h}} \cdot \rho\left(\frac{(\mathbf{u}^{\mathrm{h}})_{n+1-\theta} - (\mathbf{u}^{\mathrm{h}})_{n+1-\theta}^{\sim}}{(1-2\theta)\,\Delta t} + (\mathbf{u}^{\mathrm{h}})_{n+1-\theta} \cdot \nabla(\mathbf{u}^{\mathrm{h}})_{n+1-\theta}\right) d\Omega = 0. \tag{4.29}$$

5. Find $(\mathbf{u}^h)^{\sim}_{n+1} \in (S^h_{\mathbf{u}})_{n+1}$ such that, $\forall \mathbf{w}^h \in V^h_{\mathbf{u}}$,

$$\int_\Omega \mathbf{w}^h \cdot \rho\left(\frac{(\mathbf{u}^h)^{\sim}_{n+1} - (\mathbf{u}^h)_{n+1-\theta}}{\theta\,\Delta t} + (\mathbf{u}^h)_{n+1-\theta} \cdot \nabla(\mathbf{u}^h)_{n+1-\theta}\right) d\Omega$$
$$+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\delta}^h \cdot \rho\left(\frac{(\mathbf{u}^h)^{\sim}_{n+1} - (\mathbf{u}^h)_{n+1-\theta}}{\theta\,\Delta t} + (\mathbf{u}^h)_{n+1-\theta} \cdot \nabla(\mathbf{u}^h)_{n+1-\theta}\right) d\Omega$$
$$= 0. \tag{4.30}$$

6. Find $(\mathbf{u}^h)_{n+1} \in (S^h_{\mathbf{u}})_{n+1}$ and $(p^h)_{n+1} \in S^h_p$ such that, $\forall \mathbf{w}^h \in V^h_{\mathbf{u}}$ and $\forall q^h \in V^h_p$,

$$\int_\Omega \mathbf{w}^h \cdot \frac{\rho[(\mathbf{u}^h)_{n+1} - (\mathbf{u}^h)^{\sim}_{n+1}]}{\theta\,\Delta t}\, d\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : (\boldsymbol{\sigma}^h)_{n+1}\, d\Omega$$
$$- \int_{\Gamma_h} \mathbf{w}^h \cdot (\boldsymbol{h}^h)_{n+1}\, d\Gamma + \int_\Omega q^h\, \nabla \cdot (\mathbf{u}^h)_{n+1}\, d\Omega$$
$$+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\epsilon}^h \cdot \left\{\frac{\rho[(\mathbf{u}^h)_{n+1} - (\mathbf{u}^h)^{\sim}_{n+1}]}{\theta\,\Delta t} - \nabla \cdot (\boldsymbol{\sigma}^h)_{n+1}\right\} d\Omega = 0. \tag{4.31}$$

Remarks

6. The parameter θ is the one used in the θ-scheme (Bristeau *et al.*, 1987); for the numerical examples to be reported in this section, it is set to $\frac{1}{3}$.

7. The matrix forms corresponding to Eqs. (4.26), (4.28), (4.29), and (4.30) can be solved implicitly or explicitly as described in Tezduyar *et al.* (1990a). The matrix form of the two "Stokes substeps," i.e., Eqs. (4.27) and (4.31), are quite similar to the matrix form of the T1 formulation; they can be solved implicitly or by treating the velocity explicitly. The results reported in this section are based on the explicit treatment of all substeps. The numbers of passes used in the substeps are 4-2-2-2-4-2.

B. Numerical Examples

To have a better basis of comparison among the solutions obtained by using different elements, meshes generated with different elements are required to have the same distribution of the velocity and pressure nodes. The nodal values of the stream function and vorticity are obtained by the least-squares interpolation. For the meshes generated with the P1P1

elements, these quantities are computed from the velocity field by using the meshes generated with the Q1Q1 element. For details of the computations and the performance characteristics, see Tezduyar *et al.* (1990c).

Unsteady flow past a cylinder at Reynolds number 100

The problem set-up in this case is the same as it was for the "steady-state" case of Section III. However, this time we are interested in the unsteady behavior. The mesh used for Q1Q1 consists of 5240 elements, while the number of elements for P1P1 is 10,480. Both meshes contain 5350 velocity nodes. The periodic solution is computed by introducing a short-term perturbation to the symmetric solution. We have observed, at least for small perturbations, that the periodic solution is independent of the mode of perturbation.

Strouhal number and the time history of the lift and drag coefficients are shown in Figs. 3 and 4. Compared to the T1 formulation, the T6

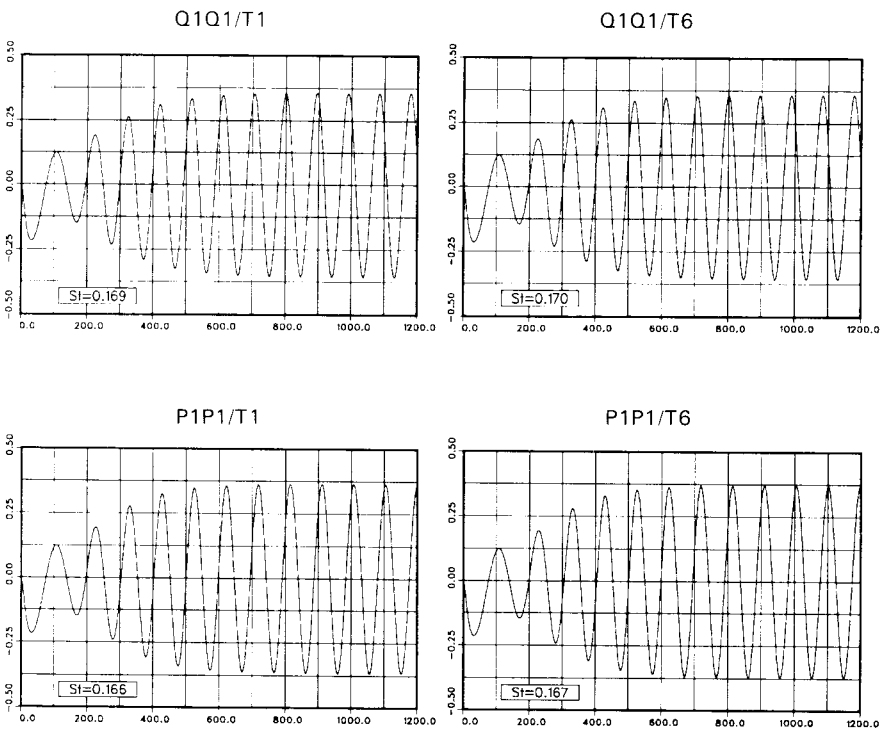

FIG. 3. Periodic solution (obtained with various formulations) for flow past a cylinder at Reynolds number 100: Strouhal number and the time history of the lift coefficient (Tezduyar *et al.*, 1990c).

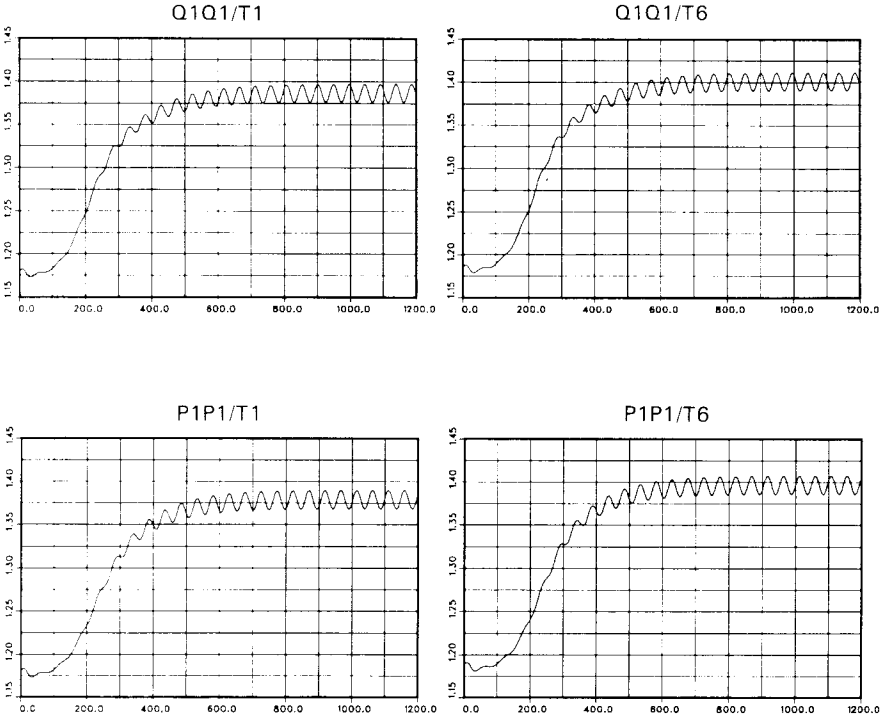

FIG. 4. Periodic solution (obtained with various formulations) for flow past a cylinder at Reynolds number 100: time history of the drag coefficient (Tezduyar *et al.*, 1990c).

formulation gives a slightly higher Strouhal number. Also, the Q1Q1 element gives a Strouhal number about 2% higher than what the P1P1 element gives. Although the lift and drag coefficients show no significant difference among different formulations, the Q1Q1 element gives a slightly higher drag coefficient than the P1P1 element, and the T6 formulation gives a slightly higher drag coefficient than the T1 formulation.

