



Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces

A.A. Johnson, T.E. Tezduyar*

AEM/AHPCRC, Supercomputer Institute, University of Minnesota, 1200 Washington Avenue South, Minneapolis, MN 55415, USA

Received 20 March 1994

Revised manuscript received 27 March 1994

Abstract

We present strategies to update the mesh as the spatial domain changes its shape in computations of flow problems with moving boundaries and interfaces. These strategies are used in conjunction with the stabilized space–time finite element formulations introduced earlier for computation of flow problems with free surfaces, two-liquid interfaces, moving mechanical components, and fluid–structure and fluid–particle interactions. In these mesh update strategies, based on the special and automatic mesh moving schemes, the frequency of remeshing is minimized to reduce the projection errors and to minimize the cost associated with mesh generation and parallelization set-up. These costs could otherwise become overwhelming in 3D problems. We present several examples of these mesh update strategies being used in massively parallel computation of incompressible flow problems.

1. Introduction

The deformable-spatial-domain/stabilized-space–time (DSD/SST) finite element formulation was introduced earlier [1, 2] to produce a general-purpose computational capability to solve compressible and incompressible flow problems with free surfaces, two-liquid interfaces, moving mechanical components, and fluid–structure and fluid–particle interactions. Because the formulation is cast in the space–time domain of this class of problems, determination of the unknown boundaries and interfaces becomes a built-in component of the overall DSD/SST strategy. In the implementations we consider here, the finite element interpolation functions are piecewise linear in both space and time, but continuous in space and discontinuous in time. For each time step, there is a corresponding space–time slab with two space–time nodes associated with each spatial node. Therefore, for each spatial nodal variable, we have two unknowns per time step. Although, in a sense, time is treated as another dimension, because the computations are carried out at one space–time slab at a time, solution of the large algebraic equation systems generated by this formulation is still kept at manageable levels. The stabilized nature of the formulation prevents numerical oscillations for high Reynolds and Mach numbers, even in the presence of shocks and/or boundary layers, and also for incompressible flow implementations using equal-order interpolation functions for velocity and pressure. The stabilization techniques used are streamline-upwind/Petrov–Galerkin formulation [3, 4] for compressible flows, and Galerkin/least-squares formulation [5] for incompressible flows.

A major issue in these computations is updating the mesh as the spatial domain changes its shape with time. There are quite a few ways of doing this, depending on the nature and magnitude of these changes in the shape of the spatial domain. The first thing that comes to mind (but probably the last in terms of effectiveness and sophistication) is to remesh (i.e. generate a new set of nodes and elements)

* Corresponding author.

every time step. The remeshing can be done by calling an automatic mesh generator, typically with triangular elements in 2D and tetrahedral elements in 3D. We reported examples of this approach earlier [6]. There are three disadvantages associated with this approach. First there is the projection errors involved in projecting the solution, every time step, from the old mesh to the new one. Then the cost of calling a mesh generator every time step. Finally, in massively (data-) parallel computation of these problems, the cost of mapping each new finite element mesh on to the parallel processors and computing the communication traces through the interconnect network of the machine. Although the costs for automatic mesh generation and parallelization set-up may not be so significant in 2D problems, they could be overwhelming in 3D problems, especially for problems with complicated geometries and parallelization set-up strategies involving partitioning techniques [7].

Quite often, for a specific problem, a special mesh moving scheme that is based on moving the nodes around according to an explicitly defined rule can be all that is needed. This eliminates remeshing altogether, and the associated projection errors and the overhead associated with mesh generation and parallelization set-up. Several examples of this approach can be found in [6, 8–14]. This strategy, of course, becomes very attractive in 3D problems. However, this strategy is limited to problems where initial meshes and subsequent movements can be handled with such explicitly defined rules of mesh motion. For example, if the initial mesh is a product of an automatic mesh generator, it would be very difficult to explicitly define a mesh motion. Also, although the initial mesh might be a structured or a semi-structured one, it may not be possible to explicitly define rules that accommodate the later changes in the geometry.

For a more general mesh moving strategy that is applicable to complicated geometries and meshes generated by automatic mesh generators, we designed an automatic mesh moving scheme in which the motion of the nodes is governed by the equations of linear elasticity, with the boundary conditions for these equations imposed by the motion of the fluid boundaries and interfaces. Similar strategies were used earlier by other researchers [15]. Normally in our strategy, we drop from the finite element formulation the Jacobian of the transformation between the physical and element domains. This translates to smaller elements being assigned larger rigidity in the mesh motion. Assuming that these smaller elements are placed at critical zones of the domain, the result is minimal deformation of the mesh at these critical zones, something that is probably desirable. Examples of this approach first appeared in [6], with conceptual examples appearing later in [8, 11]. This approach is of course applicable to 3D problems, however, it involves the added cost of solving the equations of linear elasticity. We can use this automatic mesh moving technique until the deformation of the elements becomes too large for the computations to continue to be reliable. At that time, of course, we remesh. This leads to a mesh update strategy that mixes automatic mesh moving with remeshing as needed. When we remesh is determined by checking the deformation of the elements according to some measure, and comparing these deformations to a predetermined allowable level.

In our favorite mesh update strategy, we combine the automatic mesh moving plus remeshing as needed with special mesh moving. Sometimes it is quite effective to have structured or semi-structured meshes in some zones of the domain, and unstructured meshes in other zones. For example, Reu and Ying [16] used, in the context of the finite volume method, a hybrid grid around an airfoil to investigate dynamic stall. Our structured or semi-structured meshes are updated with special mesh moving schemes, and our unstructured meshes with an automatic mesh moving scheme. This not only reduces the cost associated with the solution of nodal motions in the automatic mesh moving schemes, but also leaves, if desirable, certain zones of the mesh deformation-free. Earlier examples of this approach can be found in [8, 11]. When remeshing is needed, it will very likely need to be performed only for the unstructured zones; this reduces the overhead associated with remeshing. For example, in a flow problem involving more than one airfoil in relative motion, each airfoil can have a structured mesh around it, and an unstructured mesh elsewhere in the computational domain. The structured mesh around each airfoil goes through rigid body motion with that airfoil while the unstructured mesh is updated with the automatic mesh moving scheme. It is important to note here that regardless of whether the mesh is structured, unstructured or semi-structured, our parallel implementations are based on the assumption that the mesh is unstructured. Therefore the parallel performance is somewhat insensitive to the complexity of the mesh.

In Section 2 of this article, we briefly review the DSD/SST finite element formulation for

incompressible flows. The automatic mesh moving scheme is described in Section 3. In Section 4 we present several numerical examples.

2. The deformable-spatial-domain/stabilized-space–time finite element formulation

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} denotes the number of space dimensions. The velocity, $\mathbf{u}(\mathbf{x}, t)$ and pressure, $p(\mathbf{x}, t)$ are governed by the Navier–Stokes equations of incompressible flows

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

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

where ρ is the density of the fluid, and $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\rho\nu\boldsymbol{\varepsilon}(\mathbf{u}), \quad \boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad (3)$$

where ν is the fluid kinematic viscosity and \mathbf{I} is the identity tensor. The Dirichlet and Neumann-type boundary conditions are represented by

$$\mathbf{u} = \mathbf{g} \quad \text{on } (\Gamma_t)_g, \quad (4)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \quad \text{on } (\Gamma_t)_h, \quad (5)$$

where $(\Gamma_t)_g$ and $(\Gamma_t)_h$ are complementary subsets of the boundary Γ_t . The initial condition is a divergence-free velocity field specified over the entire domain

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_0 \quad \text{on } \Omega_0. \quad (6)$$

To construct the finite element function spaces for the space–time formulation, we first 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 steps $0 = t_0 < t_1 < \dots < t_N = T$. With $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$, we 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 in the case with Γ_t , surface P_n is decomposed into $(P_n)_g$ and $(P_n)_h$. For each space–time slab, we define the following finite element function spaces for the velocity, pressure, and their variations

$$(S_u^h)_n = \{ \mathbf{u}^h \mid \mathbf{u}^h \in [H^{1h}(Q_n)]^{n_{sd}}, \mathbf{u}^h \doteq \mathbf{g}^h \quad \text{on } (P_n)_g \}, \quad (7)$$

$$(V_u^h)_n = \{ \mathbf{w}^h \mid \mathbf{w}^h \in [H^{1h}(Q_n)]^{n_{sd}}, \mathbf{w}^h \doteq \mathbf{0} \quad \text{on } (P_n)_g \}, \quad (8)$$

$$(S_p^h)_n = (V_p^h)_n = \{ p^h \mid p^h \in H^{1h}(Q_n) \}. \quad (9)$$

Here $H^{1h}(Q_n)$ represents the finite-dimensional function space over the space–time slab Q_n . Over the element domain, this space is formed by using first-order polynomials in space and also in time. The interpolation function spaces are continuous in space, but discontinuous in time.

The stabilized, space–time formulation is written as follows: given $(\mathbf{u}^h)_n$, find $\mathbf{u}^h \in (S_u^h)_n$ and $p^h \in (S_p^h)_n$ such that $\forall \mathbf{w}^h \in (V_u^h)_n$ and $\forall q^h \in (V_p^h)_n$

$$\begin{aligned} & \int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f} \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) dQ - \int_{(P_n)_h} \mathbf{w}^h \cdot \mathbf{h}^h dP \\ & + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h dQ + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-) d\Omega \\ & + \sum_{e=1}^{(n_{cl})_n} \int_{Q_n^e} \frac{\tau}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h \right) - \nabla \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h) \right] \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f} \right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \right] dQ \\ & + \sum_{e=1}^{(n_{cl})_n} \int_{Q_n^e} \delta \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h dQ = 0. \end{aligned} \quad (10)$$

This process is applied sequentially to all space–time slabs Q_1, Q_2, \dots, Q_{N-1} . In Eq. (10), the following notation is being used

$$(\mathbf{u}^h)_n^\pm = \lim_{\varepsilon \rightarrow 0} \mathbf{u}(t_n \pm \varepsilon), \quad (11)$$

$$\int_{Q_n} (\dots) dQ = \int_{I_n} \int_{\Omega_t} (\dots) d\Omega dt, \quad (12)$$

$$\int_{P_n} (\dots) dP = \int_{I_n} \int_{\Gamma_t} (\dots) d\Gamma dt. \quad (13)$$

The computations start with

$$(\mathbf{u}^h)_0^- = \mathbf{u}_0. \quad (14)$$

REMARKS

- (1) The first four integrals in the variational formulation given by Eq. (10) constitute the standard Galerkin formulation of the problem. The fifth integral weakly enforces the continuity of the velocity field across space–time slabs.
- (2) The sixth integral is the least-squares term corresponding to the momentum equation. This term provides stability for advection-dominated flows. It also provides stability when equal-order interpolation functions are used for velocity and pressure. The definition of the parameter τ can be found in [12].
- (3) The seventh integral enhances the stability at high Reynolds number flows. This is a least-squares term corresponding to the continuity equation, and the parameter δ is defined in [12].
- (4) The addition of the stabilizing terms to the Galerkin formulation does not compromise the consistency of the formulation, since these involve the residuals as factors.

