Contemporary Mathematics
Volume **180**, 1994

A One Shot Domain Decomposition/Fictitious Domain method for the Navier–Stokes Equations

ROLAND GLOWINSKI, TSORNG–WHAY PAN AND JACQUES PÉRIAUX

ABSTRACT. In this paper which is motivated by computation on parallel MIMD machines, we address the numerical solution of some class of elliptic problems by a combination of domain decomposition and fictitious domain methods. We take advantage of the fact that the Steklov–Poincare 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 conjugate gradient type with good parallelization properties which force simultaneously the matching at the subdomain interfaces and the actual boundary conditions. Preliminary results obtained on a KSR machine are presented. A similar methodology has been applied to simulate viscous flows around obstacles modelled by Navier–Stokes equations.

1. Introduction

Fictitious domain methods for Partial Differential Equations have shown recently a most interesting potential for solving complicated problems from Science and Engineering (see, e.g., [**1, 2**] for some impressive illustrations of the above statement). The main reason of popularity of fictitious domain methods (sometime called *domain imbedding methods*; cf. [**3**]) is that they allow the use of fairly structured meshes on a simple shape auxiliary domain containing the actual one, allowing therefore the use of fast solvers. In [**4, 5**], we have used Lagrange multiplier and finite element methods combined with fictitious domain techniques to compute the numerical solutions of elliptic problems with Dirichlet boundary conditions and simulate some nonlinear time dependent problems, namely the flow of a viscous–plastic medium in a cylindrical pipe and time dependent external incompressible viscous flow modelled by the Navier–Stokes equations.

1991 *Mathematics Subject Classification.* Primary 65M60; Secondary 76D05.
The final version of this paper will be submitted for publication elsewhere.

In this paper motivated by computation on parallel MIMD machines, we address the numerical solution of a class of elliptic problems by a combination of domain decomposition and fictitious domain methods. From the fact that the Steklov–Poincare operators associated with the subdomain interfaces and with the fictitious domain treatment of internal boundaries have very similar properties; we derive fast solution methods of conjugate gradient type with good parallelization properties which enforce simultaneously the matching at the subdomain interfaces and the actual boundary conditions. A similar methodology has been applied to simulate viscous flows around obstacles modelled by Navier–Stokes equations. In Section 2, we describe the formulation of a family of Dirichlet problems and discuss an equivalent formulation which is at the basis of the domain decomposition/fictitious domain methods. Preliminary results obtained on a KSR1 machine are presented. In Section 3 we apply a similar methodology to simulate external incompressible viscous flow modelled by Navier–Stokes equations.

2. One Shot DD/FD Method for the Dirichlet problem

2.1. Formulation of the Dirichlet problem

We consider the following elliptic problem:

$$\alpha u - \nu \Delta u = f \; in \, \Omega \setminus \bar{\omega}, \tag{2.1}$$

$$u = g_0 \; on \, \gamma, \; u = g_1 \; on \, \Gamma, \tag{2.2}$$

where Ω is a "box" domain in $\mathbb{R}^d (d \geq 1)$, ω is a bounded domain in $\mathbb{R}^d (d \geq 1)$ such that $\omega \subset\subset \Omega$ (e.g., see Fig. 2.1 in which $\Omega = \Omega_1 \cup \Omega_2$ and $\omega = \omega_1 \cup \omega_2$), Γ (resp., γ) is the boundary $\partial\Omega$ (resp., $\partial\omega$), $\alpha \geq 0$, 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 (2.1)–(2.2) has a unique solution. The equivalent variational formulation of problem (2.1)–(2.2) is

$$\int_{\Omega_0} (\alpha u v + \nu \nabla u \cdot \nabla v) \, dx = \int_{\Omega_0} f v \, dx, \; \forall v \in V_0; \; u \in V_g, \tag{2.3}$$

where $\Omega_0 = \Omega \setminus \bar{\omega}$, $V_g = \{v | v \in H^1(\Omega_0), v = g_0 \; on \, \gamma, v = g_1 \; on \, \Gamma\}$ and $V_0 = \{v | v \in H^1(\Omega_0), v = 0 \; on \, \gamma \cup \Gamma\}$.