The periodic solution flow patterns corresponding to the crest value of the lift coefficient are shown in Figs. 5–8. The patterns corresponding to the trough value of the lift coefficient are simply the mirror images, with respect to the horizontal centerline, of the patterns shown in Figs. 5–8. The solutions obtained with different formulations are very similar. However, it can be seen, upon close comparison, that the T6 formulation is less dissipative than the T1 formulation and that the Q1Q1 element is less dissipative than the P1P1 element. On comparing these solutions with the

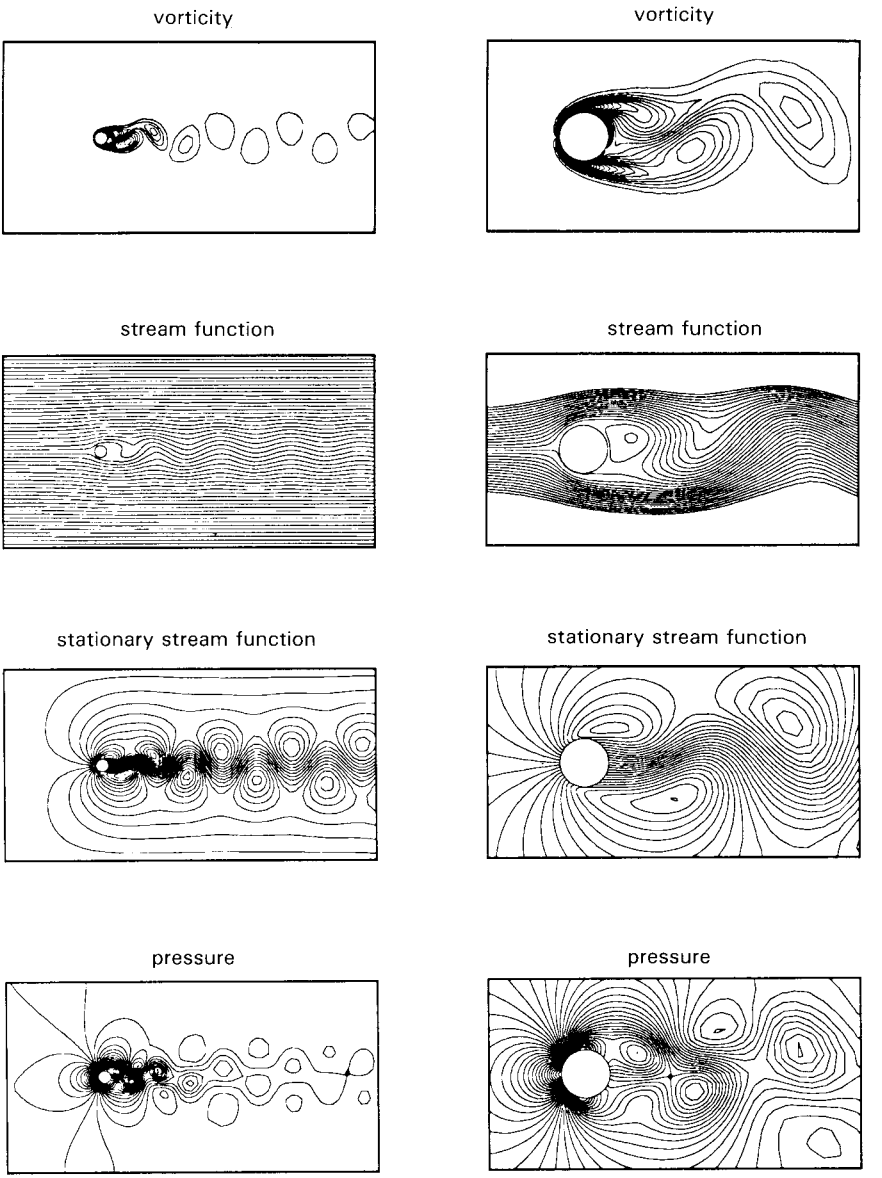

FIG. 5. Periodic solution (obtained with Q1Q1/T1) for flow past a cylinder at Reynolds number 100: flow patterns corresponding to the crest value of the lift coefficient (Tezduyar *et al.*, 1990c).

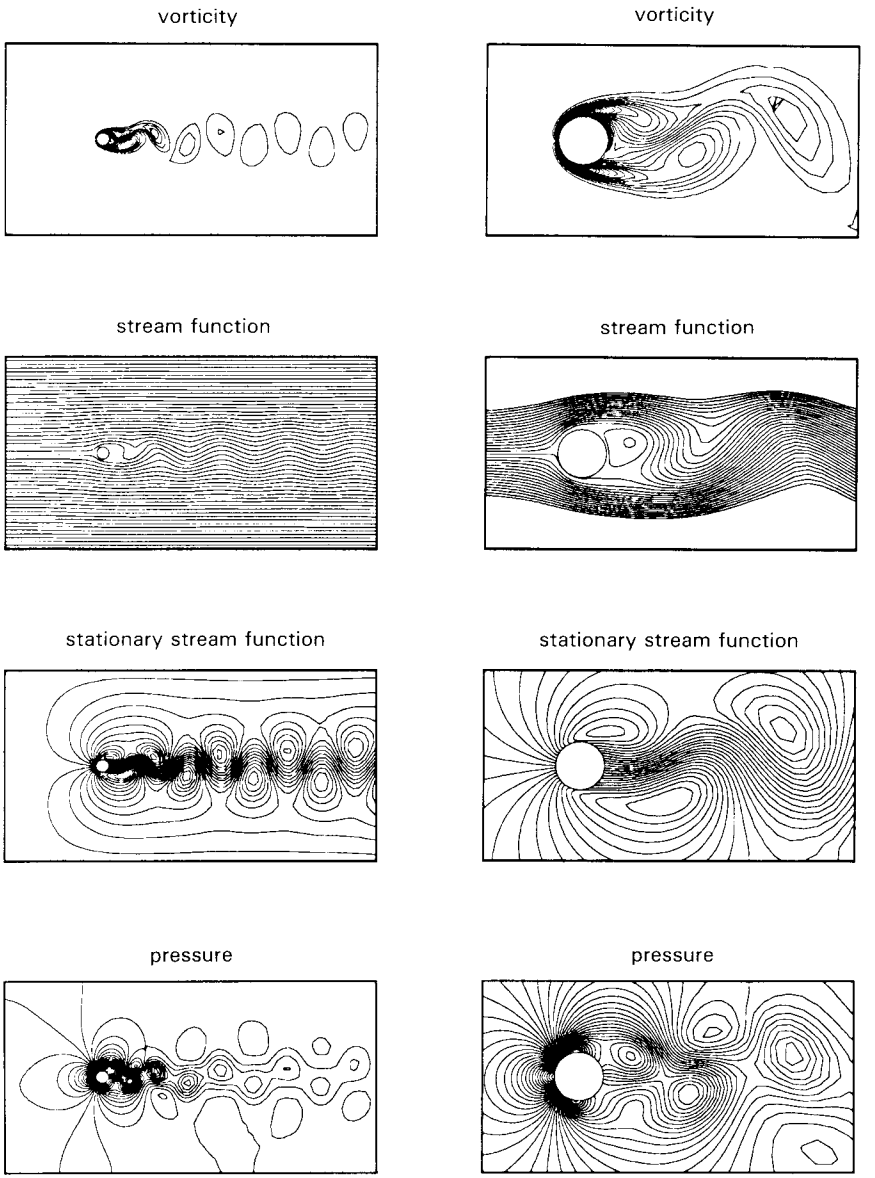

FIG. 6. Periodic solution (obtained with Q1Q1/T6) for flow past a cylinder at Reynolds number 100: flow patterns corresponding to the crest value of the lift coefficient (Tezduyar *et al.*, 1990c).

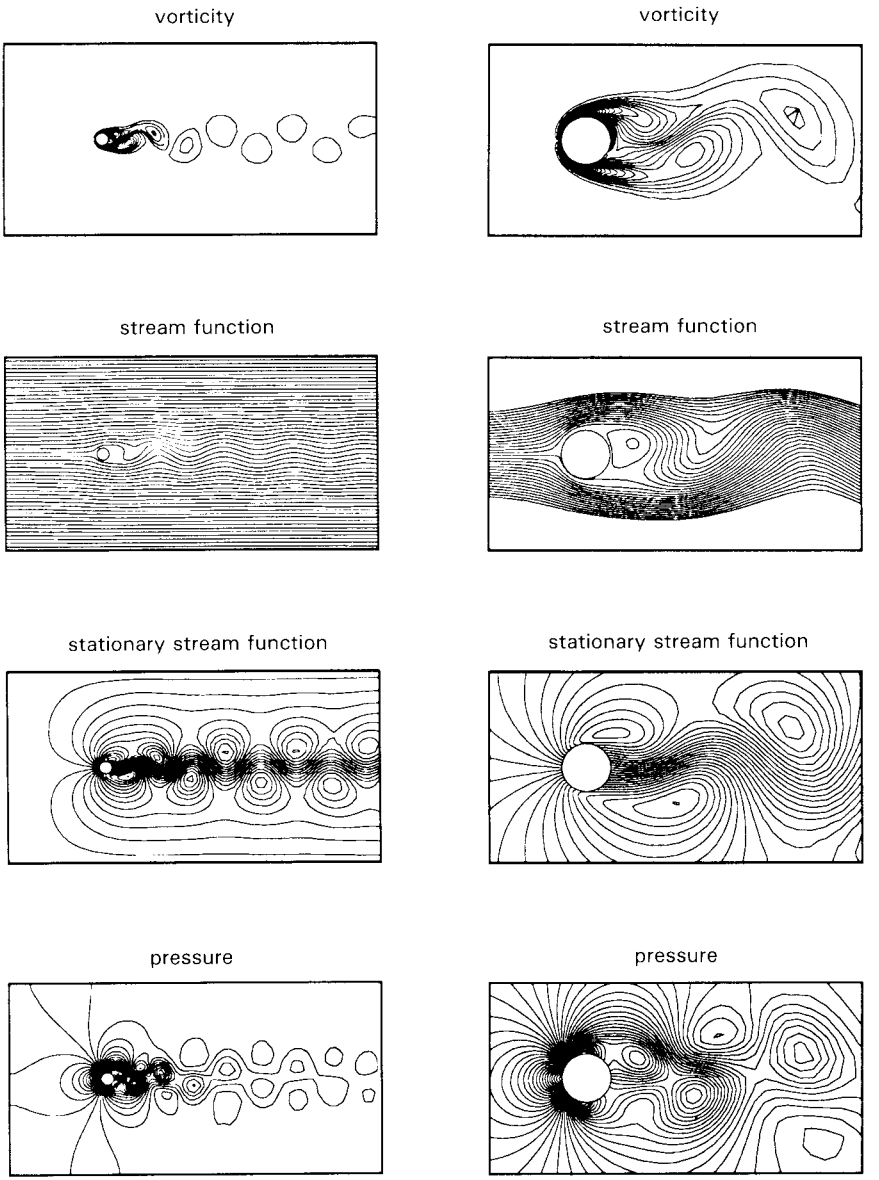

FIG. 7. Periodic solution (obtained with P1P1/T1) for flow past a cylinder at Reynolds number 100: flow patterns corresponding to the crest value of the lift coefficient (Tezduyar *et al.*, 1990c).

