ELSEVIER

Comput. Methods Appl. Mech. Engrg. 174 (1999) 261–274

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

The Shear-Slip Mesh Update Method

M. Behr, T. Tezduyar*

Army HPC Research Center, Mechanical Engineering and Materials Science, Rice University – MS 321, 6100 Main Street, Houston, TX 77005-1892, USA

Received 27 January 1998; revised 9 March 1998

Abstract

The Shear-Slip Mesh Update Method, designed to handle certain classes of flow problems with moving boundaries and interfaces, is presented. Specifically, we focus on problems with large but regular boundary displacements, such as straight-line translation or rotation. These motions are accommodated by using a thin layer of deforming space–time elements, together with limited remeshing without any projection at space–time slab interfaces. As examples, 2D flow around two counter-rotating squares and 3D flow past a propeller are presented.

1. Introduction

The Deformable-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation was introduced in [1,2] to model flow problems with moving boundaries and interfaces. In this general class of problems, those involving a single translating or rotating object in an unbounded domain can be modeled by attaching the coordinate frame to the object. This way, the finite element mesh can be constructed over this translating or rotating frame of reference. However, if the problem involves another object which is fixed or in relative motion with respect to the first one, this simple approach fails.

An alternative strategy for handling problems with changing spatial domains, involves moving the mesh by employing the mesh moving strategies discussed in [3]. This is the approach most frequently used so far in our DSD/SST computations. Movement of the mesh is sometimes followed by *remeshing*, defined as creating a new finite element mesh with a new node connectivity. Remeshing with automatic mesh generators can be prohibitively expensive in 3D simulations. It also necessitates projection of the solution from the old mesh to the new one, and this generally results in smearing of the features of the solution. In many situations, however, the frequency of remeshing needed to maintain the mesh integrity can be reduced substantially. Sometimes remeshing can be totally eliminated. Many cases falling into this category have been successfully simulated [3] using the DSD/SST approach, including the cases with free-surface flows [4]. The acceptable level of changes in the shape of the spatial domain during the course of these simulations can be characterized as either *small* or *medium*. At such levels remeshing and its associated penalties do not become substantial.

As we briefly reported earlier in [5], with the Shear-Slip Mesh Update Method (SSMUM), the DSD/SST formulation can be very effectively applied to modeling of geometries which undergo *large* but regular deformations, such as straight-line translations or rotations. This is accomplished by letting, at each time step, the elements in a thin zone of the mesh to undergo 'shear' deformation. This zone is remeshed frequently via regeneration of element connectivity. The idea behind the special-purpose mesh design is the matching of the

* Corresponding author.

0045-7825/99/$ – see front matter
PII: S0045-7825(98)00299-0

node positions for the old (deformed) and new (good-quality) meshes, so that projection is not necessary. Since only a small part of the overall connectivity is being regenerated, the computational cost is greatly reduced. Application of this method to a 3D computation involving translation and a structured mesh was presented in [6] in the context of two trains passing each other in a tunnel. Here we focus on spinning geometries and hybrid structured/unstructured meshes.

In the context of finite difference and finite volume methods, the Chimera overset grid methodology [7] is widely used to handle the class of problems we are dealing with here. While it is certainly possible to use this approach with finite element formulations, the technique would have clear disadvantages. The cost of reverse mapping required to find a background cell for each of the overlapping nodes can be high for curvilinear grids. Inherent in the method is projection of the solutions between grids, which suffers from the same diffusion of features as finite element remeshing. However, special grid designs similar to the ones discussed here in finite element context, can circumvent these disadvantages. Such methods would not be viewed as equivalent to the technique presented in this paper, because there is no finite difference or finite volume counterpart of the space–time approach.

In Section 2, we review the Navier–Stokes equations of incompressible flows and describe the turbulence model. The stabilized finite element formulation is presented in Section 3. The SSMUM is discussed in Section 4. In Section 5, we present simulation of 2D flow past two counter-rotating square cylinders and 3D flow past a propeller. We end with concluding remarks in Section 6.

2. Governing equations

The problems under consideration are governed by the time-dependent Navier–Stokes equations of incompressible flows. We consider a viscous, incompressible fluid occupying at an instant $t \in (0, T)$ a bounded region $\Omega_t \subset \mathbb{R}^{n_{sd}}$, with boundary Γ_t, where n_{sd} is the number of space dimensions. In the DSD/SST formulation, the spatial domain may change with respect to time, and the subscript t indicates such time-dependence. The symbols $\boldsymbol{u}(\boldsymbol{x}, t)$ and $p(\boldsymbol{x}, t)$ represent the velocity and pressure. The external forces (e.g. the gravity) are represented by $\boldsymbol{f}(\boldsymbol{x}, t)$. The momentum and mass balance equations can be written as follows:

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

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

where the density ρ is assumed to be constant. The stress tensor $\boldsymbol{\sigma}$ can be decomposed into its isotropic and deviatoric parts:

$$\boldsymbol{\sigma}(\boldsymbol{u}, p) = -p\boldsymbol{I} + \boldsymbol{T}\,. \tag{3}$$

We consider only the Newtonian fluids, for which the deviatoric stress is related linearly to the strain rate tensor:

$$\boldsymbol{T} = 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u})\,, \qquad \boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}})\,, \tag{4}$$

where μ is the dynamic viscosity. The Dirichlet and Neumann-type boundary conditions are represented as

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } (\Gamma_t)_g\,, \tag{5}$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} = \boldsymbol{h} \quad \text{on } (\Gamma_t)_h\,, \tag{6}$$

where $(\Gamma_t)_g$ and $(\Gamma_t)_h$ are complementary subsets of the boundary Γ_t.

The initial condition consists of a divergence-free velocity field specified over the entire domain:

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

Because typical meshes are not able to resolve the flow features well enough to fully capture turbulence effects at high Reynolds numbers, introduction of a turbulence model becomes necessary. In the present study, a

simple Smagorinsky turbulence model is employed. In this model, the kinematic viscosity $\nu = \mu/\rho$ is augmented by an eddy viscosity ν_t which is defined as

$$\nu_t = (Ch)^2(2\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}))^{1/2}, \tag{8}$$

where C is a constant and h is the element length, defined here as the maximum of the edge lengths for the element (see [8] for details and references, and [9] for early definition).

3. Stabilized finite element formulation

The computations presented here are based on the space–time finite element method, in which the weak form of the governing equations is written over the associated space–time domain of the problem. Space–time formulations are naturally suited to problems involving moving boundaries and interfaces, as the deformation of spatial elements is taken into account automatically. In order to construct the finite element function spaces for the space–time method, we partition the time interval $(0, T)$ into subintervals $I_n = (t_n, t_{n+1})$, where t_n and t_{n+1} belong to an ordered series of time levels $0 = t_0 < t_1 < \cdots < t_N = T$. Let $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$. We will define the space–time slab Q_n as the domain enclosed by the surfaces Ω_n, Ω_{n+1}, and P_n, where P_n is the surface described by the boundary Γ_t as t traverses I_n. As it is the case with Γ_t, surface P_n is decomposed into $(P_n)_g$ and $(P_n)_h$ with respect to the type of boundary condition (Dirichlet and Neumann) being applied.

