Fictitious Domain Methods for Viscous Flow Simulation

R.. Glowinsky   A.J. Kearsley
T.W. Pan   J. Periaux

May 1995

TR95-16

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
MAY 1995

2. REPORT TYPE

3. DATES COVERED
00-00-1995 to 00-00-1995

4. TITLE AND SUBTITLE
Fictitious Domain Methods for Viscous Flow Simulation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computational and Applied Mathematics Department ,Rice University,6100 Main Street MS 134,Houston,TX,77005-1892

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		28	
unclassified	unclassified	unclassified			

FICTITIOUS DOMAIN METHODS FOR VISCOUS FLOW SIMULATION

Roland Glowinski[1], **Anthony J. Kearsley**[2], **Tsorng-Whay Pan**[3], **Jacques Periaux**[4]

Abstract

We discuss the fictitious domain solution of the Navier–Stokes equations modelling unsteady incompressible viscous flow. The method is based on a Lagrange multiplier treatment of the boundary conditions to be satisfied and is particularly well suited to the treatment of no–slip boundary conditions. This approach allows the use of structured meshes and fast specialized solvers for problems on complicated geometries. Another interesting feature of the fictitious domain approach is that it allows the solution of optimal shape problems without regriding. The resulting methodology is applied to the solution of flow problems including external viscous flow past oscillating rigid body and vortex dynamics of two-dimensional flow modelled by the incompressible Navier–Stokes equations and then to an optimal shape problem for Stokes and Navier–Stokes flows.

1. INTRODUCTION

Fictitious domain methods for partial differential equations are showing interesting possibilities for solving complicated problems motivated by applications from science and engineering (see, for example, [1] and [2] for some impressive illustrations of the above statement). The main reason for the popularity of fictitious domain methods (sometimes called *domain embedding methods*; cf. [3]) is that they allow the use of fairly structured meshes on a simple shape auxiliary domain containing the actual one, therefore allowing the use of fast solvers.

In this article which follows [4–6], we consider the fictitious domain solution of the Navier–Stokes equations modelling the unsteady incompressible Newtonian viscous fluids and apply the resulting methodology to the solution of optimal shape problems for Stokes and Navier–Stokes flows.

The principle of fictitious domain methods discussed here is to solve the problem

[1] Department of Mathematics, University of Houston, Houston, Texas 77204 USA, Université P. et M. Curie, Paris, and CERFACS, Toulouse, France

[2] Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251-1892, USA

[3] Department of Mathematics, University of Houston, Houston, Texas 77204, USA

[4] Dassault Aviation, 92214 Saint–Cloud, France

in a large domain (containing the actual one) with a very simple shape, the fictitious domain, and to impose the boundary condition by the introduction of a Lagrange multiplier on the actual boundary. Its advantage is to allow the numerical treatment on a fixed structured mesh, independent of the actual boundary of bodies which might be moving. Thus the time–consuming construction of a boundary-fitted mesh for each different position of moving bodies can be skipped.

The methods discussed here go far beyond the related work in [7] where only the steady Stokes problem was considered (in the particular case where the boundary of the actual domain is compatible with the finite element mesh used in the auxiliary domain). The methods described in the following sections do not require a strong coupling between the actual boundary discretization and the grid used in the auxiliary domain. It also relies on the splitting methods described in, e.g., [8]–[12]; with these methods one can decouple the numerical treatments of the incompressibility and of the advection, and take advantage of this fact to use the embedding approach in the (linear) incompressibility step only, the advection being treated in the larger domain without concern –in some sense– for the actual boundary.

The content of this article is as follows: In Section 2 we discuss fictitious domain methods for Dirichlet problems and its numerical solution by combination of domain decomposition and fictitious domain methods; then in Section 3 we consider fictitious domain methods for incompressible Navier–Stokes equations. In Section 4 the simulation of external incompressible viscous flow past an oscillating rigid body modelled by Navier–Stokes equations is discussed. In Section 5 we apply the methods of the above sections to study the vortex dynamics of a two–dimensional viscous flow taking place in a disk. In Section 6 we address the fictitious domain solution of optimal shape problems for Stokes and Navier–Stokes flows. Finally, in Section 7 we conclude the paper with some observations and comments on future work.

2. FICTITIOUS DOMAIN METHODS FOR THE DIRICHLET PROBLEM

2.1 A model problem

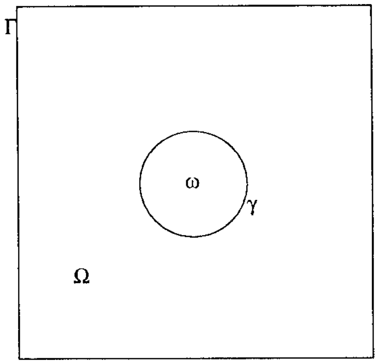

Fig. 1.

We consider the following Dirichlet problem:

$$\alpha u - \nu \Delta u = f \text{ in } \Omega \setminus \bar{\omega}, \tag{1}$$
$$u = g_0 \text{ on } \gamma, \tag{2}$$
$$u = g_1 \text{ on } \Gamma, \tag{3}$$

where Ω is a "box" shaped domain in $\mathbb{R}^d (d \geq 1)$, ω is a bounded domain in $\mathbb{R}^d (d \geq 1)$ such that ω is contained in Ω (e.g., see Fig. 1), Γ (resp., γ) is the boundary $\partial\Omega$ (resp., $\partial\omega$), we suppose that γ is smooth; $\alpha \geq 0$ and $\nu > 0$; finally, f, g_0, and g_1 are given functions defined over $\Omega \setminus \bar{\omega}$, γ and Γ, respectively. If f, g_0, and g_1 are smooth enough, problem (1)–(3) has a unique solution.

A fictitious domain method was already proposed for problem (1)–(3) in [4]. For simplicity, we shall assume that $f \in L^2(\Omega \setminus \bar{\omega})$. We imbed $\Omega \setminus \bar{\omega}$ in Ω and define

$$V_{g_1} = \{v | v \in H^1(\Omega), v = g_1 \text{ on } \Gamma\}. \tag{4}$$

We then have equivalence between (1)–(3) and the following *saddle-point* problem:

Find $\{\tilde{u}, \lambda\} \in V_{g_1} \times L^2(\gamma)$ *such that*

$$\int_\Omega (\alpha \tilde{u} v + \nu \nabla \tilde{u} \cdot \nabla v)\, dx = \int_\Omega \tilde{f} v \, dx + \int_\gamma \lambda v \, d\gamma, \ \forall v \in H_0^1(\Omega), \tag{5}$$

$$\int_\gamma \mu (\tilde{u} - g_0)\, d\gamma = 0, \ \forall \mu \in L^2(\gamma), \tag{6}$$

where $\tilde{f}$ is a $L^2(\Omega)$–extension of f and satisfies $\tilde{f}|_{\Omega\setminus\bar{\omega}} = f$. It can be shown that $\tilde{u}|_{\Omega\setminus\bar{\omega}}$ is the solution of problem (1)–(3) and $\tilde{u} = g$ on γ. In (5)–(6), more precisely we have that the multiplier λ is equal to $\nu[\partial\tilde{u}/\partial\mathbf{n}]|_\gamma$ (i.e., ν times the jump of the normal derivative of $\tilde{u}$ at γ).

Problem (5)–(6) can be solved by the general conjugate gradient methodology proposed in [4]. For the cases where $\omega \subset \mathbb{R}^2$ with a smooth boundary γ, we have obtained by *Fourier Analysis* a quasi–optimal preconditioned conjugate gradient algorithm for the solution of (5)–(6) (see [4]).

2.2 Domain decomposition/fictitious domain approach

Motivated by computation with nonmatching grids on MIMD machines, we consider the numerical solution of Dirichlet problems by a combination of domain decomposition and fictitious domain methods. We take advantage of the fact that the Steklov–Poincaré operators associated with the subdomain interfaces and with the fictitious domain treatment of internal boundaries have very similar properties. We use these properties to derive fast solution methods of the conjugate gradient type with good parallelization properties which simultaneously force the matching at subdomain interfaces and the actual boundary conditions.

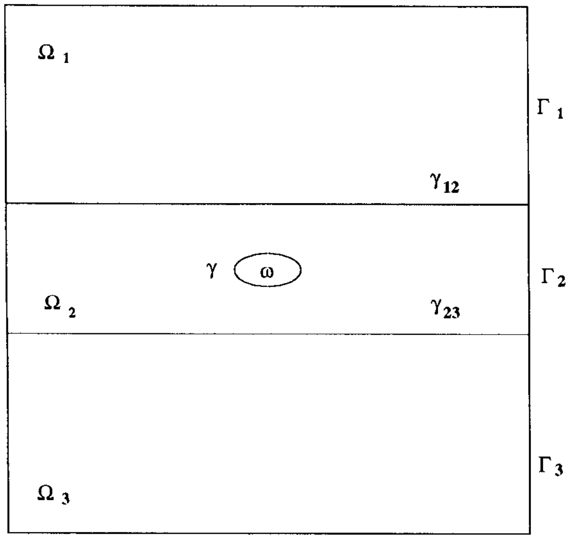

Fig. 2.

We consider the case where ω is a bounded domain, and a three subdomain decomposition like the one in Fig. 2 where $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$; we denote by γ_{12} (resp., γ_{23}) the interface between Ω_1 and Ω_2 (resp., Ω_2 and Ω_3), and by γ the boundary of ω; and let $\Gamma_i = \Gamma \cap \partial\Omega_i$ for $i = 1, 2, 3$. To obtain the equivalent fictitious formulation of (1)–(3), we imbed $\Omega \backslash \bar{\omega}$ in Ω and define the following spaces

$$V_{g_1}^i = \{v | v \in H^1(\Omega_i), v = g_1 \text{ on } \Gamma_i\}, \quad V_0^i = \{v | v \in H^1(\Omega_i), v = 0 \text{ on } \Gamma_i\},$$

for $i = 1, 2, 3$ and let

$$V = V_{g_1}^1 \times V_{g_1}^2 \times V_{g_1}^3, \quad V_0 = V_0^1 \times V_0^2 \times V_0^3, \quad \Lambda = L^2(\gamma) \times L^2(\gamma_{12}) \times L^2(\gamma_{23}).$$

Problem (1)–(3) is equivalent to the following system

Find $(u_1, u_2, u_3) \in V$, $(\lambda, \lambda_{12}, \lambda_{23}) \in \Lambda$ *such that*

$$\sum_{i=1}^{3} \int_{\Omega_i} (\alpha u_i v_i + \nu \nabla u_i \cdot \nabla v_i)\, dx = \sum_{i=1}^{3} \int_{\Omega_i} \tilde{f} v_i\, dx + \int_{\gamma_{12}} \lambda_{12}(v_2 - v_1)\, ds \tag{7}$$

$$+ \int_{\gamma_{23}} \lambda_{23}(v_3 - v_2)\, ds + \int_{\gamma} \lambda v_2\, d\gamma, \ \forall (v_1, v_2, v_3) \in V_0,$$

$$\int_{\gamma} \mu(u - g_0)\, d\gamma = 0, \ \forall \mu \in L^2(\gamma), \tag{8}$$

$$\int_{\gamma_{12}} \mu_{12}(u_2 - u_1)\, d\gamma = 0, \ \forall \mu_{12} \in L^2(\gamma_{12}), \tag{9}$$

$$\int_{\gamma_{23}} \mu_{23}(u_3 - u_2)\, d\gamma = 0, \ \forall \mu_{23} \in L^2(\gamma_{23}), \tag{10}$$

where $\tilde{f}$ is a $L^2(\Omega)$ extension of f. We have equivalence in the sense that if relations (7)–(10) hold then $u_i = u|_{\Omega_i}$, for $i = 1, 2, 3$, where u is solution of (5)–(6), and conversely.

