Parallel Computational Fluid Dynamics
Towards Teraflops, Optimization and Novel Formulations
D. Keyes, A. Ecer, J. Periaux, N. Satofuka and P. Fox (Editors)

Parallel Solution of Multibody Store Separation Problems by a Fictitious Domain Method

T.-W. Pan[a], V. Sarin[b], R. Glowinski[a], J. Périaux[c] and A. Sameh[b]

[a]Department of Mathematics, University of Houston, Houston, TX 77204, USA

[b]Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

[c]Dassault Aviation, 92214 Saint-Cloud, France

The numerical simulation of interaction between fluid and complex geometries, e.g., multibody store separation, is computationally expensive and parallelism often appears as the only way toward large scale of simulations, even if we have a fast Navier-Stokes solver. The method we advocate here is a combination of a distributed Lagrange multiplier based fictitious domain method and operator splitting schemes. This method allows the use of a *fixed structured finite element grid* on a simple shape auxiliary domain containing the actual one for the entire fluid flow simulation. It can be easily parallelized and there is no need to generate a new mesh at each time step right after finding the new position of the rigid bodies. Numerical results of multibody store separation in an incompressible viscous fluid on an SGI Origin 2000 are presented.

1. FORMULATION

In this article, we consider the numerical simulation of multibody store separation in an incompressible viscous fluid by a distributed Lagrange multiplier/fictitious domain method (see refs. [1, 2]). The motion of the rigid body, such as the NACA0012 airfoil, is not known a priori and is due to the hydrodynamical forces and gravity. In the simulation we do not need to compute the hydrodynamical forces explicitly, since the interaction between fluid and rigid bodies is implicitly modeled by the global variational formulation at the foundation of the methodology employed here. This method offers an alternative to the ALE methods investigated in [3], [4], and [5].

Let us first describe the variational formulation of a distributed Lagrange multiplier based fictitious domain method. Let $\Omega \subset \mathbb{R}^d (d = 2, 3$; see Figure 1 for a particular case where $d = 2)$ be a space region; for simplicity we suppose that Ω is filled with a Newtonian viscous incompressible fluid (of density ρ_f and viscosity ν_f) and contains a moving rigid body B of density ρ_s; the incompressible viscous flow is modeled by the Navier-Stokes equations and the motion of the ball is described by the Euler's equations (an almost

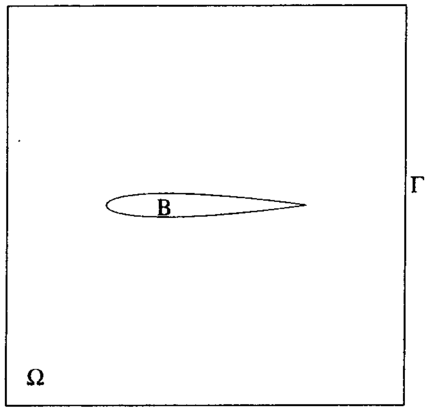

Figure 1: An example of two-dimensional flow region with one rigid body.

direct consequence of Newton's laws of motion). With the following functional spaces

$$W_{\mathbf{g}_0}(t) = \{\mathbf{v}|\mathbf{v} \in H^1(\Omega)^d,\ \mathbf{v} = \mathbf{g}_0(t) \ on\ \Gamma\},$$
$$L_0^2(\Omega) = \{q|q \in L^2(\Omega),\ \int_\Omega q d\mathbf{x} = 0\}, \quad \Lambda(t) = H^1(B(t))^d,$$

the *fictitious domain formulation with distributed Lagrange multipliers* for flows around freely moving rigid bodies (see [2] for detail) is as follows