3. The automatic mesh moving scheme

Consider an elastic body occupying a bounded region $\Omega \subset \mathbb{R}^{n_{sd}}$ with boundary Γ . The motion of the mesh $\mathbf{v}(\mathbf{x})$ is governed by the equilibrium equations of elasticity:

$$\nabla \cdot \boldsymbol{\sigma}^* + \mathbf{f}^* = 0, \quad (15)$$

where \mathbf{f} is the prescribed body force. The stress tensor $\boldsymbol{\sigma}^*$ is related to the elastic strain tensor $\boldsymbol{\varepsilon}^*$ as follows

$$\boldsymbol{\sigma}^* = \lambda(\text{tr } \boldsymbol{\varepsilon}^*)\mathbf{I} + 2\mu\boldsymbol{\varepsilon}^*, \quad (16)$$

where λ and μ are the Lamé elastic constants. The strain tensor $\boldsymbol{\varepsilon}^*$ is related to the displacement gradients by the expression

$$\boldsymbol{\varepsilon}^* = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T). \quad (17)$$

The Dirichlet and Neumann-type boundary conditions for the mesh motion are imposed by the motion of the boundaries and interfaces, and are represented by

$$\mathbf{v} = \mathbf{g}^* \quad \text{on } (\Gamma)_{g^*}, \quad (18)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma}^* = \mathbf{h}^* \quad \text{on } (\Gamma)_{h^*}, \quad (19)$$

where $(\Gamma)_{g^*}$ and $(\Gamma)_{h^*}$ are complementary subsets of the boundary Γ .

The finite element function spaces for the automatic mesh moving scheme are constructed as follows:

$$(S_v^h) = \{\mathbf{v}^h \mid \mathbf{v}^h \in [H^{1h}(\Omega)]^{n_{sd}}, \mathbf{v}^h = (\mathbf{g}^*)^h \quad \text{on } \Gamma_{g^*}\}, \quad (20)$$

$$(V_v^h) = \{\mathbf{w}^h \mid \mathbf{w}^h \in [H^{1h}(\Omega)]^{n_{sd}}, \mathbf{w}^h = \mathbf{0} \quad \text{on } \Gamma_{g^*}\}, \quad (21)$$

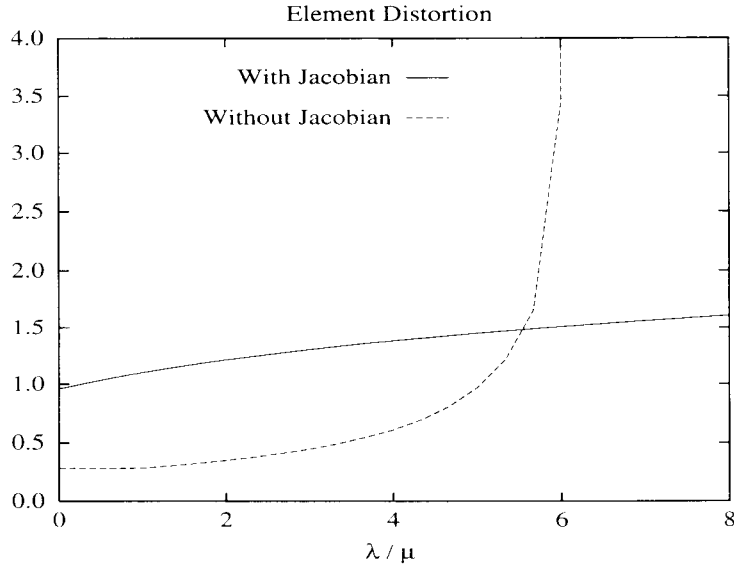


Fig. 1. Element distortion as a function of λ/μ .

The automatic mesh moving scheme can then be formulated as follows: find $\mathbf{v}^h \in S_v^h$ such that $\forall \mathbf{w}^h \in V_v^h$

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\mathbf{w}^h) : \boldsymbol{\sigma}^*(\mathbf{v}^h) \, d\Omega - \int_{\Omega} \mathbf{w}^h \cdot \mathbf{f}^* \, d\Omega = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^* \, d\Gamma. \quad (22)$$

REMARKS

- (5) If $\mathbf{f}^* = \mathbf{0}$ and $\mathbf{h}^* = \mathbf{0}$, by dividing the equations by μ , the number of elastic constants can be reduced to a single one: λ/μ .
- (6) It is desirable to preserve the structure of the mesh in the more refined areas, and have the deformations of the mesh weighted towards the larger elements. To accomplish this, we would like to have a variable stiffness coefficient where this coefficient would be larger for small elements and vice versa. We implement this by dropping from Eq. (22) the Jacobian of the transformation between the physical and element domains. By doing this, in areas where refinement is important, small elements (which are more susceptible to distortion) retain their shape better.

The effect of varying λ/μ on the distortion of the mesh can be seen in Fig. 1. For a given deformation of the domain, we vary the elastic constant λ/μ and plot the maximum of the changes in the aspect ratio (here defined as [the maximum edge length squared]/[the area]) for all the elements in the domain (where 0.0 means no distortion). It can be seen that the formulation with the Jacobian left out results in less distortions. A similar, relative behavior can be seen if we monitor the changes in the element areas instead of the changes in the aspect ratios.

4. Numerical examples

In this section, several examples of flow problems involving moving boundaries and/or interfaces are presented. A number of mesh update strategies are used to facilitate the boundary and interface motions that are involved in these problems.

All of these simulations were performed on the CM-5 with vector execution units. The partitions we used ranged from 32 to 512 processing nodes. The GMRES search technique with a diagonal preconditioner was used to solve the large linear equation systems involved in the computations.

A viscous drop falling in a viscous fluid

This problem is an example of a mesh update strategy using the automatic mesh moving scheme alone on a fairly structured mesh. The problem involves a heavier fluid (the drop) falling through a lighter fluid. As the drop falls, it changes shape until a terminal velocity is reached. This is an axisymmetric simulation, with surface tension (see [2]) present at the fluid interface (surface of the drop). This problem was previously reported in [6].

There are two types of mesh motions taking place in this problem. The first one is the translation of the mesh as a whole, where the entire mesh is moved so as to track the center of gravity of the drop as it falls. On top of this global translation, the automatic mesh moving scheme is used to track the surface of the drop with respect to the center of gravity. The nodes at the interface are moved so that no fluid crosses this boundary. We accomplish this by matching the normal velocity of the interface nodes with the computed normal fluid velocity at the same location. The tangential velocity of the nodes on the interface is free for us to choose, and we specify that velocity in such a way that we keep the spacing of the interface nodes uniform throughout the computation.

The non-dimensional parameters for the simulation are

$$\begin{aligned} \rho &= 1.0, & \hat{\rho} &= 1.444, \\ \nu &= 0.00577, & \hat{\nu} &= 0.0115, \\ g &= 1.0, & \gamma &= 0.0667, & d_0 &= 1.0, \end{aligned}$$

where γ is the surface tension coefficient, g is gravity and d_0 is the diameter of the initial, spherical drop. The parameters with $\hat{\cdot}$ are those corresponding to the fluid inside the drop. In Fig. 2 we show a close-up of the mesh at the initial and terminal stages. Fig. 3 shows the shape and location of the drop at 5 different instants during the simulation. The top image (on the left) shows the drop shape at the initial stage (a sphere), and the bottom one at terminal velocity. Also in Fig. 3 (on the right), we see the vorticity and stream function inside and outside of the drop at terminal velocity. The values of the significant non-dimensional parameters at terminal velocity are

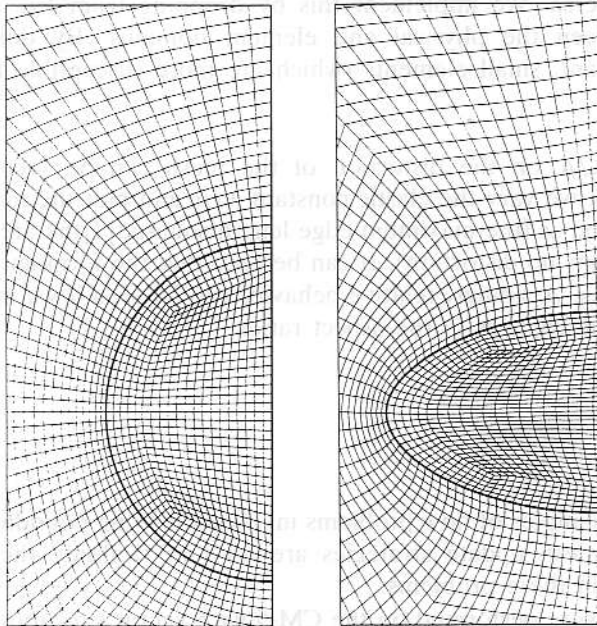


Fig. 2. A viscous drop falling in a viscous fluid: close-up view of the finite element mesh (8151 nodes and 7865 elements) at the initial and terminal stages.

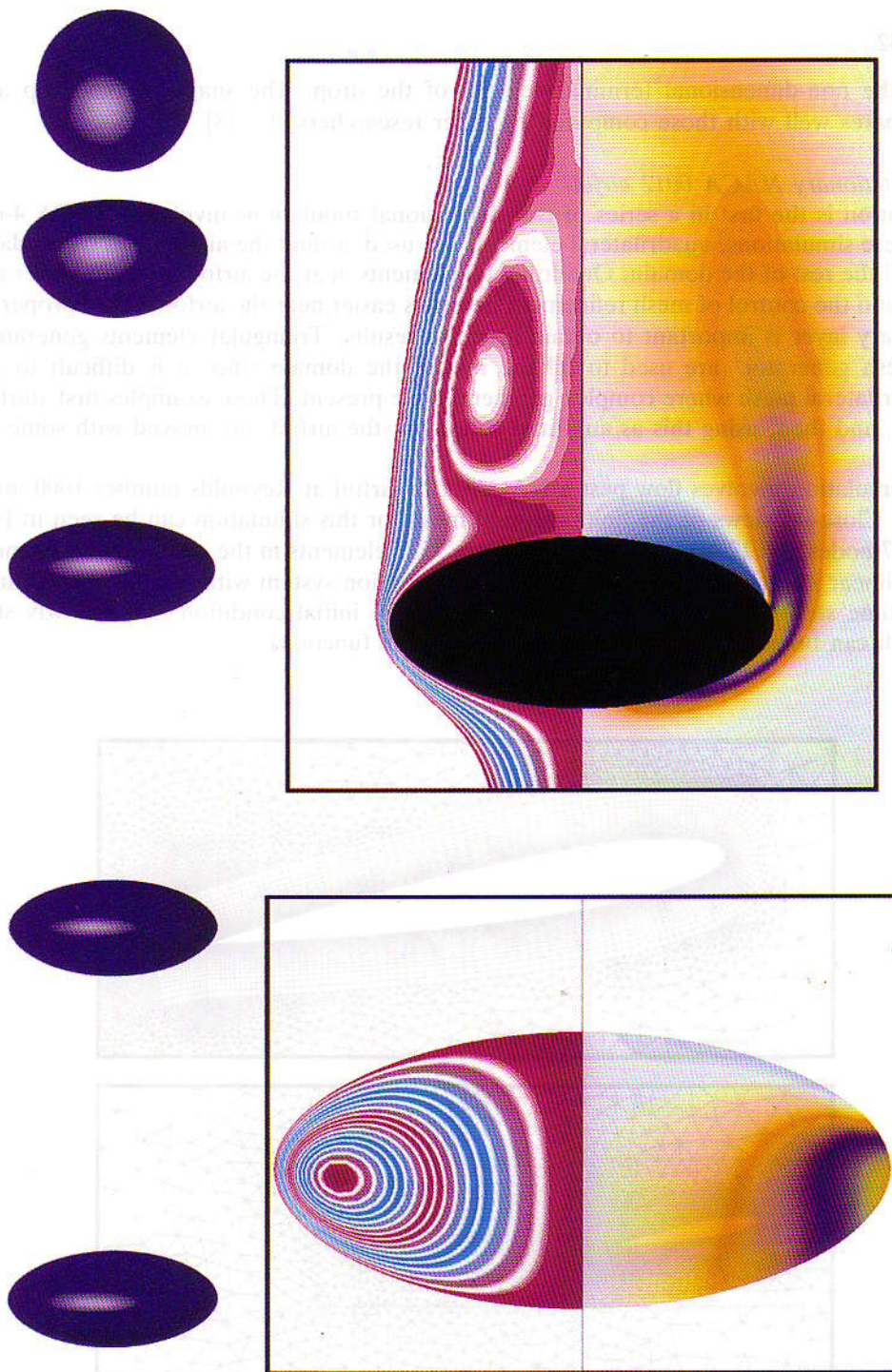


Fig. 3. A viscous drop falling in a viscous fluid: the drop shape and the vorticity and stream function at terminal velocity.

$$\begin{aligned} \text{Re} &= 112, \\ \text{We} &= \frac{U^2}{\gamma} = 6.24, \\ C_d &= 1.42, \end{aligned} \tag{23}$$

where U is the non-dimensional terminal velocity of the drop. The shape of the drop at terminal velocity compares well with those computed by other researchers [17, 18].