Due to the combination of the two methods, there are two Lagrange multipliers in (7)–(10). The function λ which is a *Lagrange multiplier* associated with the *boundary condition* $u = g_0$ on γ is essentially the jump of $\nu(\partial u/\partial n)$ at γ, and the function λ_{12} (resp., γ_{23}) which can be viewed as a *Lagrange multiplier* associated with the *interface boundary condition* $u_1 = u_2$ (resp., $u_2 = u_3$) on γ_{12} (resp., γ_{23}) is nothing but the function $\nu(\partial u/\partial n_1)|_{\gamma_{12}} = -\nu(\partial u/\partial n_2)|_{\gamma_{12}}$ (resp., $\nu(\partial u/\partial n_2)|_{\gamma_{23}} = -\nu(\partial u/\partial n_3)|_{\gamma_{23}}$) where n_i is the outward normal unit vector of Ω_i at interfaces γ_{12} and γ_{23}.

There are three different conjugate gradient algorithms which have different parallelization properties and can be parallelized on MIMD machines [13]. Here we apply the one called the one shot method to solve the saddle–point system (7)–(10). In [14], the numerical experiments of the one shot method to two–dimensional and three–dimensional test problems have been done on a KSR1 of MIMD type.

The one shot method is the following:

$$\lambda^0 = (\lambda^0, \lambda^0_{12}, \lambda^0_{23}) \in \Lambda \textit{ given;} \tag{11}$$

solve

$$\textit{Find } (u_1^0, u_2^0, u_3^0) \in V, \textit{ such that}$$
$$\begin{aligned}\sum_{i=1}^{3} \int_{\Omega_i} (\alpha u_i^0 v_i + \nu \nabla u_i^0 \cdot \nabla v_i)\, dx &= \sum_{i=1}^{3} \int_{\Omega_i} \tilde{f} v_i \, dx + \int_{\gamma_{12}} \lambda^0_{12}(v_2 - v_1)\, ds \\ &+ \int_{\gamma_{23}} \lambda^0_{23}(v_3 - v_2)\, ds + \int_{\gamma} \lambda^0 v_2 \, d\gamma, \ \forall (v_1, v_2, v_3) \in V_0,\end{aligned} \tag{12}$$

set $\quad \mathbf{g}^0 = (g^0, g^0_{12}, g^0_{23}) = ((u_2^0 - g_0)|_{\gamma}, (u_2^0 - u_1^0)|_{\gamma_{12}}, (u_3^0 - u_2^0)|_{\gamma_{23}}),$

and set $\quad \mathbf{w}^0 = (w^0, w^0_{12}, w^0_{23}) = (g^0, g^0_{12}, g^0_{23}).$

For $n \geq 0$, *knowing* λ^n, $\mathbf{g}^n$, $\mathbf{w}^n$, *compute* λ^{n+1}, $\mathbf{g}^{n+1}$, $\mathbf{w}^{n+1}$ *as follows:*

solve

$$\textit{Find } (\bar{u}_1^n, \bar{u}_2^n, \bar{u}_3^n) \in V_0, \textit{ such that}$$
$$\begin{aligned}\sum_{i=1}^{3} \int_{\Omega_i} (\alpha \bar{u}_i^n v_i + \nu \nabla \bar{u}_i^n \cdot \nabla v_i)\, dx &= \int_{\gamma_{12}} w^n_{12}(v_2 - v_1)\, ds \\ &+ \int_{\gamma_{23}} w^n_{23}(v_3 - v_2)\, ds + \int_{\gamma} w^n v_2 \, d\gamma, \ \forall (v_1, v_2, v_3) \in V_0,\end{aligned} \tag{13}$$

and set $\quad \bar{\mathbf{g}}^n = (\bar{g}^n, \bar{g}^n_{12}, \bar{g}^n_{23}) = (\bar{u}_2^n|_{\gamma}, (\bar{u}_2^n - \bar{u}_1^n)|_{\gamma_{12}}, (\bar{u}_3^n - \bar{u}_2^n)|_{\gamma_{23}}),$

We then compute

$$\rho_n = \frac{\int_{\gamma} |g^n|^2 \, d\gamma + \int_{\gamma_{12}} |g^n_{12}|^2 \, ds + \int_{\gamma_{23}} |g^n_{23}|^2 \, ds}{\int_{\gamma} \bar{g}^n w^n \, d\gamma + \int_{\gamma_{12}} \bar{g}^n_{12} w^n_{12} \, ds + \int_{\gamma_{23}} \bar{g}^n_{23} w^n_{23} \, ds}, \tag{14}$$

and set

$$\lambda^{n+1} = \lambda^n - \rho_n \mathbf{w}^n, \tag{15}$$

$$u_i^{n+1} = u_i^n - \rho_n \bar{u}_i^n, \ for\ i = 1, 2, 3, \tag{16}$$

$$\mathbf{g}^{n+1} = \mathbf{g}^n - \rho_n \bar{\mathbf{g}}^n. \tag{17}$$

If $\|\mathbf{g}^{n+1}\|/\|\mathbf{g}^0\| \leq \epsilon$*, take* $\lambda = \lambda^{n+1}$, $u_1 = u_1^{n+1}$, $u_2 = u_2^{n+1}$*, and* $u_3 = u_3^{n+1}$*; if not compute*

$$\gamma_n = \|\mathbf{g}^{n+1}\|^2/\|\mathbf{g}^n\|^2, \tag{18}$$

and set $\quad \mathbf{w}^{n+1} = \mathbf{g}^{n+1} + \gamma_n \mathbf{w}^n$.

Do $n = n + 1$ *and go to* (13).

In algorithm (11)–(18), $\|\mathbf{g}\|^2 = \int_\gamma |g|^2\, d\gamma + \int_{\gamma_{12}} |g_{12}|^2\, ds + \int_{\gamma_{23}} |g_{23}|^2\, ds$ for any $\mathbf{g} = (g, g_{12}, g_{23}) \in \Lambda$. We have taken $\epsilon = 10^{-8}$ in the stopping test. When $\omega \subset \mathbb{R}^2$ with smooth boundary, a preconditioned one shot method has been discussed in [13].

2.3 Numerical experiments

We consider the following test problem:

$$-\Delta u = 0 \text{ in } \Omega \setminus \bar{\omega}, \tag{19}$$

$$u = 0 \text{ on } \gamma, \tag{20}$$

$$u = -x\cos\theta + y\sin\theta \text{ on } \Gamma, \tag{21}$$

where $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$, ω is a NACA0012 airfoil centered at $(0, 0)$ with zero degree of angle of attack whose chord length is 0.35, and θ is an incident angle. In numerical experiments, $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ where $\Omega_1 = (-0.5, 0.5) \times (0.125, 0.5)$, $\Omega_2 = (-0.5, 0.5) \times (-0.125, 0.125)$, $\Omega_3 = (-0.5, 0.5) \times (-0.5, -0.125)$.

The finite dimensional spaces $V_{g_1 h}^i$ and V_{0h}^i which approximate $V_{g_1}^i$ and V_0^i, respectively, for $i = 1, 2, 3$ are defined as follows:

$$V_{g_1 h}^i = \{v_h | v_h \in C^0(\bar{\Omega}_i), v_h = g_{1h} \ on\ \Gamma_i, v_h|_T \in P_1,\ \forall T \in \mathcal{T}_h^i\},$$
$$V_{0h}^i = \{v_h | v_h \in C^0(\bar{\Omega}_i), v_h = 0 \ on\ \Gamma_i, v_h|_T \in P_1,\ \forall T \in \mathcal{T}_h^i\},$$

where g_{1h} is an approximation of g_1 given by (21), $\mathcal{T}_h^i$ is a triangulations (see Fig. 3) of Ω_i for $i = 1, 2, 3$ and P_1 is the space of the polynomials in x_1, x_2 of degree ≤ 1. For the above finite dimensional spaces, we can use different mesh size in each subdomain (see Fig. 3).

The spaces $L^2(\gamma)$, $L^2(\gamma_{12})$ and $L^2(\gamma_{23})$ are approximated by the finite dimensional space Λ_h, Λ_{h12}, and Λ_{h23}, respectively, as follows:

$$\begin{aligned}\Lambda_h &= \{\mu_h|\mu_h \in L^\infty(\gamma), \mu_h \textit{ is constant on the segment joining}\\ &\qquad \textit{2 consecutive mesh points on } \gamma\},\\ \Lambda_{h12} &= \{\mu_h|\mu_h \in L^\infty(\gamma_{12}), \mu_h \textit{ is constant on the segment joining}\\ &\qquad \textit{2 consecutive mesh points on } \gamma_{12}\},\\ \Lambda_{h23} &= \{\mu_h|\mu_h \in L^\infty(\gamma_{23}), \mu_h \textit{ is constant on the segment joining}\\ &\qquad \textit{2 consecutive mesh points on } \gamma_{23}\}.\end{aligned}$$

A particular choice of mesh points on γ is visualized in Fig. 3. The mesh points on γ_{12} and γ_{23} are the midpoints of the edges located on the interfaces of the finest triaugulation [15] (e.g., see Fig. 4).

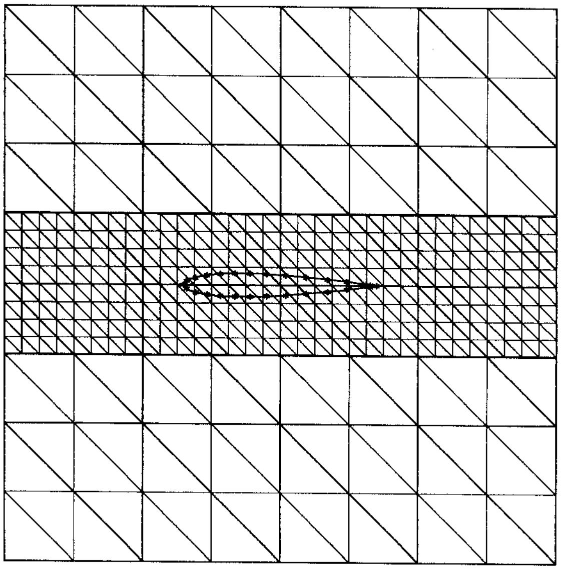

Fig. 3: Example of meshes on Ω_1, Ω_2, and Ω_3 with mesh sizes $h = 1/8$, $h = 1/32$, and $h = 1/8$ respectively. Mesh points on γ are marked by "*".