For a.e. $t > 0$, *find* $\{\mathbf{u}(t), p(t), \mathbf{V_G}(t), \mathbf{G}(t), \boldsymbol{\omega}(t), \boldsymbol{\lambda}(t)\}$ *such that*

$$\mathbf{u}(t) \in W_{\mathbf{g}_0}(t),\ p(t) \in L_0^2(\Omega),\ \mathbf{V_G}(t) \in \mathbb{R}^d,\ \mathbf{G}(t) \in \mathbb{R}^d,\ \boldsymbol{\omega}(t) \in \mathbb{R}^3,\ \boldsymbol{\lambda}(t) \in \Lambda(t)$$

and

$$\begin{cases} \rho_f \int_\Omega \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} d\mathbf{x} + \rho_f \int_\Omega (\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} \cdot \mathbf{v} d\mathbf{x} - \int_\Omega p \boldsymbol{\nabla} \cdot \mathbf{v} d\mathbf{x} \\ \quad +2\nu_f \int_\Omega \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) dx - < \boldsymbol{\lambda}, \mathbf{v} - \mathbf{Y} - \boldsymbol{\theta} \times \overrightarrow{\mathbf{Gx}} >_{\Lambda(t)} \\ \quad +(1 - \frac{\rho_f}{\rho_s})[M \frac{d\mathbf{V_G}}{dt} \cdot \mathbf{Y} + (\mathbf{I} \frac{d\omega}{dt} + \omega \times \mathbf{I}\omega) \cdot \boldsymbol{\theta}] \\ = (1 - \frac{\rho_f}{\rho_s}) M \mathbf{g} \cdot \mathbf{Y} + \rho_f \int_\Omega \mathbf{g} \cdot \mathbf{v} d\mathbf{x},\ \forall \mathbf{v} \in H_0^1(\Omega)^d,\ \forall \mathbf{Y} \in \mathbb{R}^d,\ \forall \boldsymbol{\theta} \in \mathbb{R}^3, \end{cases} \tag{1}$$

$$\int_\Omega q \boldsymbol{\nabla} \cdot \mathbf{u}(t) d\mathbf{x} = 0,\ \forall q \in L^2(\Omega), \tag{2}$$
$$\frac{d\mathbf{G}}{dt} = \mathbf{V_G}, \tag{3}$$
$$< \boldsymbol{\mu}, \mathbf{u}(t) - \mathbf{V_G}(t) - \boldsymbol{\omega}(t) \times \overrightarrow{\mathbf{G}(t)\mathbf{x}} >_{\Lambda(t)} = 0,\ \forall \boldsymbol{\mu} \in \Lambda(t), \tag{4}$$
$$\mathbf{V_G}(0) = \mathbf{V}^0,\ \boldsymbol{\omega}(0) = \boldsymbol{\omega}^0,\ \mathbf{G}(0) = \mathbf{G}^0; \tag{5}$$
$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}),\ \forall \mathbf{x} \in \Omega \backslash \overline{B(0)}\ and\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{V}^0 + \boldsymbol{\omega}^0 \times \overrightarrow{\mathbf{G}^0\mathbf{x}},\ \forall \mathbf{x} \in \overline{B(0)}. \tag{6}$$

In (1)-(6), $\mathbf{u}(=\{u_i\}_{i=1}^d)$ and p denote velocity and pressure respectively, $\boldsymbol{\lambda}$ is a Lagrange multiplier, $\mathbf{D}(\mathbf{v})=(\nabla\mathbf{v}+\nabla\mathbf{v}^t)/2$, $\mathbf{g}$ is the gravity, $\mathbf{V_G}$ is the translation velocity of the mass center of the rigid body B, $\boldsymbol{\omega}$ is the angular velocity of B, M is the mass of the rigid body, $\mathbf{I}$ is the inertia tensor of the rigid body at $\mathbf{G}$, $\mathbf{G}$ is the center of mass of B; $\boldsymbol{\omega}(t)=\{\omega_i(t)\}_{i=1}^3$ and $\boldsymbol{\theta}=\{\theta_i\}_{i=1}^3$ if $d=3$, while $\boldsymbol{\omega}(t)=\{0,0,\omega(t)\}$ and $\boldsymbol{\theta}=\{0,0,\theta\}$ if $d=2$. From the rigid body motion of B, $\mathbf{g}_0$ has to satisfy $\int_\Gamma \mathbf{g}_0\cdot\mathbf{n}d\Gamma=0$, where $\mathbf{n}$ denotes the unit vector of the outward normal at Γ (we suppose the *no-slip* condition on ∂B). We also use, if necessary, the notation $\phi(t)$ for the function $\mathbf{x}\to\varphi(\mathbf{x},t)$.