For each space–time slab Q_n, we define the following finite-dimensional trial solution ($(\mathcal{S}_{\boldsymbol{u}}^h)_n$ and $(\mathcal{S}_p^h)_n$) and test function ($(\mathcal{V}_{\boldsymbol{u}}^h)_n$ and $(\mathcal{V}_p^h)_n$) spaces for the velocity and pressure:

$$(\mathcal{S}_{\boldsymbol{u}}^h)_n = \{\boldsymbol{u}^h \mid \boldsymbol{u}^h \in [H^{1h}(Q_n)]^{n_{sd}},\ \boldsymbol{u}^h = \boldsymbol{g}^h \text{ on } (P_n)_g\}, \tag{9}$$

$$(\mathcal{V}_{\boldsymbol{u}}^h)_n = \{\boldsymbol{w}^h \mid \boldsymbol{w}^h \in [H^{1h}(Q_n)]^{n_{sd}},\ \boldsymbol{w}^h = \boldsymbol{0} \text{ on } (P_n)_g\}, \tag{10}$$

$$(\mathcal{S}_p^h)_n = (\mathcal{V}_p^h)_n = \{p^h \mid p^h \in H^{1h}(Q_n)\}, \tag{11}$$

where $H^{1h}(Q_n)$ represents the finite-dimensional function space constructed over the space–time slab Q_n by using first-order polynomials in space and in time. The interpolation function spaces are discontinuous in time and continuous in space.

The stabilized space–time formulation of Eqs. (1) and (2) can then be written as follows: given $(\boldsymbol{u}^h)_n^-$, find $\boldsymbol{u}^h \in (\mathcal{S}_{\boldsymbol{u}}^h)_n$ and $p^h \in (\mathcal{S}_p^h)_n$ such that $\forall\, \boldsymbol{w}^h \in (\mathcal{V}_{\boldsymbol{u}}^h)_n$ and $\forall\, q^h \in (\mathcal{V}_p^h)_n$:

$$\begin{aligned}
&\int_{Q_n} \boldsymbol{w}^h \cdot \rho\left(\frac{\partial \boldsymbol{u}^h}{\partial t} + \boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h - \boldsymbol{f}\right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\boldsymbol{w}^h) : \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h)\, \mathrm{d}Q \\
&\quad + \int_{Q_n} q^h \nabla \cdot \boldsymbol{u}^h \, \mathrm{d}Q + \int_{\Omega_n} (\boldsymbol{w}^h)_n^+ \cdot \rho((\boldsymbol{u}^h)_n^+ - (\boldsymbol{u}^h)_n^-)\, \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{MOM}} \frac{1}{\rho}\left[\rho\left(\frac{\partial \boldsymbol{w}^h}{\partial t} + \boldsymbol{u}^h \cdot \nabla \boldsymbol{w}^h\right) - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{w}^h, q^h)\right] \\
&\quad \cdot \left[\rho\left(\frac{\partial \boldsymbol{u}^h}{\partial t} + \boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h - \boldsymbol{f}\right) - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h)\right] \mathrm{d}Q \\
&\quad + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{CONT}} \nabla \cdot \boldsymbol{w}^h \rho \nabla \cdot \boldsymbol{u}^h \, \mathrm{d}Q = \int_{(P_n)_h} \boldsymbol{w}^h \cdot \boldsymbol{h}^h \, \mathrm{d}P .
\end{aligned} \tag{12}$$

The following notation is being used in Eq. (12):

$$(\boldsymbol{u}^h)_n^{\pm} = \lim_{\varepsilon \to 0} \boldsymbol{u}(t_n \pm \varepsilon), \tag{13}$$

$$\int_{Q_n} \ldots \mathrm{d}Q = \int_{I_n} \int_{\Omega_t^h} \ldots \mathrm{d}\Omega \, \mathrm{d}t, \tag{14}$$

$$\int_{P_n} \ldots \mathrm{d}P = \int_{I_n} \int_{\Gamma_t^h} \ldots \mathrm{d}\Gamma \, \mathrm{d}t \,. \tag{15}$$

The definitions of the stabilization parameters τ_{MOM} and τ_{CONT} can be found in [10].

The solution to Eq. (12) is obtained for all of the space–time slabs $Q_0, Q_1, \ldots, Q_{N-1}$ sequentially, and the computations start with

$$(\boldsymbol{u}^h)_0^- = \boldsymbol{u}_0^h \,. \tag{16}$$

The nonlinear system given by Eq. (12) is solved with the Newton–Raphson method. The large linear system that arises at each iteration of the Newton–Raphson algorithm is solved in turn by using an iterative solution technique with the Generalized Minimum Residual (GMRES) [11] update method.

4. Mesh update method

The strategy presented here is based on special-purpose mesh designs which combine regions of rigid non-deforming elements with layers of shear-absorbing deforming elements. This accommodates the motions consisting of straight line translation and/or rotation. In this approach, a translating object is embedded in a strip (in 2D) or a tube (in 3D) of rigid elements that move 'glued' to that object. Similarly, a rotating object is embedded in a disk of rigid elements which rotate 'glued' to that object. These non-deforming regions are immersed in another set of non-deforming elements spanning the exterior boundaries, as shown in Fig. 1. A layer of shear-absorbing elements connects these moving and non-moving regions. The thickness of this layer can span one or more elements, with multi-element layers requiring more complex implementations but providing greater flexibility. A single-element shear-slip layer is illustrated in Fig. 2 in the context of a 2D spatial domain with a translating object. Shown here are two space–time slabs, in spatial projection in the left picture, and viewed from inside of the shear-slip layer in the right picture. The notation identifying the distinct time levels is the same as the one used in Section 3. The quadrilateral elements on the right, coded blue, belong to the stationary exterior mesh, while the elements on the left, coded purple, belong to the translating mesh which follows the moving object. The direction of the movement is upward in the 2D view, and away from the viewer in the 3D view. Evolution of a single shear-absorbing element during the two time steps illustrated here is highlighted in red. The spatial outline of that element deforms during the first step, and then re-connects to new purple nodes, which moved into proper positions. This process is carried out repeatedly, with changes in element connectivity, but with no projection. Note that the spatial mesh at the lower level of the second time step is of the same quality as the one at the lower level of the first time step.

REMARK 1. Even though re-connecting of the shear-slip elements alters the spatial interpolation of the solution at the slab boundaries, it appears that many important features of the solution are preserved. Notably, the integral of divergence of the 2D velocity field is unchanged after the re-connect illustrated here.

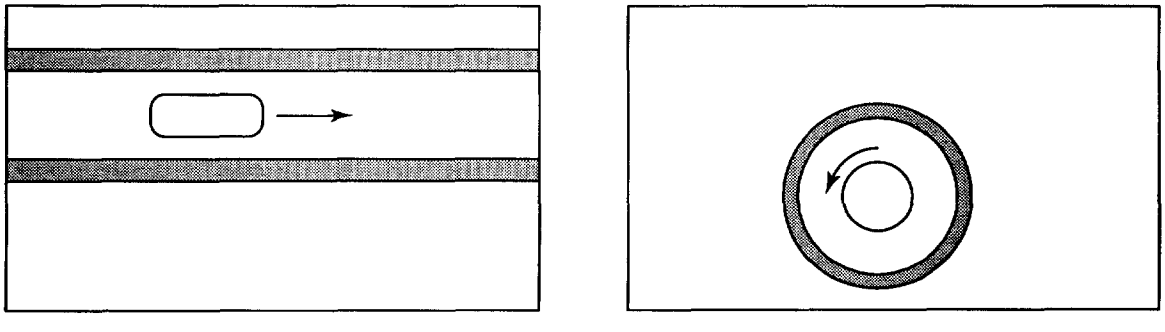

Fig. 1. Sketches of domain decomposition for the SSMUM. Regions of deforming elements are shown in grey.