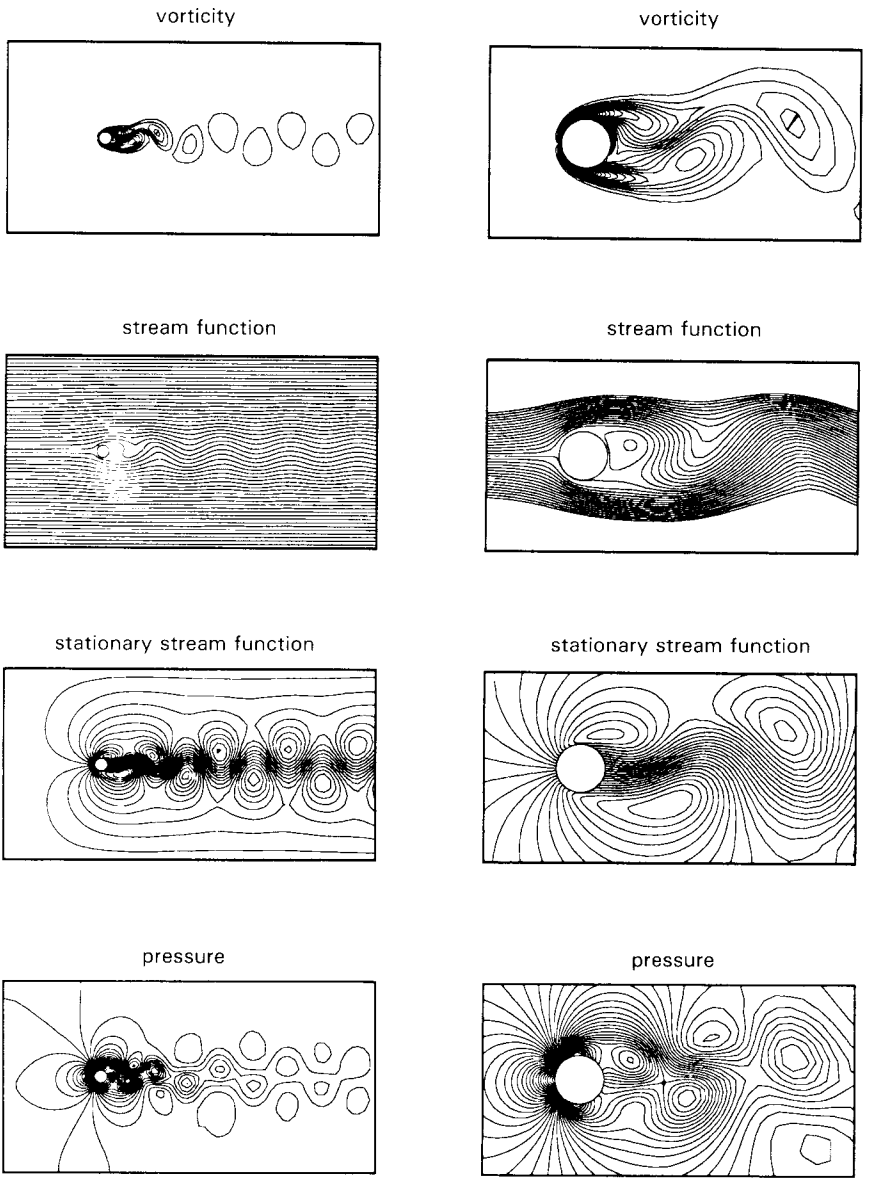

FIG. 8. Periodic solution (obtained with P1P1/T6) for flow past a cylinder at Reynolds number 100: flow patterns corresponding to the crest value of the lift coefficient (Tezduyar *et al.*, 1990c).

ones reported in Tezduyar *et al.* (1990b), it can be observed that the solutions obtained with the Q1Q1 and P1P1 elements are very close to the ones obtained with the pQ2P1 and Q1P0/T6 elements.

V. Application to Moving Boundaries and Interfaces: The DSD/ST Procedure

A. The Method

It was first shown in Tezduyar *et al.* (1990d,e) that the stabilized space-time finite element formulation described in Section III can be effectively applied to fluid dynamics computations involving moving boundaries and interfaces. The variational formulation associated with the DSD/ST (Deforming-Spatial-Domain/Space-Time) procedure is only slightly different than the one given by Eq. (3.5) in Section III. Because the spatial domains are now time-dependent, the subscript t that was dropped from the symbols such as Ω_t and Γ_t needs to be reinstated. Furthermore, we let $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$, and define the space-time slab Q_n as the domain enclosed by the surfaces Ω_n, Ω_{n+1}, and P_n (see Fig. 9). The variational formulation replacing the one given by Eq. (3.5) can then be written as follows:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h\right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\, dQ \\
&\quad - \int_{(P_n)_h} \mathbf{w}^h \cdot \boldsymbol{h}\, dP - \int_{(P_n)_{AB}} \mathbf{w}^h \cdot \mathbf{n}_A \gamma / R_A\, dP \\
&\quad + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h\, dQ + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot ((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-)\, d\Omega \\
&\quad + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau \left[\rho\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right) - \nabla \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h)\right] \\
&\quad \cdot \left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h\right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\right] dQ = 0, \qquad (5.1)
\end{aligned}$$

where $(P_n)_{AB}$ is the space-time surface described by the boundary $(\Gamma_t)_{AB}$ as t traverses the time interval (t_n, t_{n+1}).

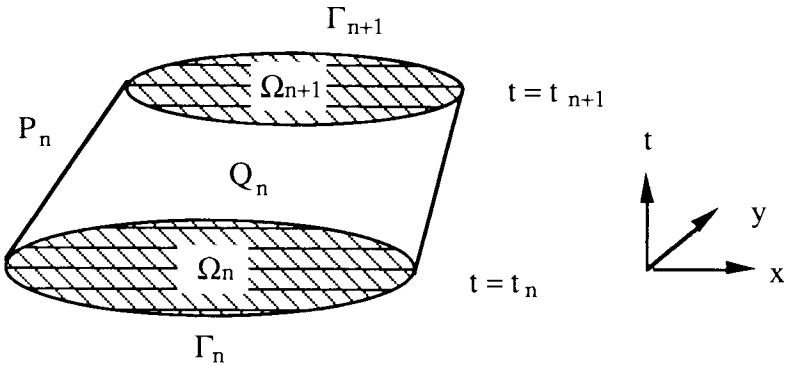

FIG. 9. The space-time slab for the DSD/ST formulation (Tezduyar *et al.*, 1990d).

Remarks

8. The kinematical conditions at the interface $(\Gamma_t)_{AB}$ are automatically satisfied because the discretized subdomains $(\Omega_t)_A$ and $(\Omega_t)_B$ share the same nodes at this interface.

9. The additional term (i.e., the fourth integral) in Eq. (5.1) enforces the dynamical conditions associated with the interfaces and free-surfaces in the presence of surface tension effects. If the interface is to be interpreted as the free-surface of a single fluid, then the fluid is assumed to occupy subdomain $(\Omega_t)_A$. This variational formulation can of course be easily extended to more than two fluids.

10. For two-liquid flows, the solution and variational function spaces for pressure should include the functions that are discontinuous across the interface.

As a special case of drifting solid objects, let us now consider a drifting cylinder. The cylinder moves with unknown linear velocity components V_1 and V_2 and angular velocity $\dot{\Theta}$. The temporal evolutions of these additional unknowns depend on the flow field and can be described by writing the Newton's law for the cylinder:

$$\frac{dV_1}{dt} = \frac{D(V_1, V_2, \dot{\Theta}, \mathbf{U})}{m}, \tag{5.2}$$

$$\frac{dV_2}{dt} = \frac{L(V_1, V_2, \dot{\Theta}, \mathbf{U})}{m}, \tag{5.3}$$

$$\frac{d\dot{\Theta}}{dt} = \frac{T(V_1, V_2, \dot{\Theta}, \mathbf{U})}{J}, \tag{5.4}$$

where D, L, and T are the drag, lift and torque on the cylinder, respectively, while m and J are its mass and polar moment of inertia. The vector of nodal values of velocity and pressure is denoted by $\mathbf{U}$. Temporal discretization of Eqs. (5.2)–(5.4) leads to a set of equations which, in an abstract form, can be written as

$$\mathbf{V} - \mathbf{V}^- = \Delta t\, \mathbf{D}(\mathbf{V}^-, \mathbf{V}, \mathbf{U}). \tag{5.5}$$

Here, $\mathbf{V}$ (unknown) and $\mathbf{V}^-$ (known) are vectors representing the motion of the cylinder, respectively, inside the current space–time slab and at the end of the previous one. The current slab thickness $t_{n+1} - t_n$ is Δt. For linear-in-time interpolation, Eq. (5.5) takes the form

$$\begin{bmatrix} (V_1)^-_{n+1} \\ (V_2)^-_{n+1} \\ (\dot{\Theta})^-_{n+1} \\ (V_1)^+_n \\ (V_2)^+_n \\ (\dot{\Theta})^+_n \end{bmatrix} - \begin{bmatrix} (V_1)^-_n \\ (V_2)^-_n \\ (\dot{\Theta})^-_n \\ (V_1)^-_n \\ (V_2)^-_n \\ (\dot{\Theta})^-_n \end{bmatrix} = \Delta t \begin{bmatrix} \frac{1}{2m}(D^-_n + D^-_{n+1}) \\ \frac{1}{2m}(L^-_n + L^-_{n+1}) \\ \frac{1}{2J}(T^-_n + T^-_{n+1}) \\ \frac{1}{6m}(D^-_n - D^-_{n+1}) \\ \frac{1}{6m}(L^-_n - L^-_{n+1}) \\ \frac{1}{6J}(T^-_n - T^-_{n+1}) \end{bmatrix}. \tag{5.6}$$