Remark 1. The hydrodynamics forces and torque imposed on the rigid body by the fluid are built in (1)-(6) implicitly (see [2] for detail), hence we do not need to compute them explicitly in the simulation. Since in (1)-(6) the flow field is defined on the entire domain Ω, it can be computed with a simple structured grid. Then by (4), we can enforce the rigid body motion in the region occupied by the rigid bodies via Lagrange multipliers.

Remark 2. In the case of Dirichlet boundary conditions on Γ, and taking the incompressibility condition $\nabla\cdot\mathbf{U}=0$ into account, we can easily show that

$$2\nu_f\int_\Omega \mathbf{D}(\mathbf{U}):\mathbf{D}(\mathbf{v})d\mathbf{x}=\nu_f\int_\Omega \nabla\mathbf{U}:\nabla\mathbf{v}d\mathbf{x},\ \forall\mathbf{v}\in W_0, \tag{7}$$

which, from a *computational* point of view, leads to a substantial simplification in (1)-(6).

2. APPROXIMATION

Concerning the space approximation of the problem (1)-(6) by finite element methods, we use P_1isoP_2 and P_1 finite elements for the velocity field and pressure, respectively (see [6] for details). Then for discretization in time we apply an operator-splitting technique à la Marchuk-Yanenko [7] to decouple the various computational difficulties associated with the simulation. In the resulting discretized problem, there are three major subproblems: (i) a divergence-free projection subproblem, (ii) a linear advection-diffusion subproblem, and (iii) a rigid body motion projection subproblem. Each of these subproblems can be solved by conjugate gradient methods (for further details, see ref. [2]).

3. PARALLELIZATION

For the divergence-free projection subproblems, we apply a conjugate gradient algorithm preconditioned by the discrete equivalent of $-\Delta$ for the homogeneous Neumann boundary condition; such an algorithm is described in [8]. In this article, the numerical solution of the Neumann problems occurring in the treatment of the divergence-free condition is achieved by a parallel multilevel Poisson solver developed by Sarin and Sameh [9].

The advection-diffusion subproblems are solved by a least-squares/conjugate-gradient algorithm [10] with two or three iterations at most in the simulation. The arising linear systems associated with the discrete elliptic problems have been solved by the Jacobi iterative method, which is easy to parallelize.

Finally, the subproblems associated with rigid body motion projection can also be solved by an Uzawa/conjugate gradient algorithm (in which there is no need to solve any elliptic problems); such an algorithm is described in [1] and [2].

Due to the fact that the distributed Lagrange multiplier method uses *uniform meshes* on a rectangular domain and relies on matrix-free operations on the velocity and pressure unknowns, this approach simplifies the distribution of data on parallel architectures and ensures very good load balancing. The basic computational kernels comprising of vector operations such as additions and dot products, and matrix-free matrix-vector products yield nice scalability on distributed shared memory computers such as the SGI Origin 2000.

