Computer Physics Communications 73 (1992) 93–112
North-Holland

Computer Physics Communications

Notes on the stabilized space–time finite-element formulation of unsteady incompressible flows *

S. Mittal and T.E. Tezduyar

Department of Aerospace Engineering and Mechanics, Army High-Performance Computing Research Center, and Minnesota Supercomputer Institute, University of Minnesota, 1200 Washington Avenue South, Minneapolis, MN 55415, USA

This paper gives a review of our research efforts on the stabilized space-time finite element formulation of unsteady incompressible flows, including those involving moving boundaries are interfaces. Iterative solution techniques employed to solve the equation systems resulting from the space–time finite element discretization of these flow problems are also reviewed. Results are presented for certain unsteady flow problems, including large-amplitude sloshing and flows past oscillating cylinders.

1. Introduction

This manuscript is a written version of the lectures of the second author at the Ninth Summer School on Computing Techniques in Physics, held at Skalský dvůr, Czechoslovakia. The lectures focused on review of our research efforts on the stabilized space–time finite element formulation of unsteady incompressible flows, including those involving moving boundaries and interfaces. Iterative solution techniques employed to solve the equation systems resulting from the space–time finite element discretization of these flow problems are also reviewed. Most of the material in this manuscript has been extracted from recent paper by Tezduyar et al. [1,2], Liou and Tezduyar [3], Tezduyar [4], Tezduyar and Mittal [5], and Mittal et al. [6]. Results are presented for certain unsteady flow problems, including large-amplitude sloshing and flows past oscillating cylinders.

The space–time formulation reviewed in this article is used in conjunction with the GLS (Galerkin/Least-squares) stabilization. The GLS stabilization prevents 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. In this kind of stabilization, a series of stabilizing terms are added to the Galerkin formulation of the problem. These terms can be obtained by minimizing the sum of the squared residual of the momentum equation integrated over each element domain. The GLS stabilization leads to a consistent formulation, in the sense that an exact solution still satisfies the stabilized formulation. Consequently, it introduces minimal numerical diffusion, and therefore results in solutions with minimal loss of accuracy. This approach has been successfully applied to Stokes flows by Hughes and Franca [7], to compressible flows by Hughes, Franca and Hulbert [8] and Shakib [9], and to incompressible flows at finite Reynolds numbers by Tezduyar et al. [1,2],

Correspondence to: T.E. Tezduyar, Minnesota Supercomputer Institute, University of Minnesota, 1200 Washington Avenue South, Minneapolis, MN 55415, USA.

* This work was sponsored by NASA-JSC under grant NAG 9-449, by NSF under grant MSM-8796352, and by ALCOA Foundation. Partial support for this work has also come from the Army Research Office contract number DAAL03-89-C-0038 with the Army High Performance Computing Research Center at the University of Minnesota.

Liou and Tezduyar [3], Mittal et al. [6,10] and Hansbo and Szepessy [11].

The space–time finite element formulation with the GLS stabilization has recently been used for various problems with fixed spatial domains. These authors are most familiar with the work of Shakib [9], Hansbo and Szepessy [11], Hughes et al. [12], and Hughes and Hulbert [13]. The fundamentals of the space–time formulation, its implementation, and the associated stability and accuracy analysis can be found in these references. In the space–time formulation, the finite element discretization is applied not only spatially but also temporally. Consequently, the deformation of the spatial domain is taken into account automatically. This feature of the stabilized space–time formulation was first exploited by Tezduyar et al. [1,2]. They introduced the DSD/ST (deforming-spatial-domain/space–time) procedure and applied it 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 frequency of remeshing is minimized. 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. Minimizing the frequency of remeshing also results in an increase in the parallelization potential of the computations.

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 [3] by using the generalized minimal residual (GMRES) [14] iteration algorithm with the clustered element-by-element (CEBE) preconditioners.

The CEBE preconditioning is a generalized version of the standard element-by-element (EBE) preconditioning. The EBE preconditioners, which were first introduced by Hughes, Levit and Winget [15] and Hughes, Winget, Levit and Tezduyar [16], are defined as sequential products of element level matrices. The iterative computations with EBE preconditioners are performed in an element-by-element fashion and are highly vectorizable (see Hughes and Ferencz [17]). In the CEBE preconditioning the elements are partitioned into clusters of elements, with a desired number of elements in each cluster, and the iterations are performed in a cluster-by-cluster fashion. The number of clusters should be viewed as an optimization parameter to minimize the computational cost (both memory and CPU time). By specifying the number of clusters, one can select an algorithm anywhere in the spectrum of algorithms ranging from the direct solution technique (when the number of clusters is one) to the standard element-by-element method (when the number of clusters is same as the number of elements). Parallel implementation of the CEBE preconditioning is very similar to that of the grouped element-by-element (GEBE) [18,19] preconditioning.

Computation of time-dependent incompressible flow problems on fixed spatial domains can also be performed by using the finite element discretization in space only, rather than in both space and time. In this case first the GLS stabilization for the steady-state equations of incompressible flows is considered. Then in the definition of the stabilizing terms, the residual of the steady-state equations is replaced with the ones for the time-dependent equations. These stabilizing terms are added to the Galerkin formulation of the time-dependent equations. Furthermore, if, at the element interiors, the contribution to the weighting function from the viscous terms is neglected (it is identically zero for linear velocity interpolation) one obtains a formulation with the combination of SUPG (streamline-upwind/Petrov–Galerkin) and PSPG (pressure-stabilizing/Petrov–Galerkin) stabilizations.

The SUPG formulation, which prevents the numerical oscillations caused by the presence of dominant advection terms, was introduced by

Hughes and Brooks [20]. A comprehensive description of the formulation, together with various numerical examples, can be found in Brooks and Hughes [21]. For hyperbolic systems in general, and compressible Euler equations in particular, the SUPG stabilization was first reported by Tezduyar and Hughes [22]. The implementation of the SUPG formulation in Brooks and Hughes [21] 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. [23].

It was shown that (see Brezzi and Pitkaranta [24], and Hughes et al. [25]), with proper stabilization, elements which do not satisfy the Brezzi condition can still be used for Stokes flow problems. The Petrov–Galerkin stabilization proposed in Hughes et al. [25] is achieved, just like in the SUPG stabilization, by adding to the Galerkin formulation a series of integrals over element domains. The PSPG stabilization proposed in Tezduyar et al. [26] is a generalization, to finite Reynolds number flows, of the Petrov–Galerkin stabilization proposed in Hughes et al. [25] for Stokes flows. In Tezduyar et al., [26], 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. With the T6 scheme, on the other hand, the SUPG stabilization is applied only to the steps involving the advective terms, while the PSPG stabilization is applied only to the steps involving the pressure terms. Both schemes were implemented in Tezduyar et al. [26] based on the Q1Q1 (Bilinear velocity and pressure) and P1P1 (linear velocity and pressure) elements, and were successfully applied to a set of nearly standard test problems.

In the SUPG, PSPG and GLS stabilizations the stabilizing terms added involve the residual of the momentum equation as a factor. Consequently, 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, PSPG and GLS stabilizations that numerical oscillations are prevented without introducing excessive numerical diffusion (i.e. without "over-stabilizing") and therefore without compromising the accuracy of the solution.

Results are presented for applications to certain unsteady flows including those involving moving boundaries and interfaces, such as large-amplitude sloshing, liquid drops and flows past oscillating cylinders. In the flows involving oscillating cylinders, either the motion is prescribed or it needs to be determined as part of the solution.

2. The governing equations of unsteady incompressible flows

Let $\Omega_t \in \mathbb{R}^{n_{sd}}$ 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:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) - \nabla\cdot\boldsymbol{\sigma} = \boldsymbol{0}, \quad \text{on } \Omega_t \forall t \in (0, T \tag{1}$$

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

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

$$\boldsymbol{\sigma}(p, \boldsymbol{u}) = -p\boldsymbol{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u}), \tag{3}$$

with

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \tfrac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}\right) \tag{4}$$

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

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } (\Gamma_t)_g \ \forall t \in (0, T) \tag{5}$$

The "natural" boundary conditions associated with (1) are the conditions on the stress components, and these are the conditions assumed to be imposed at the remaining part of the boundary:

$$\boldsymbol{n}\cdot\boldsymbol{\sigma}=\boldsymbol{h} \quad \text{on } (\Gamma_t)_h \ \forall t\in(0\ T). \tag{6}$$

The homogeneous version of (6), which corresponds to the "traction-free" (i.e. zero normal and shear stress) conditions, is often imposed at the outflow boundaries. As initial condition, a divergence-free velocity field $\boldsymbol{u}_0(\boldsymbol{x})$ is specified over the domain Ω, at $t=0$:

$$\boldsymbol{u}(\boldsymbol{x},0)=\boldsymbol{u}_0(\boldsymbol{x}) \quad \text{on } \Omega_0 \tag{7}$$

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

The kinematical conditions at the interface $(\Gamma_t)_{AB}$ 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:

$$\boldsymbol{n}_A\cdot\boldsymbol{\sigma}_A+\boldsymbol{n}_B\cdot\boldsymbol{\sigma}_B=\boldsymbol{n}_A\gamma/R_A \quad \text{on } (\Gamma_t)_{AB}\ \forall t\in(0,T), \tag{8}$$

where $\boldsymbol{n}_A$ and $\boldsymbol{n}_B$ 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 $\boldsymbol{n}_A$ points towards the center of curvature. The condition (8) is applicable also to 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.