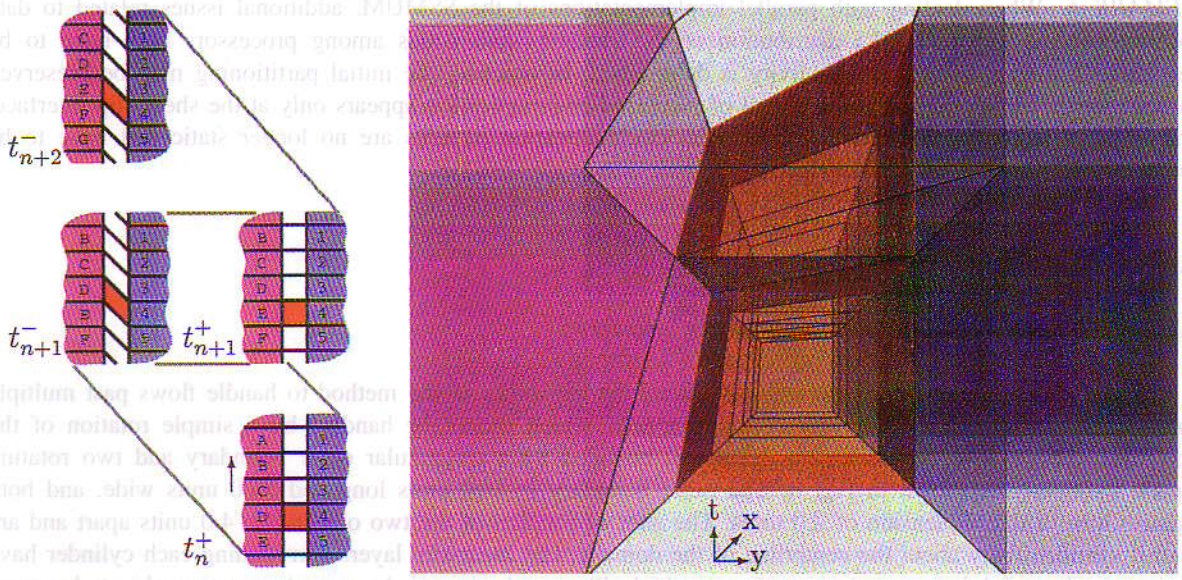

Fig. 2. Shear-absorbing element layer: a 2D view (left) and a space–time view (right). The space–time view shows the two space–time slabs as the left part of the spatial domain translates away from the viewer.

REMARK 2. In practical applications, it is sometimes desirable to place the shear-slip layer as close to the moving boundaries as possible. This happens for example when two or more objects are in motion in proximity of each other. This is greatly simplified by using unstructured meshes for the rigid portions of the domain, as is done in the subsequent examples.

REMARK 3. The SSMUM used in conjunction with triangular or tetrahedral meshes can greatly benefit from judicious design of the initial mesh. It is possible to generate the shear-slip layer mesh in such a way that the sheared mesh at the end of the time step will be of comparable quality as the initial mesh. On the other hand, similar, but improper, designs will lead to excessive mesh deformations. This is illustrated in Fig. 3.

REMARK 4. It may not be necessary to re-connect the elements in the shear-slip layer at every time step. Choosing a smaller time step, for a given mesh resolution and rotational speed, is still possible, leading to lower frequency of re-connects. Conversely, selecting a wider shear-slip layer, more than one element thick, will allow the use of larger time steps. These possibilities, although slightly more complicated in implementation, lend considerable flexibility to the proposed mesh update method.

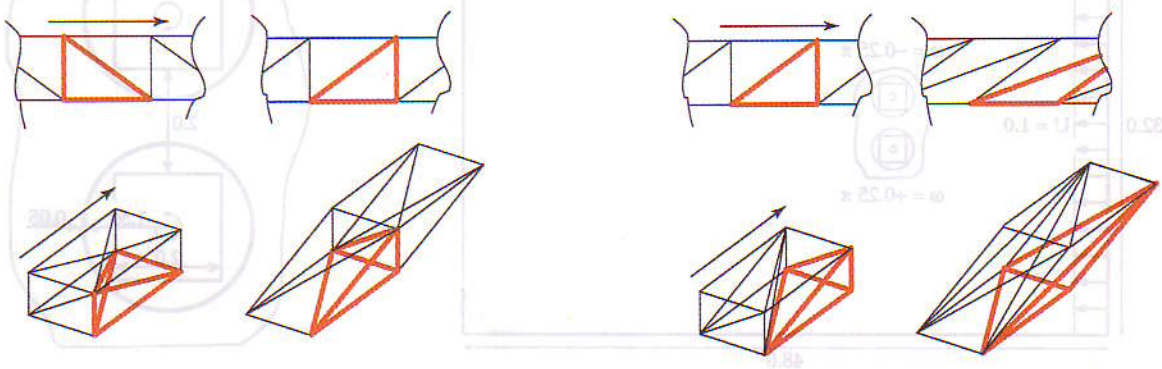

Fig. 3. Initial mesh design and its influence on the quality of the deformed mesh: proper (left) and improper (right) subdivisions.

REMARK 5. When dealing with parallel implementations of the SSMUM, additional issues related to data locality become apparent. The distribution of the elements and nodes among processors may have to be regenerated after new mesh connectivity is determined. In practice, the initial partitioning may be preserved throughout the simulation, as the problem of increased communication appears only at the shear-slip interface, and is often negligible. Nevertheless, the data communication patterns are no longer static and have to be recomputed following each re-connect.

5. Numerical examples

5.1. 2D flow past two counter-rotating square cylinders

We begin with this test computation to illustrate the capability of the method to handle flows past multiple rotating or translating objects, representing situations which cannot be handled by a simple rotation of the coordinate frame. The computational domain is bounded by a rectangular outer boundary and two rotating square cylinders, as shown in Fig. 4. The outer boundary is 48.0 units long and 36.0 units wide, and both squares have a side dimension of 2.0 units. The axes of rotation of the two objects are 4.0 units apart and are placed symmetrically about the centerline of the domain. The shear-slip layers surrounding each cylinder have an inner radius of 1.5 units and are 0.05 units thick. Placing the layers closer to the rotating objects becomes necessary if the gap between the objects becomes smaller. However, based on our numerical experimentations, this does not seem to affect the quality of the solution. The mesh consists of 31 928 space–time nodes and 31 492 triangular elements, and is shown in Fig. 5. Each shear-slip layer is one element thick and has 160 segments in the circumferential direction. Joining the moving and fixed regions of the mesh, it shears during each time step and, at the end of a time step, re-connects to the new nodes belonging to the rotating interior discs. A free-stream velocity of 1.0 is assumed, entering horizontally from the left. The Reynolds number based on the upstream velocity and the size of the squares is 400. Zero normal velocity and zero shear stress conditions are specified at the upper and lower boundaries, and a traction-free condition is imposed at the outflow. The steady state solution at Reynolds number 50 serves as the initial condition for the unsteady flow, which is computed for 3000 time steps with a time step size of 0.05. The top and bottom squares are rotating counter-clockwise and clockwise, respectively, each with a rotational velocity magnitude of 0.25π. Fig. 6 shows, at nine equally spaced instants spanning one full revolution, the vorticity field in the vicinity of the squares. We observe a regular vortex shedding from the corners of the cylinders as they move opposite to the flow direction and enter the gap, and also a lower frequency shedding corresponding to flow past a compound object. The flow through the gap appears to be switching direction bias at irregular intervals, as evidenced by the