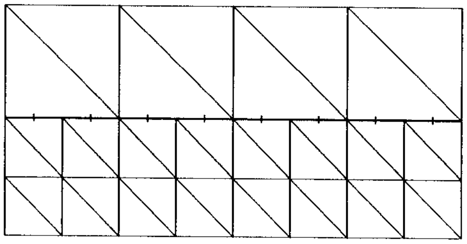

Fig. 4: Example of mesh on interface between subdomain with nonmatching grids. Mesh points on interface are marked by "|"

The solution of (19)–(21) is the stream function of the potential flow of incident angle θ. In the numerical experiments, we have taken $\theta = 45°$ and the mesh sizes are $h_1 = 1/128$, $h_2 = 1/512$, and $h_3 = 1/128$ for the subdomains Ω_1, Ω_2, and Ω_3 respectively. We applied the one shot method with nonmatching grids to (19)–(21) and the corresponding numerical results are shown in Fig. 5. Those obtained by the fictitious domain method with uniform grid of mesh size $h = 1/512$ are also shown in Fig. 5. With the one shot method and nonmatching grids, we can successfully use fine grid in the region close to the body and coarse grid in the region far from the body. We observe the excellent agreement between these results.

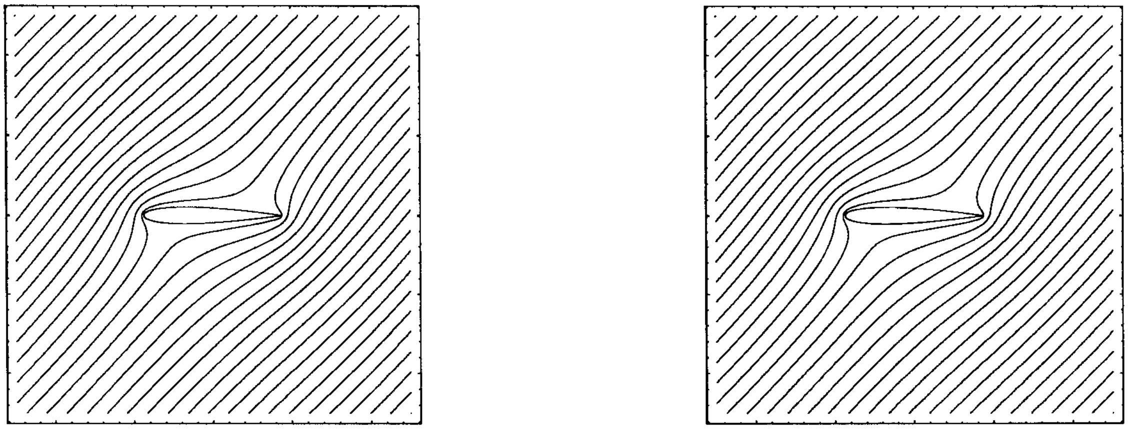

Fig. 5: The left figure shows the streamlines obtained by the one shot method with nonmatching grids and $h_1 = 1/128$, $h_2 = 1/512$, and $h_3 = 1/128$ for the subdomains Ω_1, Ω_2, and Ω_3 shown in Fig. 3. The right figure shows the streamlines obtained by the fictitious domain method with uniform grid of mesh size $h = 1/512$.

3. FICTITIOUS DOMAIN METHODS FOR THE NAVIER–STOKES EQUATIONS

3.1 The Navier–Stokes Equations

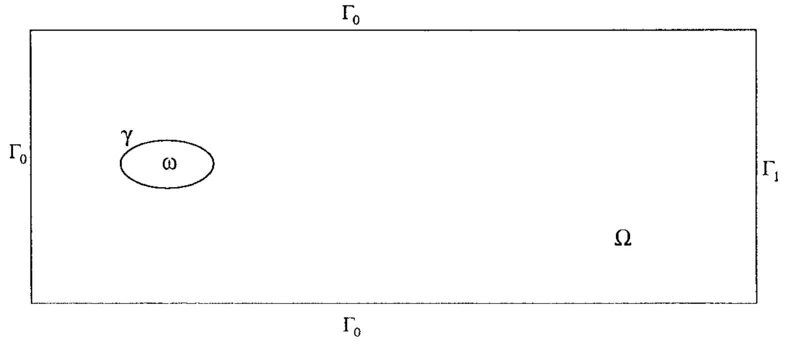

Fig. 6.

In [5, 6], we have considered external incompressible viscous flow modeled by the Navier–Stokes equations with either Dirichlet boundary conditions or Neumann boundary conditions downstream. Using the notation of Fig. 6, we consider the following problem

$$\frac{\partial \mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} \; in \; \Omega \setminus \overline{\omega(t)}, \tag{22}$$

$$\nabla\cdot\mathbf{u} = 0 \; in \; \Omega \setminus \overline{\omega(t)}, \tag{23}$$

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x}\in\Omega\setminus\overline{\omega(t)}, (with \; \nabla\cdot\mathbf{u}_0 = 0), \tag{24}$$

$$\mathbf{u} = \mathbf{g}_0(t) \; on \; \Gamma_0, \; \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p = \mathbf{g}_1(t) \; on \; \Gamma_1, \tag{25}$$

$$\mathbf{u} = \mathbf{g}_2(t) \; on \; \gamma(t). \tag{26}$$

In (22)–(26), Ω is bounded domains in $\mathbb{R}^d(d\geq 2)$, $\omega(t)$ could be a moving rigid body in $\mathbb{R}^d(d\geq 2)$ (see Fig. 6), Γ (resp., $\gamma(t)$) is the boundary of Ω (resp., $\omega(t)$) with $\Gamma = \Gamma_0\cup\Gamma_1$, $\Gamma_0\cap\Gamma_1 = \emptyset$, and $\int_{\Gamma_1} d\Gamma > 0$, $\mathbf{n}$ is the outer normal unit vector at Γ_1, $\mathbf{u} = \{u_i\}_{i=1}^{i=d}$ is the flow velocity, p is the pressure, $\mathbf{f}$ is a density of external forces, $\nu(>0)$ is a viscosity parameter, and $(\mathbf{v}\cdot\nabla)\mathbf{w} = \{\sum_{j=1}^{j=d} v_j \frac{\partial w_i}{\partial x_j}\}_{i=1}^{i=d}$.

To obtain the equivalent fictitious domain formulation for the Navier–Stokes equations, we embed $\Omega\setminus\overline{\omega(t)}$ in Ω and define

$$\mathbf{V}_{\mathbf{g}_0} = \{\mathbf{v}|\mathbf{v}\in(\mathbf{H}^1(\Omega))^d, \mathbf{v} = \mathbf{g}_0 \; on \; \Gamma_0\}, \quad \mathbf{V}_0 = \{\mathbf{v}|\mathbf{v}\in(\mathbf{H}^1(\Omega))^d, \mathbf{v} = \mathbf{0} \; on \; \Gamma_0\},$$

and $\Lambda(t) = (\mathbf{L}^2(\gamma(t)))^d$. We observe that if $\mathbf{U}_0$ is an extension of $\mathbf{u}_0$ with $\nabla\cdot\mathbf{U}_0 = 0$ in Ω, and if $\tilde{\mathbf{f}}$ is an extension of $\mathbf{f}$, we have equivalence between (22)–(26) and the following problem:

$$\left\{\begin{aligned} & For \; t>0, \; find \; \mathbf{U}(t)\in\mathbf{V}_{\mathbf{g}_0}, \; P(t)\in\mathbf{L}^2(\Omega), \; \lambda(t)\in\Lambda(t) \; such \; that \\ & \int_\Omega \frac{\partial\mathbf{U}}{\partial t}\cdot\mathbf{v}\,d\mathbf{x} + \nu\int_\Omega \nabla\mathbf{U}\cdot\nabla\mathbf{v}\,d\mathbf{x} \\ & \quad + \int_\Omega(\mathbf{U}\cdot\nabla)\mathbf{U}\cdot\mathbf{v}\,d\mathbf{x} - \int_\Omega P\nabla\cdot\mathbf{v}\,d\mathbf{x} \\ & = \int_\Omega \tilde{\mathbf{f}}\cdot\mathbf{v}\,d\mathbf{x} + \int_{\Gamma_1}\mathbf{g}_1\cdot\mathbf{v}\,d\Gamma + \int_{\gamma(t)}\lambda\cdot\mathbf{v}\,d\gamma, \; \forall\mathbf{v}\in\mathbf{V}_0, \; a.e. \; t>0, \end{aligned}\right. \tag{27}$$

$$\nabla\cdot\mathbf{U}(t) = 0 \text{ in } \Omega, \tag{28}$$

$$\mathbf{U}(\mathbf{x},0) = \mathbf{U}_0(\mathbf{x}), \; \mathbf{x}\in\Omega, \; (\text{with } \; \nabla\cdot\mathbf{U}_0 = 0), \tag{29}$$

$$\mathbf{U}(t) = \mathbf{g}_2(t) \text{ on } \gamma(t), \tag{30}$$

in the sense that $\mathbf{U}|_{\Omega\setminus\overline{\omega(t)}} = \mathbf{u}$, $P|_{\Omega\setminus\overline{\omega(t)}} = p$. Above, we have used the notation $\phi(t)$ for the function $\mathbf{x}\to\phi(\mathbf{x},t)$.

Concerning the multiplier λ, its interpretation is simple since it is equal to the jump of $\nu(\partial\mathbf{U}/\partial\mathbf{n}) - \mathbf{n}P$ at $\gamma(t)$. A closely related approach (limited to the steady

Stokes problem) is discussed in [7]. We observe that the effect of the actual geometry is concentrated on $\int_{\gamma(t)} \lambda \cdot \mathbf{v}\, d\gamma$ in the right–hand–side of (27), and on (30).

3.2 Time discretization by operator splitting

To solve (27)–(30), we shall consider a time discretization by an *operator splitting method*, like the ones discussed in, e.g., [8]–[12]. With these methods we are able to *decouple* the nonlinearity and the incompressibility in the Navier–Stokes/fictitious domain problems (27)–(30). In the following, we apply the *θ–scheme* (cf. [12]) to (27)–(30) with time step $\triangle t > 0$. Let $\Lambda^s = (\mathbf{L}^2(\gamma(s\triangle t)))^d$ and denote by ϕ^s either an approximation of or the function $\phi(s\triangle t)$. We obtain the following time discretization scheme:

$$\mathbf{U}^0 = \mathbf{U}_0 \textit{ is given;} \tag{31}$$

for $n \geq 0$, *knowing* $\mathbf{U}^n$, *find* $\mathbf{U}^{n+\theta} \in \mathbf{V}_{\mathbf{g}_0^{n+\theta}}, P^{n+\theta} \in \mathbf{L}^2(\Omega), \lambda^{n+\theta} \in \Lambda^{n+\theta}$ *such that*

$$\left\{\begin{aligned} &\int_\Omega \frac{\mathbf{U}^{n+\theta} - \mathbf{U}^n}{\theta\triangle t} \cdot \mathbf{v}\, d\mathbf{x} + \alpha\nu \int_\Omega \nabla \mathbf{U}^{n+\theta} \cdot \nabla \mathbf{v}\, d\mathbf{x} - \int_\Omega P^{n+\theta} \nabla \cdot \mathbf{v}\, d\mathbf{x} \\ &= \int_\Omega \tilde{\mathbf{f}}^{n+\theta} \cdot \mathbf{v}\, d\mathbf{x} - \int_\Omega (\mathbf{U}^n \cdot \nabla)\mathbf{U}^n \cdot \mathbf{v}\, d\mathbf{x} + \int_{\gamma^{n+\theta}} \lambda^{n+\theta} \cdot \mathbf{v}\, d\gamma \\ &-\beta\nu \int_\Omega \nabla \mathbf{U}^n \cdot \nabla \mathbf{v}\, d\mathbf{x} + \int_{\Gamma_1} (\alpha \mathbf{g}_1^{n+\theta} + \beta \mathbf{g}_1^n) \cdot \mathbf{v}\, d\Gamma, \ \forall \mathbf{v} \in \mathbf{V}_0, \end{aligned}\right. \tag{32}$$