Flow past a stationary NACA 0012 airfoil

This simulation is the first in a series of two-dimensional simulations involving NACA 4-digit series airfoils. In these simulations, quadrilateral elements are used around the airfoil, and triangular elements are used to fill the rest of the domain. Quadrilateral elements near the airfoil result in better accuracy in those zones, and the control of mesh refinement becomes easier near the airfoil where proper resolution of the boundary layer is important to obtain accurate results. Triangular elements generated with an automatic mesh generator¹ are used to fill the rest of the domain since it is difficult to generate a regular, quadrilateral mesh where complex geometries are present. These examples first start with fixed airfoil results, and then, using this as an initial condition, the airfoils are moved with some prescribed motion.

This first simulation involves flow past a NACA 0012 airfoil at Reynolds number 1000 and angle of attack of 10° . Close-up views of the finite element mesh for this simulation can be seen in Fig. 4. This mesh has 8117 nodes and 12 416 elements. There are 3572 elements in the quadrilateral element region. At each non-linear iteration of a given time step, an equation system with roughly 45 000 unknowns is solved. The time step used in this simulation is 0.01. The initial condition is the nearly steady-state solution which can be seen in Fig. 5 (vorticity and stream function).

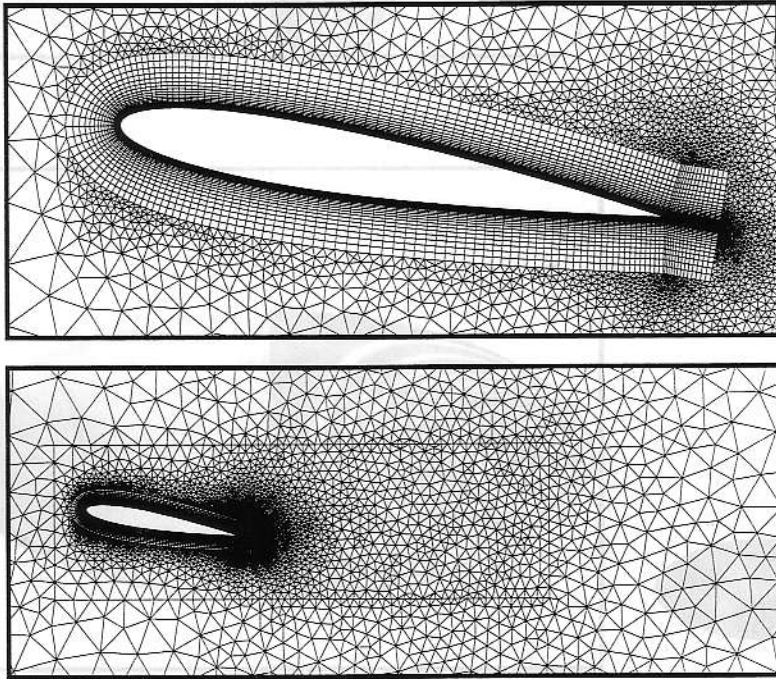


Fig. 4. Flow past a stationary NACA 0012 airfoil: close-up views of the finite element mesh (8117 nodes and 12 416 elements).

¹ Except for the case of the double NACA 4412 airfoil mesh, all of the triangular element meshes were generated by the *Emc*², a mesh generator from INRIA, France [19]. The triangular mesh for the double NACA 4412 airfoils was generated by one of our own mesh generators.

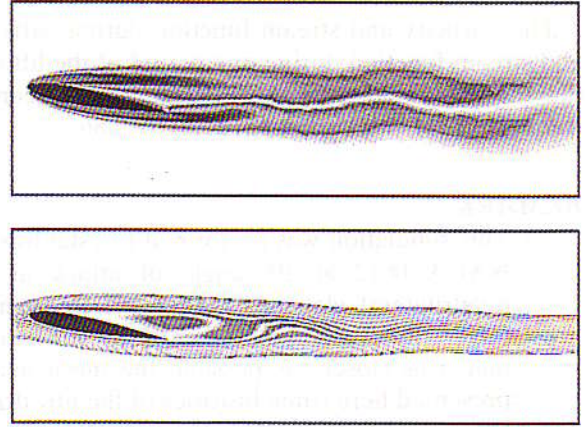
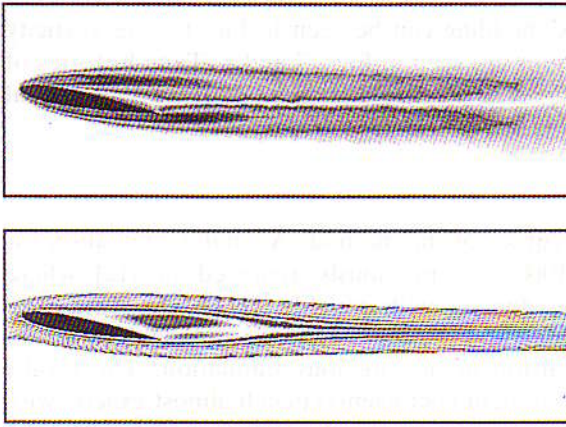


Fig. 5. Flow past a stationary NACA 0012 airfoil at $Re = 1000$: vorticity and stream function at initial condition (steady-state solution).

Fig. 6. Flow past a stationary NACA 0012 airfoil at $Re = 1000$: vorticity and stream function during early vortex shedding.

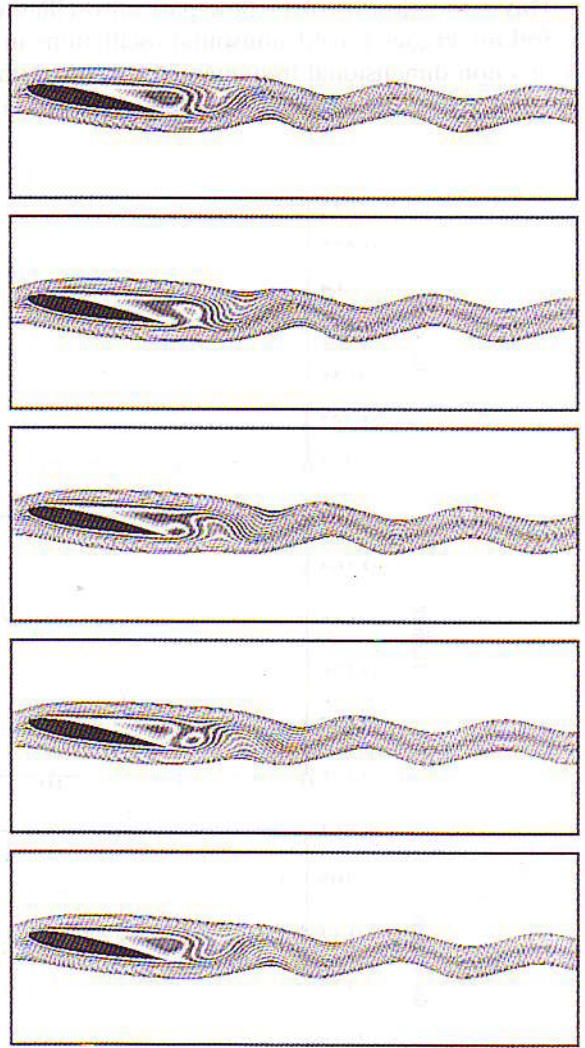
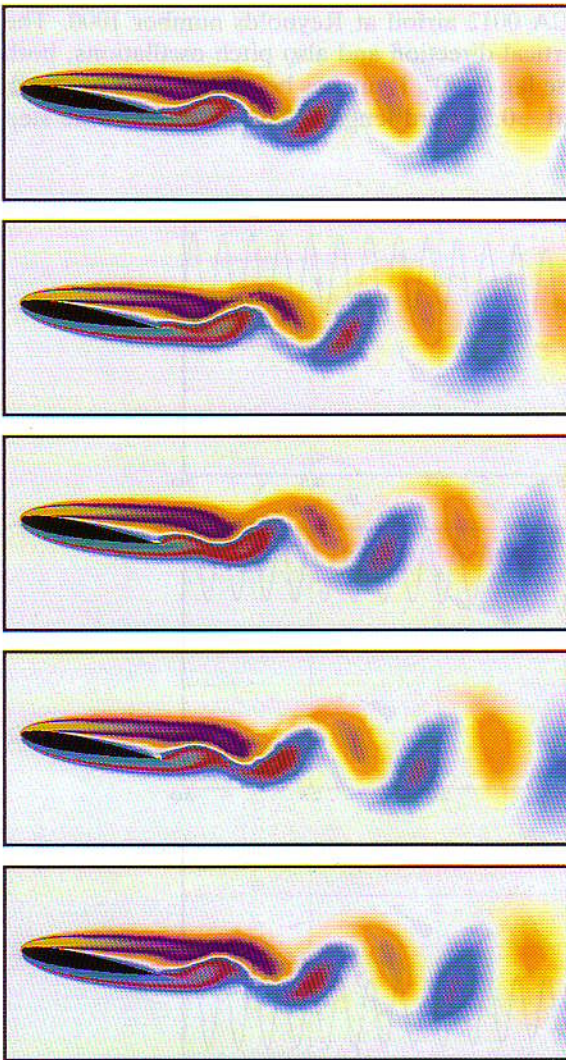


Fig. 7. Flow past a stationary NACA 0012 airfoil at $Re = 1000$: vorticity at various instants during one period of the lift oscillations.

Fig. 8. Flow past a stationary NACA 0012 airfoil at $Re = 1000$: stream function at various instants during one period of the lift oscillations.

The vorticity and stream function during early vortex shedding can be seen in Fig. 6. The vorticity and stream function during one period of the lift oscillations are seen in Figs. 7 and 8. Time histories of the lift, drag and moment coefficients are seen in Fig. 9. The Strouhal number (non-dimensional frequency) for the vortex shedding is 0.86.