4. NUMERICAL RESULTS

In this article, the parallelized code of algorithm (1)-(6) has been applied to simulate multibody store separation in a 2D channel with non-spherical rigid bodies. There are three NACA0012 airfoils in the channel. The characteristic length of the fixed NACA0012 airfoil is 1.25 and those of the two moving ones are 1. The x_1 and x_2 dimensions of the channel are 16.047 and 4 respectively. The density of the fluid is $\rho_f = 1.0$ and the density of the particles is $\rho_s = 1.1$. The viscosity of the fluid is $\nu_f = 0.001$. The initial condition for the fluid flow is $\mathbf{u} = \mathbf{0}$. The boundary condition on $\partial\Omega$ of velocity field is

$$\mathbf{u}(x_1, x_2, t) = \begin{cases} \mathbf{0}, & if\ x_2 = -2,\ or,\ 2, \\ \begin{pmatrix} 0 \\ (1.0 - e^{-50t})(1 - x_2^2/4) \end{pmatrix}, & if\ x_1 = -4,\ or,\ 16.047 \end{cases}$$

for $t \geq 0$. Hence the Reynolds number is 1000 with respect to the characteristic length of the two smaller airfoils and the maximal in-flow speed. The initial mass centers of the three NACA0012 airfoils are located at $(0.5, 1.5)$, $(1, 1.25)$, and $(-0.25, 1.25)$. Initial velocities and angular velocities of the airfoils are zeroes. The time step is $\Delta t = 0.0005$. The mesh size for the velocity field is $h_v = 2/255$. The mesh size for pressure is $h_p = 2h_v$.

An example of a part of the mesh for the velocity field and an example of mesh points for enforcing the rigid body motion in NACA0012 airfoils are shown in Figure 2. All three NACA0012 airfoils are fixed up to $t = 1$. After $t = 1$, we allow the two smaller airfoils to move freely. These two smaller NACA0012 airfoils keep their stable orientations when they are moving downward in the simulation. Flow field visualizations and density plots of the vorticity obtained from numerical simulations (done on 4 processors) are shown in Figures 3 and 4.

In Table 1, we have observed overall algorithmic speed-up of 15.08 on 32 processors, compared with the elapsed time on one processor. In addition, we also obtain an impressive about thirteen-fold increase in speed over the serial implementation on a workstation, a DEC alpha-500au, with 0.5 GB RAM and 500MHz clock speed.

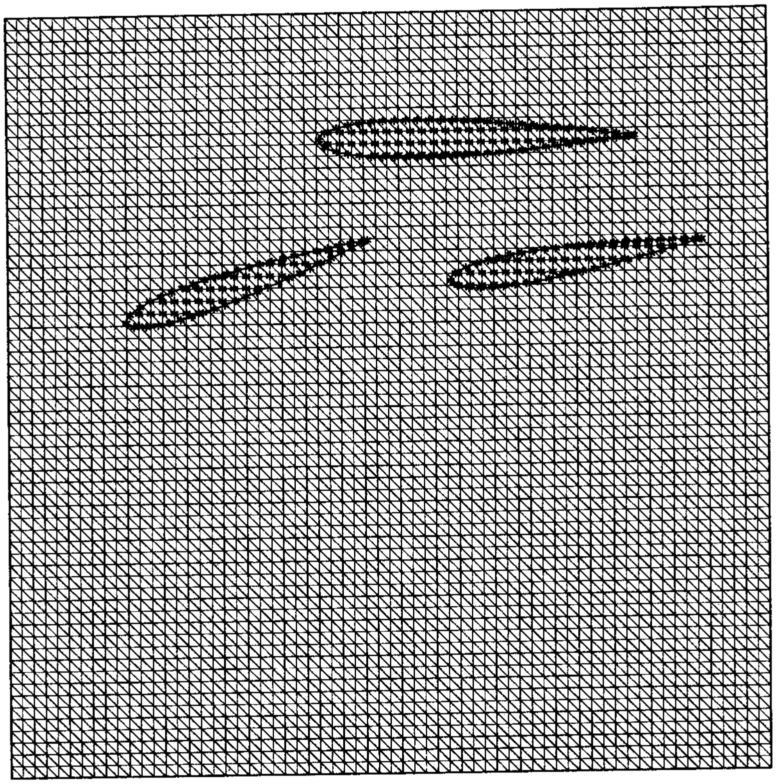

Figure 2. Part of the velocity mesh and example of mesh points for enforcing the rigid body motion in the NACA0012 airfoils with $h_v = 3/64$.