$$\nabla \cdot \mathbf{U}^{n+\theta} = 0 \text{ in } \Omega, \tag{33}$$

$$\mathbf{U}^{n+\theta} = \mathbf{g}_2^{n+\theta} \text{ on } \gamma^{n+\theta}; \tag{34}$$

next, find $\mathbf{U}^{n+1-\theta} \in \mathbf{V}_{\mathbf{g}_0^{n+1-\theta}}$ *such that*

$$\left\{\begin{aligned} &\int_\Omega \frac{\mathbf{U}^{n+1-\theta} - \mathbf{U}^{n+\theta}}{(1-2\theta)\triangle t} \cdot \mathbf{v}\, d\mathbf{x} + \beta\nu \int_\Omega \nabla \mathbf{U}^{n+1-\theta} \cdot \nabla \mathbf{v}\, d\mathbf{x} \\ &\qquad + \int_\Omega (\mathbf{U}^{n+1-\theta} \cdot \nabla)\mathbf{U}^{n+1-\theta} \cdot \mathbf{v}\, d\mathbf{x} \\ &= \int_\Omega \tilde{\mathbf{f}}^{n+1-\theta} \cdot \mathbf{v}\, d\mathbf{x} + \int_{\gamma^{n+\theta}} \lambda^{n+\theta} \cdot \mathbf{v}\, d\gamma + \int_\Omega P^{n+\theta} \nabla \cdot \mathbf{v}\, d\mathbf{x} \\ &\quad -\alpha\nu \int_\Omega \nabla \mathbf{U}^{n+\theta} \cdot \nabla \mathbf{v}\, d\mathbf{x} + \int_{\Gamma_1} (\alpha \mathbf{g}_1^{n+\theta} + \beta \mathbf{g}_1^{n+1-\theta}) \cdot \mathbf{v}\, d\Gamma, \ \forall \mathbf{v} \in \mathbf{V}_0; \end{aligned}\right. \tag{35}$$

finally, find $\mathbf{U}^{n+1} \in \mathbf{V}_{\mathbf{g}_0^{n+1}}, P^{n+1} \in \mathbf{L}^2(\Omega), \lambda^{n+1} \in \Lambda^{n+1}$ *such that*

$$\left\{\begin{aligned} &\int_\Omega \frac{\mathbf{U}^{n+1} - \mathbf{U}^{n+1-\theta}}{\theta\triangle t} \cdot \mathbf{v}\, d\mathbf{x} + \alpha\nu \int_\Omega \nabla \mathbf{U}^{n+1} \cdot \nabla \mathbf{v}\, d\mathbf{x} - \int_\Omega P^{n+1} \nabla \cdot \mathbf{v}\, d\mathbf{x} \\ &= \int_\Omega \tilde{\mathbf{f}}^{n+1} \cdot \mathbf{v}\, d\mathbf{x} - \int_\Omega (\mathbf{U}^{n+1-\theta} \cdot \nabla)\mathbf{U}^{n+1-\theta} \cdot \mathbf{v}\, d\mathbf{x} + \int_{\gamma^{n+1}} \lambda^{n+1} \cdot \mathbf{v}\, d\gamma \\ &-\beta\nu \int_\Omega \nabla \mathbf{U}^{n+1-\theta} \cdot \nabla \mathbf{v}\, d\mathbf{x} + \int_{\Gamma_1} (\alpha \mathbf{g}_1^{n+1} + \beta \mathbf{g}_1^{n+1-\theta}) \cdot \mathbf{v}\, d\Gamma, \ \forall \mathbf{v} \in \mathbf{V}_0, \end{aligned}\right. \tag{36}$$

$$\nabla \cdot \mathbf{U}^{n+1} = 0 \text{ in } \Omega, \tag{37}$$

$$\mathbf{U}^{n+1} = \mathbf{g}_2^{n+1} \text{ on } \gamma^{n+1}, \tag{38}$$

where $\alpha + \beta = 1, 0 < \alpha, \beta < 1$ and $0 < \theta < 1/2$. With the choice of $\theta = 1 - 1/\sqrt{2}$, $\alpha = 2 - \sqrt{2}$ and $\beta = \sqrt{2} - 1$, the time discretization seems to be unconditionally stable (see [12]).

In Section 3.3 the conjugate gradient solution of the *Stokes/fictitious domain problems* (32)–(34) and (36)–(38) will be discussed. Concerning problem (35) it is worth noticing that we have been taking advantage of the time discretization by operator splitting to treat the advection in the larger domain Ω without concern –in some sense– for the constraint $\mathbf{u} = \mathbf{g}$ at γ. Problem (35) can be solved by least–squares methods [12] and is also well suited to solution methods based on higher order upwinding on regular meshes, or on the backward method of characteristics (see, e.g., [16]).

3.3 Iterative solution of the Stokes/fictitious domain problem

Problems (32)–(34) and (36)–(38) are particular cases of the following Stokes/fictitious domain problem:

$$\left\{ \begin{array}{l} \textit{Find } \mathbf{U} \in \mathbf{V}_{\mathbf{g}_0},\ P \in \mathbf{L}^2(\Omega),\ \lambda \in \Lambda \textit{ such that} \\ \alpha \displaystyle\int_\Omega \mathbf{U} \cdot \mathbf{v}\, d\mathbf{x} + \nu \int_\Omega \nabla \mathbf{U} \cdot \nabla \mathbf{v}\, d\mathbf{x} - \int_\Omega P \nabla \cdot \mathbf{v}\, d\mathbf{x} \\ = \displaystyle\int_\Omega \mathbf{f} \cdot \mathbf{v}\, d\mathbf{x} + \int_\gamma \lambda \cdot \mathbf{v}\, d\gamma + \int_{\Gamma_1} \mathbf{g}_1 \cdot \mathbf{v}\, d\Gamma,\ \forall \mathbf{v} \in \mathbf{V}_0, \end{array} \right. \tag{39}$$

$$\nabla \cdot \mathbf{U} = 0 \textit{ in } \Omega, \tag{40}$$

$$\mathbf{U} = \mathbf{g}_2 \textit{ on } \gamma, \tag{41}$$

where, in (39), $\alpha(> 0)$ is the reciprocal of a partial time step. In (39)–(41), P (resp., λ) appears to be a Lagrange multiplier associated with (40) (resp., (41)).

We can solve the above saddle–point system (39)–(41) by a conjugate gradient algorithm (called the one shot method) driven by the pressure P and the multiplier λ, simultaneously. Let us consider a bilinear form $b(\cdot,\cdot)$, symmetric and strongly elliptic over Λ (we may choose $b(\lambda, \mu) = \int_\gamma \lambda \cdot \mu\, d\gamma$, $\forall \lambda$, $\mu \in \Lambda$). The following algorithm is the one shot method driven by the multipliers P and λ:

$$\{P^0, \lambda^0\} \in \mathbf{L}^2(\Omega) \times \Lambda \textit{ given;} \tag{42}$$

solve the following Dirichlet problem

$$\left\{ \begin{array}{l} \textit{Find } \mathbf{U}^0 \in \mathbf{V}_{\mathbf{g}_0}, \textit{ such that} \\ \alpha \displaystyle\int_\Omega \mathbf{U}^0 \cdot \mathbf{v}\, d\mathbf{x} + \nu \int_\Omega \nabla \mathbf{U}^0 \cdot \nabla \mathbf{v}\, d\mathbf{x} = \int_\Omega \mathbf{f} \cdot \mathbf{v}\, d\mathbf{x} + \\ \displaystyle\int_\gamma \lambda^0 \cdot \mathbf{v}\, d\gamma + \int_\Omega P^0 \nabla \cdot \mathbf{v}\, d\mathbf{x} + \int_{\Gamma_1} \mathbf{g}_1 \cdot \mathbf{v}\, d\Gamma,\ \forall \mathbf{v} \in \mathbf{V}_0, \end{array} \right. \tag{43}$$

set $r_1^0 = \nabla \cdot \mathbf{U}^0$, $\mathbf{r}_2^0 = (\mathbf{U}^0 - \mathbf{g}_2)|_\gamma$ *and define* $\mathbf{g}^0 = \{g_1^0, \mathbf{g}_2^0\}$ *as follows*

$$g_1^0 = \alpha\phi^0 + \nu r_1^0, \tag{44}$$

with ϕ^0 *the solution of*

$$\left\{\begin{array}{l} -\Delta\phi^0 = r_1^0 \text{ in } \Omega, \\ \dfrac{\partial\phi^0}{\partial\mathbf{n}} = 0 \text{ on } \Gamma_0;\ \phi^0 = 0 \text{ on } \Gamma_1; \end{array}\right. \tag{45}$$

$$\left\{\begin{array}{l} \text{Find } \mathbf{g}_2^0 \in \Lambda \text{ such that} \\ b(\mathbf{g}_2^0, \mu) = \displaystyle\int_\gamma \mathbf{r}_2^0 \cdot \mu \, d\gamma,\ \forall\mu \in \Lambda. \end{array}\right. \tag{46}$$

We take $\mathbf{w}^0 = \{w_1^0, \mathbf{w}_2^0\} = \{g_1^0, \mathbf{g}_2^0\}$.

Then for $n \geq 0$, *assuming that* $\{P^n, \lambda^n\}, \mathbf{U}^n, r_1^n, \mathbf{r}_2^n, \mathbf{w}^n, \mathbf{g}^n$ *are known, compute* $\{P^{n+1}, \lambda^{n+1}\}$, $\mathbf{U}^{n+1}$, r_1^{n+1}, $\mathbf{r}_2^{n+1}$, $\mathbf{w}^{n+1}$, $\mathbf{g}^{n+1}$ *as follows:*

solve

$$\left\{\begin{array}{l} \text{Find } \bar{\mathbf{U}}^n \in \mathbf{V}_0, \text{ such that} \\ \alpha \displaystyle\int_\Omega \bar{\mathbf{U}}^n \cdot \mathbf{v} \, d\mathbf{x} + \nu \int_\Omega \nabla\bar{\mathbf{U}}^n \cdot \nabla\mathbf{v} \, d\mathbf{x} \\ \qquad = \displaystyle\int_\gamma \mathbf{w}_2^n \cdot \mathbf{v} \, d\gamma + \int_\Omega w_1^n \nabla \cdot \mathbf{v} \, d\mathbf{x},\ \forall\mathbf{v} \in \mathbf{V}_0; \end{array}\right. \tag{47}$$

set $\bar{r}_1^n = \nabla \cdot \bar{\mathbf{U}}^n$, $\bar{\mathbf{r}}_2^n = \bar{\mathbf{U}}^n|_\gamma$ *and define* $\bar{\mathbf{g}}^n = \{\bar{g}_1^n, \bar{\mathbf{g}}_2^n\}$ *as follows:*

$$\bar{g}_1^n = \alpha\bar{\phi}^n + \nu\bar{r}_1^n, \tag{48}$$

with $\bar{\phi}^n$ *the solution of*

$$\left\{\begin{array}{l} -\Delta\bar{\phi}^n = \bar{r}_1^n \text{ in } \Omega, \\ \dfrac{\partial\bar{\phi}^n}{\partial\mathbf{n}} = 0 \text{ on } \Gamma_0;\ \bar{\phi}^n = 0 \text{ on } \Gamma_1; \end{array}\right. \tag{49}$$

$$\left\{\begin{array}{l} \text{Find } \bar{\mathbf{g}}_2^n \in \Lambda \text{ such that} \\ b(\bar{\mathbf{g}}_2^n, \mu) = \displaystyle\int_\gamma \bar{\mathbf{r}}_2^n \cdot \mu \, d\gamma,\ \forall\mu \in \Lambda. \end{array}\right. \tag{50}$$