3. The stabilized space–time finite element formulation

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 $0=t_0<t_1<\cdots<t_N=T$. It was first shown by Tezduyar et al. [1,2] that the stabilized space–time finite element formulation can be effectively applied to fluid dynamics computations involving moving boundaries and interfaces. In this formulation the spatial domains at various time levels are allowed to vary. We let $\Omega_n=\Omega_{t_n}$ and $\Gamma_n=\Gamma_{t_n}$, and define the space–time slab Q_n as the space–time domain enclosed by the surfaces Ω_n, Ω_{n+1} and P_n (see fig. 1). Here P_n, the lateral surface of Q_n, is the surface described by the boundary Γ, as t traverses I_n. Similar to the way it was represented by eqs. (5) and (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

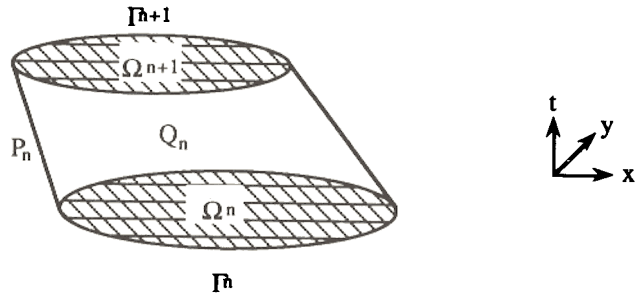

Fig. 1. The space–time slab for the DSD/ST formulation.

interpolation function spaces for the velocity and pressure:

$$(S_u^h)_n = \left\{ u^h \mid u^h \in \left[H^{1h}(Q_n) \right]^{n_{sd}} \right.$$
$$\left. u^h = g^h \text{ on } (P_n)_g \right\}, \tag{9}$$

$$(V_u^h)_n = \left\{ w^h \mid w^h \in \left[H^{1h}(Q_n) \right]^{n_{sd}} \right.$$
$$\left. w^h = 0 \text{ on } (P_n)_g \right\}, \tag{10}$$

$$(S_p^h)_n = (V_p^h)_n = \{ q^h \mid q^h \in H^{1h}(Q_n) \}. \tag{11}$$

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 (1)–(8) can be written as follows: start with

$$(u^h)_0^- = (u_0)^h; \tag{12}$$

sequentially for $Q_1, Q_2, \ldots, Q_{N-1}$, given $(u^h)_n^-$, find $u^h \in (S_u^h)_n$ and $p^h \in (S_p^h)_n$, such that

$\forall w^h \in (V_u^h)_n$ and $\forall q^h \in (V_p^h)_n$,

$$\int_{Q_n} w^h \cdot \rho \left(\frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h \right) dQ$$
$$+ \int_{Q_n} \varepsilon(w^h) : \sigma(p^h, u^h)\, dQ$$
$$\int_{(P_n)_h} w^h \cdot h\, dP - \int_{(P_n)_{AB}} w^h \cdot n_A \gamma / R_A\, dP$$
$$+ \int_{Q_n} q^h \nabla \cdot u^h\, dQ$$
$$+ \int_{\Omega} (w^h)_n^+ \cdot \rho \left((u^h)_n^+ - (u^h)_n^- \right) d\Omega$$
$$+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau \frac{1}{\rho} \left[\rho \left(\frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) \right.$$
$$\left. - \nabla \cdot \sigma(q^h, w^h) \right] \cdot \left[\rho \left(\frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h \right) \right.$$
$$\left. - \nabla \cdot \sigma(p^h, u^h) \right] dQ = 0, \tag{13}$$

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}).

In the variational formulation given by (13), the following notation is being used:

$$(u^h)_n^{\pm} = \lim_{\delta \to 0} u^h(t_n \pm \delta), \tag{14}$$

$$\int_{Q_n} (\quad) dQ = \int_{t_n} \int_{\Omega} (\quad) d\Omega\, dt$$

$$\int_{P_n} (\quad) dP = \int_{I_n} \int_{\Gamma} (\quad) d\Gamma\, dt \tag{16}$$

Remarks

1. If we were to consider a spatial finite element discretization, rather than a space–time one, the Galerkin formulation of (1)–(8) would have consisted of the first five integrals (their spatial versions of course) appearing in equation (13). In the space–time formulation, because the interpolation functions are discontinuous in time, the sixth integral in eq. (13) enforces, weakly, the temporal continuity of the velocity field. The remaining series of integrals in eq. (13) 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) and the pressure-stabilizing/Petrov–Galerkin (PSPG) procedure employed for incompressible flows [26]. It is with such stabilization procedures that it is possible to use elements which have equal-order interpolation functions for velocity and pressure, and which 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. (13).

The coefficient τ used in this formulation is obtained by a simple multi-dimensional general-

ization of the optimal τ given in Shakib [9] for one-dimensional space–time formulation. The expression for the τ used in this formulation is

$$\tau = \left(\left(\frac{2\|u^h\|}{h} \right)^2 + \left(\frac{4\nu}{h^2} \right)^2 \right)^{-1/2} \quad (17)$$

where ν is the kinematic viscosity, and h is the spatial element length. For derivation of τ for higher-order elements see Franca et al. [27].

Remarks

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 which are piecewise linear in time) of a problem is approximately four times more compared to 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 (see section 4).

5. 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 byproduct 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. [2]. We use the same mesh moving scheme for all the results involving flow past oscillating cylinders to be presented in this article.

4. The clustered element-by-element (CEBE) method

It was pointed out before, in remark 4, that the memory needed for the global matrices involved in the space–time method is quite substantial. It was shown in Liou and Tezduyar [3] that the clustered element-by-element (CEBE) preconditioners, together with the generalized minimal residual (GMRES) method [14] can be effectively used to reduce the associated cost significantly. In this section we review the clustered element-by-element method.

After linearization of the fully discretized equations, the following systems needs to be solved for the nodal values of the unknowns:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \quad (18)$$

We rewrite (18) in a scaled form

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}, \quad (19)$$

where

$$\tilde{\boldsymbol{A}} = \boldsymbol{W}^{-1/2}\boldsymbol{A}\boldsymbol{W}^{-1/2}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{W}^{1/2}\boldsymbol{x},$$

$$\tilde{\boldsymbol{b}} = \boldsymbol{W}^{-1/2}\boldsymbol{b}.$$

The scaling matrix $\boldsymbol{W}$ is defined as

$$\boldsymbol{W} = \text{diag } \boldsymbol{A}$$

With this definition of $\boldsymbol{W}$, diag $\tilde{\boldsymbol{A}}$ becomes an identity matrix.

For the formulations presented in this article, the matrix $\boldsymbol{A}$ is not in general symmetric and positive-definite. Therefore, the proposed CEBE preconditioner will be used in conjunction with the GMRES (generalized minimum residual) method; an outline of the GMRES method used is given below.

0. Set the iteration counter $m = 0$, and start with an initial guess $\tilde{\mathbf{x}}_0$,

i. Calculate the residual scaled with the preconditioner matrix $\tilde{P}$:

$$\tilde{r}_m = \tilde{P}^{-1}\left(\tilde{A}\tilde{x}_m - \tilde{b}\right). \tag{24}$$

ii. Construct the Krylov vector space:

$$e^{(1)} = \tilde{r}_m / \|\tilde{r}_m\|, \tag{25}$$

$$f^{(j)} = \tilde{P}^{-1}\tilde{A}e^{(j-1)} - \sum_{i=1}^{j-1} (\tilde{P}^{-1}\tilde{A}e^{(j-1)}, e^{(i)})e^{(i)},$$

$$2 \le j \le k, \tag{26}$$

$$e^{(j)} = f^{(j)} / \| f^{(j)} \|, \tag{27}$$

where k is the dimension of the Krylov space and $e^{(i)}$, $i = 1, 2, \ldots, k$, are the basis vectors.

iii. Update the unknown vector:

$$\tilde{x}_{m+1} = \tilde{x}_m + \sum_{j=1}^{k} s_j e^{(j)}, \tag{28}$$

where $s = \{s_j\}$ is the solution of the equation system

$$Qs = z, \tag{29}$$

with

$$Q = \left[\left(\tilde{P}^{-1}\tilde{A}e^{(i)}, \tilde{P}^{-1}\tilde{A}e^{(j)} \right) \right], \quad 1 \le i, j \le k, \tag{30}$$

$$z = \left\{ \left(\tilde{P}^{-1}\tilde{A}e^{(i)}, -\tilde{r}_m \right) \right\}, \quad 1 \le i \le k \tag{31}$$

iv. For next iteration, set $m \leftarrow m + 1$ and goto i: The iterations continue until $\|\tilde{r}_m\|$ falls below a predetermined value. It should be noted that the matrix Q is symmetric and positive-definite.