REMARK

- (7) This simulation was performed to establish the accuracy of the method. A similar simulation for NACA 0012 at 10° angle of attack at $Re = 1000$ was previously reported in [12] where quadrilateral elements were used throughout the domain with roughly 22 000 elements. The quadrilateral element region of the mesh in the simulation presented here was in fact designed to match as closely as possible the mesh near the airfoil in the previous simulation. The results presented here (time histories of the lift, drag and moment coefficients) match almost exactly with what was computed in the previous simulation, and thus, shows that this method is equally accurate near the airfoil, even though much fewer nodes are used overall.

Flow past an oscillating NACA 0012 airfoil

This simulation involves flow past an oscillating NACA 0012 airfoil at Reynolds number 1000. The airfoil undergoes forced sinusoidal oscillations in the vertical direction and also pitch oscillations, both with a non-dimensional frequency of 0.5, but with a phase lag of 90° . It pitches between angles of attack 10° and -10° , and vertically oscillates between $+0.5$ and -0.5 chord lengths. The mesh and time step

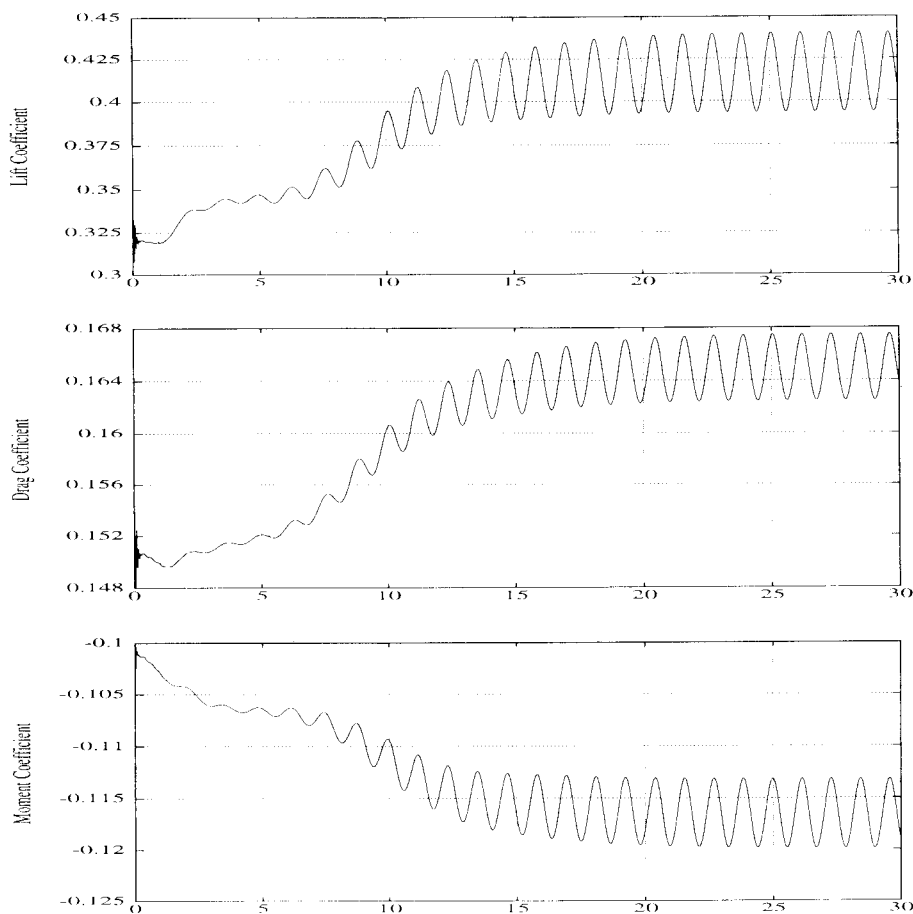


Fig. 9. Flow past a stationary NACA 0012 airfoil at $Re = 1000$: time histories of the lift, drag and moment coefficients.

size used in this simulation are exactly the same as those used in the stationary NACA 0012 airfoil simulation.

The automatic mesh moving scheme (with the Jacobian left out of the formulation) is used to facilitate the motion of the airfoil. In these motions, the quadrilateral element region undergoes rigid-body motion with the airfoil, and the triangular element region takes up the deformations. The mesh motion equations are solved using the finite element method, and this leads to an equation system with roughly 15 000 unknowns. This system is solved once at each time step.

The vorticity and stream function during one cycle of the airfoil motion can be seen in Figs. 10 and 11. The mesh during one cycle of the airfoil motion can be seen in Fig. 12. Time histories of the lift, drag and moment coefficients are shown in Fig. 13. It can be seen (as expected) that the lift and moment coefficients are symmetric about the zero axis (since this is a symmetric motion), and, from the time history of the drag coefficient, it can be seen that this motion produces a net thrust force. Detailed computational studies have been performed to analyze thrust-producing airfoil motions (see [20, 21]) where many parameters such as frequency, pivot location and amplitudes are varied.

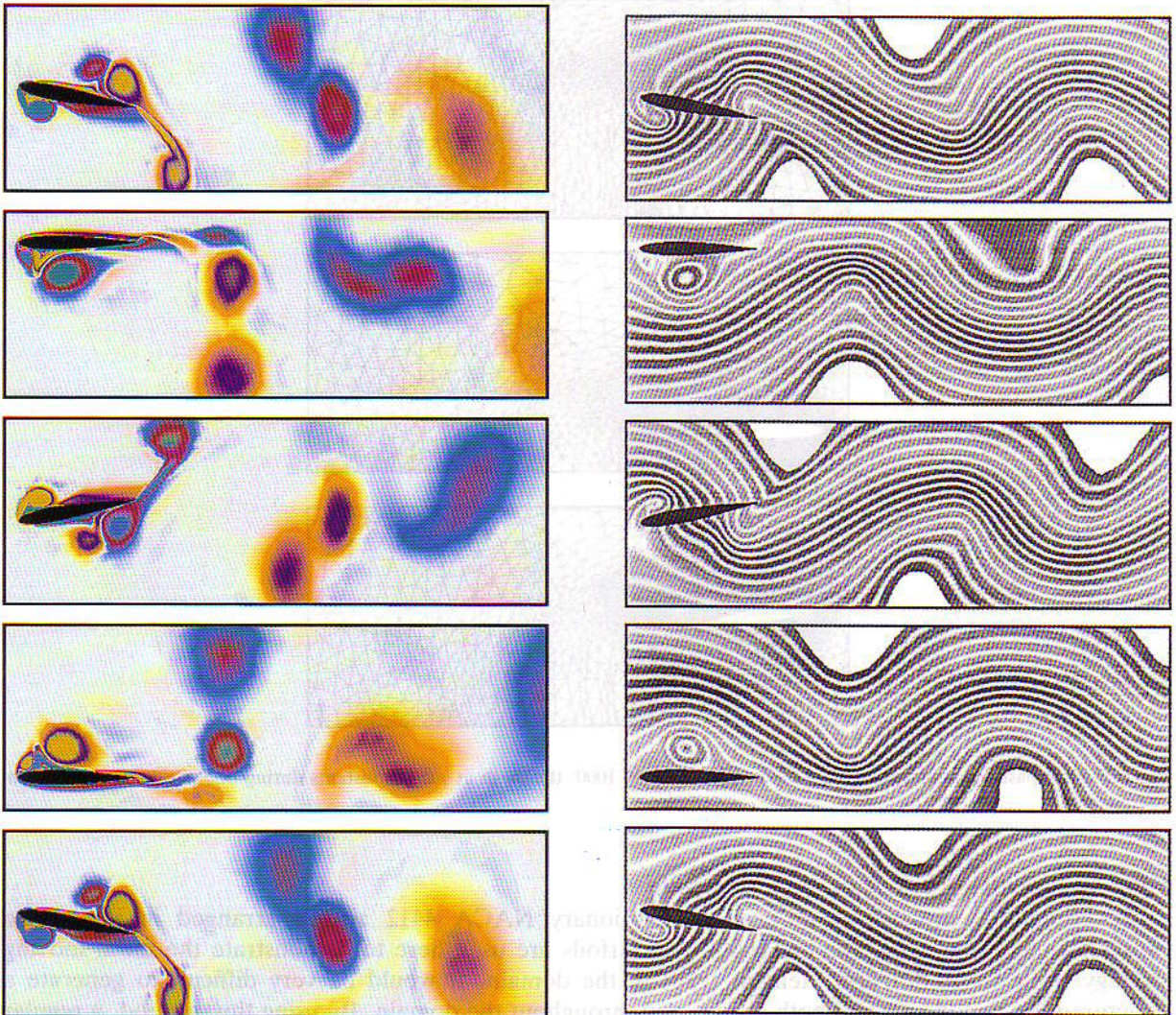


Fig. 10. Flow past an oscillating NACA 0012 airfoil at $Re = 1000$: vorticity at various instants during one period of the motion.

Fig. 11. Flow past an oscillating NACA 0012 airfoil at $Re = 1000$: stream function at various instants during one period of the motion.

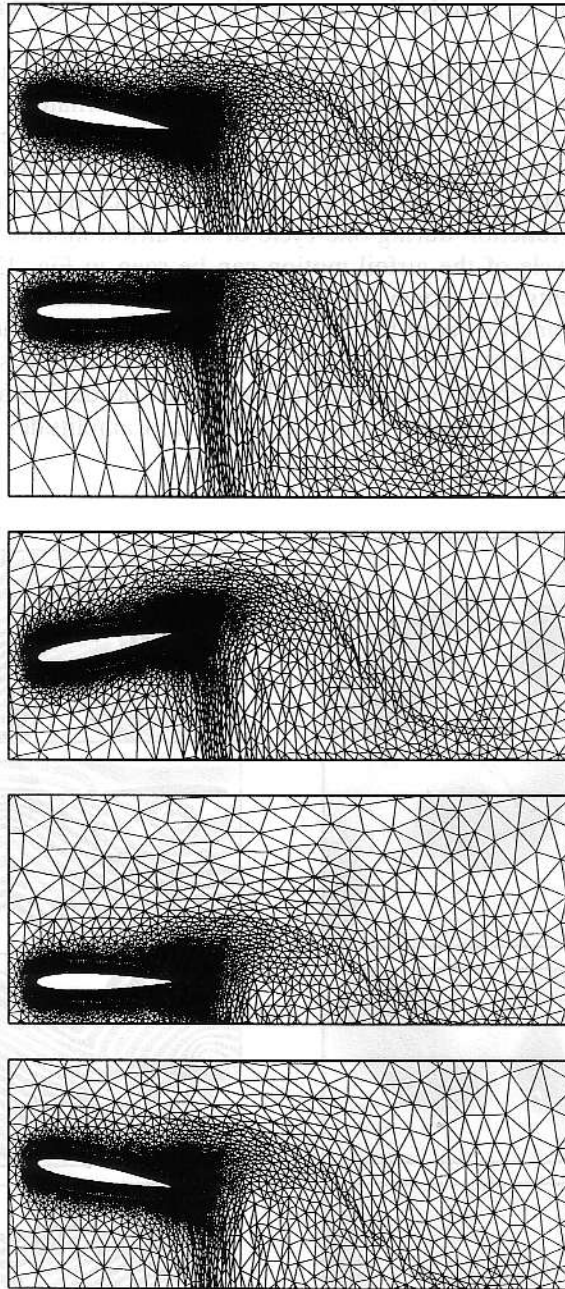


Fig. 12. Flow past an oscillating NACA 0012 airfoil at $Re = 1000$: the mesh at various instants during one period of the motion.