We then compute

$$\rho_n = (\textstyle\int_\Omega r_1^n g_1^n \, d\mathbf{x} + \int_\gamma \mathbf{r}_2^n \cdot \mathbf{g}_2^n \, d\gamma)/(\int_\Omega \bar{r}_1^n w_1^n \, d\mathbf{x} + \int_\gamma \bar{\mathbf{r}}_2^n \cdot \mathbf{w}_2^n \, d\gamma), \tag{51}$$

and set

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \rho_n \bar{\mathbf{U}}^n, \tag{52}$$

$$P^{n+1} = P^n - \rho_n w_1^n, \quad \lambda^{n+1} = \lambda^n - \rho_n \mathbf{w}_2^n, \tag{53}$$

$$r_1^{n+1} = r_1^n - \rho_n \bar{r}_1^n, \quad \mathbf{r}_2^{n+1} = \mathbf{r}_2^n - \rho_n \bar{\mathbf{r}}_2^n, \tag{54}$$

$$g_1^{n+1} = g_1^n - \rho_n \bar{g}_1^n, \quad \mathbf{g}_2^{n+1} = \mathbf{g}_2^n - \rho_n \bar{\mathbf{g}}_2^n. \tag{55}$$

If $(\int_\Omega r_1^{n+1} g_1^{n+1}\, d\mathbf{x} + \int_\gamma \mathbf{r}_2^{n+1} \cdot \mathbf{g}_2^{n+1}\, d\gamma)/(\int_\Omega r_1^0 g_1^0\, d\mathbf{x} + \int_\gamma \mathbf{r}_2^0 \cdot \mathbf{g}_2^0\, d\gamma) \leq \epsilon$, *take* $P = P^{n+1}$, $\mathbf{U} = \mathbf{U}^{n+1}$, *and* $\lambda = \lambda^{n+1}$. *If not, compute*

$$\gamma_n = (\int_\Omega r_1^{n+1} g_1^{n+1}\, d\mathbf{x} + \int_\gamma \mathbf{r}_2^{n+1} \cdot \mathbf{g}_2^{n+1}\, d\gamma)/(\int_\Omega r_1^n g_1^n\, d\mathbf{x} + \int_\gamma \mathbf{r}_2^n \cdot \mathbf{g}_2^n\, d\gamma), \tag{56}$$

and set $\quad \mathbf{w}^{n+1} = \mathbf{g}^{n+1} + \gamma_n \mathbf{w}^n.$

Do $n = n + 1$ *and go back to* (47).

4. FICTITIOUS DOMAIN METHOD FOR FLOW PAST A MOVING RIGID BODY

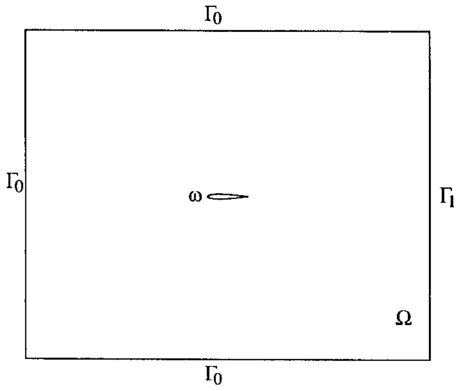

Fig. 7.

In this section we consider an incompressible viscous flow modelled by the Navier–Stokes equations past a moving rigid body, a NACA0012, by a fictitious domain method. In the test problem, let $\Omega = (-0.625, 0.625) \times (-0.5, 0.5)$ (see Fig. 7) and ω be a NACA0012 airfoil centered at $(0, 0)$ (half–chord point) which is forced to pitch about its center with the prescribed angular displacement defined by

$$\theta(t) = \theta_o - \theta_a \cos \pi t, \tag{57}$$

where $\theta_o = \theta_a = 10°$ and the chord length is 0.125. Thus the angle of attack is between 0° to 20°. The boundary conditions are defined as follows:

$$\mathbf{u} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \; on\, \Gamma_0, \quad \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{n}p = \mathbf{0} \text{ on } \Gamma_1. \tag{58}$$

When constructing finite dimensional subspaces, we can skip the time-consuming construction of a boundary-fitted mesh for each different position of the moving body and use a fixed uniform structured mesh $\mathcal{T}_h$ which is a triangulation of Ω (see, e.g., Fig. 8). For finite dimensional subspaces approximating $\mathbf{V}_{\mathbf{g}_0}$ (here let $\mathbf{g}_0 = (g_0^1, g_0^2) = (1, 0)$) and $\mathbf{V}_0$, we choose

$$\mathbf{V}_{\mathbf{g}_0 h} = \{\mathbf{v}_h | \mathbf{v}_h \in V_h^1 \times V_h^2\}, \quad \mathbf{V}_{0h} = \{\mathbf{v}_h | \mathbf{v}_h \in H_{0h}^1 \times H_{0h}^1\}, \tag{59}$$

where

$$V_h^i = \{\phi_h | \phi_h \in C^0(\bar{\Omega}), \phi_h|_T \in P_1,\ \forall T \in \mathcal{T}_h, \phi_h = g_{0h}^i \text{ on } \Gamma_0\}, \text{for } i = 1, 2, \tag{60}$$

$$H_{0h}^1 = \{\phi_h | \phi_h \in C^0(\bar{\Omega}), \phi_h|_T \in P_1,\ \forall T \in \mathcal{T}_h, \phi_h = 0 \text{ on } \Gamma_0\}. \tag{61}$$

In (60) and (61) P_1 is the space of the polynomials in x_1, x_2 of degree ≤ 1, and g_{0h}^1 (resp., g_{0h}^2) is an approximation of g_0^1 (resp., g_0^2). A traditional way of approximating the pressure is to take it in the space

$$H_{2h}^1 = \{\phi_h | \phi_h \in C^0(\bar{\Omega}), \phi_h|_T \in P_1,\ \forall T \in \mathcal{T}_{2h}\},$$

where $\mathcal{T}_{2h}$ is a triangulation twice coarser than $\mathcal{T}_h$. Concerning the space Λ_h^s approximating Λ^s, we define it by

$$\Lambda_h^s = \{\mu_h | \mu_h \in (L^\infty(\gamma^s))^2, \mu_h \textit{ is constant on the segment}$$
$$\textit{joining 2 consecutivemesh points on } \gamma^s\}.$$

A particular choice for mesh points on γ^s for different angles of attack is visualized on Fig. 8.

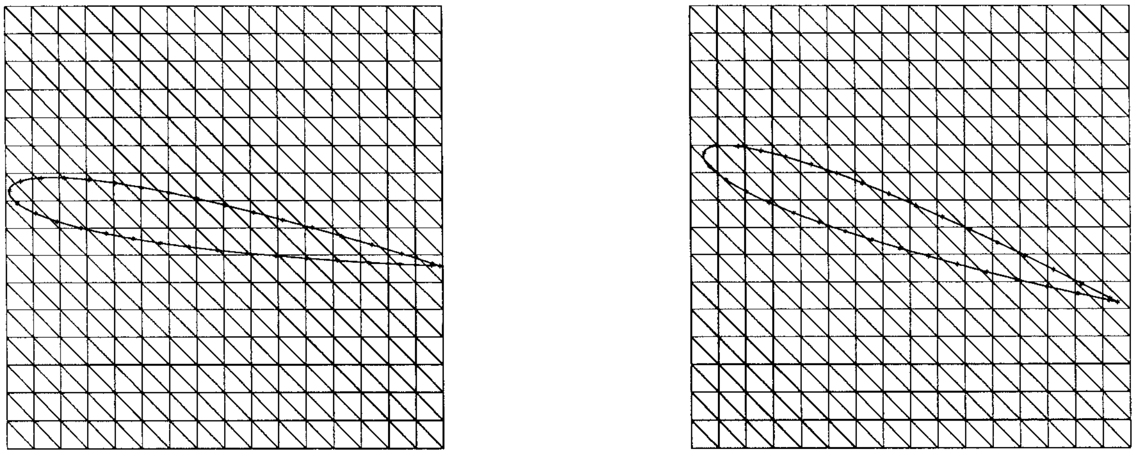

Fig. 8: Part of the triangulation of Ω with $h = 1/128$ and mesh points marked by "*" on NACA0012 with 10° (left) and 20° (right) degrees of angle of attack.

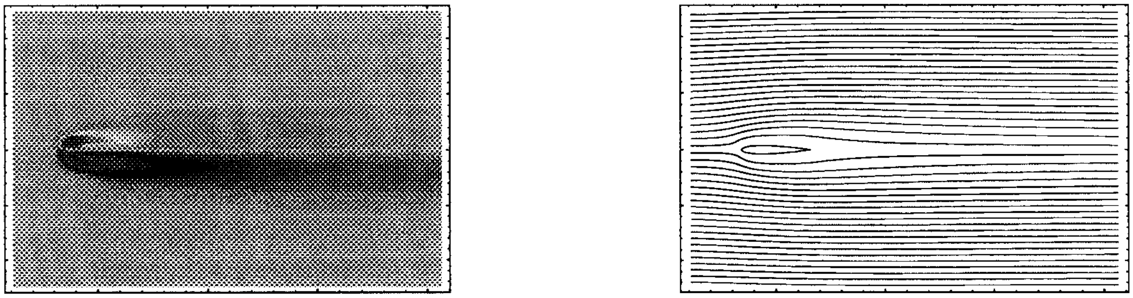

Fig. 9: Vorticity distribution and stream lines of the initial velocity field.