5. CONCLUSION

We have presented in this article a distributed Lagrange multiplier based fictitious domain method for the simulation of flow with moving boundaries. Some preliminary experiments of parallelized code have shown the potential of this method for the direct simulation of complicated flow. In the future, our goal is to develop portable 3D code with the ability to simulate large scale problems on a wide variety of architectures.

Table 1 :

Elapsed time/time step and algorithmic speed-up on a SGI Origin 2000

	Elapsed Time	Algorithmic speed-up
1 processor*	146.32 sec.	1
2 processors	97.58 sec.	1.50
4 processors	50.74 sec.	2.88
8 processors	27.25 sec.	5.37
16 processors	15.82 sec.	9.25
32 processors	9.70 sec.	15.08

* The sequential code took about 125.26 sec./time step on a DEC alpha-500au.

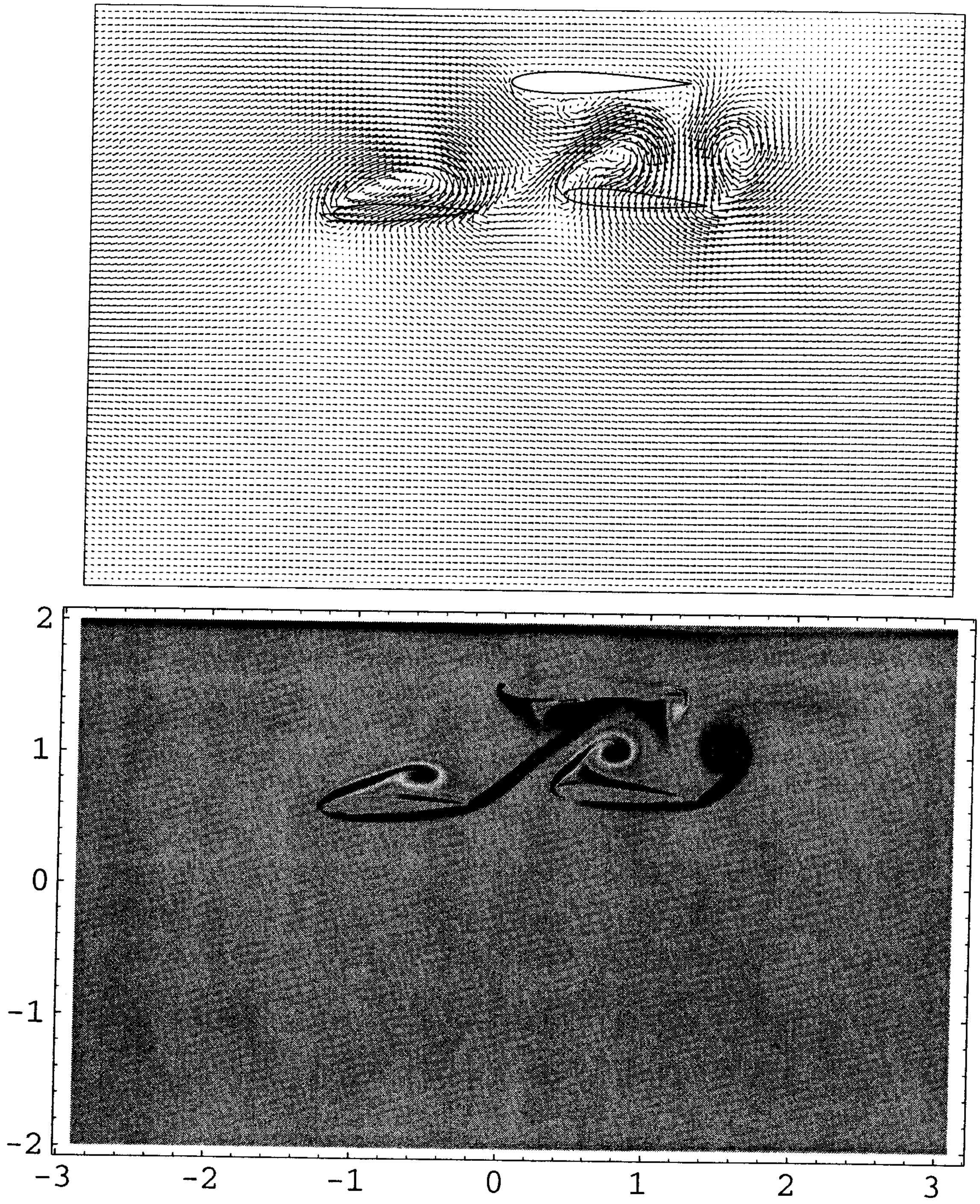

Figure 3. Flow field visualization (top) and density plot of the vorticity (bottom) around the NACA0012 airfoils at $t = 1.5$.