Flow past two stationary NACA 4412 airfoils

This simulation involves flow past two stationary NACA 4412 airfoils arranged in a bi-plane configuration at Reynolds number 5000. Two airfoils are used here to demonstrate the mesh moving strategy while using different element types in the domain. It would be very difficult to generate a structured mesh alone around both airfoils and throughout the domain. By using this method, a regular structured mesh can be generated around each airfoil, essentially for all positions or orientations they may be in. Airfoil A is located at $(+0.125, -0.250)$ with angle of attack 20° , while Airfoil B is located at $(-0.125, +0.250)$ with angle of attack 25° .

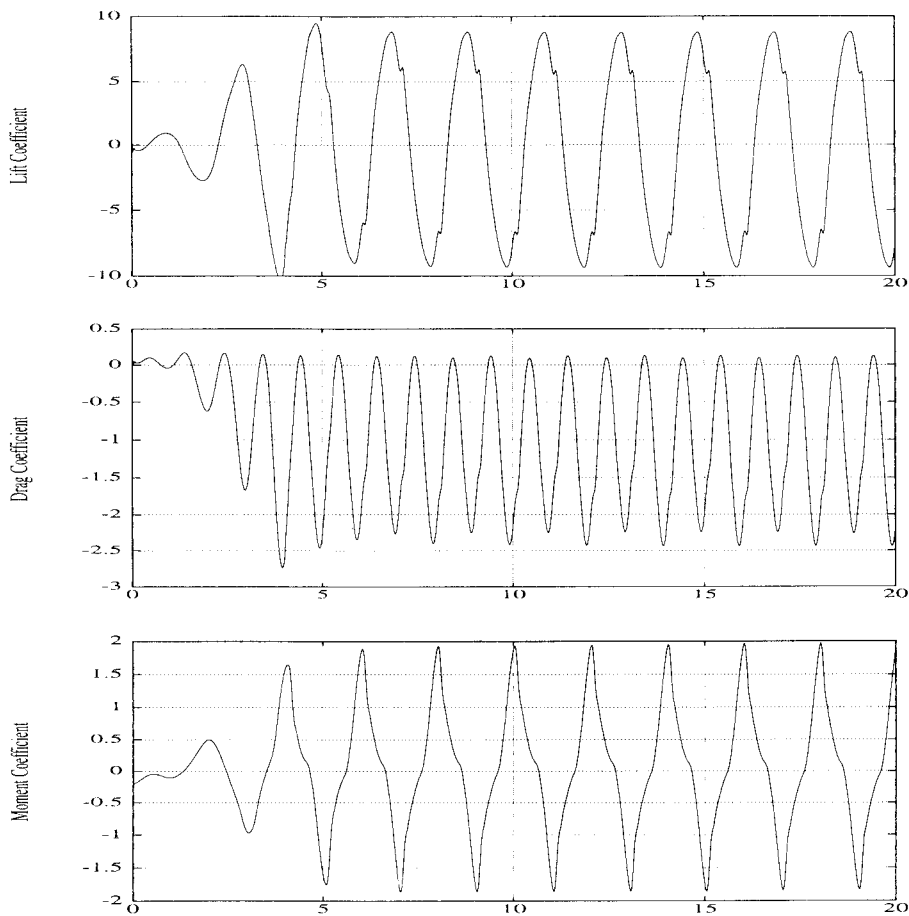


Fig. 13. Flow past an oscillating NACA 0012 airfoil at $Re = 1000$: time histories of the lift, drag and moment coefficients.

Close-up views of the finite element mesh for this simulation can be seen in Fig. 14. This mesh contains 7310 nodes and 9994 elements. There are 4352 elements in the two quadrilateral element regions. At each non-linear iteration of a given time step, an equation system with roughly 42 000 unknowns is solved. The time step size used in this simulation is 0.02. The initial condition is the nearly steady-state solution which can be seen in Fig. 15 (vorticity and stream function).

The vorticity and stream function during early vortex shedding can be seen in Fig. 16. Vorticity and stream function during roughly one period of the lift oscillations are seen in Figs. 17 and 18. Time histories of the lift, drag and moment coefficients for Airfoil A and Airfoil B are seen in Figs. 19 and 20, respectively.

Flow past two NACA 4412 airfoils with one oscillating

This simulation again involves flow past two NACA 4412 airfoils arranged in a bi-plane configuration at Reynolds number 5000. However, this time Airfoil B undergoes a forced oscillation between angles of attack 15° and 25° with a non-dimensional frequency of 0.5. The mesh and time step size used in this simulation are exactly the same as those used in the simulation of the two stationary airfoils of the previous case.

As before, the automatic mesh moving method is used to facilitate the motions of the airfoil. The quadrilateral element region around Airfoil B undergoes rigid-body motion with the airfoil, while the quadrilateral element region around Airfoil A is deformation-free. The triangular element region takes up the deformations.

The vorticity and stream function during one cycle of the airfoil motion can be seen in Figs. 21 and

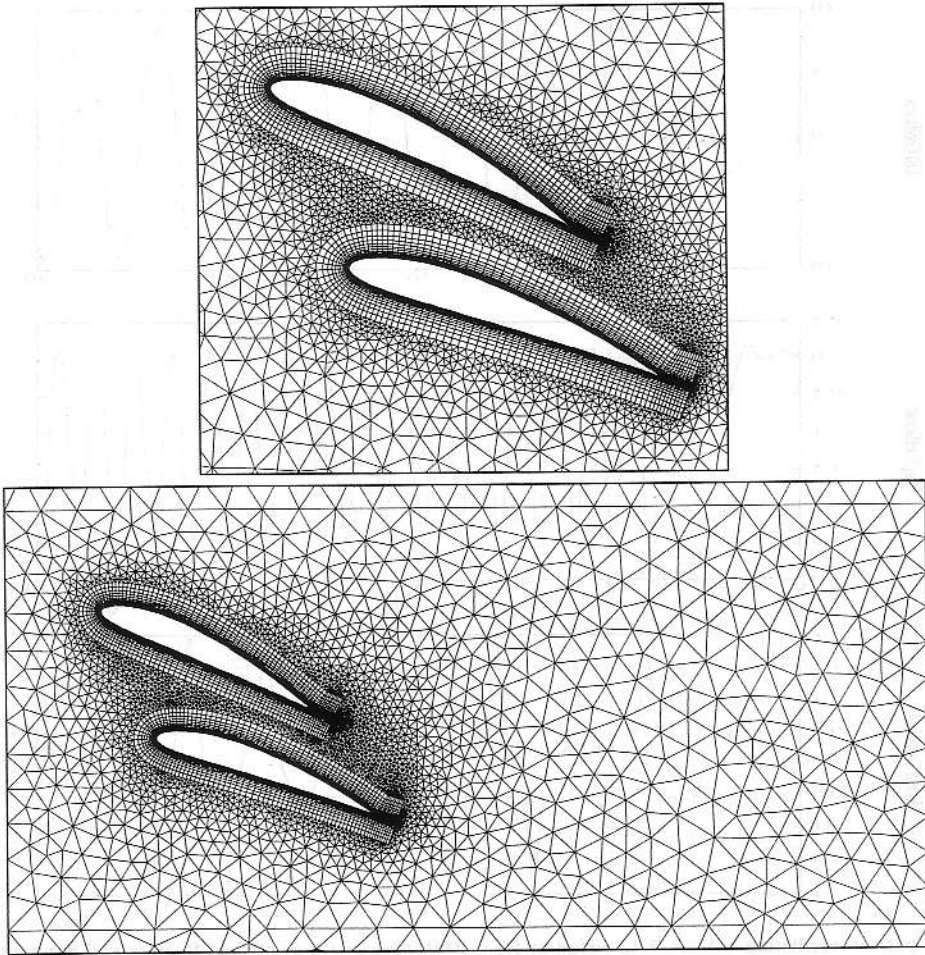


Fig. 14. Flow past two NACA 4412 airfoils: close-up views of the finite element mesh (7310 nodes and 9994 elements).

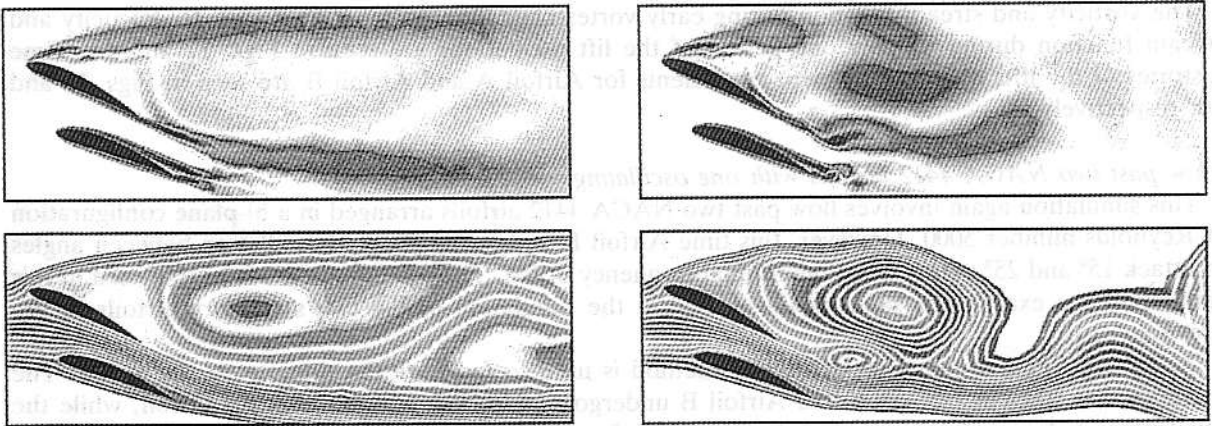


Fig. 15. Flow past two NACA 4412 airfoils at $Re = 5000$: vorticity and stream function at initial condition (steady-state solution).

Fig. 16. Flow past two NACA 4412 airfoils at $Re = 5000$: vorticity and stream function during early vortex shedding.