2.2. Domain decomposition/fictitious domain approach

For simplicity we consider the case where ω is the union of two disjoint bounded domains, ω_1 and ω_2, and Ω is the union of two subdomains Ω_1 and Ω_2 (see Fig. 2.1); we denote by γ_0 the interface between Ω_1 and Ω_2, by γ_1 (resp., γ_2) the boundary of ω_1 (resp., ω_2), and let $\Gamma_1 = \Gamma \cap \partial\Omega_1$ and $\Gamma_2 = \Gamma \cap \partial\Omega_2$. Combining the fictitious domain method discussed in [**4**] to a domain decomposition method (see, e.g., [**6**]) and applying to the solution of problem (2.1)–(2.2), (2.3), we obtain the following equivalent problem:

$$
(2.4)\qquad \begin{cases} Find\, u_i \in V_g^i,\ \lambda_i \in L^2(\gamma_i),\ \lambda_d \in L^2(\gamma_0)\ such\, that \\ \int\limits_{\Omega_i} (\alpha u_i v_i + \nu \nabla u_i \cdot \nabla v_i)\, dx = \int\limits_{\Omega_i} \tilde{f} v_i\, dx \\ + \int\limits_{\gamma_i} \lambda_i v_i\, d\gamma + (-1)^i \int\limits_{\gamma_0} \lambda_d v_i\, d\gamma,\ \forall v_i \in V_0^i,\ for\, i = 1,2, \end{cases}
$$

$$
(2.5)\qquad \int\limits_{\gamma_i} \mu_i (u_i - g_0)\, d\gamma = 0\,, \forall \mu_i \in L^2(\gamma_i),\ for\, i = 1,2,
$$

$$
(2.6)\qquad \int\limits_{\gamma_0} \mu_d (u_2 - u_1)\, d\gamma = 0\,, \forall \mu_d \in L^2(\gamma_0),
$$

where $\tilde{f}$ is a $L^2(\Omega)$ extension of f, $V_g^i = \{v | v \in H^1(\Omega_i), v = g_1\, on\, \Gamma_i\}$ and $V_0^i = \{v | v \in H^1(\Omega_0), v = 0\, on\, \Gamma_i\}$ for $i = 1,2$. We have equivalence in the sense that if relations (2.4)–(2.6) hold then $u_i = u|_{\Omega_i}$, for $i = 1,2$, and conversely.

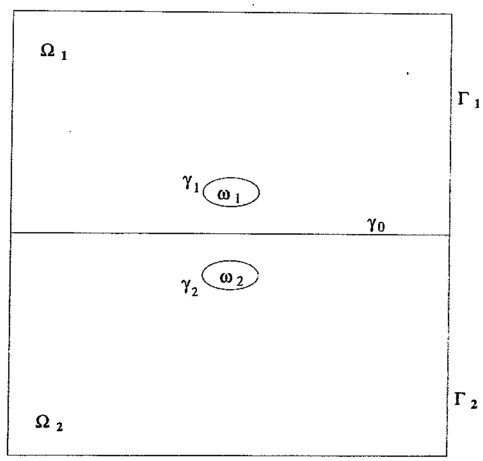

Figure 2.1.

In (2.4)–(2.6), the function λ_i which is a *Lagrange multiplier* associated with the *boundary condition* $u = g_0$ on γ_i is essentially the jump of $\nu \frac{\partial u}{\partial n}$ at γ_i for $i = 1,2$ and the function λ_d which can be viewed as a *Lagrange multiplier* associated with the *interface boundary condition* $u_1 = u_2$ on γ_0 is nothing but the function $\nu \frac{\partial u}{\partial n_2}|_{\gamma_0} = -\nu \frac{\partial u}{\partial n_1}|_{\gamma_0}$, where n_i is the normal unit vector at γ_0, outward to Ω_i.

Due to the combination of the two methods, there are *two Lagrange multipliers* associated with the *boundary conditions* and with the matching of solution at the *subdomain interfaces*, respectively. We can solve the saddle–point system (2.4)–(2.6) by a conjugate gradient algorithm driven by the multiplier associated with the boundary conditions, the one driven by the multiplier associated with the matching at the subdomain interfaces, or by the one called the *one shot method* driven by the *two multipliers* at the same time [7]. These methods have different parallelization properties and can be parallelized on MIMD machines. The one driven by the multiplier associated with the boundary conditions was discussed in [7] and its speed on a KSR1 is much slower than that of the one shot method for the two subdomains decomposition shown in Fig. 2.1. In Section 2.3 we are would like to test the one shot method with more subdomains.