Remark

6. The convergence rate of this algorithm depends on the condition number of the matrix $\tilde{P}^{-1}\tilde{A}$. Therefore one would like to select a preconditioner that involves minimal inversion cost, and provides, within cost limitations, an optimal representation of $\tilde{A}$.

Let ϵ denote the set of all elements resulting from the finite element discretization of the computational domain Ω into subdomains Ω^e, $e = 1, 2, \ldots, h_{el}$, where n_{el} is the number of elements. The clustered element-by-element method is based on the clustering of the elements set ϵ into ϵ_J, $J = 1, 2, \ldots N_{cl}$, where N_{cl} is the number of the clusters, such that

$$\epsilon = \bigcup_{J=1}^{N_{cl}} \epsilon_J, \tag{32}$$

$$\varnothing = \bigcap^{N_{cl}} \epsilon_J. \tag{33}$$

The global coefficient matrix $\tilde{A}$ is expressed as

$$\tilde{A} = \sum_{J=1}^{N_{cl}} \tilde{A}_J, \tag{34}$$

with the cluster matrix $\tilde{A}_J$ defined as

$$\tilde{A}_J = \sum_{e \in \epsilon_J} \tilde{A}^e, \tag{35}$$

where $\tilde{A}^e$ is the element level matrix.

Consider the factorization of the matrix $(I + \tilde{B}_J)$:

$$\left(I + \tilde{B}_J\right) = \hat{L}_J \hat{U}_J, \quad J = 1, 2, \ldots, N_{cl}, \tag{36}$$

where

$$\tilde{B}_J = \tilde{A}_J - \tilde{W}_J, \quad J = 1, 2, \ldots, N_{cl}, \tag{37}$$

and $\hat{L}_J$ and $\hat{U}_J$ are the lower and upper triangular factors of $(I + \tilde{B}_J)$. The CEBE preconditioner is defined as

$$\tilde{P} = \prod_{J=1}^{N_{cl}} \hat{L}_J \prod_{J=N_{cl}}^{1} \hat{U}_J \tag{38}$$

Remarks

7. The convergence of the algorithm depends on numbering of the clusters but not on the numbering of the elements within each cluster. By treating each cluster as a super-element, we can identify the clustered element-by-element procedure as a generalization of the standard element-by-element method.

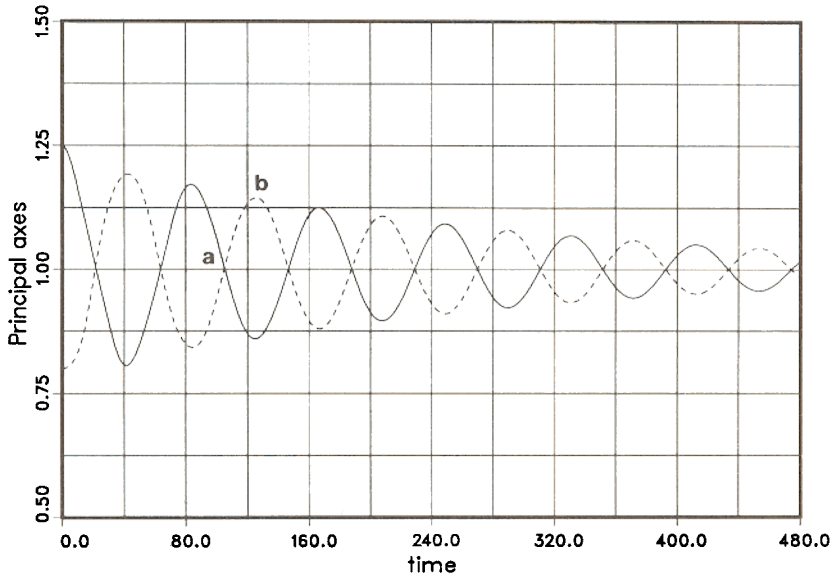

Fig. 2. Pulsating drop: time histories of the axial dimensions of the drop.

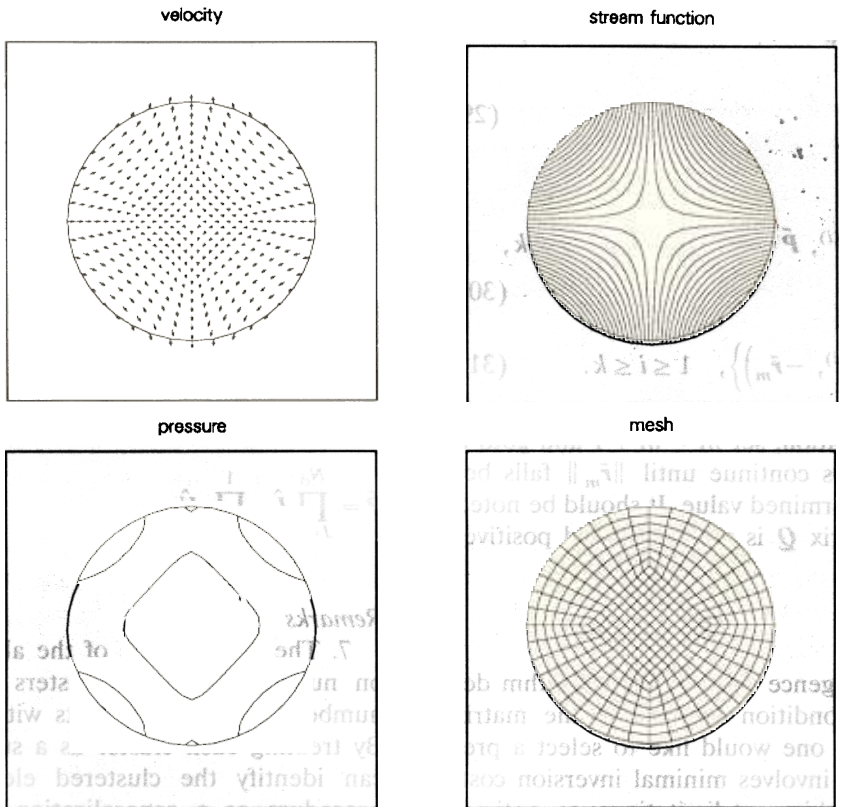

Fig. 3a. Pulsating drop: flow field and finite element mesh corresponding (approximately) to point "a" in fig. 2.

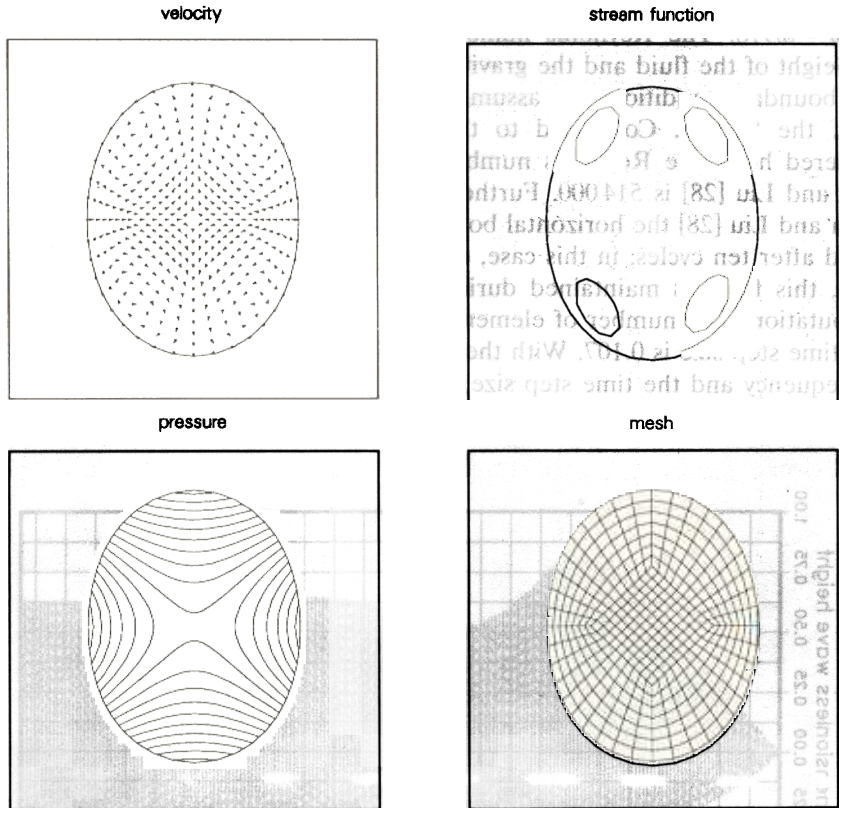

Fig. 3b. Pulsating drop: flow field and finite element mesh corresponding (approximately) to point "b" in fig. 2.