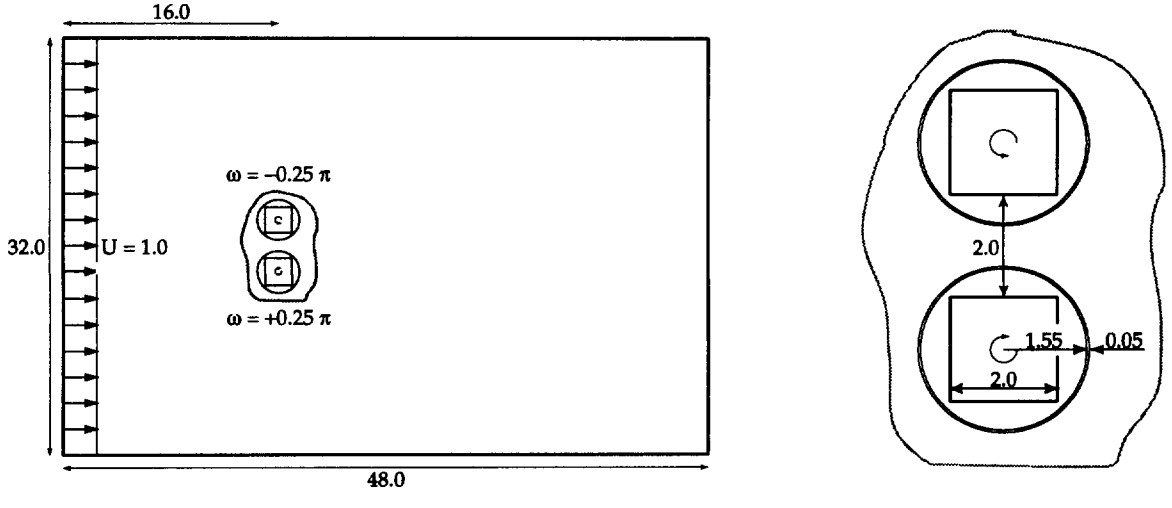

Fig. 4. 2D flow past two counter-rotating square cylinders: computational domain.

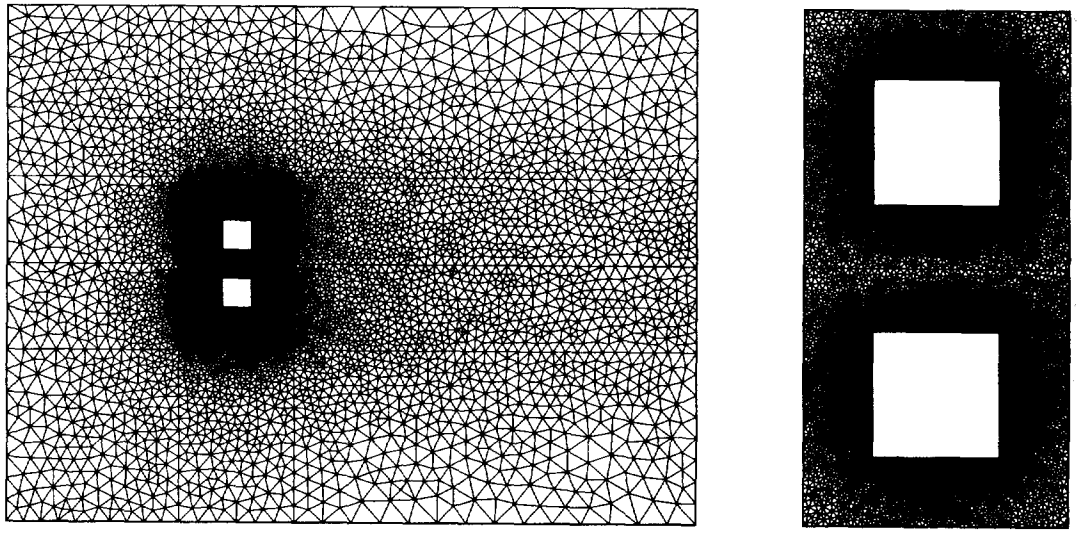

Fig. 5. 2D flow past two counter-rotating square cylinders: initial mesh and close-up.

drag and lift coefficient plots shown in Figs. 7 and 8. This behavior is characteristic of flows past groups of obstacles, e.g. double plates [12] or staggered circular cylinders [13]. One can note that the effect of the layer of deforming elements is hardly visible in the solution.

This computation was carried out on the IBM SP2. At every times step the coupled, nonlinear equations are solved with 4 Newton–Raphson iterations. The coupled, linear equations that need to be solved at each Newton–Raphson iteration are solved also iteratively, with the GMRES update techniques with a Krylov space size of 40.

5.2. 3D flow past a rotating propeller

Extending the method to 3D simulations, we simulate here a high-Reynolds number flow past a rotating propeller. Although a similar simulation can be performed by using a rotating frame of reference and a non-deforming mesh, we view our computation as a first step towards modeling of more complex geometries in which the moving part interacts with other stationary or moving objects.

The computational domain, shown in Fig. 9, has dimensions $50.0 \times 40.0 \times 40.0$. The propeller rotates around the centerline of this domain (which coincides with the x-axis) and consists of a conical afterbody, a spherical tip, and seven blades with radius of 2.915. The overall length of the propeller fairing is 5.0. The shear-slip layer is an axisymmetric, closed shell with interior radii 3.5 and 1.5 and a uniform thickness of 0.1. Its center coincides with the center of the propeller fairing.