2.3. Performance on a KSR1 machine

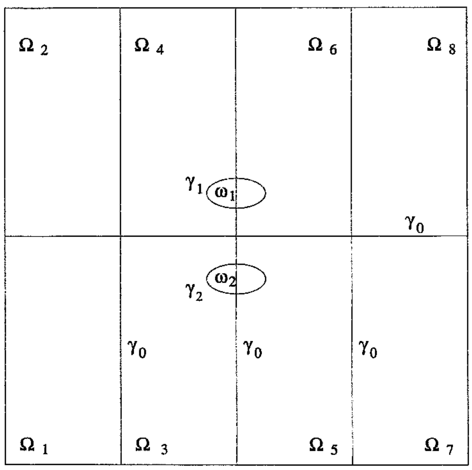

Figure 2.2.

We consider problem (2.1)–(2.2) with $\alpha = \nu = 1$ as test problem and let $u(x,y) = x^2+y^2$ be the solution of the test problem. Then $f(x,y) = x^2+y^2-4$. Let $\omega = \omega_1 \cup \omega_2$ where $\omega_i = \{(x,y)|\frac{(x-1.0)^2}{(1/8)^2} + \frac{(y-c_i)^2}{(1/16)^2} < 1\}$, for $i = 1,2$, c_1 =1.1875, c_2=0.8125; take $\Omega = (0,2) \times (0,2)$.

In the numerical experiments, we consider the case where Ω is the union of eight subdomains $\Omega_1, \ldots,$ to Ω_8 (see Fig. 2.2). The finite dimensional spaces V_{gh}^i and V_{0h}^i of V_g^i and V_0^i, respectively for $i = 1,\ldots,8$ are as follows:

$$V_{gh}^i = \{v_h | v_h \in V_g^i \cap C^0(\bar{\Omega}_i), v_h = g_h \, on \, \Gamma_i, v_h|_T \in P_1 \, \forall T \in \mathcal{T}_h^i\},$$
$$V_{0h}^i = \{v_h | v_h \in V_0^i \cap C^0(\bar{\Omega}_i), v_h = 0 \, on \, \Gamma_i, v_h|_T \in P_1 \, \forall T \in \mathcal{T}_h^i\},$$

where g_h is an approximation of g, $\mathcal{T}_h^i$ is a triangulations of Ω_i for $i = 1,\ldots,8$ and P_1 is the space of the polynomials in x, y of degree ≤ 1. For $i = 0,1,2$, the finite dimensional space Λ_h^i of $L^2(\gamma_i)$ is defined as follows:

$$\Lambda_h^i = \{\mu_h | \mu_h \in L^\infty(\gamma_i), \mu_h \, is \, constant \, on \, the \, segment \, joining \, 2 \, consecutive \, mesh \, points \, on \, \gamma_i\}.$$

The choice of mesh points on γ_1 and γ_2 are shown on Fig. 2.3. For stability reason –the so called LBB inf–sup condition– the length of each segment on γ_1 and γ_2 has to be chosen greater than the meshsize h. The obvious choice for the mesh points on γ_0 are the midpoints of the edges located on γ_0 (see Fig. 2.3).