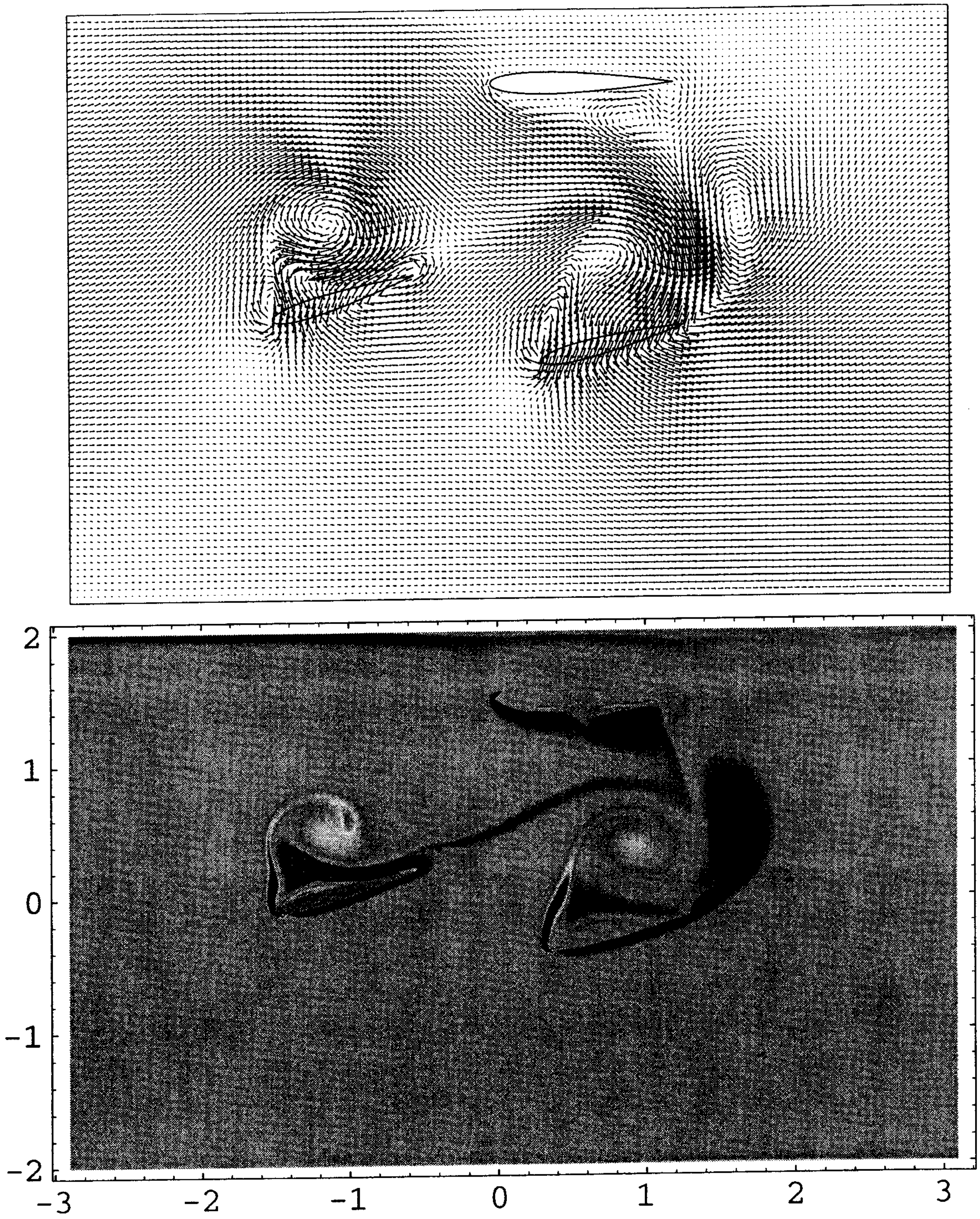

Figure 4. Flow field visualization (top) and density plot of the vorticity (bottom) around the NACA0012 airfoils at $t = 2$.

6. ACKNOWLEDGMENTS

We acknowledge the helpful comments and suggestions of E. J. Dean, V. Girault, J. He, Y. Kuznetsov, B. Maury, and G. Rodin, and also the support of the department of Computer Science at the Purdue University concerning the use of an SGI Origin 2000. We acknowledge also the support of the NSF (grants CTS-9873236 and ECS-9527123) and Dassault Aviation.

REFERENCES

[1] R. Glowinski, T.-W. Pan, T. Hesla, D.D. Joseph, J. Périaux, A fictitious domain method with distributed Lagrange multipliers for the numerical simulation of particulate flows, in J. Mandel, C. Farhat, and X.-C. Cai (eds.), Domain Decomposition Methods 10, AMS, Providence, RI, 1998, 121-137.

[2] R. Glowinski, T.W. Pan, T.I. Hesla, and D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Internat. J. of Multiphase Flow, 25 (1999), 755-794.

[3] H.H. Hu, Direct simulation of flows of solid-liquid mixtures, Internat. J. Multiphase Flow, 22 (1996), 335-352.