Based on the general expression (5.5), we can write the incremental form of (5.6) as

$$-\Delta t \left(\frac{\partial \mathbf{D}}{\partial \mathbf{U}}\right) \Delta \mathbf{U} + \left[\mathbf{I} - \Delta t \left(\frac{\partial \mathbf{D}}{\partial \mathbf{V}}\right)\right] \Delta \mathbf{V} = \mathbf{R}_{\mathbf{V}}(\mathbf{U}, \mathbf{V}). \tag{5.7}$$

Equation (5.7) is of course coupled with the incremental form of the discrete equation system resulting from (5.1):

$$(\mathbf{M}^*_{\mathbf{UU}})\, \Delta \mathbf{U} + (\mathbf{M}^*_{\mathbf{UV}})\, \Delta \mathbf{V} = \mathbf{R}_{\mathbf{U}}(\mathbf{U}, \mathbf{V}). \tag{5.8}$$

In computations reported in this section, the system (5.7)–(5.8) is solved by

a block iteration scheme in which the term $(\partial \mathbf{D}/\partial \mathbf{V})$ is neglected. During each iteration, Eq. (5.8) is solved for $\Delta \mathbf{U}$ only, using the value of $\mathbf{V}$ from the previous iteration; and then $\mathbf{V}$ is updated by (5.7) while $\mathbf{U}$ is held constant. However, the full system can, in principle, be solved simultaneously to take advantage of larger time steps afforded by a fully implicit method. Iterating on the solution will still be needed not only because of the nonlinear nature of (2.1) but also because of the dependence of the element domains Q_n^e on the vector $\mathbf{V}$.

Remark

11. In the DSD/ST procedure, to facilitate the motion of free-surfaces, interfaces, and solid boundaries, we need to move the boundary nodes with the normal component of the velocity at those nodes. Except for this restriction, we have the freedom to move all the nodes any way we would like to. With this freedom, we can move the mesh in such a way that we only need to remesh when it becomes necessary to do so to prevent unacceptable degrees of mesh distortion and potential entanglements. By minimizing the frequency of remeshing, we minimize the projection errors expected to be introduced by remeshing. In fact, for some computations, as a by-product of moving the mesh, we may be able to get a limited degree of automatic mesh refinement, again with minimal projection errors. For example, a mesh moving scheme suitable for a single cylinder drifting in a bounded flow domain is described in Tezduyar *et al.* (1990e).

B. Numerical Examples

All solutions presented in this section were obtained with linear-in-time interpolation functions. For the details of the computations, see Tezduyar *et al.* (1990e).

Free-surface wave propagation

This is a problem that was considered in Hughes *et al.* (1981). Initially, the fluid is stationary, and occupies a long rectangular region with dimensions $L \times D$, where $L = 949.095$ and $D = 10$. The flow is assumed to be inviscid, and both the density and the gravity are set to 1.0. The mesh consists of 320 elements, with two elements through depth. The wave is generated by prescribing the velocity along the left-hand boundary of the

domain according to the expression $u_1 = (Hc/D)\,\mathrm{sech}^2(c\kappa t/D - 4)$, where $c = [g(D + H)]^{1/2}$ and $\kappa = (3H/4D)^{1/2}$, with $g = 1$ and $H = 0.86$. The time step size is 1.789. Figure 10 shows the solutions obtained at various time steps. After 160 time steps, the wave retains 94.4% of its initial amplitude. This solution compares well with those presented in Hughes *et al.* (1981).

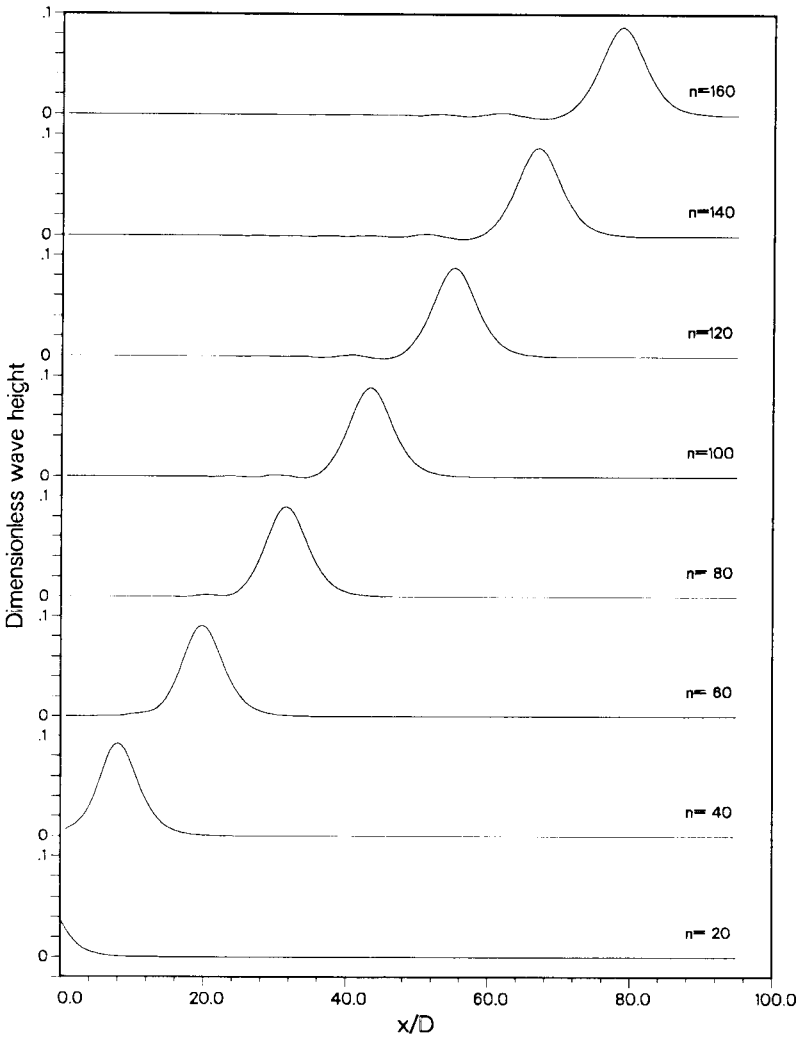

FIG. 10. Free-surface wave propagation: time history of the surface wave (Tezduyar *et al.*, 1990e).

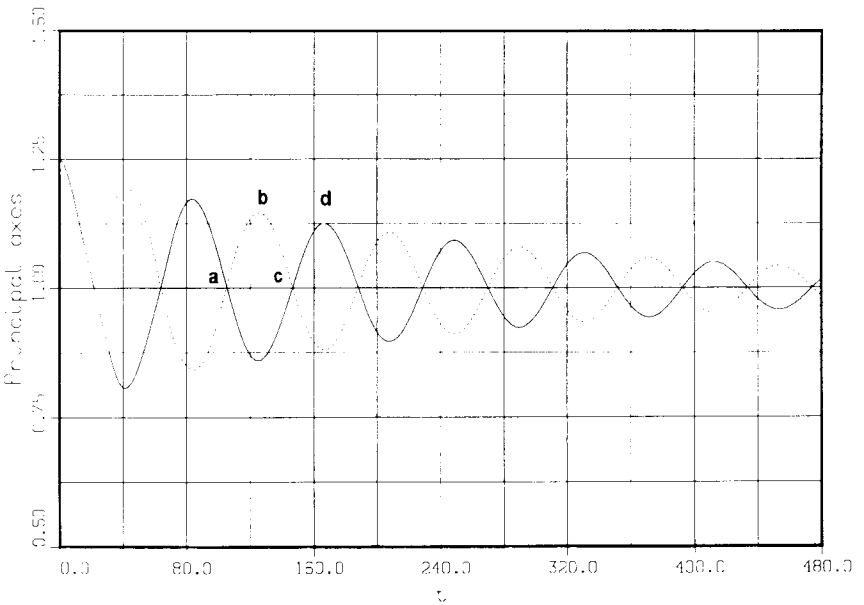

FIG. 11. Pulsating drop: time history of the axial dimensions of the drop (Tezduyar *et al.*, 1990e).

Pulsating drop

In this problem, the drop is initially of elliptical shape with axial dimensions 1.25 (horizontal) and 0.80 (vertical). The density, viscosity, and the surface tension coefficient are 1.0, 0.001 and 0.001, respectively. The effect of gravity is neglected. The number of elements is 380, and the time step size is 1.0. Figure 11 shows the time history of the axial dimensions of the drop. Figures 12a, 12b, 12c, and 12d show the flow field and finite element mesh corresponding, approximately, two points *a*, *b*, *c*, and *d* in Fig. 11.

Large-amplitude sloshing

This problem is similar to the one that was considered in Huerta and Liu (1988). Initially, the fluid is stationary and occupies a 2.667×1.0 rectangular region. The density and viscosity are 1.0 and 0.002, respectively. The gravity is 1.0, and the surface tension is neglected. The wave is created by applying a horizontal body force of $A \sin(\omega t)$, where $A = 0.01$ and $\omega = 0.978$. The Reynolds number (based on the height of the fluid and the gravity) is 514. Inviscid boundary conditions are assumed at the walls of the "tank." Compared to the problem considered here, the

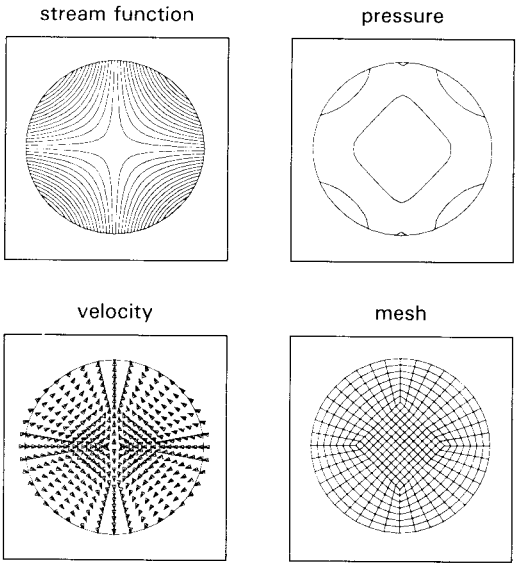

FIG. 12. (a) Pulsating drop: flow field and finite element mesh corresponding (approximately) to point *a* in Fig. 11 (Tezduyar *et al.*, 1990e).