8. We have the option of storing the cluster matrices and their inverses, or recomputing them as they are needed.

9. The method is highly vectorizable and parallelizable.

5. Numerical examples

All solutions using the space–time formulation, presented here, were obtained with linear-in-time interpolation functions. For the details of the computations see Tezduyar et al. [1,2] and Mittal et al. [6].

5.1. 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 2 shows the time histories of the axial dimensions of the drop. Figures 3a and 3b show the field and finite element mesh corresponding, approximately, to points "a", and "b" in fig. 2.

5.2. Large-amplitude sloshing

This problem is similar to the one that was considered in Huerta and Liu [28]. Initially the fluid is stationary and occupies a 2.667×1.0 rectangular region. The density and viscosity are 1.0 and 0.002. 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 Reynolds number used in Huerta and Liu [28] is 514 000. Furthermore, in Huerta and Liu [28] 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 4 shows the time histories 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 5a and 5b show the flow field and finite element mesh corresponding, approximately, to points "a", and "b" in fig. 4.

5.3. Unsteady flows past a circular cylinder

In all cases, the computational values for the cylinder radius and the free-stream velocity are, respectively, 1.0 and 0.125; a time step of 1.0 is

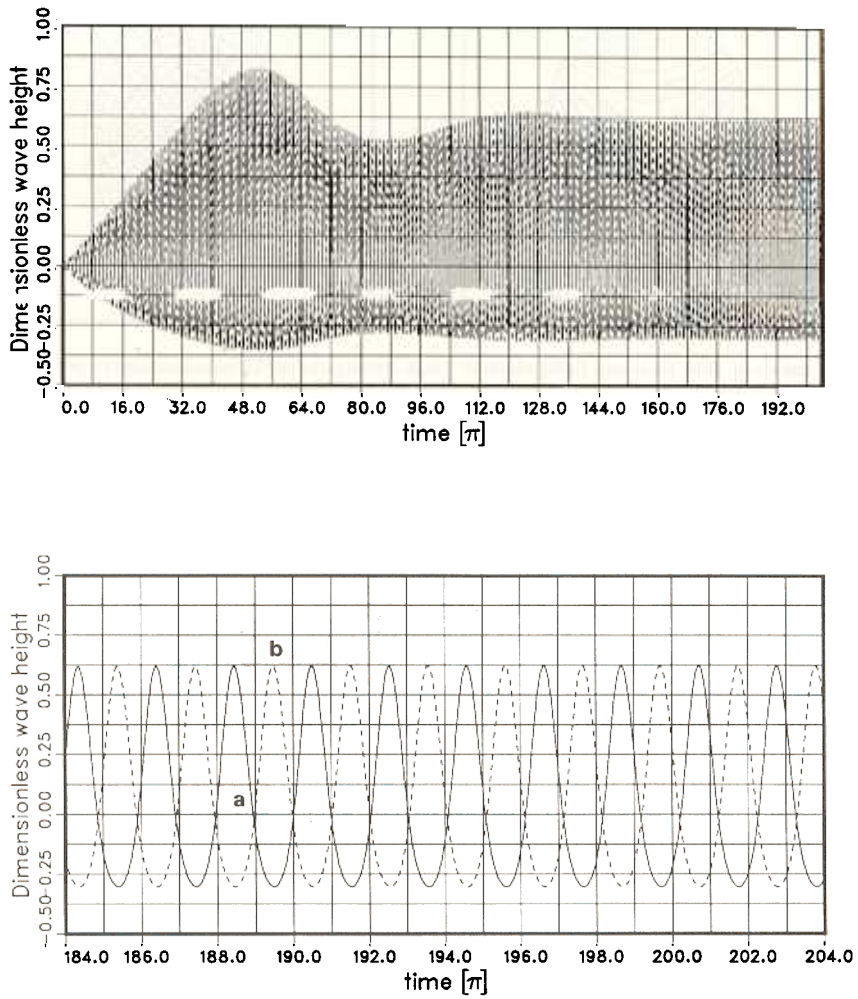

Fig. 4. Large-amplitude sloshing: time histories of the vertical location (relative to the stationary level) of the free-surface along the left- and right-hand-sides of the "tank".

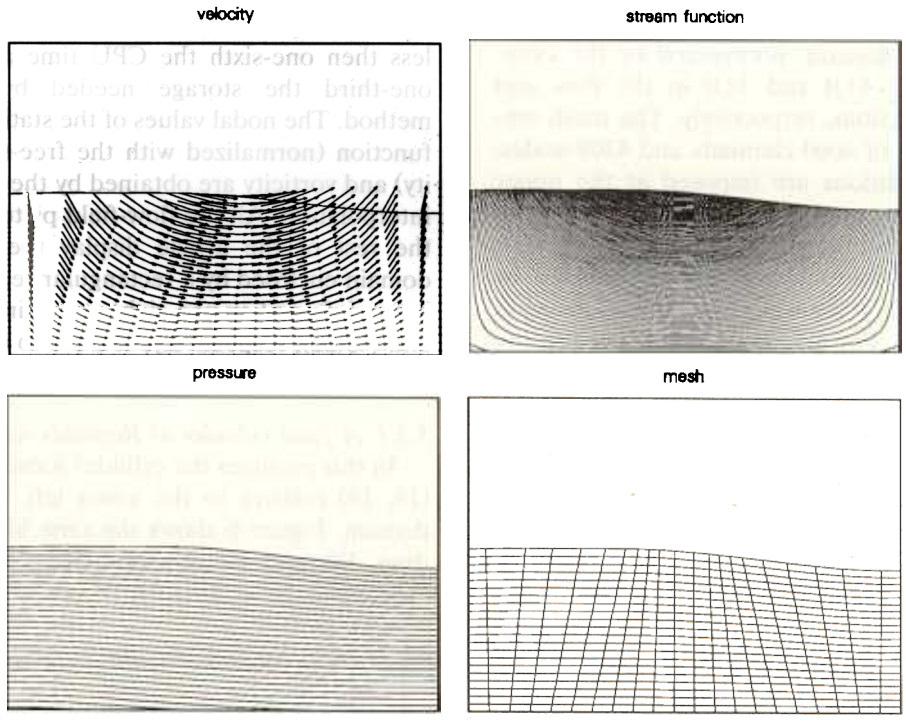

Fig. 5a. Large-amplitude sloshing: flow field and finite element mesh corresponding (approximately) to point "a" in fig. 4.

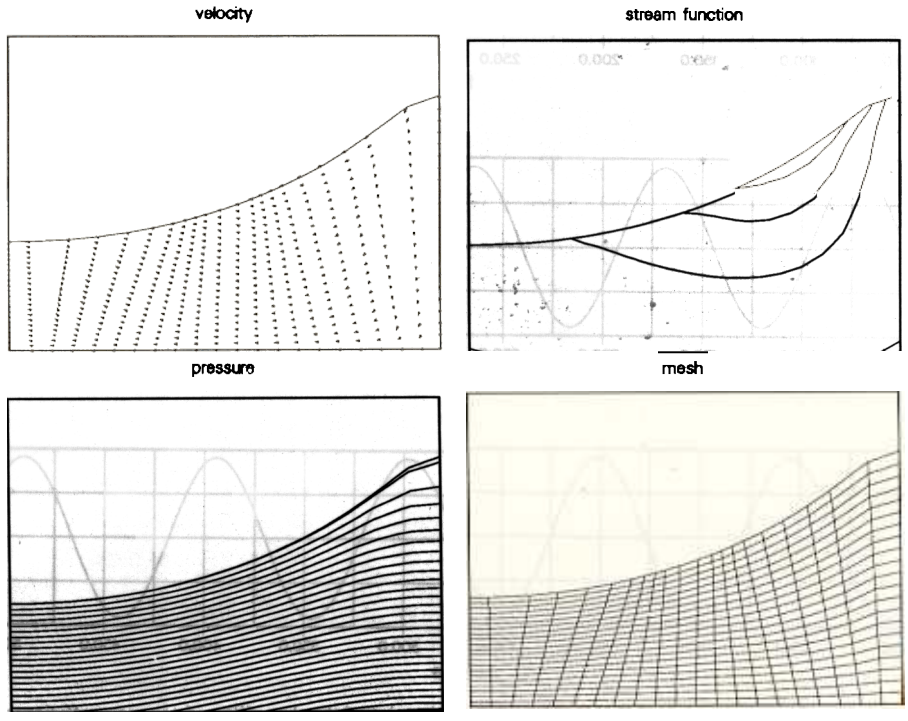

Fig. 5b. Large-amplitude sloshing: flow field and finite element mesh corresponding (approximately) to point "b" in fig. 4.