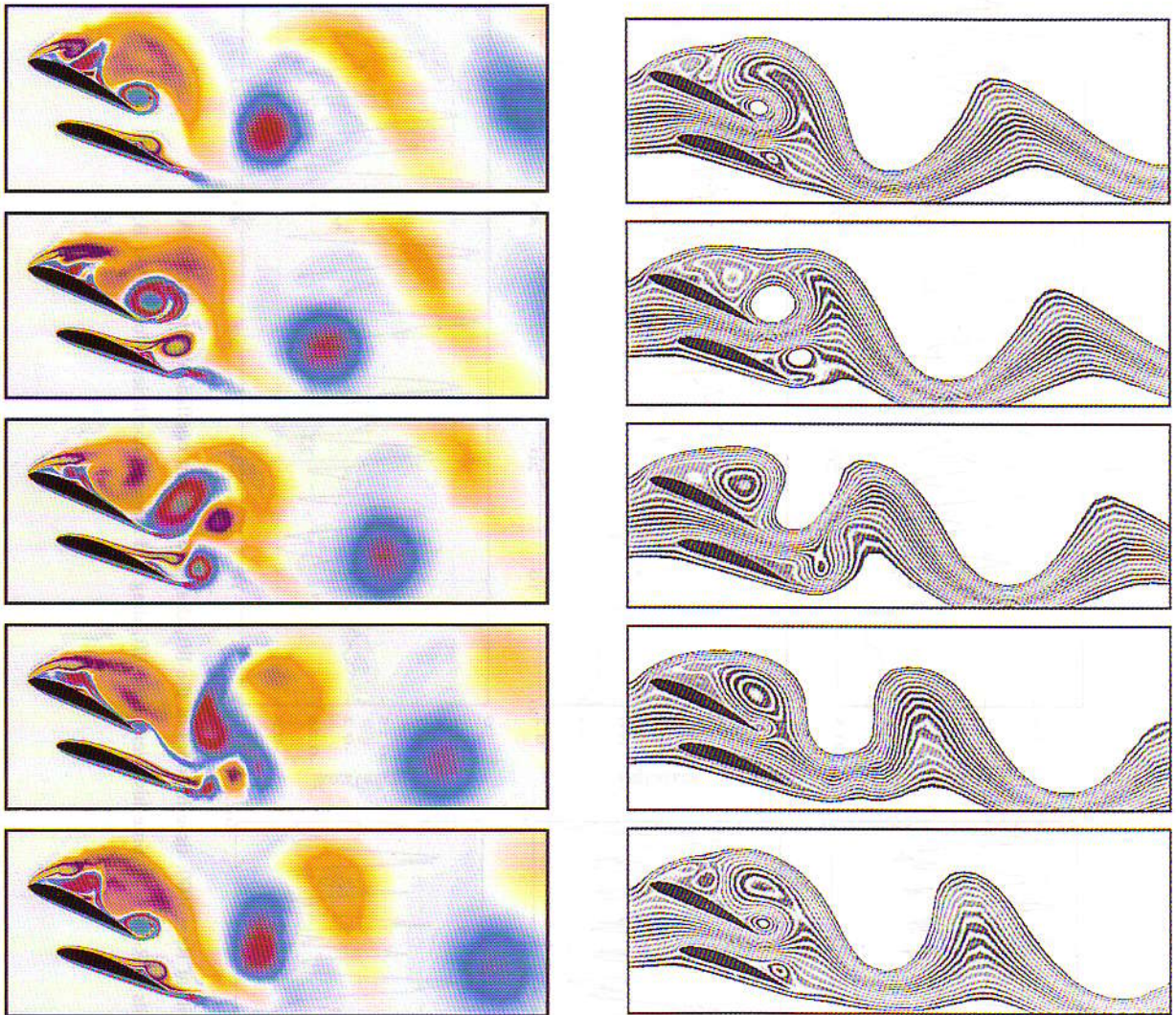


Fig. 17. Flow past two NACA 4412 airfoils at $Re = 5000$: vorticity at various instants during one period of the lift oscillations.

Fig. 18. Flow past two NACA 4412 airfoils at $Re = 5000$: stream function at various instants during one period of the lift oscillations.

22. A close-up view of the mesh at two instants during one half cycle of the airfoil motion is shown in Fig. 23. Time histories of the lift, drag and moment coefficient for Airfoil A and Airfoil B can be seen in Figs. 24 and 25, respectively.

Flow through a sluice gate

This last problem involves the instantaneous release of a gate which separates the deep and shallow regions of a reservoir. The gravity causes a surge of fluid, which is initially at rest, to go through the gate and create a large breaking wave in the shallow reservoir. The computations are carried out until the tip of the breaking wave comes in contact with the fluid. This problem was previously computed by other researchers with the MAC method [22]. The non-dimensional parameters for this simulation are as follows

$$\begin{aligned} \nu &= 0.0343, & h &= 1.0, \\ \rho &= 1.0, & g &= 1.0, \\ H_{\text{begin}} &= 9.1, & H_{\text{end}} &= 7.6, \end{aligned}$$

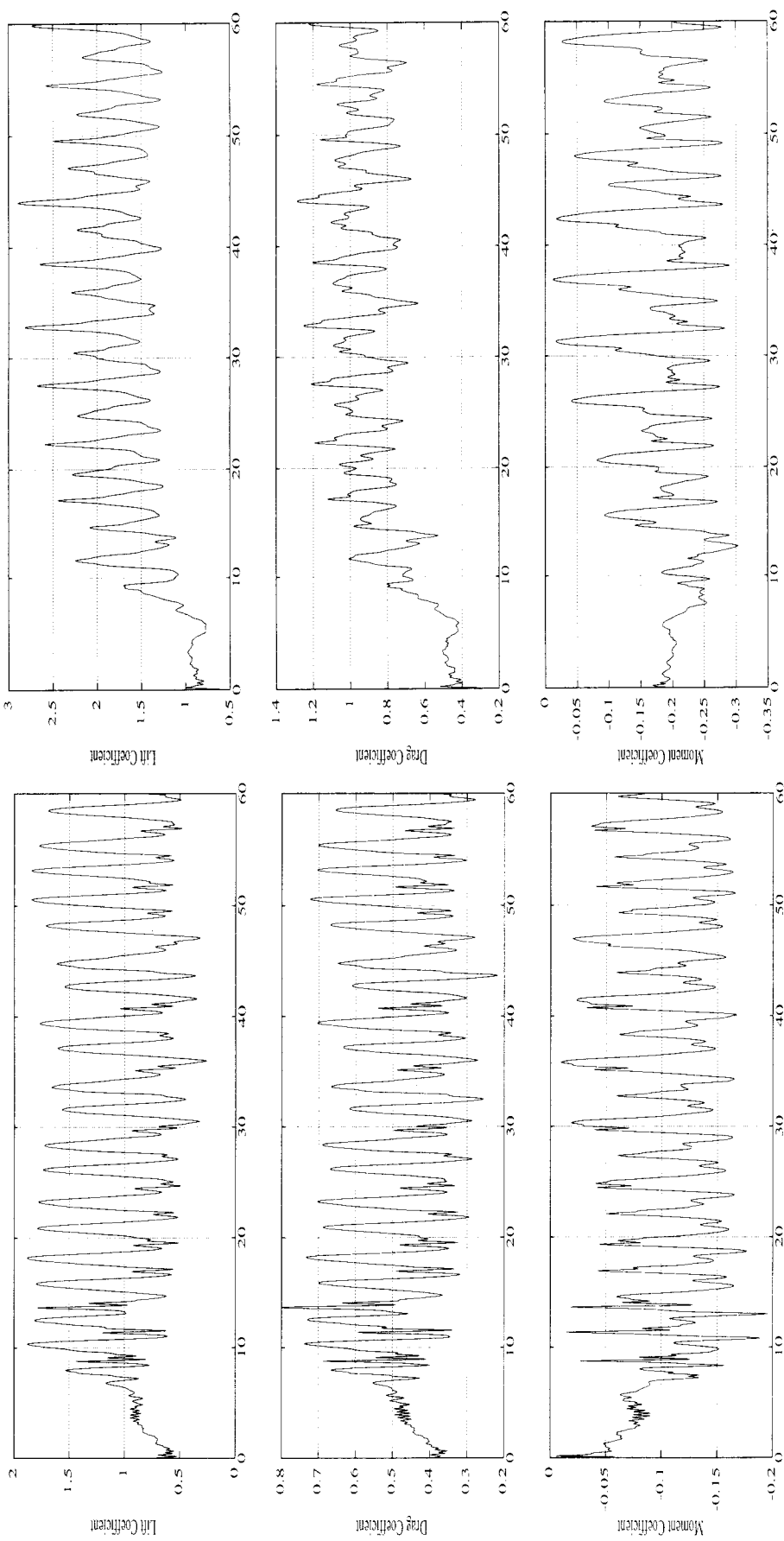


Fig. 19. Flow past two NACA 4412 airfoils at $Re = 5000$: time histories of the lift, drag and moment coefficients for Airfoil A.

Fig. 20. Flow past two NACA 4412 airfoils at $Re = 5000$: time histories of the lift, drag and moment coefficients for Airfoil B.

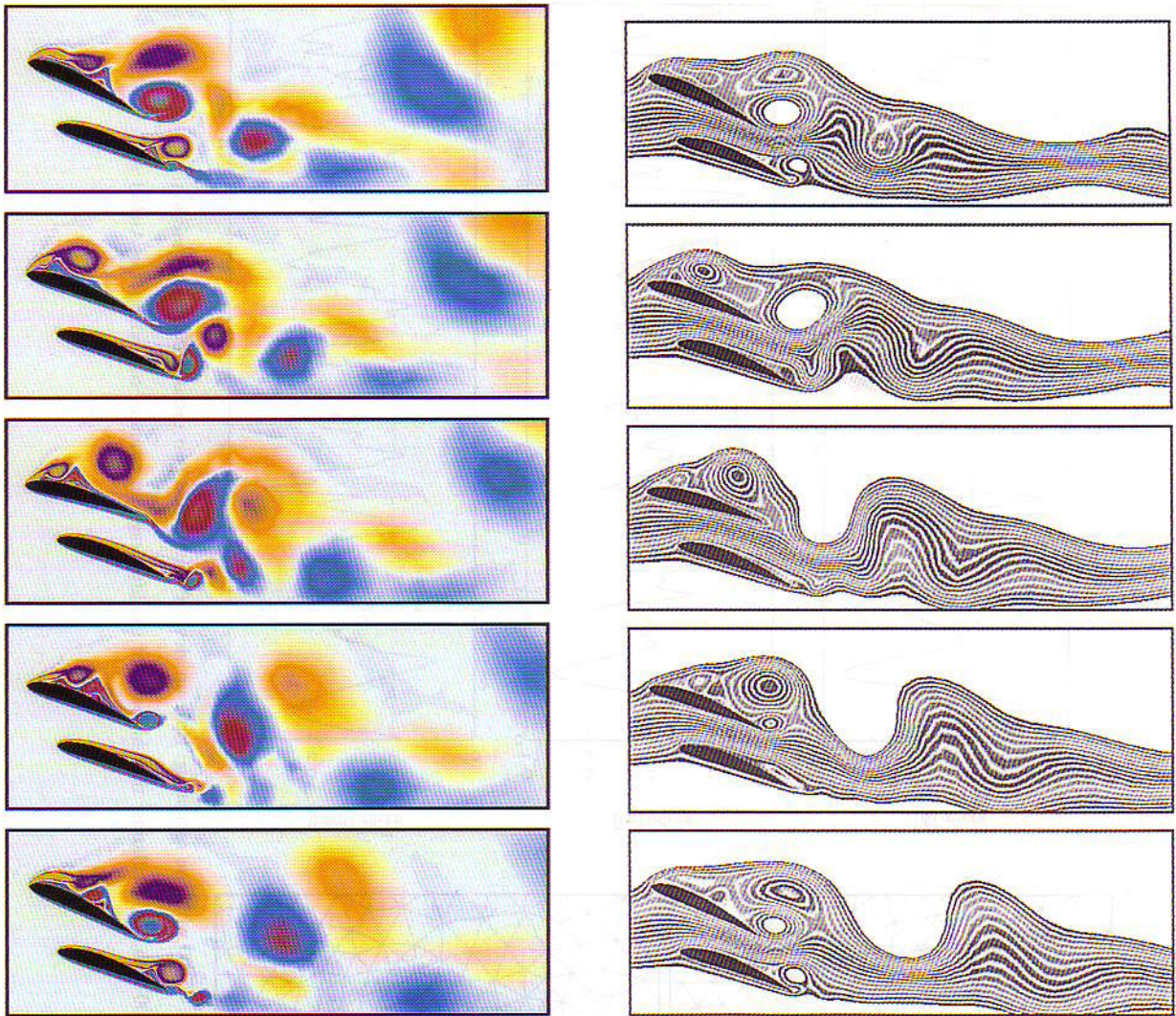


Fig. 21. Flow past two NACA 4412 airfoils at $Re = 5000$ with one oscillating: vorticity at various instants during one period of the motion.