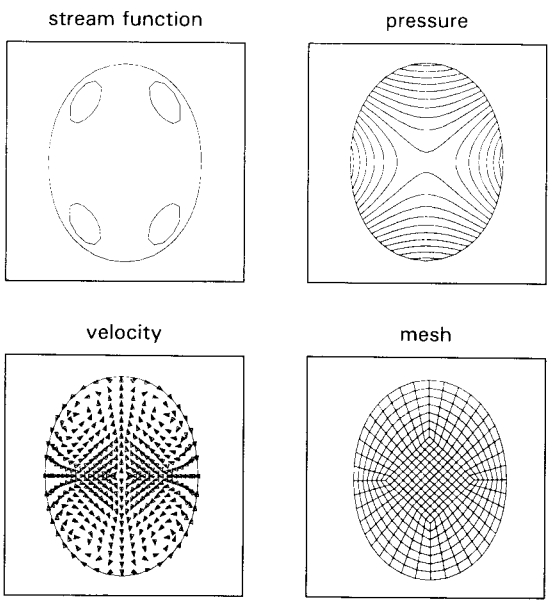

FIG. 12. (b) Pulsating drop: flow field and finite element mesh corresponding (approximately) to point *b* in Fig. 11 (Tezduyar *et al.*, 1990e).

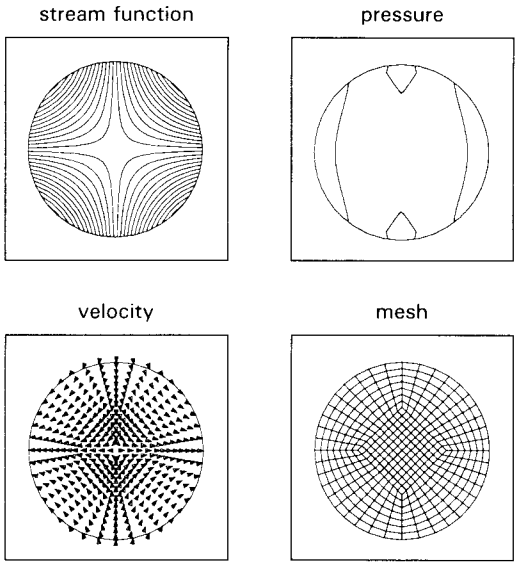

FIG. 12. (c) Pulsating drop: flow field and finite element mesh corresponding (approximately) to point *c* in Fig. 11 (Tezduyar *et al.*, 1990e).

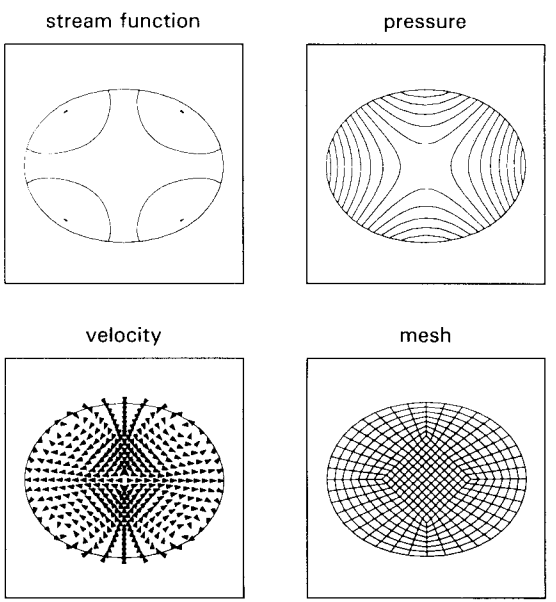

FIG. 12. (d) Pulsating drop: flow field and finite element mesh corresponding (approximately) to point *d* in Fig. 11 (Tezduyar *et al.*, 1990e).

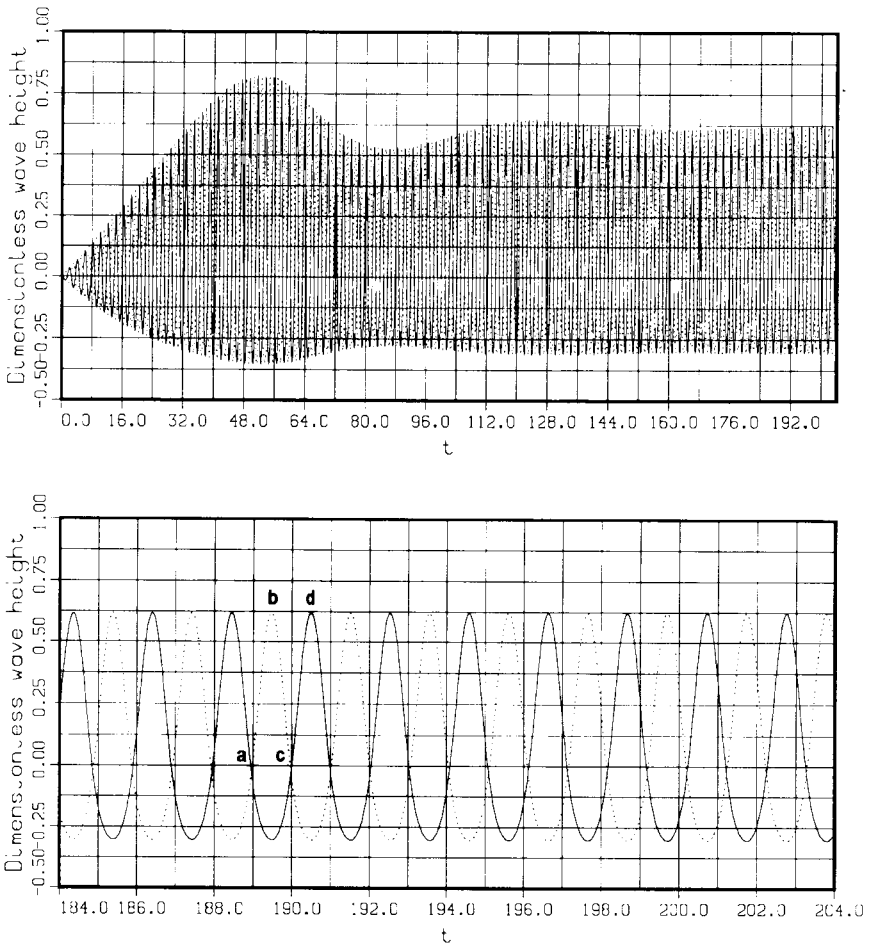

FIG. 13. Large-amplitude sloshing: time history of the vertical location (relative to the stationary level) of the free-surface along the left- and right-hand sides of the "tank" (Tezduyar *et al.*, 1990e).

Reynolds number used in Huerta and Liu (1988) is 514,000. Furthermore, in Huerta and Liu (1988) the horizontal body force is removed after ten cycles; in this case, on the other hand, this force is maintained during the entire computation. The number of elements is 441, and the time step size is 0.107. With these values of the frequency and the time step size, a single period of the forcing function takes 60 time steps. Figure 13 shows the time history of the vertical location (relative to the stationary level of 1.0) of the free-surface along the left- and right-hand sides of the "tank." Figures 14a,

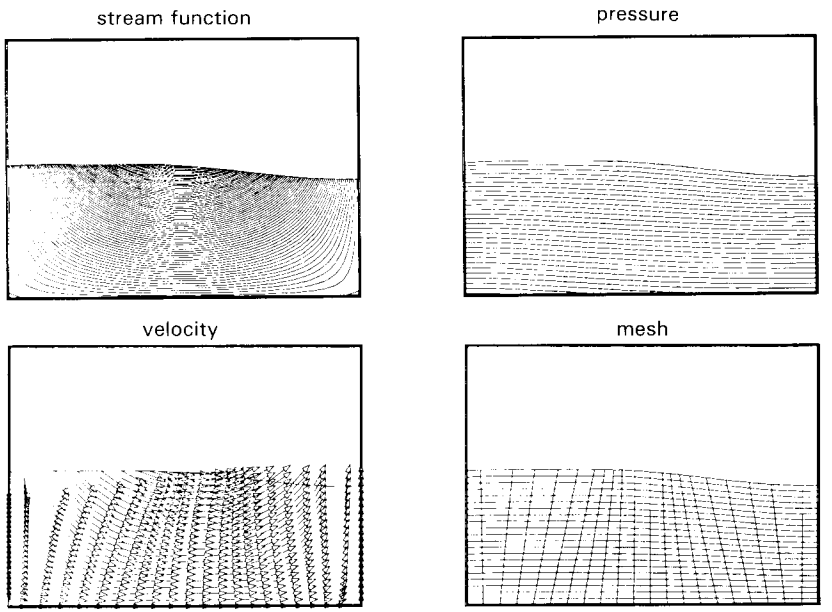

FIG. 14. (a) Large-amplitude sloshing: flow field and finite element mesh corresponding (approximately) to point *a* in Fig. 13 (Tezduyar *et al.*, 1990e).

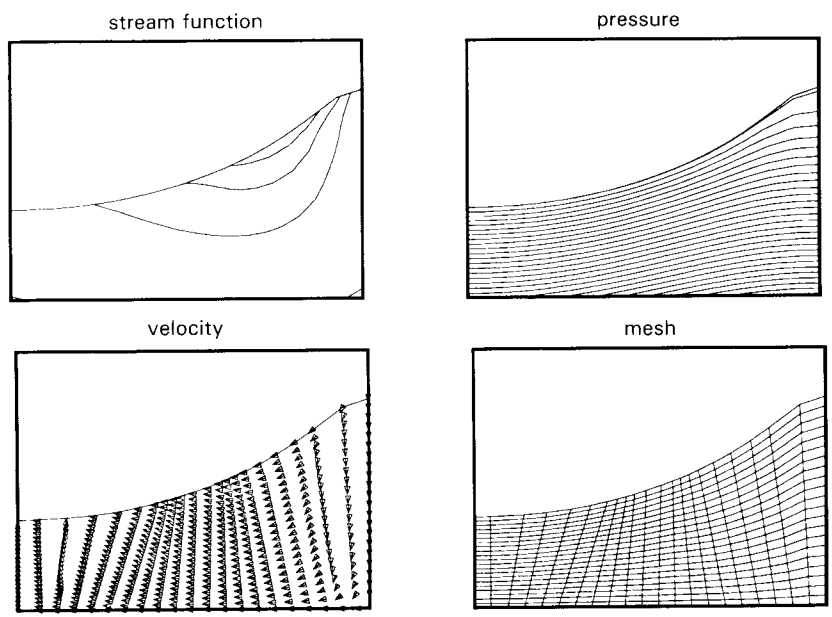

FIG. 14. (b) Large-amplitude sloshing: flow field and finite element mesh corresponding (approximately) to point *b* in Fig. 13 (Tezduyar *et al.*, 1990e).