used for the computations. The dimensions of the computational domain, normalized by the cylinder radius, are 61.0 and 32.0 in the flow and cross-flow directions, respectively. The mesh employed consists of 4060 elements and 4209 nodes. Symmetry conditions are imposed at the upper and lower computational boundaries, and the traction-free condition is imposed at the outflow boundary. The periodic solution is obtained by introducing a short term perturbation to the symmetric solution. For all computations, we use the CEBE iteration method to solve the resulting equation system. At each time step about 25 000 equations are solved simultaneously. We chose a Krylov vector space of dimension 25 and an average cluster size of 23 elements. This cluster size, determined by numerical experimentation, is nearly optimal in minimizing the CPU time. For this set of problems, the CEBE technique takes less then one-sixth the CPU time and less then one-third the storage needed by the direct method. The nodal values of the stationary stream function (normalized with the free-stream velocity) and vorticity are obtained by the least-squares interpolation. All the flow-field pictures shown in the rest of the article display the part of the domain enclosed by a rectangular region, with the lower left and upper right co-ordinates (13, 10) and (43.22), respectively, relative to the lower left corner of the domain.

5.3.1. A fixed cylinder at Reynolds number 100

In this problem the cylinder location is fixed at (16, 16) relative to the lower left corner of the domain. Figure 6 shows the time histories of the drag, lift and torque coefficients for the fixed

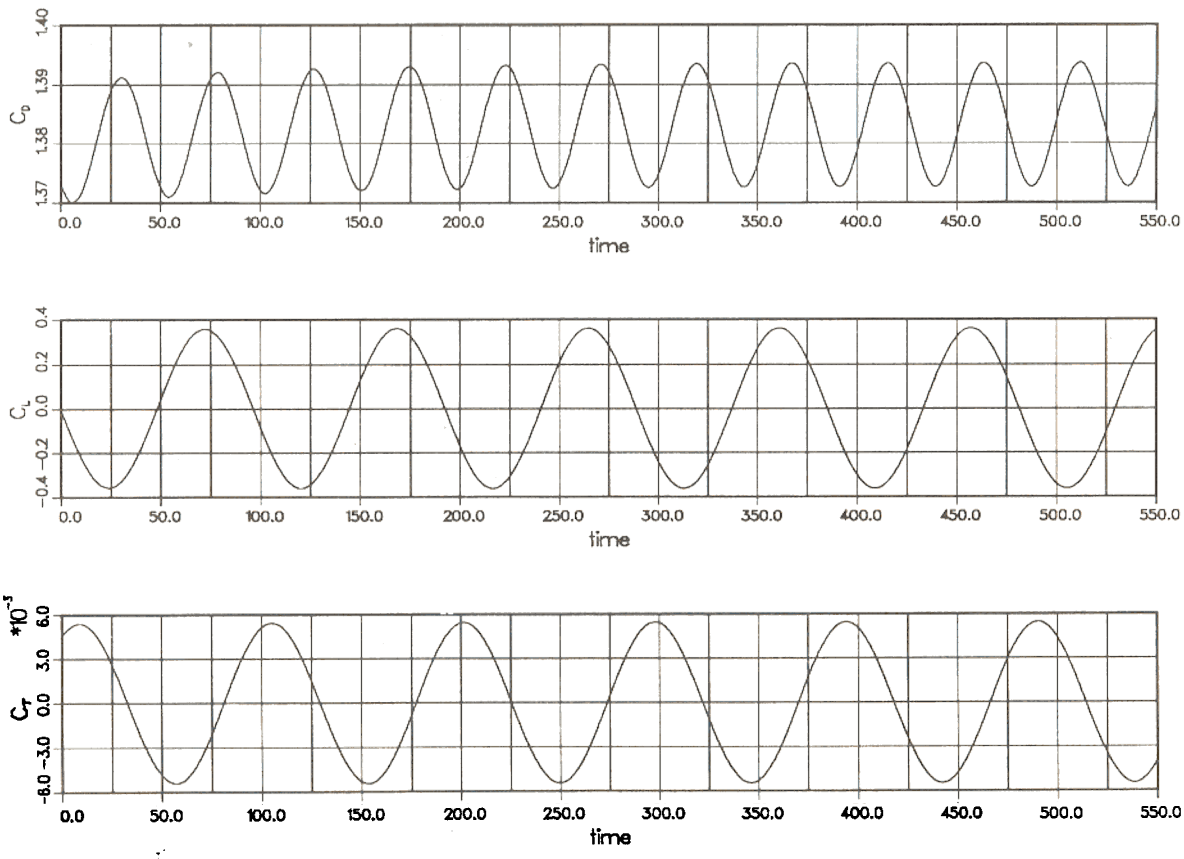

Fig. 6. Flow past a fixed cylinder at Re = 100: time histories of the drag, lift and torque coefficients.

cylinder. The Strouhal number obtained is 0.167. The difference between this value and the ones reported in Tezduyar et al. [26], computed with different formulations and on a finer mesh, is less than 2%. Figure 7 shows a sequence of frames for the vorticity during one period of the lift coefficient. The first and last frames correspond to the trough and crest of the lift coefficient, respectively; the middle frame corresponds to zero lift coefficient. As expected the first and third frames are mirror images of each other.

5.3.2. A cylinder with forced horizontal oscillations at Reynolds number 100

It is well known that at Reynolds number 100, the flow past a fixed circular cylinder leads to the classical unsymmetrical vortex shedding. In such a case the lift and torque coefficients oscillate with a frequency corresponding to the related Strouhal number, while the drag oscillates with twice that frequency.

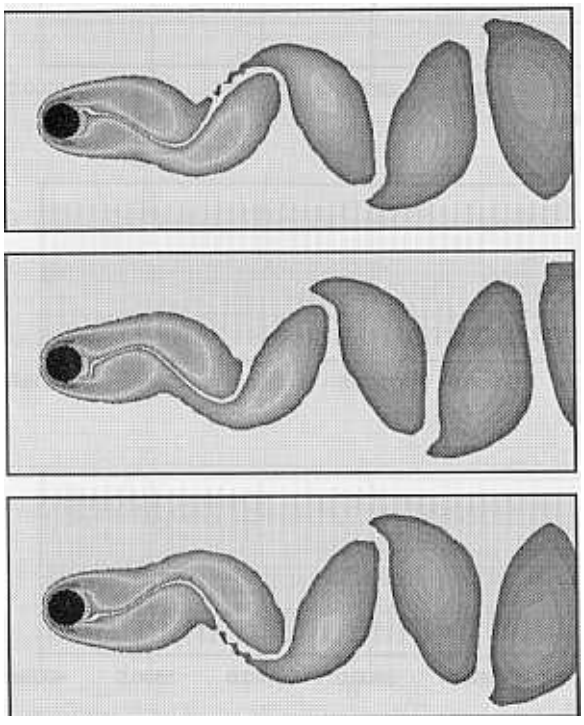

Fig. 7. Flow past a fixed cylinder at Re = 100: vorticity at various instants during one period of the lift coefficient.

The case in which the cylinder is subjected to forced horizontal oscillations shows some very interesting features. Depending on the amplitude and the frequency (f_f) of the forced oscillations of the cylinder, two modes of vortex shedding are possible. This phenomenon, for vortex-induced oscillations, has been discussed in the review papers by King [29] and Sarpkaya [30]. Oscillations with a low reduced frequency ($F_f = 2f_f a/U_\infty$, where a is the radius of the cylinder and U_∞ is the free-stream velocity) lead to unsymmetric modes of vortex shedding. For higher values of F_f, on the other hand, symmetric vortex shedding is observed. However, such a symmetric arrangement of the vortices is unstable, and consequently the vortices coalesce downstream.