The mesh consists of 305 786 space–time nodes and 939 213 tetrahedral elements, and is shown in Fig. 10. The shear-slip layer, which is one element thick, has 64 segments in the circumferential direction and 46 segments in the remaining direction. This layer goes through shear deformation during each time step, and at the end of a time step re-connects to the new nodes belonging to the rotating interior disk. The unstructured meshes in both the inner (rotating) and the outer (stationary) rigid regions of the domain were generated using an automatic mesh generator [14], while the structured mesh which fills the shear-slip region was generated manually. The different layers of the mesh are shown in cross-section in Fig. 11. A free-stream velocity of (0.1, 0.0, 0.0) is imposed at the upstream boundary which is located at 20.0 units from the front of the propeller fairing and 22.5 units from the center of the shear-slip shell. The Reynolds number based on the upstream velocity and the propeller diameter is approximately 1×10^6. A Smagorinsky turbulence model with $C = 0.15$ is employed. Zero normal velocity and zero shear stress conditions are specified at all transverse boundaries, and a traction-free condition is imposed at the outflow. The steady state solution at Reynolds number 30 serves as the initial condition for the unsteady flow, which is computed for 512 time steps with a time step size of 0.1. The

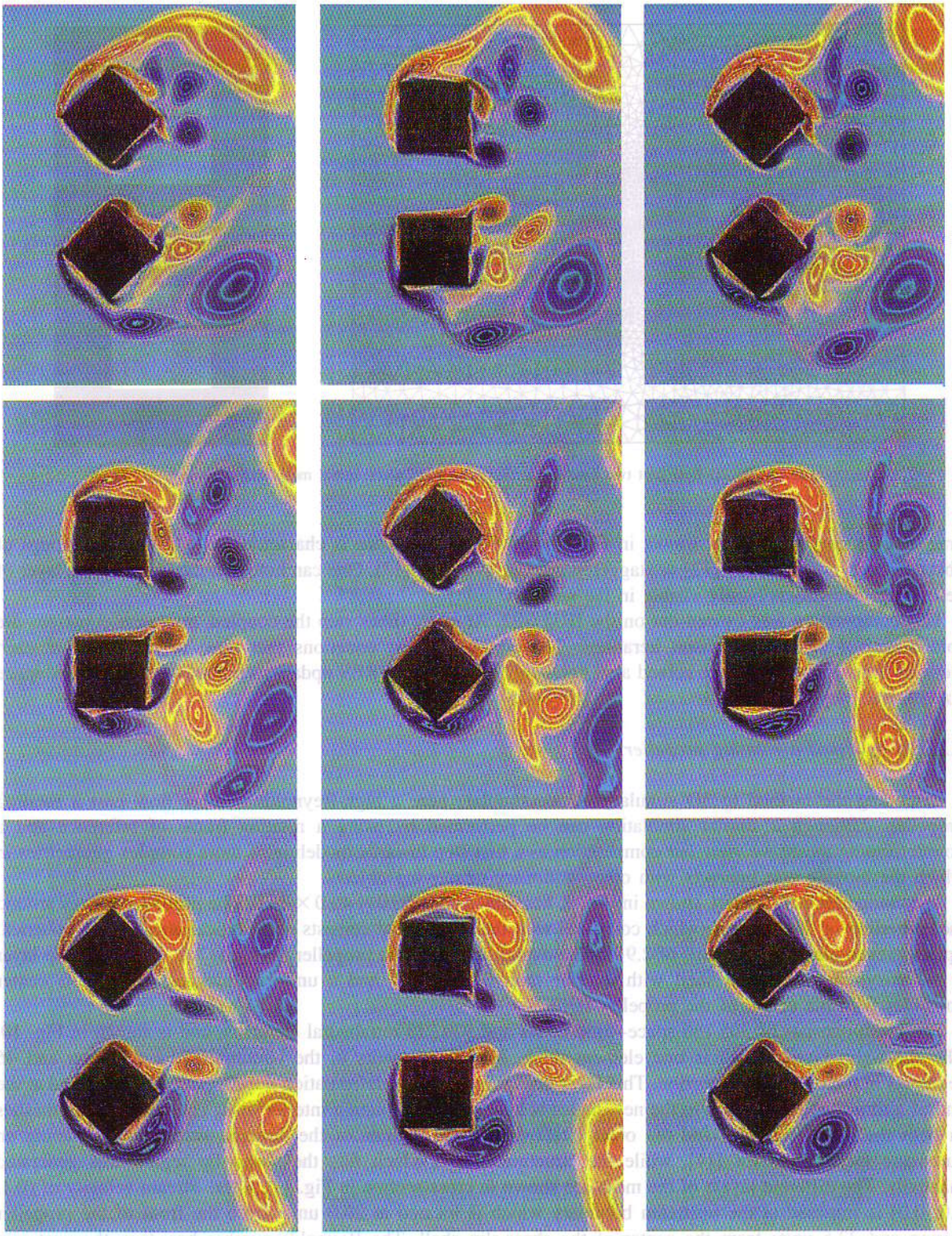

Fig. 6. 2D flow past two counter-rotating square cylinders: vorticity field at t = 125.0, 126.0, 127.0, 128.0, 129.0, 130.0, 131.0, 132.0 and 133.0 (left to right, then top to bottom).

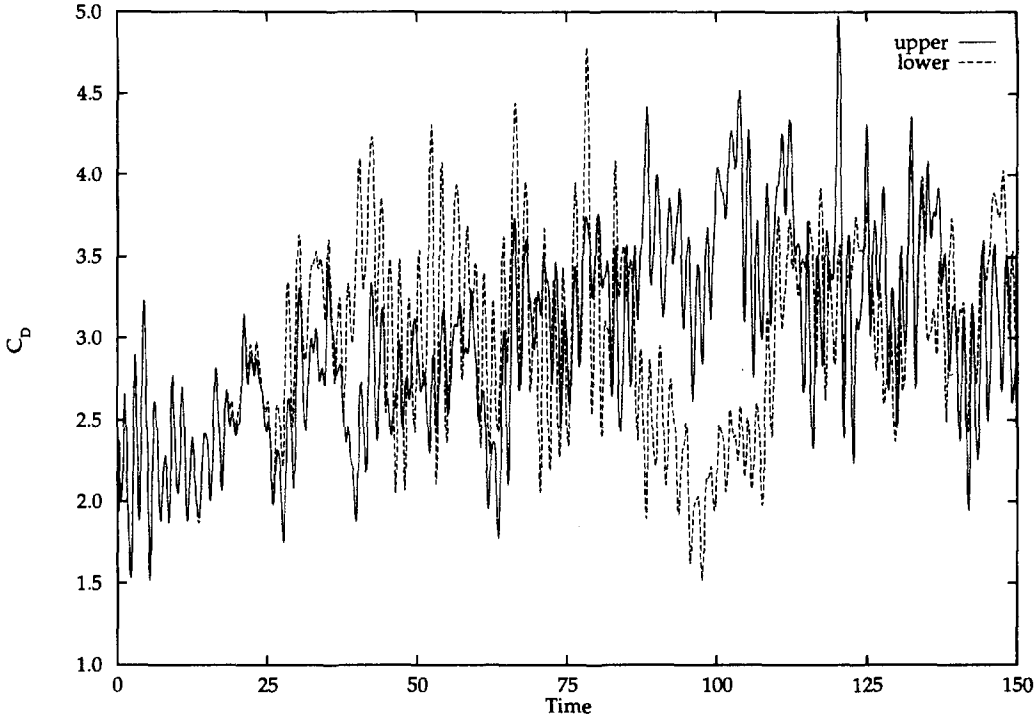

Fig. 7. 2D flow past two counter-rotating square cylinders: history of the drag coefficient.