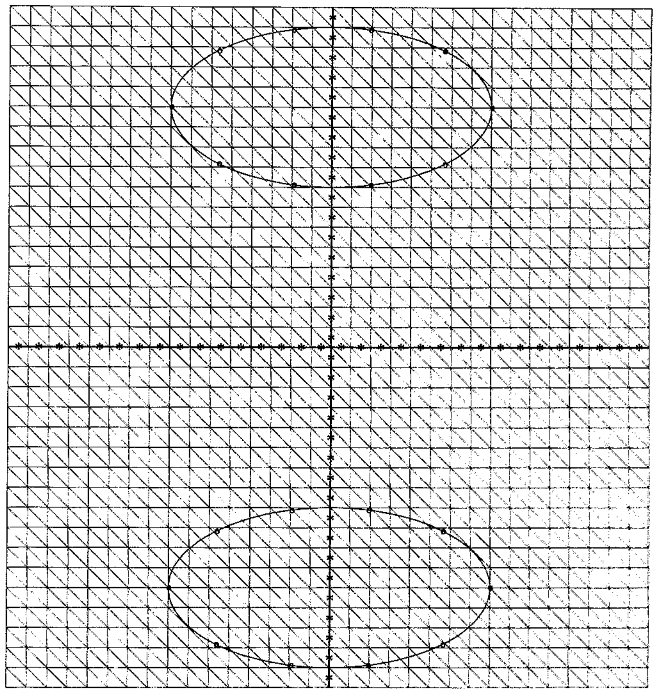

FIGURE 2.3. MESH POINTS MARKED BY "*" ON γ_1 AND γ_2 AND PART OF MESH POINTS MARKED BY "*" ON γ_0 WITH $h = 1/64$.

In the one shot method, the elliptic problems have been solved on each subdomain by a *Fast Elliptic Solver* based on *cyclic reduction* [**3, 8–10**]. Concerning implementation of the one shot method on the KSR1 machine, eight discrete elliptic problems can be solved simultaneously. For meshsize $h = 1/32$, $1/64$, $1/128$, and $1/256$, the number of iterations of the one shot method is 78, 91, 116, and 151 respectively and the number of iterations for the preconditioned one shot method is 68, 75, 88, and 92 respectively. Thus the preconditioner for two dimensional problems works very well. The CPU time per iteration of the one shot method with or without preconditioner is about the same. In Tables 2.1 and 2.2 we have shown the elapsed time and speedup per iteration of the discrete analogues of the preconditioned one shot method for different meshsizes where N_p is the number of processors used in computation. The speedup per iteration in Table 2.2 is better as the size of problem is larger.

Table 2.1. CPU per iteration on a KSR1				
N_p	h=1/32	h=1/64	h=1/128	h=1/256
1	0.291 sec.	1.130 sec.	5.251 sec.	22.608 sec.
2	0.189 sec.	0.649 sec.	2.828 sec.	11.756 sec.
4	0.125 sec.	0.394 sec.	1.543 sec.	6.177 sec.
8	0.099 sec.	0.231 sec.	0.883 sec.	3.334 sec.

Table 2.2. Speedup per iteration on a KSR1				
N_p	h=1/32	h=1/64	h=1/128	h=1/256
1	1.00	1.00	1.00	1.00
2	1.54	1.74	1.86	1.92
4	2.33	2.87	3.40	2.66
8	2.94	4.89	5.95	6.78

3. External incompressible viscous flow

3.1. Navier–Stokes equations

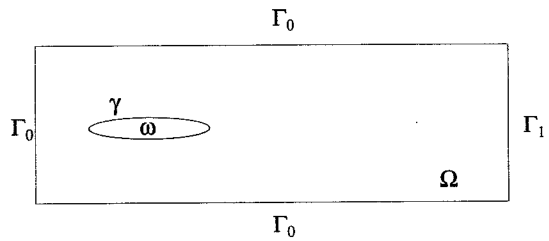

Figure 3.1.

In [**5**], we have used Lagrange multiplier/fictitious domain methods with finite element methods to simulate external incompressible viscous flow modelled by the Navier–Stokes equations. Here we would like to consider the same flow problems with a one shot method. The Navier–Stokes equations are the following:

$$\frac{\partial \mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} \; in \; \Omega\setminus\bar{\omega}, \tag{3.1}$$

$$\nabla\cdot\mathbf{u} = 0 \; in \; \Omega\setminus\bar{\omega}, \tag{3.2}$$

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x}\in\Omega\setminus\bar{\omega}, (with \; \nabla\cdot\mathbf{u}_0 = 0), \tag{3.3}$$

$$\mathbf{u} = \mathbf{g}_0 \; on \; \Gamma_0, \; \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p = \mathbf{g}_1 \; on \; \Gamma_1, \; \mathbf{u} = \mathbf{g}_2 \; on \; \gamma. \tag{3.4}$$