We simulate the flow with symmetrical shedding by forcing the cylinder to oscillate horizontally with the following prescribed, displacement (normalized by the cylinder radius):

$$X = -\cos(\omega_f t), \tag{39}$$

where $\omega_f = 2\pi f_f$. For this case, the value of f_f corresponds to a reduced frequency of 0.35. The initial condition for this simulation is prescribed as the unsteady solution for flow past a fixed cylinder at Re = 100 (from the previous example). Figure 8 shows the time histories of the drag, lift and torque coefficients and the normalized horizontal displacement and velocity (normalized by the free-stream velocity) of the cylinder. We observe that the drag coefficient for the horizontally oscillating cylinder is significantly larger than that for a fixed cylinder. Furthermore, the drag coefficient oscillates with a reduced frequency of 0.35 whereas the lift and torque coefficients approach zero. The fact that we start from an unsymmetric solution and still obtain a symmetric mode of shedding demonstrates that this mode is a stable one. Figure 9 shows a sequence of frames for the vorticity during one period of the cylinder motion. During each period two pairs of symmetric vortices are shed from the cylinder. The first and last frames correspond, respectively, to the left and right extreme positions of the cylinder, while the middle frame corresponds to the mean cylinder location.

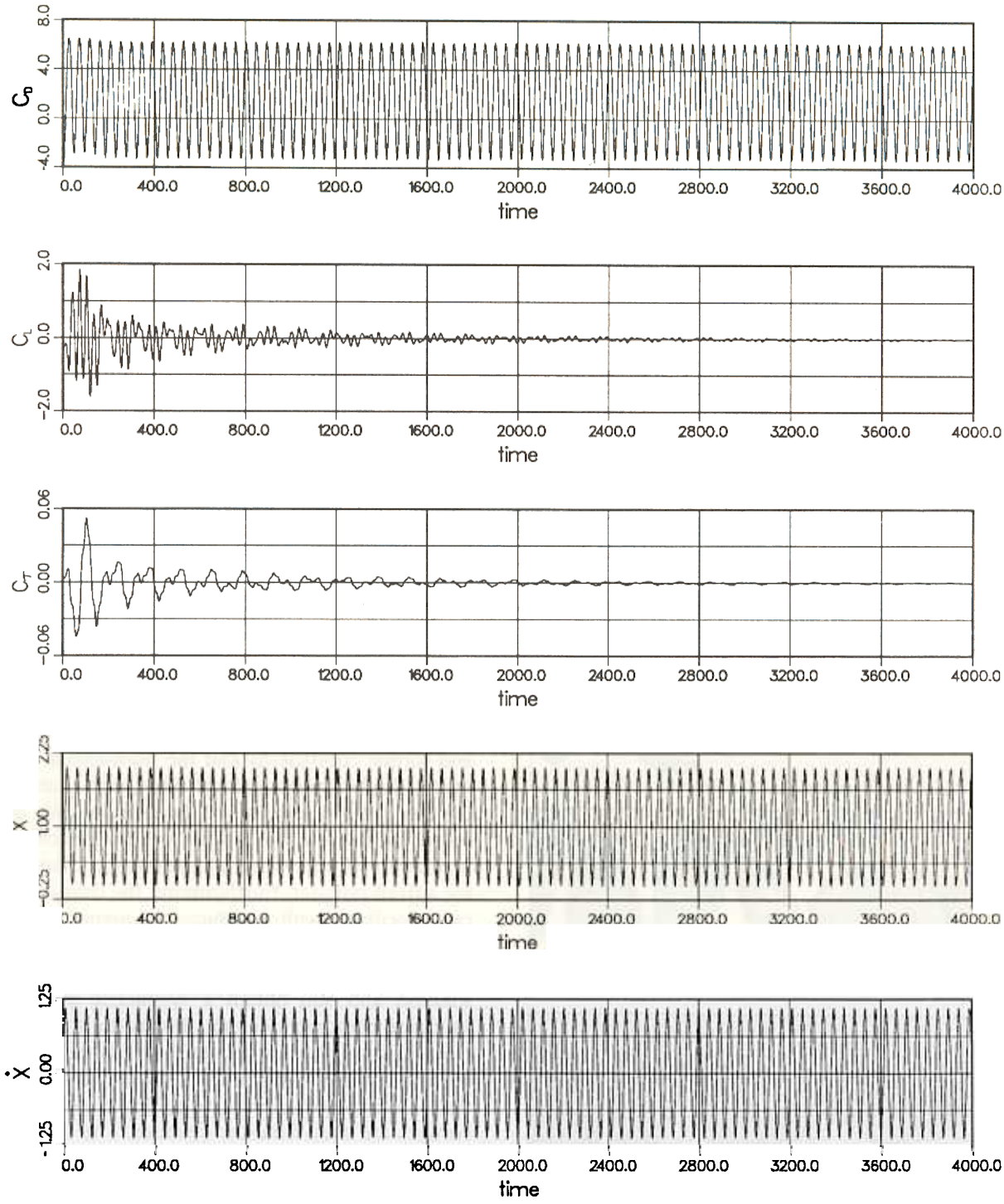

Fig. 8. Flow past a horizontally oscillating cylinder at Re = 100: time histories of the drag, lift and torque coefficients and the normalized displacement and velocity of the cylinder.

5.3.3. A cylinder with vortex-induced vertical oscillations at Reynolds number 324

In the first numerical example we observed that for sufficiently high Reynolds numbers (> 40) flow past a fixed cylinder leads to unsymmetric vortex shedding. This causes the cylinder to experience alternating lift force at a frequency corresponding to the Strouhal number for that Reynolds number. Now, if the cylinder if mounted on a flexible support, then under certain conditions it can undergo sustained oscillations with a frequency close to, or coincident with, its natural frequency. These oscillations can alter the vortex shedding mechanism which in turn can change the cylinder response and so on. This leads to a complex non-linear fluid-structure interaction phenomenon and has been addressed by several researchers [29–32]. We simulate this phenomenon for a cylinder which is allowed to move only in the vertical direction. The motion of the cylinder is governed by the following equation:

$$\ddot{Y} + 2\pi F_n \zeta \dot{Y} + (\pi F_n)^2 Y = \frac{C_L}{M} \tag{40}$$

Here $\ddot{Y}$, $\dot{Y}$ and Y are, respectively, the normalized vertical acceleration, velocity and displacement of the cylinder. The displacement and velocity of the cylinder are normalized by its radius and the free-stream velocity, respectively. M is the non-dimensional mass/unit length of the cylinder, ζ is the structural damping coefficient associated with the system, and C_L denotes the lift coefficient for the cylinder. F_n, the reduced natural frequency of the spring-mass system, is defined as

$$F_n = \frac{2 f_n a}{U_\infty} \tag{41}$$

where f_n is the actual natural frequency of the system. For our problem $F_n = 0.204$, $M = 472.74$ and $\zeta = 3.3 \times 10^{-4}$.

At Reynolds number 324 the reduced natural frequency of the spring mass system and the Strouhal number for flow past a fixed cylinder have very close values. Therefore, we decided to carry out this simulation for Reynolds number 324. The periodic solution for flow past a fixed cylinder at the same Reynolds number is used as the initial condition. Figure 10 shows, for the initial stages of the simulation, time histories of the drag, lift and torque coefficients and the normalized vertical displacement and velocity of the cylinder. We observe that the cylinder oscillates with an increasing amplitude. The drag and torque coefficients for the cylinder also increase while the lift coefficient shows a decreasing amplitude. It is interesting to note that both the mean and peak values of the drag coefficient increase with time, but the trough value remains almost constant. The quantities displayed in fig. 10 are shown in fig. 11 for a later stretch of time when the cylinder reaches a steady-state oscillation amplitude of about one radius. The cylinder oscillates with its natural frequency, and so does the torque coefficient; the drag coefficient oscillates with twice the natural frequency of the

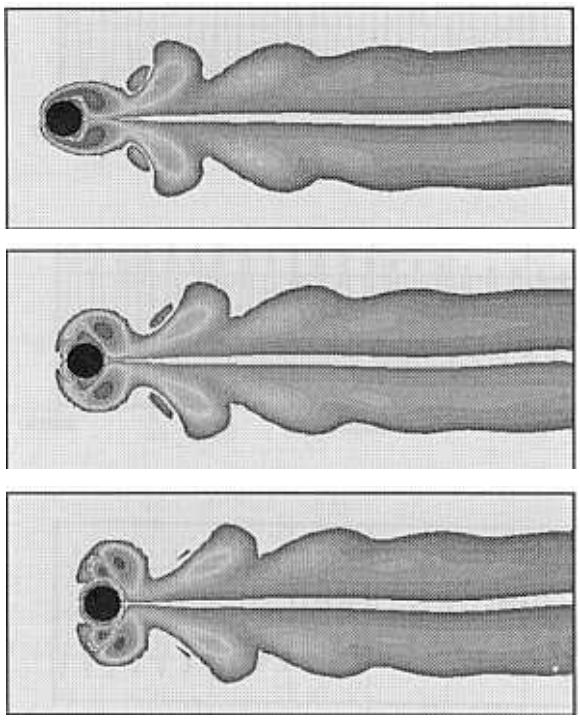

Fig. 9. Flow past a horizontally oscillating cylinder at Re = 100: vorticity at various instants during one period of the cylinder motion.