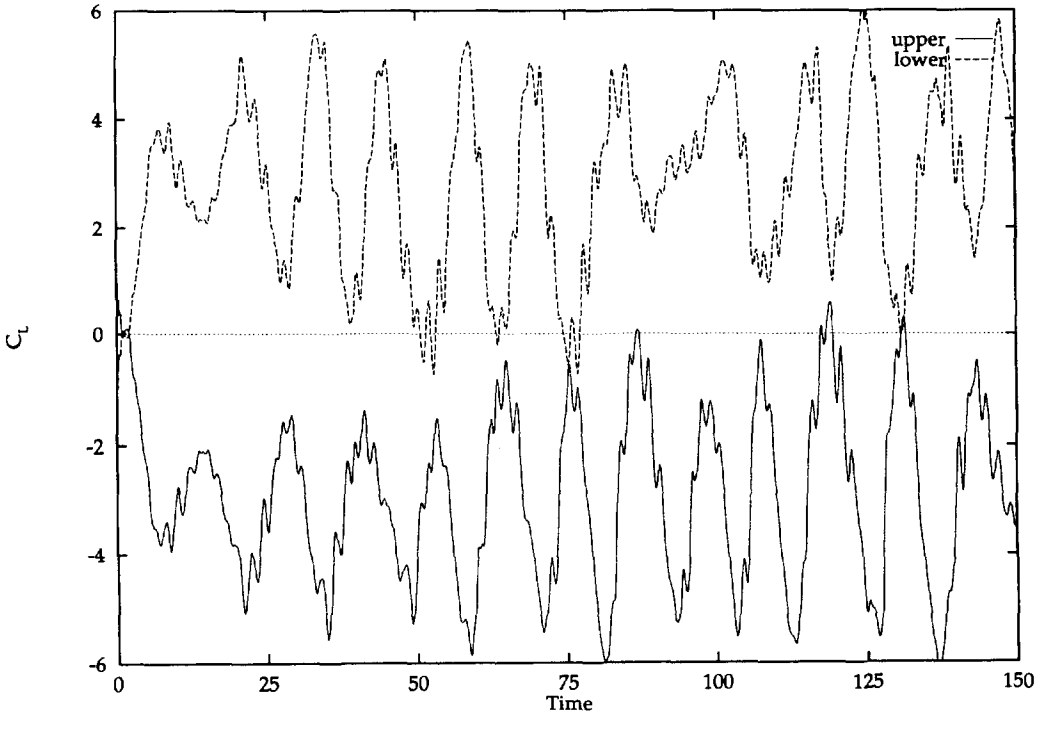

Fig. 8. 2D flow past two counter-rotating square cylinders: history of the lift coefficient.

stationary or moving objects.

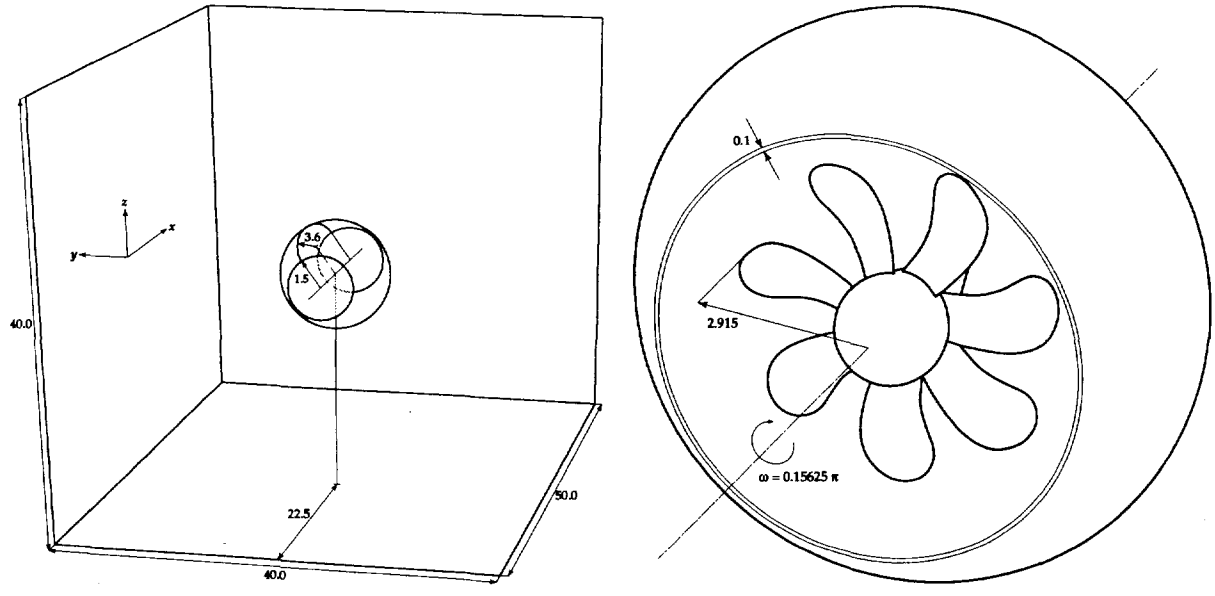

Fig. 9. 3D flow past a rotating propeller: outer boundaries of the computational domain and outer boundary of the shear-slip layer (left), and close-up view of the shear-slip layer and the propeller (right).

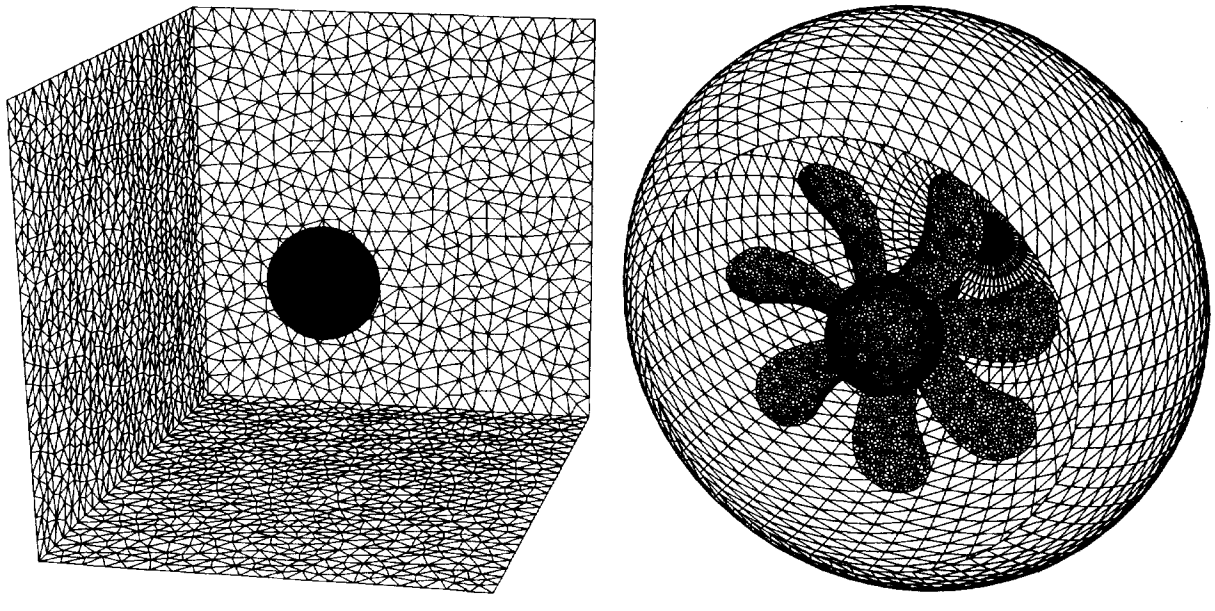

Fig. 10. 3D flow past a rotating propeller: selected boundaries of the initial mesh (left), and close-up view of the inner boundary of the shear-slip layer and the propeller (right).