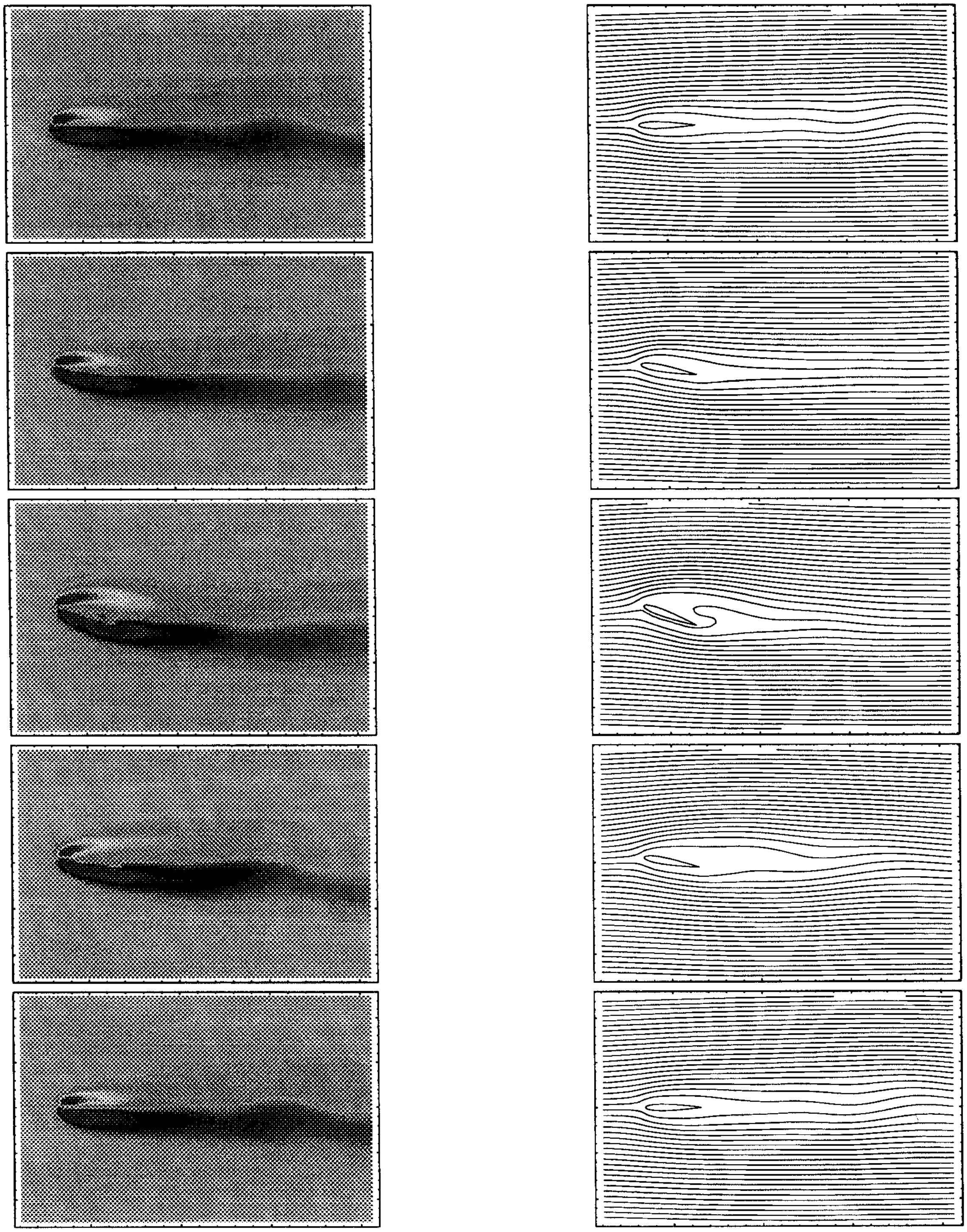

Fig. 10: Vorticity distribution and stream lines at time t =5, 5.5, 6, 6.5, 7 (from top to bottom, angles of attack are 0°, 10°, 20°, 10°, and 0°) during one period of the airfoil motion at Reynolds number=100.

Using the θ–scheme, we solve at each time step two Stokes/fictitious domain subproblems by the one shot method discussed in Section 3 and one advection–

diffusion subproblem by a least–squares/conjugate gradient algorithm [12]. In the numerical simulation, the Reynolds number is 100 (taking the chord of airfoil as characteristic length), the mesh size for the velocity field is $h_v = 1/128$, the mesh size for the pressure is $h_p = 1/64$, the time step is $\triangle t = 0.005$, and ϵ for the stopping criterion in (42)–(56) is 10^{-14}. Fig. 9 shows the vorticity distribution and stream lines of the initial condition for the simulation. It is a steady–state flow obtained at zero degree of angle of attack. Fig. 10 shows a sequence of frames for the vorticity distribution and stream lines during one period of the airfoil motion. From top to bottom, angles of attack are 0°, 10°, 20°, 10°, and 0°.

5. AN APPLICATION TO THE VORTEX DYNAMICS OF A TWO–DIMENSIONAL FLOW

Motivated by [17, 18], we would like to study the long–time evolution of the vortex dynamics of a two-dimensional incompressible viscous flow modelled by the Navier–Stokes equations. In [19], selective decay and coherent vortices in two-dimensional incompressible turbulence were studied in a square with periodic boundary conditions. In this section we consider a two-dimensional incompressible viscous flow in a disk.

In the test problem, let $\Omega = (-1.5, 1.5) \times (-1.5, 1.5)$ and ω be a disk of radius 1 centered at $(0, 0)$ (see Fig. 11). Here we imbed ω in Ω. This approach is slightly different from the one discussed in Sections 3 and 4; but we can use the same fictitious domain techniques. The Dirichlet boundary conditions on γ and Γ are defined as follows:

$$\mathbf{u} = \mathbf{0} \text{ on } \Gamma \cup \gamma. \tag{62}$$

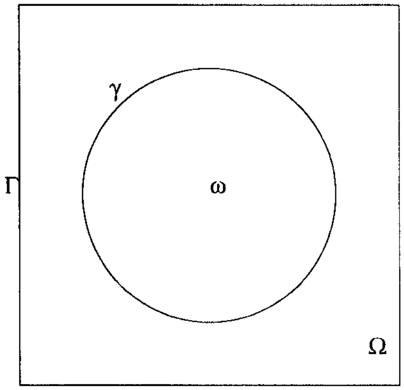

Fig. 11.

As finite dimensional subspaces of $\mathbf{V}_{\mathbf{g}_0}$ (here $\mathbf{g}_0 = (g_0^1, g_0^2) = (0, 0)$) and $\mathbf{V}_0$, we choose

$$\mathbf{V}_{\mathbf{g}_0 h} = \{\mathbf{v}_h | \mathbf{v}_h \in V_h^1 \times V_h^2\}, \quad \mathbf{V}_{0h} = \{\mathbf{v}_h | \mathbf{v}_h \in H_{0h}^1 \times H_{0h}^1\} \tag{63}$$

where

$$V_h^i = \{\phi_h | \phi_h \in C^0(\bar{\Omega}), \phi_h|_T \in P_1, \ \forall T \in \mathcal{T}_h, \phi_h = g_{0h}^i \text{ on } \Gamma\}, \text{for } i = 1, 2, \tag{64}$$

$$H_{0h}^1 = \{\phi_h | \phi_h \in C^0(\bar{\Omega}), \phi_h|_T \in P_1, \ \forall T \in \mathcal{T}_h, \phi_h = 0 \text{ on } \Gamma\}. \tag{65}$$

In (64) and (65), $\mathcal{T}_h$ is a triangulation of Ω (see, e.g., Fig. 12), P_1 is the space of the polynomials in x_1, x_2 of degree ≤ 1 and g_{0h}^1 (resp., g_{0h}^2) is approximation of g_0^1 (resp., g_0^2). A traditional way of approximating the pressure is to take it in the space

$$H_{2h}^1 = \{\phi_h | \phi_h \in C^0(\bar{\Omega}), \phi_h|_T \in P_1, \ \forall T \in \mathcal{T}_{2h}\},$$

where $\mathcal{T}_{2h}$ is a triangulation twice coarser than $\mathcal{T}_h$.

Concerning the space Λ_h approximating Λ, we define it by

$$\Lambda_h = \{\mu_h | \mu_h \in (L^\infty(\gamma))^2, \mu_h \textit{ is constant on the segment}$$
$$\textit{joining } 2 \textit{ consecutivemesh points on } \gamma\}.$$

A particular choice for the mesh points of γ is visualized on Fig. 12.

In Fig. 13 we show the vorticity and stream lines of the initial velocity field in the test problem (also obtained by the algorithm used for solving the test problem). In computing the initial condition shown in Fig. 13, we have chosen $\mathbf{u}_0^i = (\sin \pi x \cos \pi y, -\cos \pi x \sin \pi y)$ as initial velocity, the boundary conditions being

$$\mathbf{u}^i = e^{-ct}\mathbf{u}_0^i \text{ on } \Gamma \cup \gamma, \tag{66}$$

where $c = 5000$ in (66). The mesh size for the velocity field is $h_v = 1/120$, the mesh size for the pressure is $h_p = 1/60$, and the time step is $\triangle t = 0.001$. The solution obtained after 1000 time steps was retained the initial condition for the final test problem.

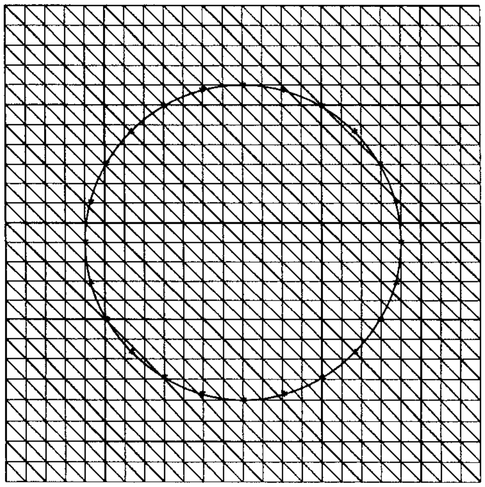

Fig. 12: Example of the triangulation of Ω with $h = 1/8$ and mesh points marked by "*" on the boundary of disk.

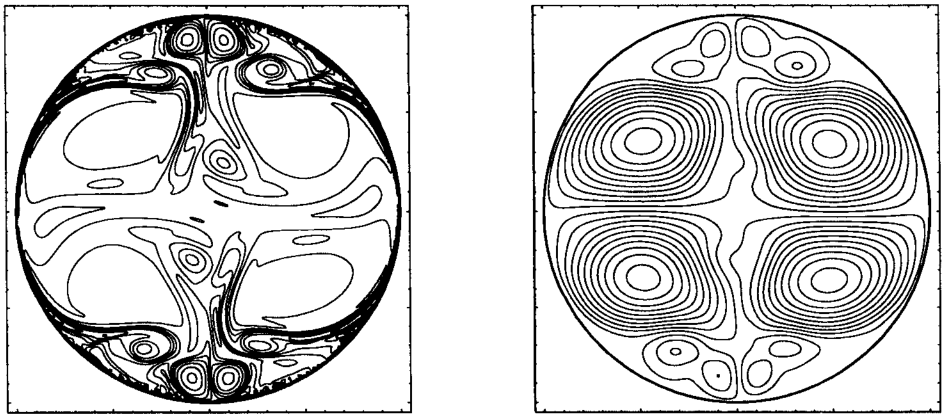

Fig. 13: Vorticity contours (left) and stream lines (right) of the initial condition.