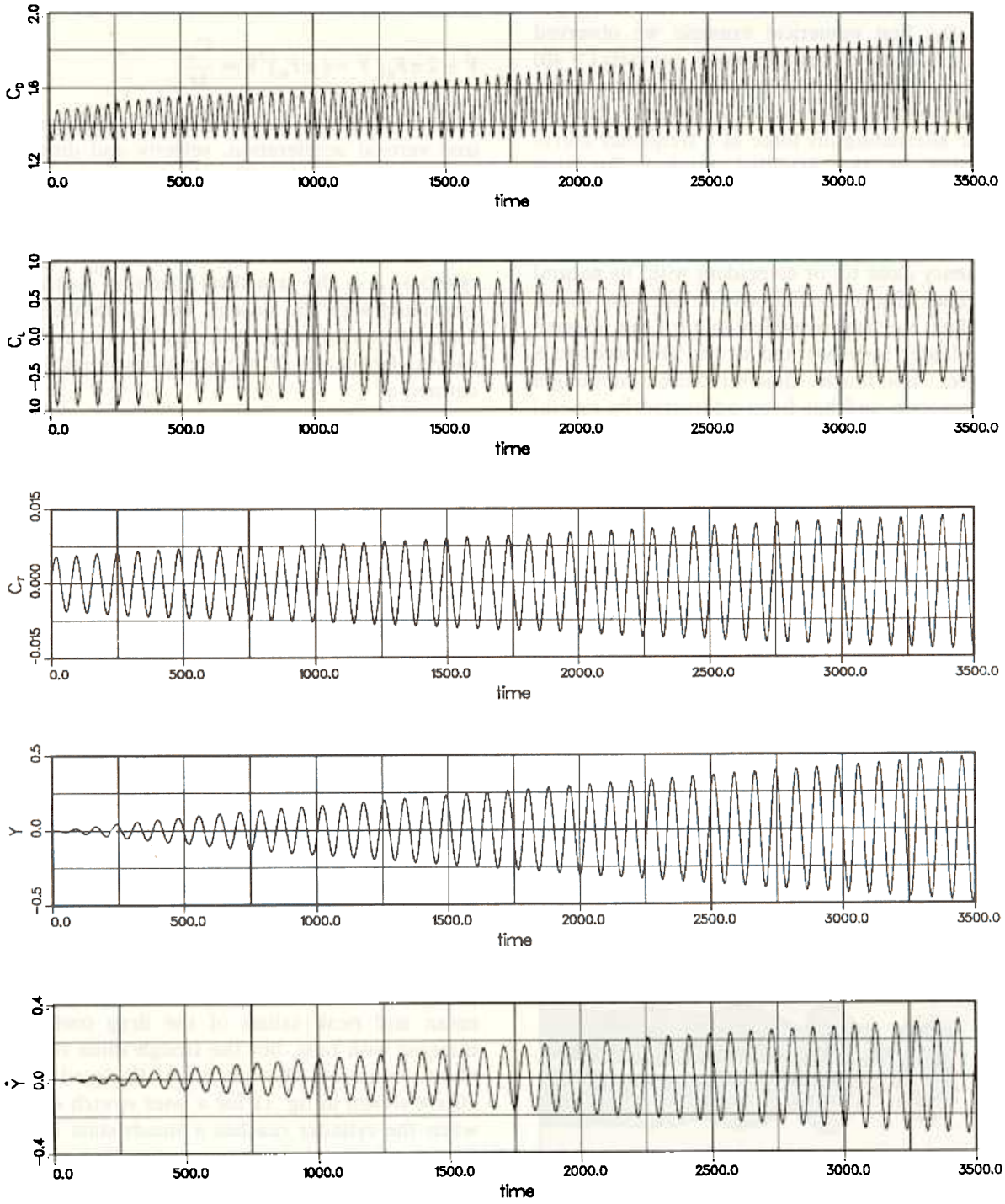

Fig. 10. Flow past a vertically oscillating cylinder at Re = 324: initial time histories of the drag, lift and torque coefficients and the normalized displacement and velocity of the cylinder.

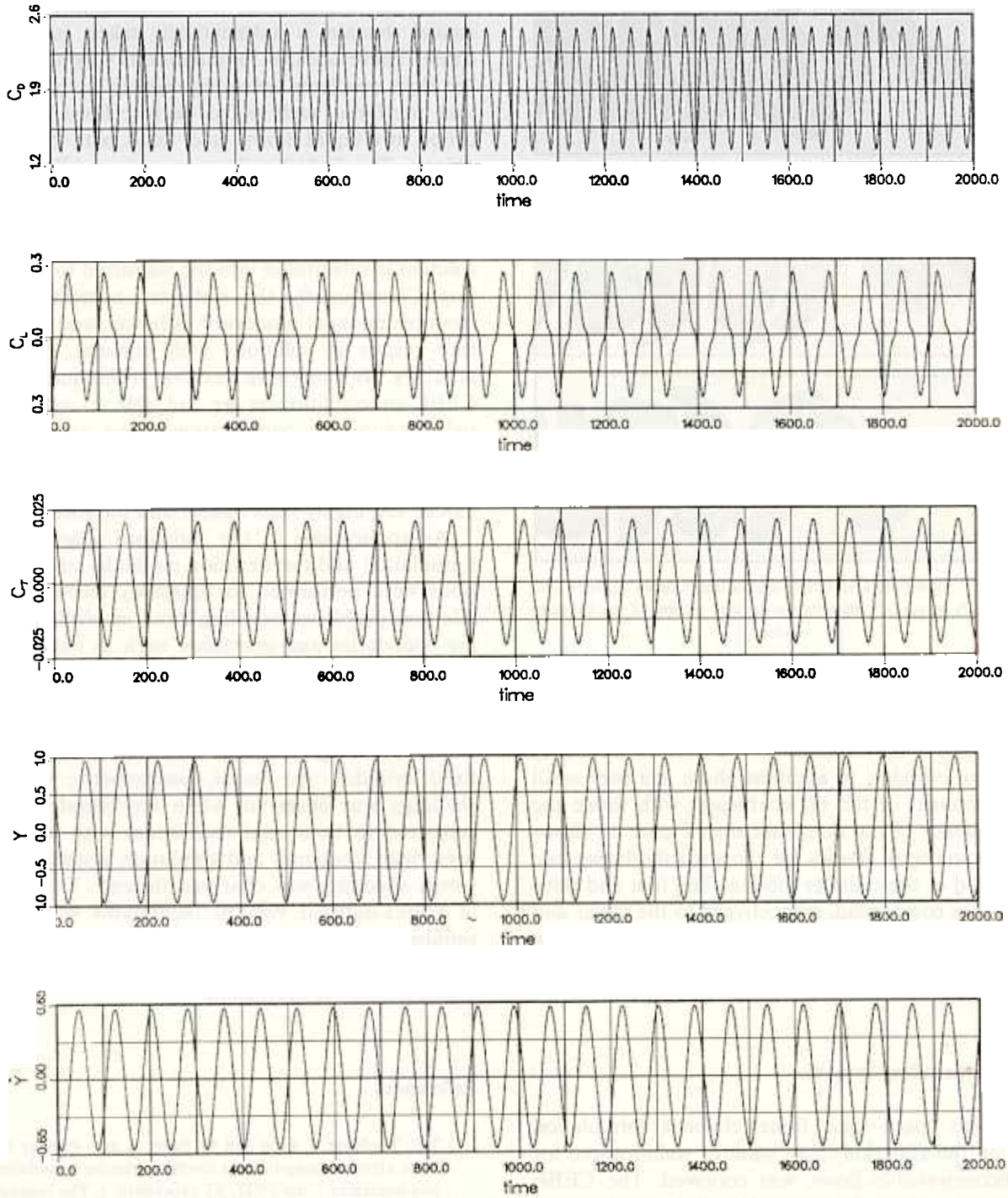

Fig. 11. Flow past a vertically oscillating cylinder at Re = 324: later time histories of the drag, lift and torque coefficients and the normalized displacement and velocity of the cylinder.