Fig. 22. Flow past two NACA 4412 airfoils at $Re = 5000$ with one oscillating: stream function at various instants during one period of the motion.

where h is the gate height, and H_{begin} and H_{end} are the heights of the deep region of the reservoir at the beginning and end of the computation.

In this problem we use automatic mesh moving plus remeshing as needed. The main mechanism is the automatic mesh moving scheme. However, this alone is not sufficient since the breaking wave creates large distortions in the mesh. Because of this, periodic remeshing of the whole domain is carried out. Also, as in the viscous drop problem, the nodes on the free surface are moved tangentially so as to keep the nodal spacing uniform. The remeshing decision is based on a combination of elemental aspect ratio and area change. This computation is carried out 133 time steps with 10 remeshes. The finite element mesh has approximately 2500 triangular elements throughout the computation. The Reynolds number for this simulation is

$$Re = \frac{\sqrt{H}}{\nu} = 88, \quad (24)$$

where H is the height of the deep reservoir at the beginning of the simulation.

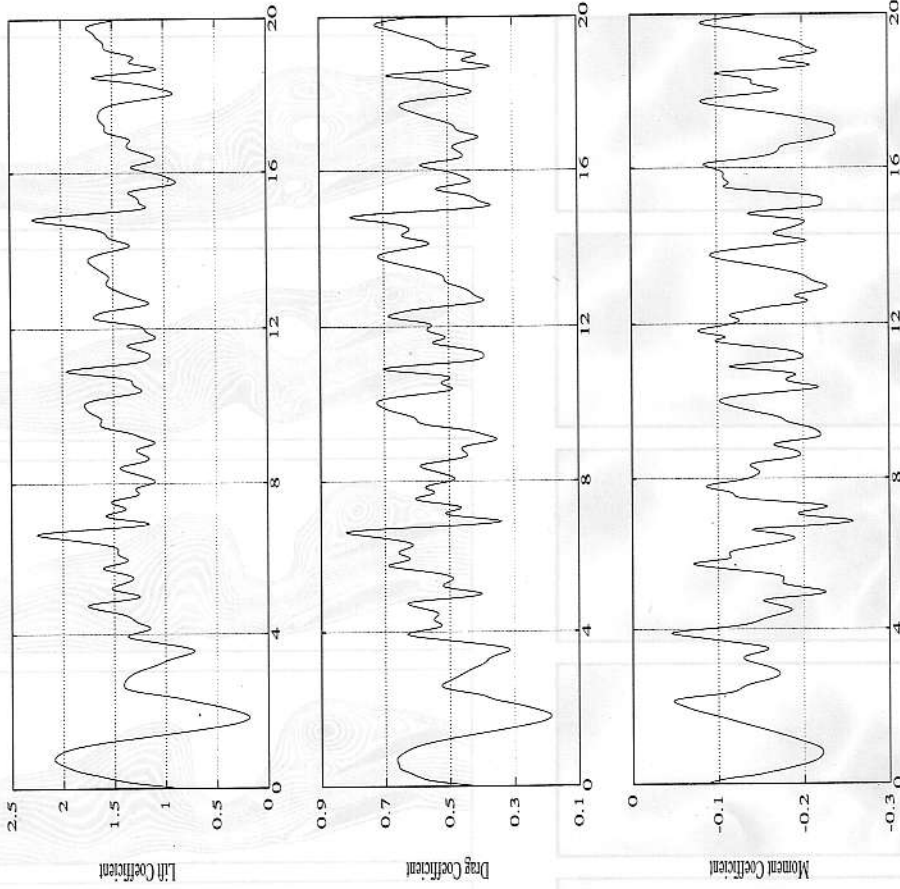
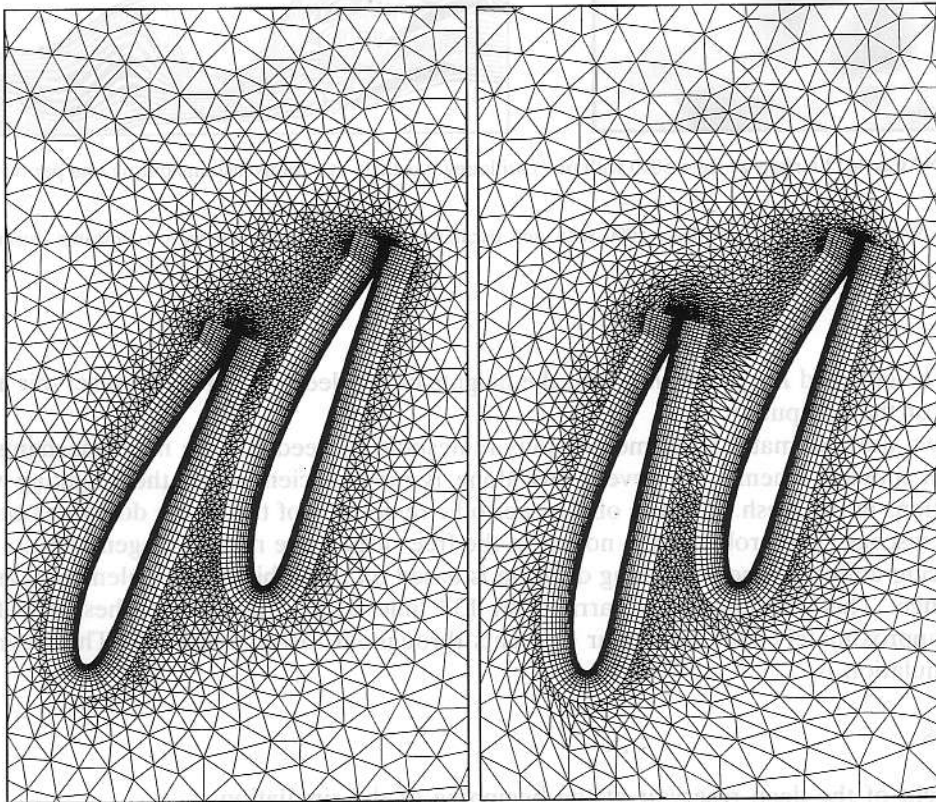


Fig. 23. Flow past two NACA 4412 airfoils at $Re = 5000$ with one oscillating: close-up view of the mesh at two instants during the motion.

Fig. 24. Flow past two NACA 4412 airfoils at $Re = 5000$ with one oscillating: time histories of the lift, drag and moment coefficients for Airfoil A.

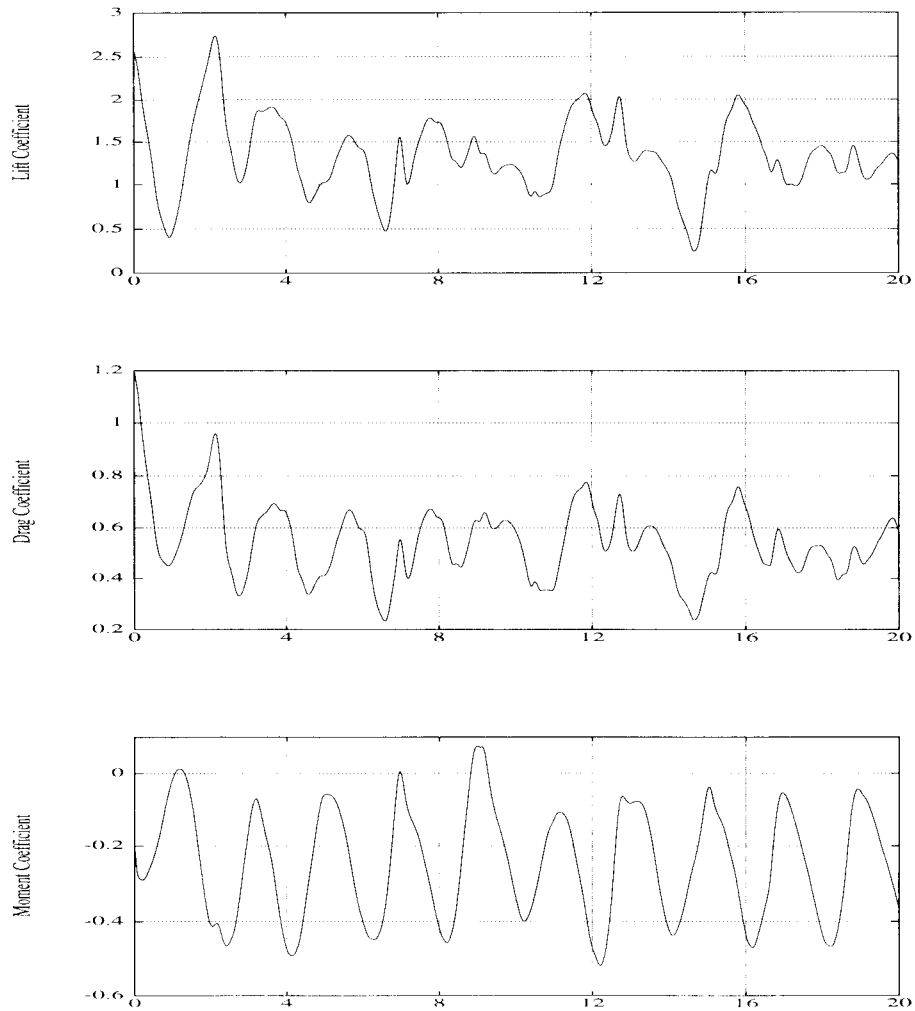


Fig. 25. Flow past two NACA 4412 airfoils at $Re = 5000$ with one oscillating: time histories of the lift, drag and moment coefficients for Airfoil B.

Fig. 26 shows close-up views of the finite element mesh at several instants during the computation. Fig. 27 shows the vorticity field at those same instants.