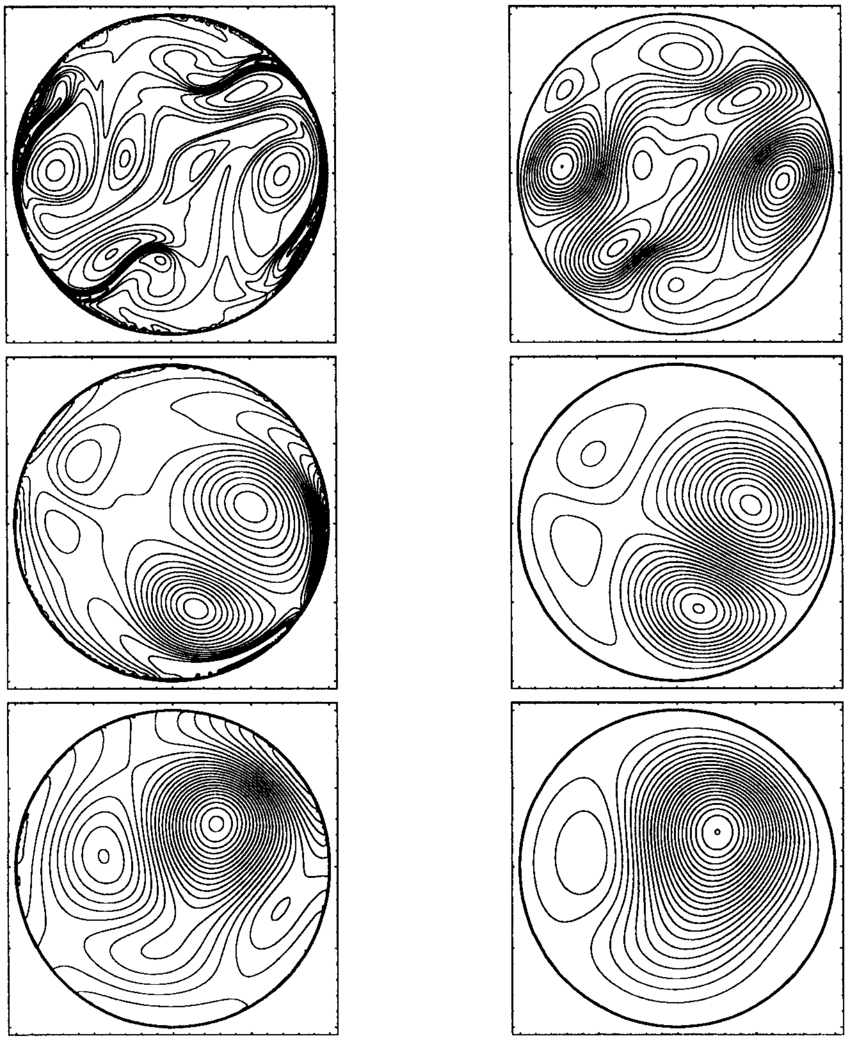

Fig. 14: Vorticity contours (left) and stream lines (right) at time $t = 30$, 170, and 557.5.

In the numerical simulation, the Reynolds number is 10011 (taking the diameter of disk as characteristic length and the maximum speed of the *initial velocity field* as characteristic speed), the mesh size for the velocity field is $h_v = 1/120$, the mesh size for the pressure is $h_p = 1/60$, and the time step is $\triangle t = 0.01$. In Fig. 14 a sequence of frames for the vorticity contours and stream lines at $t = 30$, 170, and 557.5 are shown. From $t = 0$ to about $t = 120$, vortices are stretched, collide and merge. By about $t = 120$, all major vortex-concentrations have merged into two strong counterrotating vortices; there are also two minor counterrotating vortices. Then after a long period of time ($t = 557.5$), there are two counterrotating vortices surviving. Let the flow kinetic energy be $E = \frac{1}{2}\int_\omega (u_1)^2+(u_2)^2\,d\mathbf{x}$ and the enstrophy of flow in the disk $W = \frac{1}{2}\int_\omega |\partial u_2/\partial x_1 - \partial u_1/\partial x_2|^2\,d\mathbf{x}$. The evolutions of E and W are shown in Fig. 15. At $t = 557.5$, E has decreased by a factor of 6.16×10^{-6} from its initial value which was 0.34731 and W has decreased to 1.04×10^{-6} from its initial value which was 49.588.

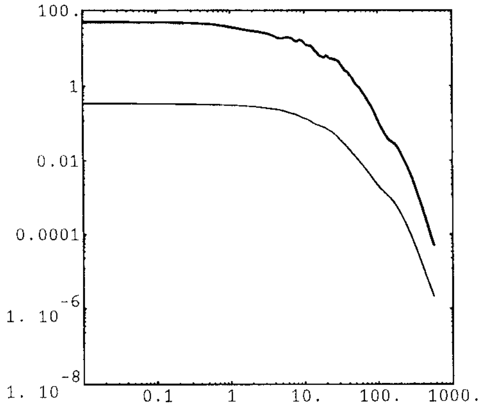

Fig. 15: Time evolution of energy E (thin line) and enstropy W (thick line).

6. Shape Optimization of the Boundary

Optimal shape design problems that are governed by partial differential equations have become a fruitful source of challenging and interesting free boundary problems. The study of particular classes of these problems has yielded solutions to fascinating applications (see for example [20–22]). Recently researchers have tried to employ the fictitious domain approach to problems of this type arising in various fields of science and engineering (see for example [23–26]). In this section we discuss such a strategy. A fictitious domain approach is employed to solve shape optimization problems for both Stokes and Navier–Stokes equations in two space dimensions.

The shape of a symmetric airfoil is given by its boundary γ. In our test problems we fix the chord length of the airfoil hence keeping the location of both the head and the tail of the airfoil fixed. We refer to [24] for more extensive numerical results of our algorithm on test problems where the location of the head and the tail of the airfoil are design or control variables, and also non–symmetric shapes were used to generate target pressures. With head and tail fixed, we parameterize the shape of the airfoil by 5 parameters, say κ_i where $i = 1, \ldots 5$. Indulging in a slight abuse of notation we can write the boundary, γ, as an explicit function of these 5 parameters. More precisely, we have $\gamma = \gamma_u(\kappa) \cup \gamma_l(\kappa)$ where the subscripts u and l denote upper and lower portions of the symmetric wing shape, respectively ($\gamma_l(\kappa)$ is a reflection of $\gamma_u(\kappa)$). In this way these parameters define the upper (and lower by symmetry) part of the airfoil by:

$$\gamma_u(\kappa) = \{(x_1, x_2) | x_2(x_1) = 5\rho(\kappa_1 x_1^{\frac{1}{2}} + \kappa_2 x_1 + \kappa_3 x_1^2 + \kappa_4 x_1^3 + \kappa_5 x_1^4)\}, \tag{67}$$

where x_1, x_2 are the coordinates of the part of the boundary of the airfoil, that part lying above the portion of the wing marked by the c in the figure below (see Fig. 16). Here the parameter ρ is a thickness parameter (for the NACA0012 we take $\rho = .12$). The airfoil is extended to be of length c' (see Fig. 16) by a closed trailing edge (see [24] for further details on this and other transformations, parameterizations and representations of the boundary).

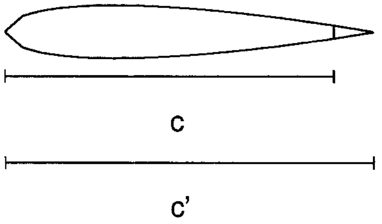

Fig. 16.

The NACA0012 is described by the parameter values,

$$\kappa^* = (0.2969, -0.126, -0.3516, 0.2843, -0.1015). \tag{68}$$

Given parameter values κ, we can employ the ideas and methodology from the preceding sections of this paper to simulate Stokes and Navier–Stokes flow over the associated κ–defined airfoil. This in turn yields a velocity field and a pressure distribution. Let p^* denote the pressure distribution associated with the NACA0012 values for κ^* in (68), hence p^* is our *target* pressure value. Now let $p(\kappa)$ represent the pressure distribution and $\mathbf{u}(\kappa)$ represent the velocity field associated with an arbitrary set of parameters κ. Similarly we can write the airfoil itself, $\omega(\kappa)$, as the region defined by the shape associated with arbitrary parameters, κ. We formulate the inverse problem of choosing κ to fit p^* in, for example, the least squares sense, subject to certain physical constraints. This inverse problem can be stated as a general nonlinear programming problem (NLP), namely

$$\min_{\kappa} ||p - p^*||_{L^2(\gamma)}, \ \textit{subject to } K(\kappa) \leq 0. \tag{69}$$

The constraint function $K(\kappa)$ in (69) implements the following constraints:

First, we require that $(\epsilon_w - x_2|_\gamma) \leq 0$ for any x_2 not an endpoint of the airfoil. Here ϵ_w is a positive minimum airfoil thickness parameter (for instance, we found that $\epsilon_w = 10^{-6}$ works well). This ensures that the upper and lower surface intersect in no more than two places (specifically at the two endpoints of the airfoil). Secondly, we require obvious physical upper and lower bounds on the volume of the airfoil, $\omega(\kappa)$, the arclength, γ, and the slopes, $\frac{\partial\gamma}{\partial x_1}$ and $\frac{\partial\gamma}{\partial x_2}$.

In practice, the overall effect of these constraints is that they trim computational work by keeping the minimization procedure away from obviously impossible airfoil shapes that yield very small but non–optimal objective function values. The importance of these constraints grows if a large number of design variables are used in conjunction with an initial guess for the design variables that results in an initial shape that is not close to the optimal shape. For the numerical results presented here, the constraints resulted in fewer simulations and hence shorter run–times, but otherwise did not drastically affect the minimization process. This was in part due to the small number of design variables and the "semi–local" starting value for the design variables (see Fig. 17 for a picture of the initial and optimal shapes). Finally, it is worth noting that for this particular test problem there were no binding constraints at the solution (i.e. $K(\kappa^*) < 0$).

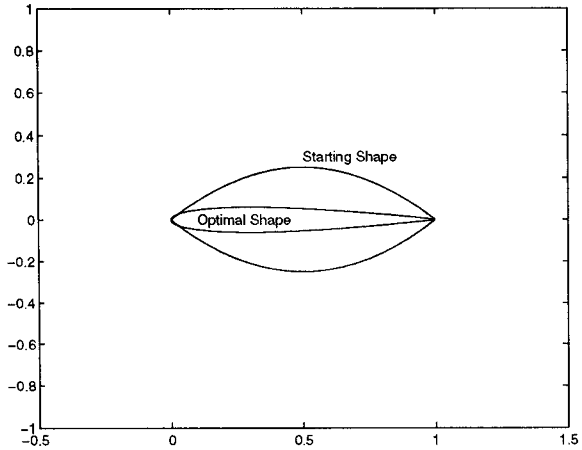

Fig. 17: The starting and optimal shapes for our test problems

6.1 Optimization Technique

While it may be possible to calculate derivatives of the objective function in (69) such a calculation would be extremely expensive and possibly inaccurate. An additional difficulty with using derivatives in the solution of (69) is the highly non–linear nature of the objective function. It can have many local minimizers and inflection points. Our numerical tests employing derivative based algorithms for the solution of (69) suggest that rapid convergence enjoyed by some gradient based methods takes place only in an extremely small neighborhood of the solution, if at all. Finally we would enjoy the luxury of using an algorithm that allowed us to replace the smooth least squares objective function in (69) with a continuous but non-differentiable one (for instance an L^∞ objective function).

Algorithms for unconstrained minimization that require no derivative information (usually referred to as *direct search* or *pattern search* methods) are far from new (see, for example, [27] and the references therein). In [28] an algorithm for unconstrained optimization that requires no derivative information was suggested. A parallel implementation of the algorithm has also been developed and tested (see [29]). Recently, this algorithm and its implementation have been modified to handle constraints (see [30]). The method samples points on the nodes of an evolving pattern, moving to the node in the pattern with the value closest to optimality. Depending on where in the pattern the point closest to optimality was located the pattern then changes, either expanding or contracting. When the pattern expands its overall size grows causing the algorithm to sample points farther from its current location. When the pattern contracts the overall size of the pattern becomes smaller; this procedure terminates when the lengths of the edges of the pattern fall below a user prescribed tolerance.