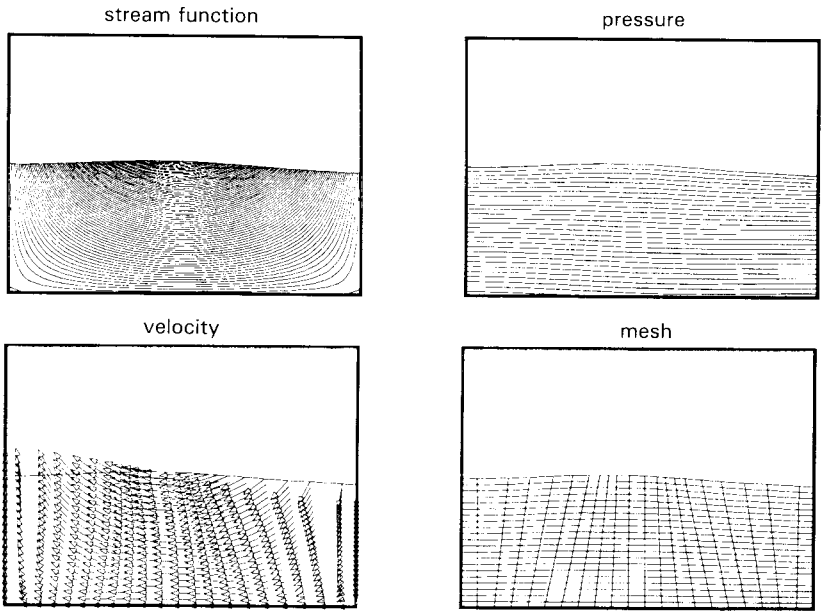

FIG. 14. (c) Large-amplitude sloshing: flow field and finite element mesh corresponding (approximately) to point *c* in Fig. 13 (Tezduyar *et al.*, 1990e).

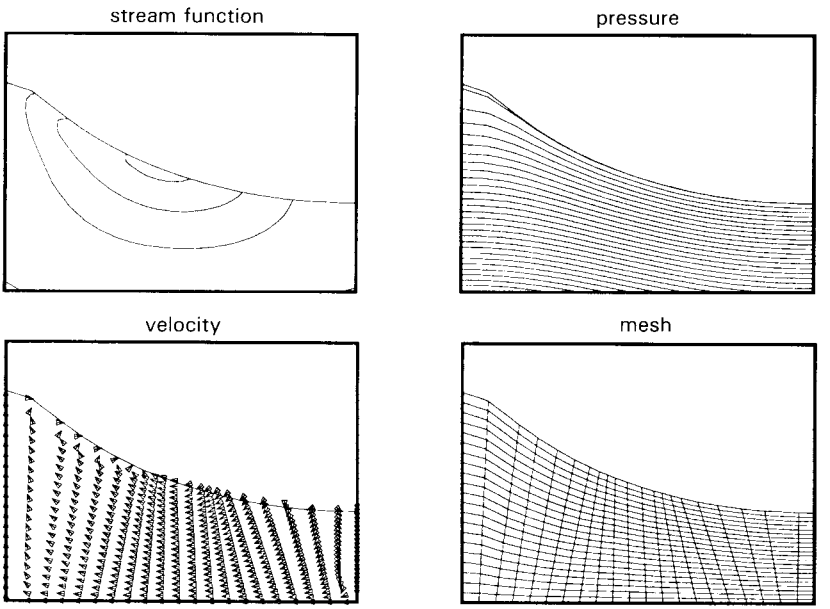

FIG. 14. (d) Large-amplitude sloshing: flow field and finite element mesh corresponding (approximately) to point *d* in Fig. 13 (Tezduyar *et al.*, 1990e).

14b, 14c, and 14d show the flow field and finite element mesh corresponding, approximately, to points *a*, *b*, *c*, and *d* in Fig. 13.

A cylinder drifting in a shear flow

This test problem involves a cylinder (with unit radius) drifting in a shear flow in a 61×32 bounded domain. The density and viscosity are 1.0 and 0.005, respectively. The upper and lower walls move with velocities 0.156 and 0.094. The upstream velocity profile is assumed to be a linear

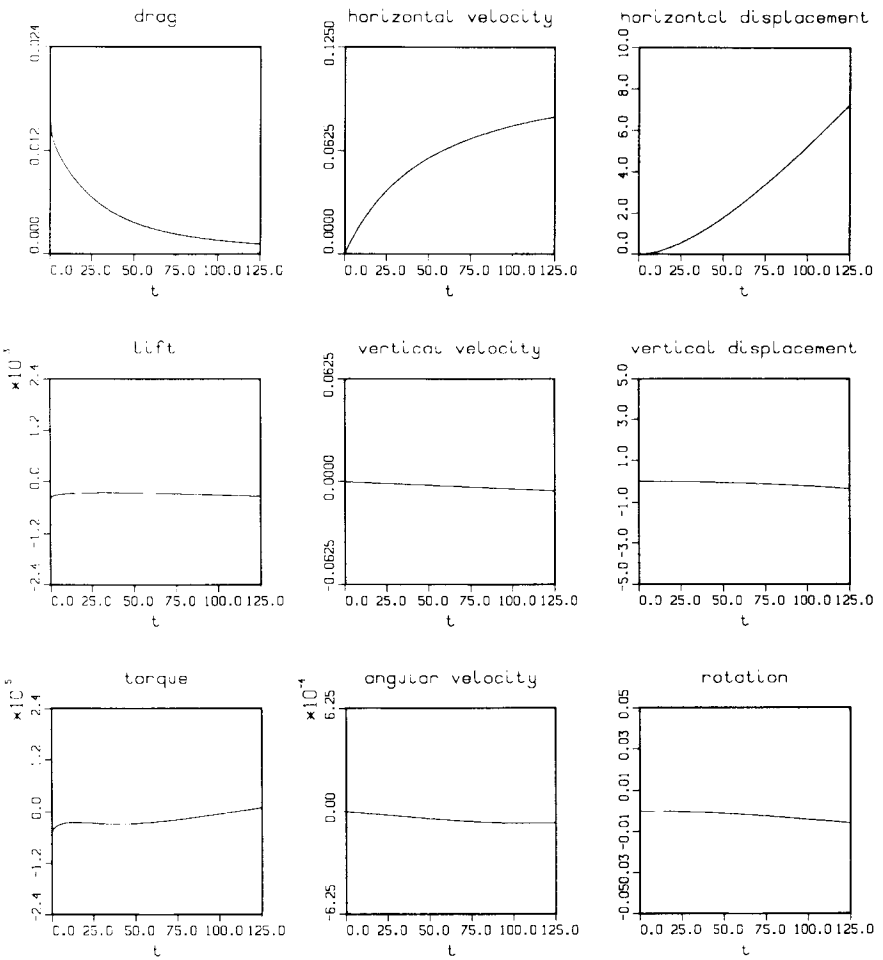

FIG. 15. A cylinder drifting in a shear flow: time history of the drag, lift, torque, linear and angular velocity components, displacement components, and rotation (Tezduyar *et al.*, 1990e).

interpolation of the velocities at the upper and lower boundaries. At the downstream boundary, normal and shear stresses are specified consistent with the undisturbed shear flow. The Reynolds number based on the average upstream velocity and the cylinder diameter is 50. The mass and polar moment of inertia of the cylinder are 2π and π, respectively. The initial condition is the steady-state solution for the fixed cylinder located at (16,16) relative to the lower left corner. The number of elements is 1152. Except for the first few time steps, the time step size is 0.125. Figure 15 shows, for the cylinder, time history of the drag, lift, torque, linear and angular velocity components, displacement components, and rotation. Figures 16a, 16b, and 16c show the flow field and finite element mesh at $t = 0.0$, 62.5, and 125.0, respectively.

VI. Concluding Remarks

In this chapter, a review of stabilization techniques used in finite element computation of incompressible flows was presented. Stabilization is needed to prevent the numerical oscillations that might be produced by the presence of dominant advection terms in the governing equations or by not using an acceptable combination of interpolation functions to represent the velocity and pressure fields. Many researchers are actively involved in development and analysis of various stabilization methods. It is important to make sure that the stabilization technique does not result in over-stabilization of the problem by introducing excessive numerical diffusion. The Galerkin/least-squares, streamline-upwind/Petrov-Galerkin, and pressure-stabilizing/Petrov-Galerkin stabilization techniques were reviewed in this article more extensively than others. All three stabilization techniques lead to formulations that are consistent. That is, the stabilization terms added to the Galerkin formulation of the problem vanish when an exact solution is substituted into the stabilized formulation. These stabilization methods introduce minimal excess diffusion, and therefore result in solutions with minimal loss of accuracy.