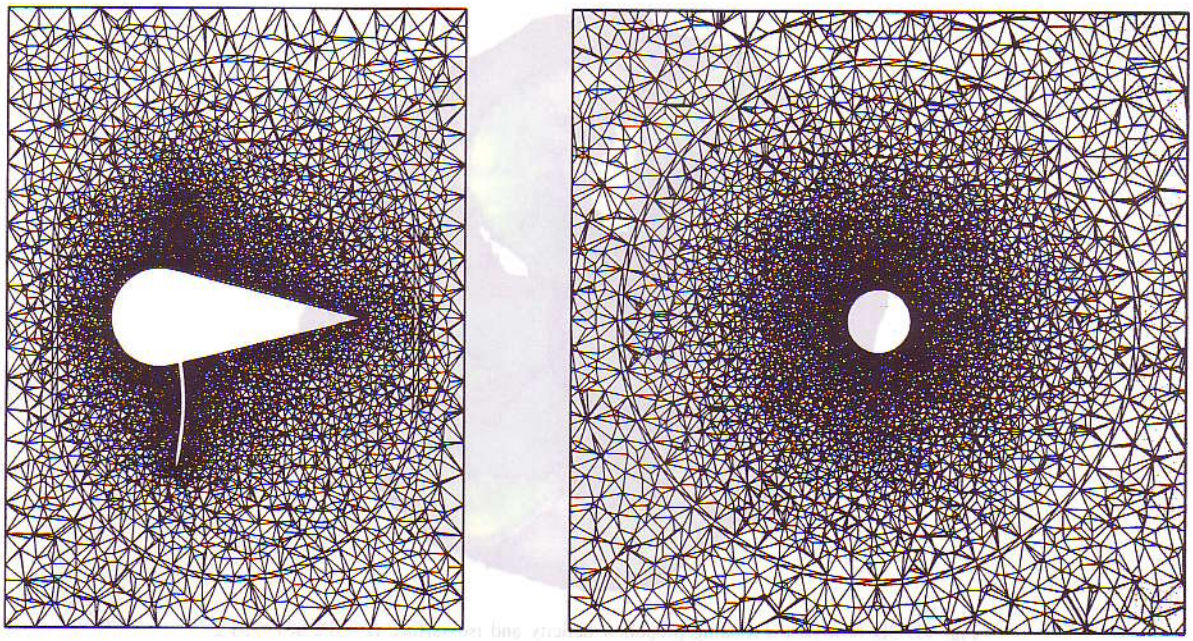

Fig. 11. 3D flow past a rotating propeller: cross-section of the mesh in the vicinity of the rotating region in the x-z plane (left), and in the y-z plane (right).

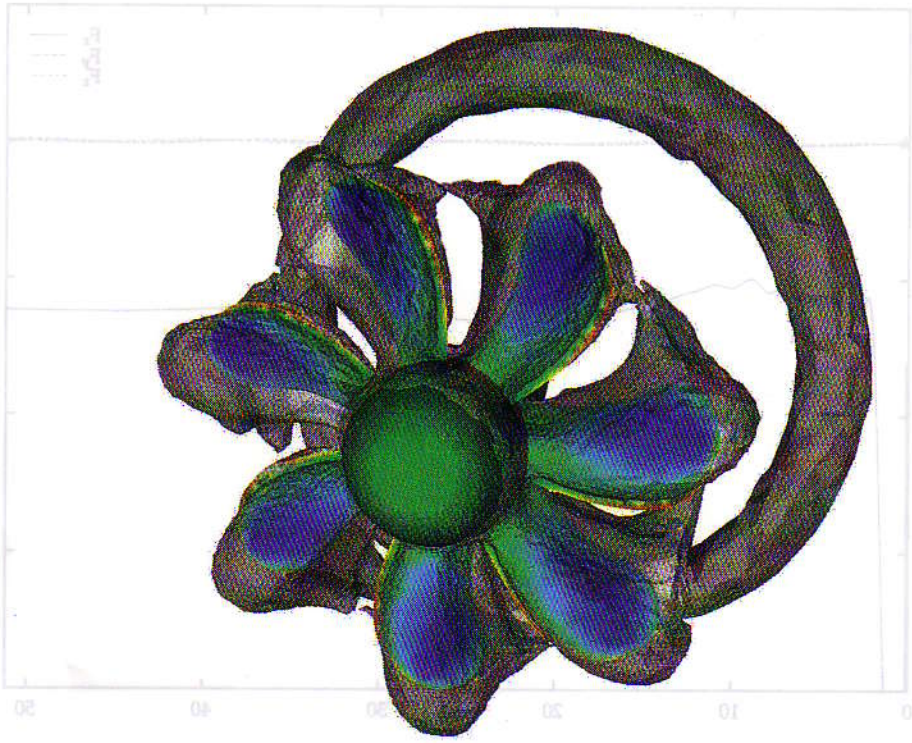

Fig. 12. 3D flow past a rotating propeller: pressure and iso-surface $p = -0.025$ at $t = 51.2$.

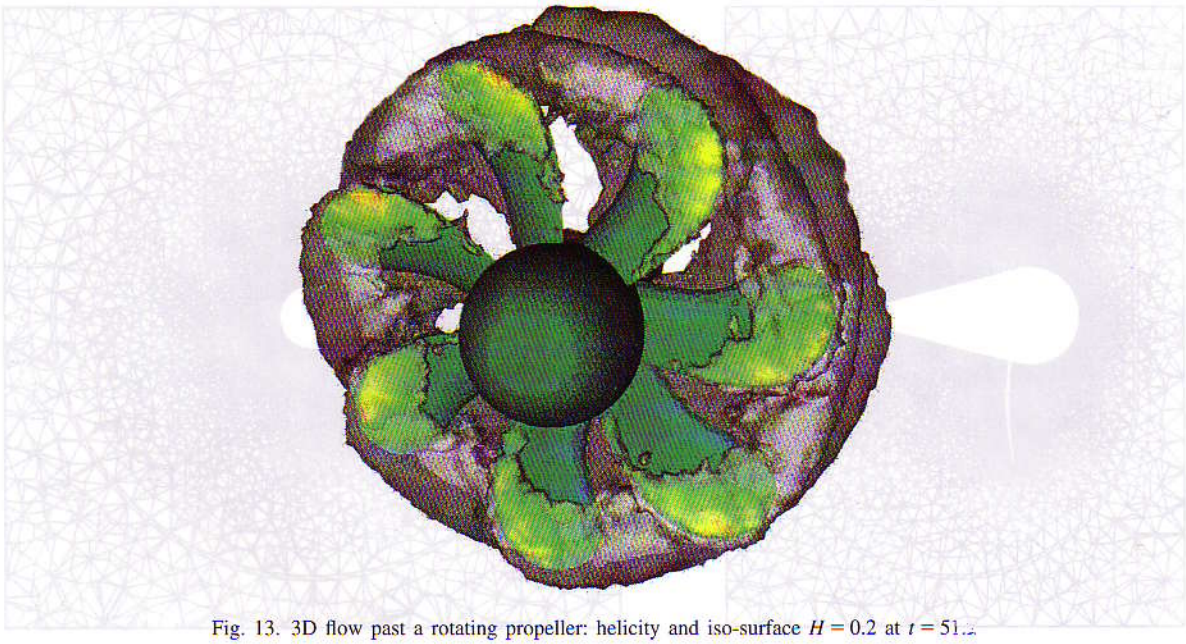

Fig. 13. 3D flow past a rotating propeller: helicity and iso-surface $H = 0.2$ at $t = 51.2$.