Pattern search methods of this sort typically do not demonstrate rapid local convergence, but they are extremely robust and far less susceptible than faster higher-order methods to the difficulties introduced when functions are non-smooth or the data is noisy. Pattern search methods are usually too slow to solve optimization problems with large numbers of parameters. Our problem (69) is interesting in that one can accurately describe the shape of an airfoil with a relatively small number of parameters. The compatibility of constrained pattern search method with our problem (69) is also interesting. After numerous numerical tests, we have not failed to locate the global minimum of (69).

6.2 Numerical Example

h_p	h_v	Δt	Reynolds	∠ of Attack	Function evaluations	Cost Function
1/64	1/128	2.5e-3	1000	0^o	2.5E2	4.9E-1
1/64	1/128	2.5e-3	500	0^o	2.5E2	1.1E-1
1/32	1/64	5.0e-3	1000	0^o	1.8E2	5.7E-1
1/32	1/64	5.0e-3	500	0^o	1.8E2	1.3E-1
1/64	1/128	2.5e-3	1000	5^o	1.9E2	9.8E-2
1/64	1/128	2.5e-3	500	5^o	1.9E2	9.4E-2
1/32	1/64	5.0e-3	1000	5^o	1.5E2	1.5E-1
1/32	1/64	5.0e-3	500	5^o	1.5E2	1.4E-1

Table 1: **Coarse Minimization: 10 points per processor**

h_p	h_v	Δt	Reynolds	∠ of Attack	Function evaluation	Cost Function
1/64	1/128	2.5e-3	1000	0^o	2.1E2	1.8E-4
1/64	1/128	2.5e-3	500	0^o	2.1E2	2.4E-4
1/32	1/64	5.0e-3	1000	0^o	1.5E2	5.7E-4
1/32	1/64	5.0e-3	500	0^o	1.5E2	1.3E-4
1/64	1/128	2.5e-3	1000	5^o	2.0E2	1.2E-4
1/64	1/128	2.5e-3	500	5^o	2.0E2	1.6E-4
1/32	1/64	5.0e-3	1000	5^o	1.6E2	1.4E-4
1/32	1/64	5.0e-3	500	5^o	1.6E2	1.1E-4

Table 2: **Fine Minimization: 25 points per processor**

Our numerical example for shape optimization requires integrating equations of the form (22)–(23) to evaluate the objective function. We consider a zero degree and five degrees angles of attack for these test problems and we keep the same boundary conditions defined by (25) and (26). Parameter values are provided in the tables below (see Table 1 and Table 2). The cost function column displays, in fact, $||p_c - p^*||_{L^2(\gamma)}/||p^*||_{L^2(\gamma)}$ where p_c is the computed pressure.

We used 16 processors of the Touchstone Delta machine to perform the minimization in parallel. A two stage coarse/fine strategy was employed to solve the shape optimization minimization; 10 points per processor were sampled with a very large search strategy initially in order to locate a vague neighborhood of the solution and then 25 points per processor were sampled with a smaller pattern search strategy to polish the quality of the solution. For more details on constrained pattern search methods see [30].

7. CONCLUSION

Compared to the previous results in [31] it is clear that the fictitious domain methodology that we advocate has been substantially progressing and looks also promising for the simulation of time dependent solution of viscous flow problems around moving rigid bodies. This solution technique appears to be well suited to the curved domains on which we prescribe Dirichlet boundary conditions that arise from this class of shape optimization problems (see [32] for some theoretical results). As always, there is still room for improvement. We mention particularly the speed up of the various iterative methods used for the solution the subproblems obtained from the time splitting. Parallelization is also an important and timely issue currently addressed (see [13, 14]).

It appears that the fictitious domain method provides an effective way of evaluating the objective function for the Stokes and Navier–Stokes flow shape optimization problem. In particular, the computational cost of regriding a mesh for every trial airfoil shape could be a prohibitive cost. Current work includes extending these ideas to more complicated objective functions and to the shape optimization problems associated with drag reduction.

Acknowledgement

We would like to acknowledge the helpful comments and suggestions of the following individuals: L. C. Cowsar, J. E. Dennis, G. H. Golub, J. Haslinger, Y. Kuznetsov, W. Lawton, P. Le Tallec, P. L. Lions, O. Pironneau, J. Pasciak, H. Resnikoff, R. A. Tapia, J. Weiss, R. O. Wells, M. F. Wheeler, O. B. Widlund, V. Torczon, and X. Zhou.

The support of the following corporations and institutions is also acknowledged: AWARE, CERFACS, CRPC, Dassault Aviation, INRIA, University of Houston, Université P. et M. Curie, Rice University. We also benefited from the support of DARPA (Contracts AFOSR F49620–89–C–0125 and AFOSR–90–0334), DRET (Grant 89424), NSF (Grants INT 8612680, DMS 8822522 and DMS 9112847) and the Texas Board of Higher Education(Grants 003652156ARP and 003652146ATP).

Use of the Touchstone Delta Machine located at Caltech was provided by the Center for Research on Parallel Computation funded by the NSF through Cooperative Grant No. CCR-9120008

REFERENCES

1. D.P. Young, R.G. Melvin, M.B. Bieterman, F.T. Johnson, S.S. Samanth, J.E. Bussoletti (1991) A locally refined finite rectangular grid finite element method. Application to Computational Physics; J. Comp. Physics, vol.92 pp.1–66
2. J.E. Bussoletti, F.T. Johnson, S.S. Samanth, D.P. Young, R.H. Burkhart (1991) EM–TRAN- AIR: Steps toward solution of general 3D Maxwell's equations *Computer Methods in Applied Sciences and Engineering*; R. Glowinski ed., pp.49–72 Nova Science, Commack, NY,
3. B.L. Buzbee, F.W. Dorr, J.A. George, G.H. Golub (1971) The direct solution of the discrete Poisson equation on irregular regions; SIAM J. Num. Anal., vol.8 pp.722–736
4. R. Glowinski, T.W. Pan, J. Periaux (1994) A fictitious domain method for Dirichlet problem and applications; Comp. Meth. Appl. Mech. Eng., vol.111 pp.283–303
5. R. Glowinski, T.W. Pan, J. Periaux (1994) A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations; Comp. Meth. Appl. Mech. Eng., vol.112 pp.133–148
6. R. Glowinski, T.W. Pan, J. Periaux (1995) A Lagrange multiplier/fictitious domain method for the Dirichlet problem. Generalization to some flow problems; Japan J. Ind. Appl. Math. (to appear)
7. C. Borgers (1990) Domain embedding methods for the Stokes equations; Num. Math., vol.57 pp.435–451
8. R. Glowinski (1984) *Numerical methods for nonlinear variational problems;* Springer–Verlag, New York
9. M.O. Bristeau, R. Glowinski, B. Mantel, J. Periaux and P. Perrier (1985) Numerical methods for incompressible and compressible Navier–Stokes problems; *Finite Elements in Fluids*, Vol. 6, R. H. Gallagher, G. Carey, J. T. Oden, and O. C. Zienkiewicz eds., Wiley, Chichester, pp.1–40
10. R. Glowinski (1985) Viscous flow simulation by finite element methods and related numerical techniques; *Progress and Supercomputing in Computational Fluid Dynamics*, E. M. Murman and S. S. Abarbanel eds., Birkhauser, Boston, pp.173–210
11. M.O. Bristeau, R. Glowinski and J. Periaux (1987) Numerical methods for the Navier–Stokes equations; Comp. Phys. Rep., vol.6 pp.73–187
12. R. Glowinski (1991) Finite element methods for the numerical simulation of incompressible viscous flow. Introduction to the control of the Navier–Stokes equations; Lectures in Applied Mathematics, AMS, Providence, RI, vol.28 pp.219–301
13. R. Glowinski, T.W. Pan, J. Periaux, A one shot domain decomposition/fictitious domain method for the solution of elliptic equations; *the Proceedings of Parallel CFD'93* (to appear)
14. R. Glowinski, T.W. Pan, J. Periaux, One shot fictitious domain/domain decomposition methods for three–dimensional elliptic problems. Parallel implentation on a KSR machine; *the Proceedings of Parallel CFD'94* (to appear)
15. E. Dean, R. Glowinski (1993) A domain decomposition method for the wave equation; *Les Grands Systemes des Sciences et de la Technologie*, J. Horowitz, J.L. Lions eds.,

Masson, Paris

16. O. Pironneau (1989) *Finite element methods for fluids;* J. Wiley, Chichester
17. E. Caglioti, P.L. Lions, C. Marchioro, M. Pulvirenti (1992) A special class of stationary flows for two-dimensional Euler equations: a statistical mechanics description; Comm. Math. Phys., vol.143 pp.501–525
18. E. Caglioti, P.L. Lions, C. Marchioro, M. Pulvirenti, A special class of stationary flows for two-dimensional Euler equations: a statistical mechanics description. Part II; Comm. Math. Phys. (to appear)
19. W.H. Matthaeus, W.T. Stribling, D. MartineZ, S. Oughton, D. Montgomery (1991) Selective decay and coherent vortices in two-dimensional incompressible turbulence; Phys. Rev. Lett., vol.66 pp.2731–2734
20. O. Pironneau (1984) *Shape Optimization;* Springer–Verlag, New York
21. D. Begis, R. Glowinski (1975) Application de la méthod des éléments finis à l'approximation d'un problème de domaine optimal. Méthodes de résolution des problèmes approchés; Appl. Math and Optimization, vol.2 pp.130–169
22. J. Haslinger, P. Neittaanmaki (1988) *Finite element approximation for optimal shape design : theory and applications;* Wiley, New York
23. J. Haslinger (1992) Embedding/control approach for solving optimal shape design problems; East-West J. Numer. Math., vol.1 pp.111–119
24. A.J. Kearsley (1995) *Optimization methods for partial differential equations from science and engineering; Ph. D. dissertation, Department of Computational & Applied Mathematics, Rice University, Houston, TX*
25. J. Haslinger, A. Klarbring (1995) Fictitious Domain/mixed finite element approach for a class of optimal shape design problems; Mathematical Modelling and Numerical Analysis (to appear)
26. J. Haslinger, K. Hoffman, M. Kocvara (1993) Control/Fictitious domain method for solving optimal shape design problems; Mathematical Modelling and Numerical Analysis, vol.27 pp.157–189
27. J. Céa (1971) *Optimisation : Théorie et Algorithmes;* Dunod, Paris
28. J.E. Dennis, V. Torczon (1991) Direct search methods on parallel machines; SIAM J. on Optimization, vol.1 pp.448-474
29. V. Torczon, PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines; Trans. Math. Software (to appear)
30. A.J. Kearsley, R.A. Tapia, V. Torczon (1993) On the use of parallel direct search methods for nonlinear programming problems; Technical Report #93-33, Department of Computational & Applied Mathematics, Rice University, Houston, TX
31. R. Glowinski, T.W. Pan, J. Periaux, M. Ravachol (1991) A fictitious domain method for the incompressible Navier–Stokes equations; *The Finite Element Method in the 90's*, E. Oñate, J. Periaux, A. Samuelson eds., Springer–Verlag, Berlin, pp.440–417
32. V. Girault, R. Glowinski, Error Analysis of a fictitious domain method applied to a Dirichlet problem; Japan J. Ind. Appl. Math. (to appear)