In (3.1)–(3.4), Ω and ω are bounded domains in $\mathbb{R}^d (d\geq 2)$ (see Fig. 3.1), Γ (resp., γ) is the boundary of Ω (resp., ω) 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 presure, $\mathbf{f}$ is a density of external forces, $\nu(>0)$ is a viscosity parameter, and

$(\mathbf{v}\cdot\nabla)\mathbf{w} = \{\sum_{j=1}^{j=d} v_j \frac{\partial w_i}{\partial x_j}\}_{i=1}^{i=d}$. For the fictitious domain formulation, we imbed $\Omega \setminus \bar{\omega}$ in Ω and define

$$\mathbf{V}_{\mathbf{g}_0} = \{\mathbf{v}|\mathbf{v} \in (\mathbf{H}^1(\Omega))^d, \mathbf{v} = \mathbf{g}_0 \, on \, \Gamma_0\}, \tag{3.5}$$

$$\mathbf{V}_0 = \{\mathbf{v}|\mathbf{v} \in (\mathbf{H}^1(\Omega))^d, \mathbf{v} = \mathbf{0} \, on \, \Gamma_0\}, \tag{3.6}$$

$$\Lambda = (\mathbf{L}^2(\gamma))^d. \tag{3.7}$$

Let $\mathbf{U}_0$ be an extension of $\mathbf{u}_0$ with $\nabla \cdot \mathbf{U}_0 = 0$ in Ω and $\tilde{\mathbf{f}}$ an extension of $\mathbf{f}$. Then we have equivalence between (3.1)–(3.4) and the following problem

For $t > 0$, find $\mathbf{U}(t) \in \mathbf{V}_{\mathbf{g}_0}$, $P(t) \in \mathbf{L}^2(\Omega)$, $\lambda(t) \in \Lambda$ such that

$$\begin{cases} \int_\Omega \frac{\partial \mathbf{U}}{\partial t} \cdot \mathbf{v} \, d\mathbf{x} + \nu \int_\Omega \nabla \mathbf{U} \cdot \nabla \mathbf{v} \, d\mathbf{x} + \int_\Omega (\mathbf{U} \cdot \nabla)\mathbf{U} \cdot \mathbf{v} \, d\mathbf{x} \\ \quad - \int_\Omega P \nabla \cdot \mathbf{v} \, d\mathbf{x} = \int_\Omega \tilde{\mathbf{f}} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Gamma_1} \mathbf{g}_1 \cdot \mathbf{v} \, d\Gamma \\ \quad + \int_\gamma \lambda \cdot \mathbf{v} \, d\gamma, \ \forall \mathbf{v} \in \mathbf{V}_0, \ a.e. \, t > 0, \end{cases} \tag{3.8}$$

$$\nabla \cdot \mathbf{U}(t) = 0 \, in \, \Omega, \ \mathbf{U}(\mathbf{x}, 0) = \mathbf{U}_0(\mathbf{x}), \ \mathbf{x} \in \Omega, \tag{3.9}$$

$$\mathbf{U}(t) = \mathbf{g}_2(t) \, on \, \gamma, \tag{3.10}$$

in the sense that $\mathbf{U}|_{\Omega\setminus\bar{\omega}} = \mathbf{u}$, $P|_{\Omega\setminus\bar{\omega}} = p$. The multiplier λ is the jump of $\nu \frac{\partial \mathbf{U}}{\partial \mathbf{n}} - \mathbf{n}P$ at γ and the effect of the actual geometry is concentrated on $\int_\gamma \lambda \cdot \mathbf{v} \, d\gamma$ in the right–hand–side of (3.8), and on (3.10).

To solve (3.8)–(3.10), we shall consider a time discretization by an operator splitting method, like the ones in, e.g., [**11–14**]. With these methods we can decouple the nonlinearity and the incompressibility in the Navier–Stokes/fictitious domain problem (3.8)–(3.10). Applying the *θ–scheme* (cf. [**14**]) to (3.8)–(3.10), we obtain *quasi–Stokes/fictitious domain subproblems* and nonlinear *advection–diffusion subproblems* (e.g., see [**5**]). In Section 3.2, a one shot method for the quasi–Stokes/FD subproblems shall be discussed. Due to the fictitious domain method and the operator splitting method, advection–diffusion subproblems may be solved in a least–squares formulation by a conjugate gradient alogrithm [**14**] in a simple shape auxiliary domain Ω without concern for the constraint $\mathbf{u} = \mathbf{g}$ at γ. Thus, advection–diffusion subproblems can be solved with domain decomposition methods.