The Galerkin/least-squares stabilization for time-dependent flow problems necessitates finite element discretization in both space and time, and this could be computationally quite costly. However, there are iteration methods designed to reduce the computational cost associated with the space-time finite element formulation. Also, for fixed spatial domains, there are ways to implement the Galerkin/least-squares formulation in a slightly different way then it is formally supposed to be, so that the

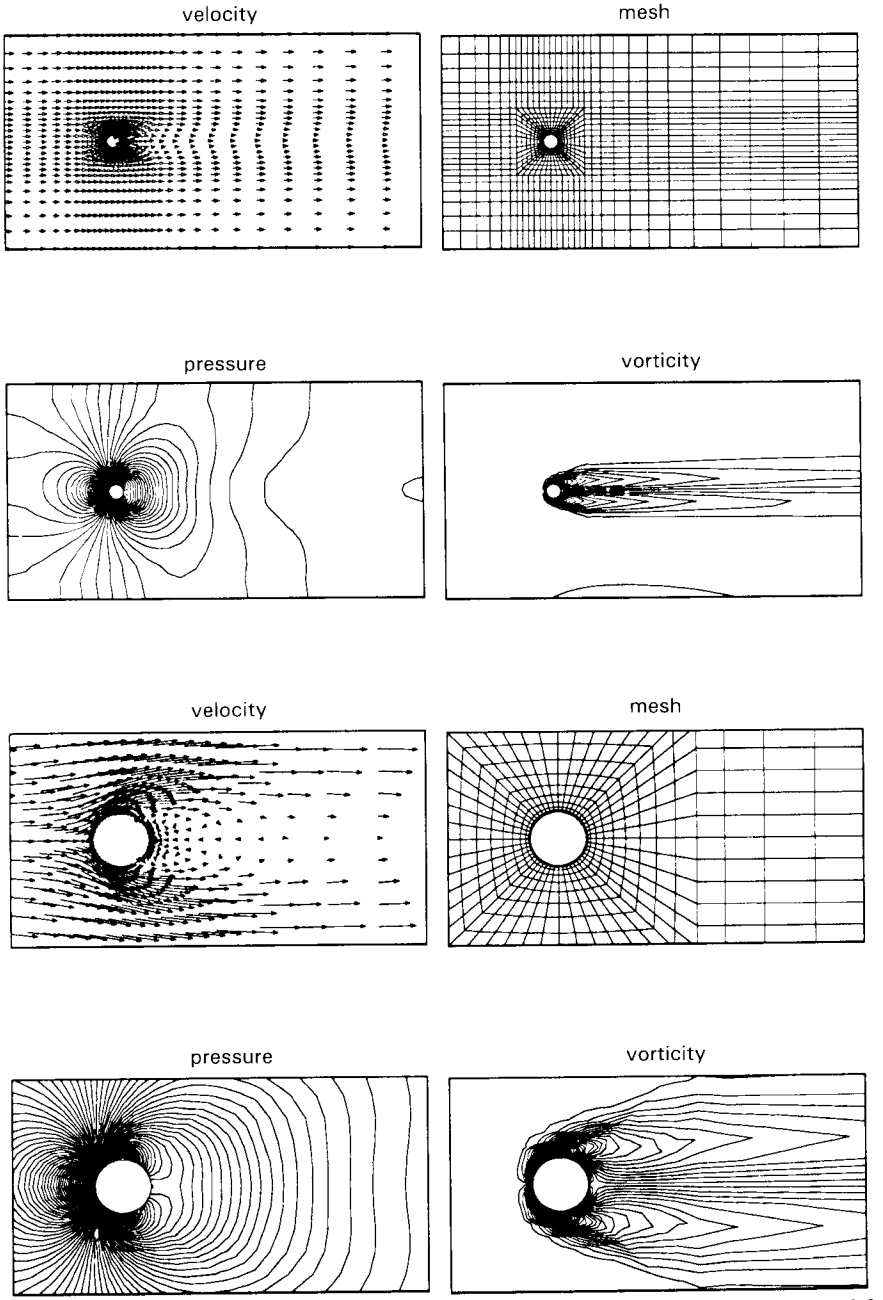

FIG. 16. (a) A cylinder drifting in a shear flow: flow field and finite element mesh at $t = 0.0$ (Tezduyar *et al.*, 1990e).

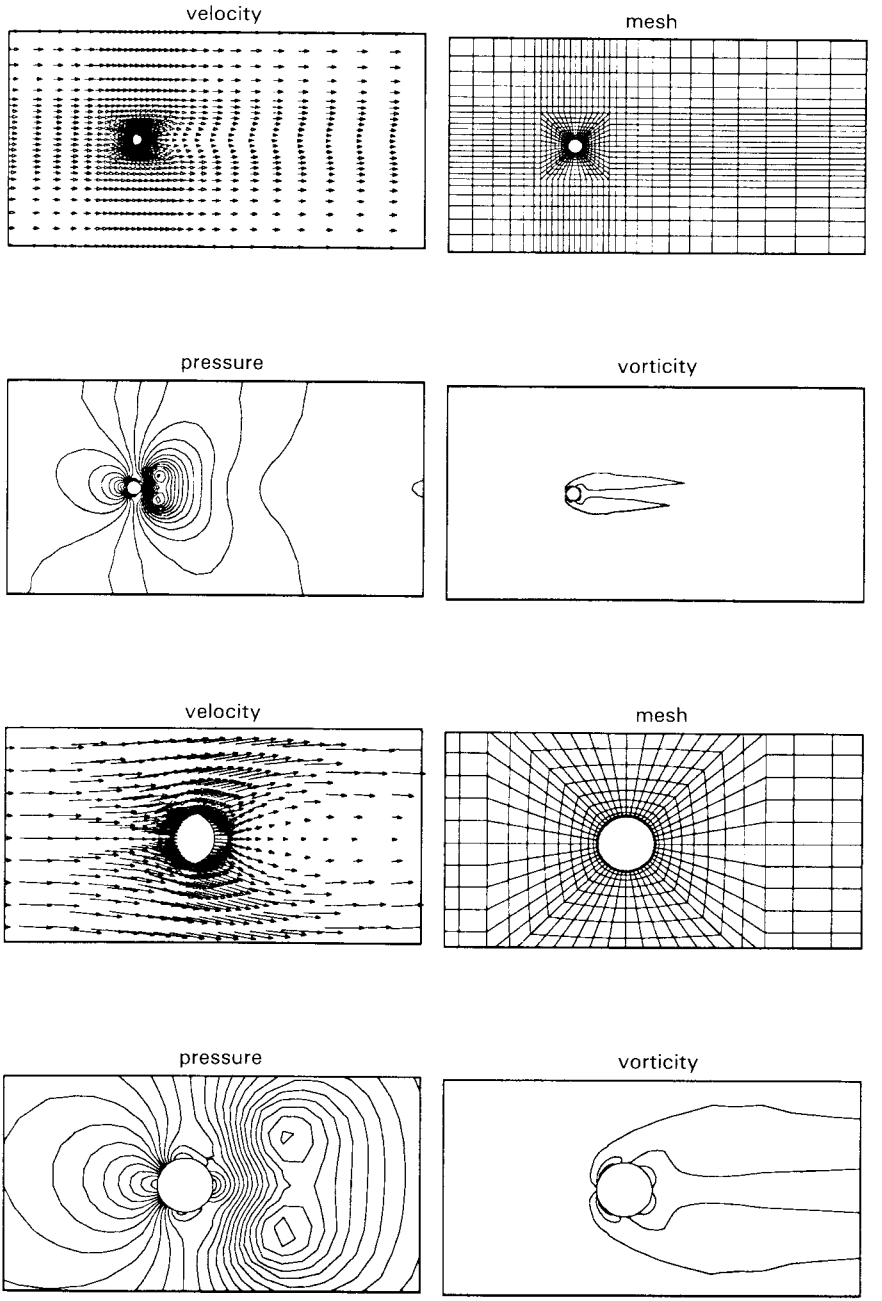

FIG. 16. (b) A cylinder drifting in a shear flow: flow field and finite element mesh at $t = 62.5$ (Tezduyar *et al.*, 1990e).

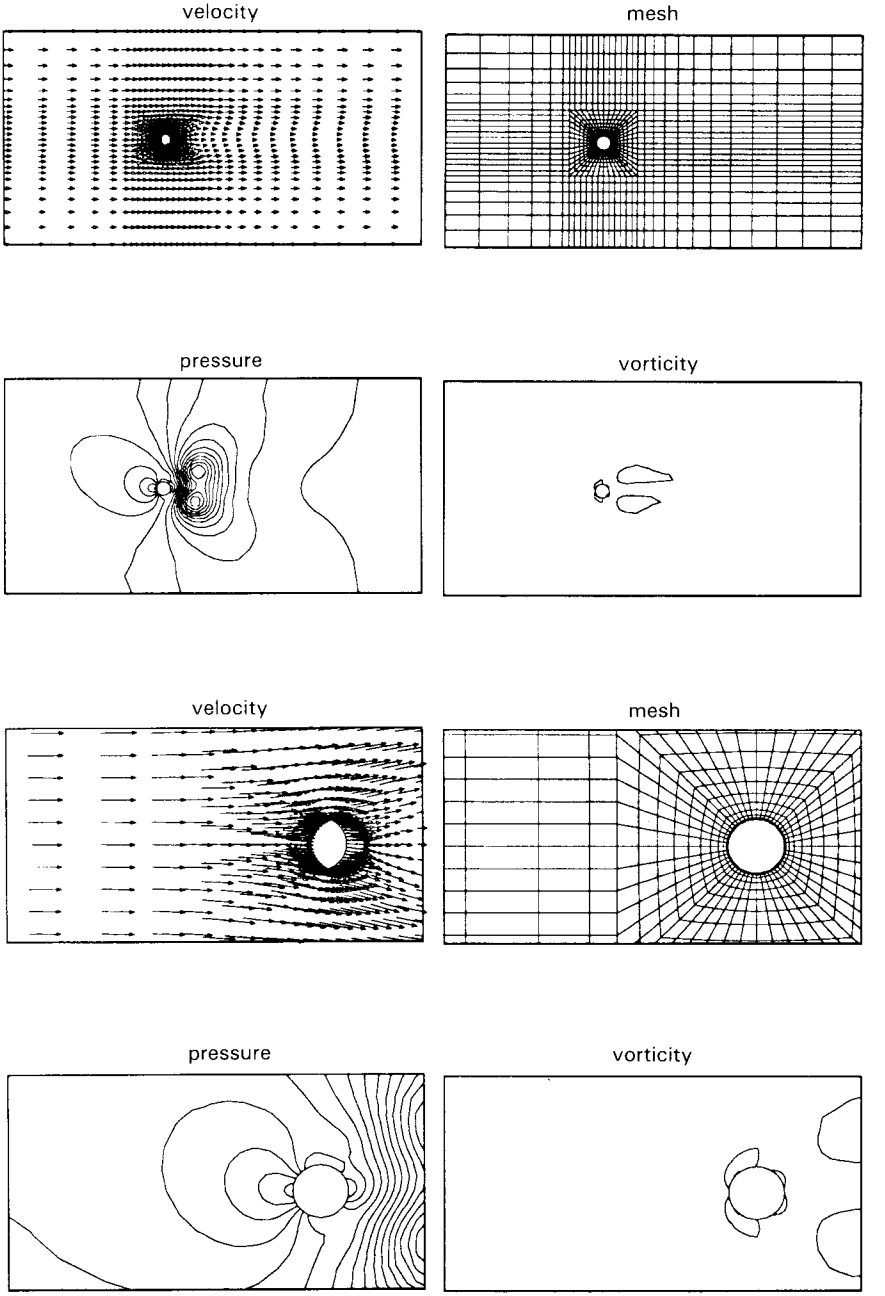

FIG. 16. (c) A cylinder drifting in a shear flow: flow field and finite element mesh at $t = 125.0$ (Tezduyar *et al.*, 1990e).

computations are performed with finite element discretization in space only. On the other hand, the combination of Galerkin/least-squares and space-time finite element formulations gives us a new and very effective strategy for computation of unsteady incompressible flows involving moving boundaries and interfaces.