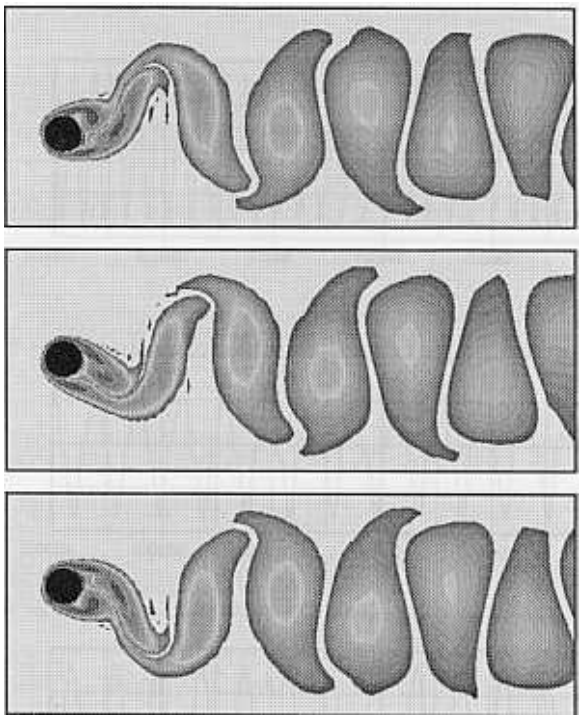

Fig. 12. Flow past a vertically oscillating cylinder at Re = 324: vorticity at various instants during one period of the cylinder motion.

cylinder. The dominant frequency for the lift coefficient corresponds to the natural frequency of the cylinder. In addition, there is a very small component of the lift coefficient with thrice the frequency of the dominant one. Figure 12 shows a sequence of frames for the vorticity during one period of the cylinder motion. The first and third frames correspond, respectively, to the lower and upper extreme positions of the cylinder, while the middle frame corresponds to the mean cylinder location.

6. Concluding remarks

The space–time finite element formulation with the Galerkin/least-squares stabilization, for incompressible flows, was reviewed. The CEBE iteration method employed to solve the equation systems resulting from these space–time finite element discretizations were also reviewed. In the space–time formulation the deformation of the spatial domain is automatically taken into account. Therefore this formulation is very suitable for flow problems involving moving boundaries and interfaces, such as free-surface flows, liquid drops, two-liquid flows, and flows with moving objects. The Galerkin/least-squares stabilization leads to a formulation which is 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. Consequently, this stabilization method introduces minimal numerical diffusion, and therefore results in solutions with minimal loss of accuracy. By employing iteration techniques with CEBE preconditioners we are able to substantially reduce the computational cost associated with solving the fully discretized equations of the space–time formulation. These iteration techniques are highly vectorizable and parallelizable.

As applications of the stabilized space–time formulation and the iteration methods, computations were performed for unsteady incompressible flow problems including those involving moving boundaries and interfaces, such as large-amplitude sloshing, liquid drops and flows past oscillating cylinders. Some interesting physical phenomena were observed as a result of the computations involving cylinders. While for flow past a fixed cylinder, the usual, unsymmetric vortex shedding was observed, when the cylinder was subjected to horizontal oscillations with certain prescribed frequency and amplitude, symmetrical vortex shedding was observed instead. The case of vortex-induced vertical oscillations was also simulated. These oscillations result in an increase in the drag and torque coefficients and a decrease in the lift coefficient.

References

[1] T.E. Tezduyar, J. Liou and M. Behr, A new strategy for finite element computations involving moving boundaries and interfaces – the DSD/ST procedure: I. The concept and the preliminary numerical tests, Comput. Methods Appl. Mech. Eng. 94 (1992) 339.

[2] T.E. Tezduyar, J. Liou, M. Behr and S. Mittal, 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, Comput. Methods Appl. Mech. Eng. 94 (1992) 353.

[3] J. Liou and T.E. Tezduyar, Computation of compressible and incompressible flows with the clustered element-by-element method, Univ. Minnesota Supercomputer Institute Research Report, UMSI 90/215 (October 1990).

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

[5] T.E. Tezduyar and S. Mittal, Finite element computations of incompressible flows, Univ. Minnesota Supercomputer Institute Research Report, UMSI 91/152 (May 1991).

[6] S. Mittal, A. Ratner, D. Hastreiter and T.E. Tezduyar, Space–time finite element computation of incompressible flows with emphasis on flows involving oscillating cylinders, Int. Video J. Eng. Research 1 (1992) 83.

[7] T.J.R. Hughes and L.P. Franca, 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. Methods Appl. Mech. Eng. 65 (1987) 85.

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

[9] F. Shakib, Finite Element Analysis of the Compressible Euler and Navier–Stokes Equations, Ph.D. Thesis, Dept. Mechanical Engineering, Stanford University (1988).

[10] S. Mittal and T.E. Tezduyar, A finite element study of incompressible flows past oscillating cylinders and airfoils, Univ. Minnesota Supercomputer Institute Research Report, UMSI 91/307 (December 1991).

[11] P. Hansbo and A. Szepessy, A velocity–pressure streamline diffusion finite element method for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng., 84 (1990) 175.

[12] T.J.R. Hughes, L.P. Franca and 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. Eng. 63 (1987) 97.

[13] T.J.R. Hughes and G.M. Hulbert, Space–time finite element methods for elastodynamics: formulations and error estimates, Comput. Methods Appl. Mech. Eng. 66 (1988) 339.

[14] Y. Saad and M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856.

[15] T.J.R. Hughes, I. Levit and J. Winget, An element-by-element solution algorithm for problems of structural and solid mechanics, Comput. Methods Appl. Mech. Eng. 36 (1983) 241.

[16] T.J.R. Hughes, J. Winget, I. Levit and T.E. Tezduyar, New alternating direction procedures in finite element analysis based upon EBE approximate factorizations, in: Recent Developments in Computer Methods for Nonlinear Solids and Mechanics, eds. S.N. Atluri and N. Perrone, AMD-Vol. 54 (ASME, New York, 1983) p. 75.

[17] T.J.R. Hughes and R.M. Ferencz, in: First Int. Symp. Domain Decomposition Methods for Partial Differential Equations, eds. R. Glowinski, G.H. Golub, and G.A. Meurant and J. Periaux (SIAM, Philadelphia, PA, 1988) p. 261.

[18] T.E. Tezduyar and J. Liou, Grouped element-by-element iteration schemes for incompressible flow computations, Comput. Phys. Commun. 53 (1989) 441.

[19] T.E. Tezduyar, J. Liou, T. Nguyen and S. Poole, Adaptive implicit–explicit and parallel element-by-element iteration schemes, in: Proc. Second Int. Symp. Domain Decomposition Methods, eds. T.F. Chan, R. Glowinski, J. Periaux and O.B. Widlund, Los Angeles, January 1988 (SIAM, Philadelphia, PA, 1988) p. 443.

[20] T.J.R. Hughes and A.N. Brooks, 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, 1979) p. 19.

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

[22] T.E. Tezduyar and T.J.R. Hughes, 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 (1982).

[23] T.E. Tezduyar, R. Glowinski and J. Liou, Petrov–Galerkin methods on multiply-connected domains for the vorticity-stream function formulation of the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 8 (1984) 1269.

[24] F. Brezzi and J. Pitkaranta, On the stabilization of finite element approximations of Stokes problem, in: Efficient Solutions of Elliptic Systems, Notes on Numerical Fluid Mechanics, Vol. 10 (Vieweg, Braunschweig, 1984) p. 11.

[25] T.J.R. Hughes, L.P. Franca and M. Balestra, 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. Methods Appl. Mech. Eng. 59 (1986) 85.

[26] T.E. Tezduyar, S. Mittal, S.E. Ray and R. Shih, Incom-

pressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity–pressure elements, Comput. Methods Appl. Mech. Eng. 95 (1992) 221.

[27] L.P. Franca, S.L. Frey and T.J.R. Hughes, Stabilized finite element methods: I. Application to the advective–diffusive model, LNCC Research Report No. 0.32/90 (October 1990) Comput. Methods Appl. Mech. Eng., to appear.

[28] A. Huerta and W.K. Liu, Viscous flow with large free surface motion, Comput. Methods Appl. Mech. Eng. 69 (1988) 277.

[29] R. King, A review of vortex shedding research and its application, Ocean Eng. 4 (June 1977) 141.

[30] T. Sarpkaya, Vortex-induced oscillations, J. Appl. Mech. 46 (June 1979) 241.

[31] R.D. Blevins, Flow-Induced Vibration 2nd ed. (Van Nostrand Reinhold, New York, 1990).

[32] O.M. Griffin and G.H. Koopmann, The vortex-excited lift and reaction forces on resonantly vibrating cylinders, J. Sound Vibration 54 (1979) 435.