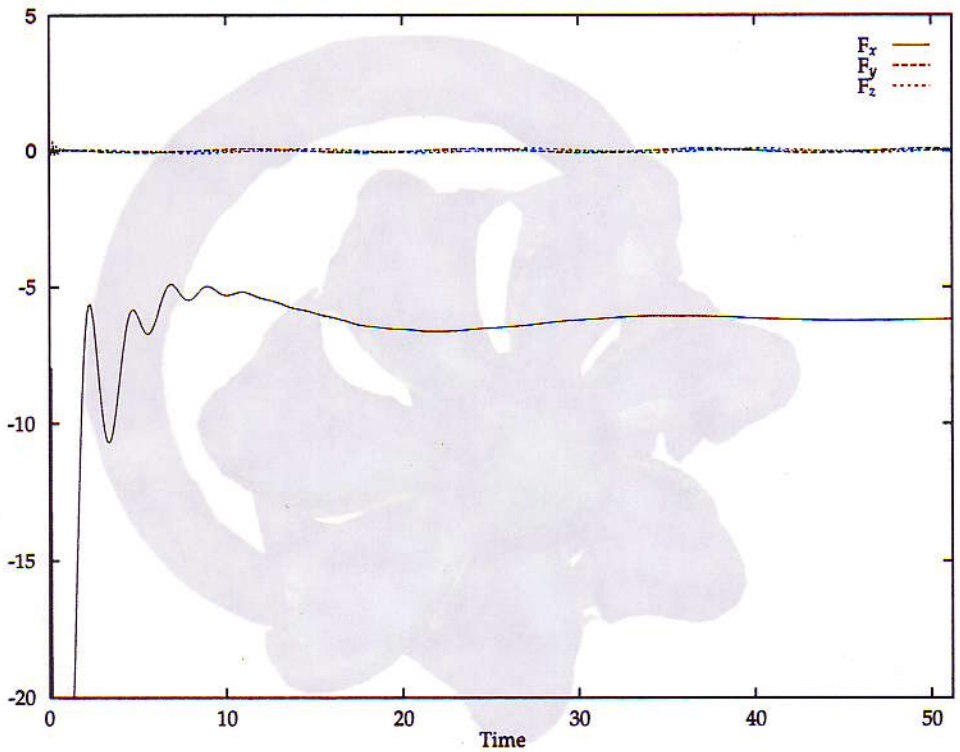

Fig. 14. 3D flow past a rotating propeller: histories of the force coefficients acting on the propeller.

propeller is rotating in clockwise direction as seen from the inflow side, with a rotational velocity magnitude of 0.125π.

Fig. 12 shows, at $t = 51.2$, the pressure field on the propeller and the pressure iso-surface corresponding to $p = -0.025$. Blade wakes and the core of a ring vortex are clearly visible. Fig. 13 shows, at the same instant, the helicity field on the propeller and the helicity iso-surface corresponding to $H = 0.2$. The histories of the force coefficient acting on the propeller are shown in Fig. 14. The coefficient scaling is based on upstream velocity and propeller diameter. The negative sign of the drag coefficient signifies that the propeller is generating thrust.

This computation has been carried out on the CRAY T3E. At every time step the coupled, nonlinear equations are solved with 4 Newton–Raphson iterations. The coupled, linear equations that need to be solved at each Newton–Raphson iteration are solved also iteratively, with the GMRES update techniques with a Krylov space size of 20.

6. Concluding remarks

We presented the Shear-Slip Mesh Update Method (SSMUM), designed to handle flow problems with moving boundaries and interfaces which involve large, regular boundary displacements. Falling into this category of motions are straight-line translation and rotation. This class of motions are accommodated by a thin layer of deforming space–time elements together with limited remeshing without projection at space–time slab interfaces. Our computations were based on the Deformable-Spatial-Domain/Stabilized Space-Time (DSD/SST) finite element formulation which automatically takes into account the deformation of the domain. A 2D flow around two counter-rotating square cylinders and a 3D flow past a propeller were presented as numerical examples.

Acknowledgment

This work was sponsored by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. CRAY T3E and IBM SP2 time was provided in part by the Minnesota Supercomputer Institute.

References

[1] 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 tests, Comput. Methods Appl. Mech. Engrg. 94 (1992) 339–351.

[2] 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, Comput. Methods Appl. Mech. Engrg. 94 (1992) 353–371.

[3] T.E. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro and M. Litke, Flow simulation and high performance computing, Comput. Mech. 18 (1996) 397–412.

[4] I. Güler, M. Behr and T.E. Tezduyar, Parallel finite element computation of free-surface flows, Comput. Mech., ~~to appear.~~ **23** (1999) 117-123.

[5] M. Behr and T.E. Tezduyar, A note on Shear-Slip Mesh Update Method, in: Lecture Notes of the Workshop on Parallel Computing in Applied Fluid Mechanics (Associazione Amici Scuola Normale Superiore, Pisa, Italy, 1997).

[6] V. Kalro and T.E. Tezduyar, Parallel finite element computation of 3D incompressible flows on MPPs, in: W.G. Habashi, ed., Solution Techniques for Large-Scale CFD Problems (John Wiley & Sons, 1995).

[7] J.A. Benek, P.G. Buning and J.L. Steger, A 3-D Chimera grid embedding technique, Paper 85-1523, AIAA, 1985.

[8] C. Kato and M. Ikegawa, Large eddy simulation of unsteady turbulent wake of a circular cylinder using the finite element method, in: I. Celik, T. Kobayashi, K.N. Ghia and J. Kurokawa, eds., Advances in Numerical Simulation of Turbulent Flows, FED-Vol. 117 (ASME, New York, 1991) 49–56.

[9] J. Smagorinsky, General circulation experiments with the primitive equations, Mthly Weather Rev. 91 (1963) 99–165.

[10] M. Behr and T.E. Tezduyar, Finite element solution strategies for large-scale flow simulations, Comput. Methods Appl. Mech. Engrg. 112 (1994) 3–24.

[11] Y. Saad and M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.

[12] M. Behr, T.E. Tezduyar and H. Higuchi, Wake interference behind two flat plates normal to the flow: A finite-element study, Theoretical Comput. Fluid Mech. 2 (1991) 223–250.

[13] S. Mittal, V. Kumar and A. Raghuvanshi, Unsteady incompressible flows past two cylinders in tandem and staggered arrangements, Int. J. Numer. Methods Fluids 25 (1997) 1315–1344.

[14] A.A. Johnson and T.E. Tezduyar, Mesh generation and update strategies for parallel computation of 3D flow problems, in: Computational Mechanics '95, Proceedings of International Conference on Computational Engineering Science, Mauna Lani, Hawaii, 1995.