References

Brezzi, F. (1973). On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers. *RAIRO Anal. Numer.* **8**, 129–151.

Brezzi, F., and Pitkaranta, J. (1984). On the stabilization of finite element approximations of Stokes problem. *Efficient Solutions of Elliptic Systems, Notes on Numerical Fluid Mechanics,* Viewig **10**, 11–19.

Bristeau, M. O., Glowinski, R., Dutto, L., Periaux, J., and Roge, G. (1990). Compressible viscous flow calculations using compatible finite element approximations. *Int. J. Numer. Meth. Fluids* **11**, 719–749.

Bristeau, M. O., Glowinski, R., and Periaux, J. (1987). Numerical methods for the Navier–Stokes equations: Applications to the simulation of compressible and incompressible viscous flows. *Comput. Phys. Rep.* **6**, 73–187.

Brooks, A. N., and Hughes, T. J. R. (1982). Streamline-upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on incompressible Navier–Stokes equation. *Comput. Meth. Appl. Mech. Eng.* **32**, 199–259.

Donea, J. (1984). A Taylor–Galerkin method for convective transport problems. *Int. J. Numer. Meth. Eng.* **20**, 101–120.

Douglas, J., and Wang, J. (1989). An absolutely stabilized finite element method for Stokes problem. *Math. Comput.* **52**, 495–508.

Franca, L. P. and Dutra do Carmo, E. G. (1989). The Galerkin-gradient-least-squares method. *Comput. Meth. Appl. Mech. Eng.* **74**, 41–54.

Franca, L. P., Frey, S. L., and Hughes, T. J. R. (1990). Stabilized finite element methods: I. Application to the advective-diffusive mode. LNCC Research Report, No. 032/90, October 1990; *Comput. Meth. Appl. Mech. Eng.,* to appear.

Franca, L. P., and Hughes, T. J. R. (1988). Two classes of mixed finite element methods. *Comput. Meth. Appl. Mech. Eng.* **69**, 89–129.

Franca, L. P., and Stenberg, R. (1990). Error analysis of some Galerkin-least-squares methods for the elasticity equations. Rapports de Recherche INRIA 1054; *SIAM J. Numer. Anal.,* to appear.

Gresho, P. M. (1990). On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *Int. J. Numer. Meth. Fluids* **11**, 587–620.

Gresho, P. M., and Chan, S. T. (1990). On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation. *Int. J. Numer. Meth. Fluids* **11**, 621–659.

Gresho, P. M., Chan, S., Upson, C., and Lee, R. (1984). A modified finite element method for solving the time-dependent incompressible Navier–Stokes equations. Part 1: Theory, Part 2: Applications. *Int. J. Numer. Meth. Fluids* **4**, 557–598; 619–640.

Hansbo, P., Szepessy, A. (1990). A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Comput. Meth. Appl. Mech. Eng.* **84,** 175-192.

Huerta, A., and Liu, W. K. (1988). Viscous flow with large free surface motion. *Comput. Meth. Appl. Mech. Eng.* **69,** 277-324.

Hughes, T. J. R., and Brooks, A. N. (1979). A multi-dimensional upwind scheme with no crosswind diffusion. *In* "Finite Element Methods for Convection Dominated Flows" (ed. T. J. R. Hughes), AMD-Vol. 34, ASME, New York, pp. 19-35.

Hughes, T. J. R., and Franca, L. P. (1987). A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. *Comput. Meth. Appl. Mech. Eng.* **65,** 85-96.

Hughes, T. J. R., Franca, L. P., and Balestra, M. (1986). A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi Condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Comput. Meth. Appl. Mech. Eng.* **59,** 85-99.

Hughes, T. J. R., Franca, L. P., and Hulbert, G. M. (1989). A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Comput. Meth. Appl. Mech. Eng.* **73,** 173-189.

Hughes, T. J. R., Franca, L. P., Mallet, M. (1987). 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. Meth. Appl. Mech. Eng.* **63,** 97-112.

Hughes, T. J. R., and Hulbert, G. M. (1988). Space-time finite element methods for elastodynamics: Formulations and error estimates. *Comput. Meth. Appl. Mech. Eng.* **66,** 339-363.

Hughes, T. J. R., Liu, W. K., and Zimmermann, T. K. (1981). Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput. Meth. Appl. Mech. Eng.* **29,** 329-349.

Hughes, T. J. R., and Tezduyar, T. E. (1984). Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Comput. Meth. Appl. Mech. Eng.* **45,** 217-284.

Johnson, C., and Saranen, J. (1986). Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Math. Comput.* **47,** 1-18.

Kawahara, M., Hirano, H., Tsubota, K., and Inagaki, K. (1982). Selective mass lumping finite element method for shallow water flow. *Int. J. Numer. Meth. Fluids* **2,** 89-112.

Liou, J., and Tezduyar, T. E. (1990). Computation of compressible and incompressible flows with the clustered element-by-element method. University of Minnesota Supercomputer Institute Research Report UMSI 90/215, October 1990. See update U1 on last page.

Lundgren, T., and Mansour, N. N. (1990). A stabilized Lagrangian finite element method for viscous free surface flows. In preparation.

Pironneau, O., and Rappaz, J. (1989). Numerical analysis for compressible viscous isentropic stationary flows. *Impact of Comput. Sci. Eng.* **1,** 109-137.

Saad, Y., and Schultz, M. H. (1983). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. Yale University Research Report YALEU/DCS/RR-254.

Shakib, F. (1988). Finite element analysis of the compressible Euler and Navier-Stokes equations. Ph.D. Thesis, Department of Mechanical Engineering, Stanford University, Stanford, California.

Silvester, D. J., and Kechkar, N. (1990). Stabilised bilinear-constant velocity-pressure finite

elements for the conjugate gradient solution of the Stokes problem. *Comput. Meth. Appl. Mech. Eng.* **79**, 71–86.

Tezduyar, T. E., and Hughes, T. J. R. (1982). Development of time-accurate finite element techniques for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Report prepared under NASA-Ames University Consortium Interchange No. NCA2-OR745-104.

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

Tezduyar, T. E., Glowinski, R., and Liou, J. (1988). Petrov–Galerkin methods on multiply-connected domains for the vorticity–stream function formulation of the incompressible Navier–Stokes equations. *Int. J. Numer. Meth. Fluids* **8**, 1269–1290.

Tezduyar, T. E., Liou, J., and Ganjoo, D. K. (1990a). Incompressible flow computations based on the vorticity–stream function and velocity–pressure formulations. *Comput. Struct.* **35**, 445–472.

Tezduyar, T. E., Shih, R., and Mittal, S. (1990e). Time-accurate incompressible flow computations with quadrilateral velocity–pressure elements. University of Minnesota Supercomputer Institute Research Report UMSI 90/143, August 1990. See update U2 below.

Tezduyar, T. E., Shih, R., Mittal, S., and Ray, S. E. (1990b). Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity–pressure elements. University of Minnesota Supercomputer Institute Research Report UMSI 90/165, September 1990. See update U3 below.

Tezduyar, T. E., Liou, J., and Behr, M. (1990d). A new strategy for finite element computations involving moving boundaries and interfaces—the DSD/ST procedure: I. The concept and the preliminary numerical tests. University of Minnesota Supercomputer Institute Research Report UMSI 90/169, September 1990. See update U4 below.

Tezduyar, T. E., Liou, J., Behr, M., and Mittal, S. (1990e). A new strategy for finite element computations involving moving boundaries and interfaces—the DSD/ST procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. University of Minnesota Supercomputer Institute Research Report UMSI 90/170, September 1990. See update U5 below.

Publication Updates

U1. J. Liou and T.E. Tezduyar, "Clustered Element-by-Element Computations for Fluid Flow", Chapter 9 in *Parallel Computational Fluid Dynamics* (ed. H.D. Simon), MIT Press, Cambridge, Massachusetts (1992) 167-187.

U2. T.E. Tezduyar, S. Mittal and R. Shih, "Time-accurate Incompressible Flow Computations with Quadrilateral Velocity-Pressure Elements", *Computer Methods in Applied Mechanics and Engineering*, **87** (1991) 363-384.

U3. T.E. Tezduyar, S. Mittal, S.E. Ray and R. Shih, "Incompressible Flow Computations with Stabilized Bilinear and Linear Equal-order-interpolation Velocity-Pressure Elements", *Computer Methods in Applied Mechanics and Engineering*, **95** (1992) 221-242.

U4. T.E. Tezduyar, M. Behr and J. Liou, "A New Strategy for Finite Element Computations Involving Moving Boundaries and Interfaces — The Deforming-Spatial-Domain/Space-Time Procedure: I. The Concept and the Preliminary Numerical Tests", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 339-351.

U5. T.E. Tezduyar, M. Behr, S. Mittal and J. Liou, "A New Strategy for Finite Element Computations Involving Moving Boundaries and Interfaces — The Deforming-Spatial-Domain/Space-Time Procedure: II. Computation of Free-surface Flows, Two-liquid Flows, and Flows with Drifting Cylinders", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 353-371.