[4] A. Johnson, T. Tezduyar, 3D Simulation of fluid-particle interactions with the number of particles reaching 100, Comp. Meth. Appl. Mech. Eng., 145 (1997), 301-321.

[5] B. Maury, R. Glowinski, Fluid particle flow: a symmetric formulation, C.R. Acad. Sci., Série I, Paris, t. 324 (1997), 1079-1084.

[6] M.O. Bristeau, R. Glowinski, J. Périaux, Numerical methods for the Navier-Stokes equations. Applications to the simulation of compressible and incompressible viscous flow, Computer Physics Reports, 6 (1987), 73-187.

[7] G.I. Marchuk, Splitting and alternating direction methods, in P.G. Ciarlet and J.L. Lions (eds.), Handbook of Numerical Analysis, Vol. I, North-Holland, Amsterdam, 1990, 197-462.

[8] R. Glowinski, Finite element methods for the numerical simulation of incompressible viscous flow. Introduction to the control of the Navier-Stokes equations, in C.R. Anderson *et al.* (eds), Vortex Dynamics and Vortex Methods, Lectures in Applied Mathematics, AMS, Providence, R.I., 28 (1991), 219-301.

[9] V. Sarin, A. Sameh, An efficient iterative method for the generalized Stokes problem, SIAM J. Sci. Comput., 19 (1998), 206-226. 2, 335-352.

[10] R. Glowinski, Numerical methods for nonlinear variational problems, Springer-Verlag, New York, 1984.