5. Concluding remarks

We presented our mesh update strategies for parallel finite element computation of flow problems with moving boundaries and interfaces. We use these strategies in conjunction with the stabilized space-time finite element formulations for flow problems with free surfaces, two-liquid interfaces, moving mechanical components, and fluid-structure and fluid-particle interactions. We designed special and automatic mesh moving schemes to minimize the frequency of remeshing. This way, we not only reduce the projection errors involved in remeshing, but also reduce the cost involved in mesh generation and parallelization set-up. Although these costs may not be so significant in simple 2D problems, they could become prohibitively high in more complicated cases, especially in 3D problems and when the parallelization set-up involves mesh partitioning techniques. Our strategies include those

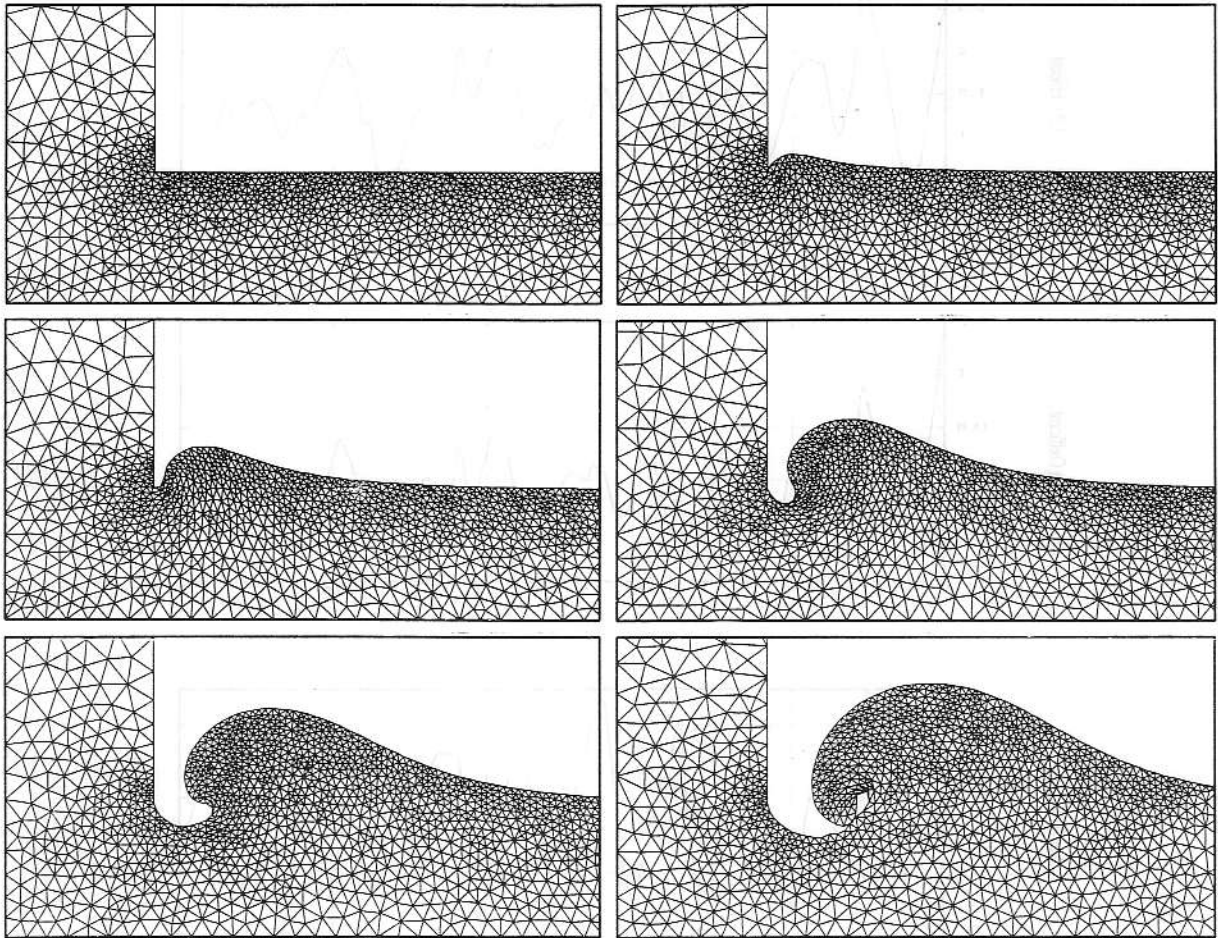


Fig. 26. Flow through a sluice gate: close-up view of the mesh at various instants during the simulation.

designed for combinations of structured and unstructured meshes. We presented several examples of incompressible flow problems computed on the Connection Machines with these mesh update strategies.

Acknowledgment

This research was sponsored by NASA-JSC under grant NAG 9-449, by NSF under grant CTS-8796352, and by ARPA under NIST contract 60NANB2D1272. Partial support for this work has also come from the ARO contract number DAAL03-89-C-0038 with the AHPCRC at the University of Minnesota.

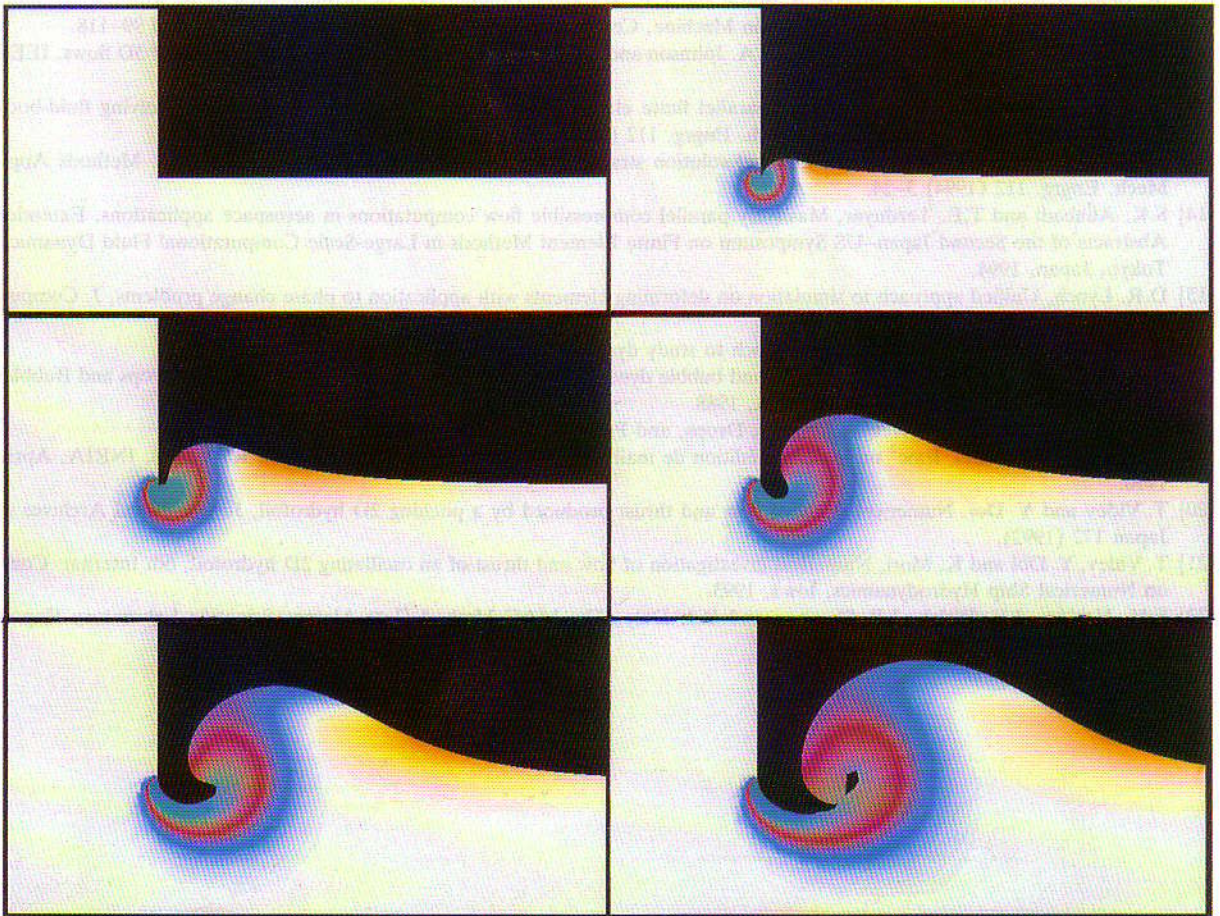


Fig. 27. Flow through a sluice gate: vorticity at various instants during the simulation.

References

- [1] T.E. Tezduyar, M. Behr and J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces – the deformable-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 boundaries and interfaces – the deformable-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 and T.J.R. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations, AIAA Paper 83-0125, Proc. AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, 1983.
- [4] S.K. Aliabadi, S.E. Ray and T.E. Tezduyar, SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations, *Comput. Mech.* 11 (1993) 300–312.
- [5] T.J.R. Hughes, L.P. Franca and G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations, *Comput. Methods Appl. Mech. Engrg.* 75 (1989) 173–189.
- [6] T.E. Tezduyar, M. Behr, S. Mittal and A.A. Johnson, Computation of unsteady incompressible flows with the stabilize finite element methods: space–time formulations, iterative strategies and massively parallel implementations, *New Methods in Transient Analysis*, AMD – Vol. 143, ASME 1992.
- [7] Z. Johan, Data parallel finite element techniques for large-scale computational fluid dynamics, Ph.D. Thesis, Department of Mechanical Engineering, Stanford University, 1992.
- [8] T.E. Tezduyar, S.K. Aliabadi, M. Behr, A.A. Johnson and S. Mittal, Massively parallel finite element computation of three-dimensional flow problems, Proc. 6th Japan Numerical Fluid Dynamics Symposium, Tokyo, Japan, 1992.
- [9] S.K. Aliabadi and T.E. Tezduyar, Space–time finite element computation of compressible flows involving moving boundaries and interfaces, *Comput. Methods Appl. Mech. Engrg.* 107 (1993) 209–223.

- [10] M. Behr, A.A. Johnson, J. Kennedy, S. Mittal and T.E. Tezduyar, Computation of incompressible flows with implicit finite element implementations on the Connection Machine, *Comput. Methods Appl. Mech. Engrg.* 108 (1993) 99–118.
- [11] T.E. Tezduyar, S.K. Aliabadi, M. Behr, A.A. Johnson and S. Mittal, Parallel finite element computation of 3D flows, ~~IEEE Computer, October (1993) 27–36.~~ *Computer*, **26** (1993) 27-36.
- [12] S. Mittal and T.E. Tezduyar, Massively parallel finite element computation of incompressible flows involving fluid-body interactions, *Comput. Methods Appl. Mech. Engrg.* 112 (1994) 253–282.
- [13] M. Behr and T.E. Tezduyar, Finite element solution strategies for large-scale flow simulations, *Comput. Methods Appl. Mech. Engrg.* 112 (1994) 3–24.
- [14] S.K. Aliabadi and T.E. Tezduyar, Massively parallel compressible flow computations in aerospace applications, *Extended Abstracts of the Second Japan–US Symposium on Finite Element Methods in Large-Scale Computational Fluid Dynamics*, Tokyo, Japan, 1994.
- [15] D.R. Lynch, Unified approach to simulation on deforming elements with application to phase change problems, *J. Comput. Phys.* 47 (1982) 187–411.
- [16] T. Reu and S.X. Ying, Hybrid grid approach to study dynamic stall, *AIAA J.* 30 (1992) 2670–2676.
- [17] L.G. Leal, *Computational studies of drop and bubble dynamics in a viscous fluid*, AIP Conf. Proc. 197: Drops and Bubbles (3rd Internat. Colloquium), Monterey, CA, 1988.
- [18] R. Clift, J. Grace and M. Weber, *Bubbles, Drops, and Particles* (Academic Press, 1978).
- [19] F. Hecht and E. Salel, Eme² un logiciel d'édition de maillages et de contours bidimensionnels, RT n° 118, INRIA, April, 1990.
- [20] T. Videv and Y. Doi, Numerical study of flow and thrust produced by a pitching 2D hydrofoil, *J. Soc. Naval Archives of Japan* 172 (1992).
- [21] T. Videv, Y. Doi and K. Mori, Numerical investigation of flow and thrust of an oscillating 2D hydrofoil, 6th Internat. Conf. on Numerical Ship Hydrodynamics, Iowa, 1993.
- [22] F.H. Harlow, J.E. Welch, J.P. Shannon and B.J. Daly, *The MAC Method* (Los Alamos Scientific Laboratory Report LA-3425, 1965).