3.2. The one shot method for the quasi–Stokes/FD subproblems

The quasi–Stokes/fictitious domain subproblem is the following:

$$\text{(3.11)}\quad \begin{cases} \textit{Find}\, \mathbf{U} \in \mathbf{V}_{\mathbf{g}_0},\ P \in \mathbf{L}^2(\Omega),\ \lambda \in \Lambda\ \textit{such that} \\ \int\limits_\Omega (\alpha \mathbf{U}\cdot\mathbf{v} + \nu\nabla\mathbf{U}\cdot\nabla\mathbf{v})\, d\mathbf{x} - \int\limits_\Omega P\nabla\cdot\mathbf{v}\, d\mathbf{x} \\ -\int\limits_\gamma \lambda\cdot\mathbf{v}\, d\gamma = \int\limits_\Omega \mathbf{f}\cdot\mathbf{v}\, d\mathbf{x} + \int\limits_{\Gamma_1} \mathbf{g}_1\cdot\mathbf{v}\, d\Gamma,\ \forall\mathbf{v}\in\mathbf{V}_0, \end{cases}$$

$$\text{(3.12)}\quad \nabla\cdot\mathbf{U} = 0\ \textit{in}\ \Omega,$$

$$\text{(3.13)}\quad \mathbf{U} = \mathbf{g}_2\ \textit{on}\ \gamma,$$

The one shot methodology has been used for solving problem (3.11)–(3.13) in which the *two Lagrange multipliers* are *the pressure* P for (3.12) and λ for *the actual boundary condition* (3.13). Here we consider a bilinear form $b(\cdot,\cdot)$ which is symmetric and elliptic over Λ. We may choose $b(\lambda,\mu) = \int_\gamma \lambda\cdot\mu\, d\gamma$, $\forall\lambda,\ \mu\in\Lambda$.

The one shot algorithm is the following:

(3.14) $\{P^0, \lambda^0\} \in \mathbf{L}^2(\Omega)\times\Lambda$ *given; solve the following Dirichlet problem:*

$$\text{(3.15)}\quad \begin{aligned} &\int\limits_\Omega (\alpha\mathbf{U}^0\cdot\mathbf{v} + \nu\nabla\mathbf{U}^0\cdot\nabla\mathbf{v})\, d\mathbf{x} = \int\limits_\Omega \mathbf{f}\cdot\mathbf{v}\, d\mathbf{x} + \\ &\int\limits_\gamma \lambda^0\cdot\mathbf{v}\, d\gamma + \int\limits_\Omega P^0\nabla\cdot\mathbf{v}\, d\mathbf{x} + \int\limits_{\Gamma_1} \mathbf{g}_1\cdot\mathbf{v}\, d\Gamma,\ \forall\mathbf{v}\in\mathbf{V}_0;\ \mathbf{U}^0\in\mathbf{V}_{\mathbf{g}_0}, \end{aligned}$$

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:*

$$\text{(3.16)}\quad g_1^0 = \alpha\phi^0 + \nu r_1^0,$$

with ϕ^0 *the solution of*

$$\text{(3.17)}\quad \begin{cases} -\Delta\phi^0 = r_1^0\ \textit{in}\ \Omega, \\ \dfrac{\partial\phi^0}{\partial\mathbf{n}} = 0\ \textit{on}\ \Gamma_0;\ \phi^0 = 0\ \textit{on}\ \Gamma_1, \end{cases}$$

$$\text{(3.18)}\quad b(\mathbf{g}_2^0,\mu) = \int\limits_\gamma \mathbf{r}_2^0\cdot\mu\, d\gamma\,\forall\mu\in\Lambda;\ \mathbf{g}_2^0\in\Lambda.$$

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, λ^n, $\mathbf{U}^n$, r_1^n, $\mathbf{r}_2^n$, $\mathbf{w}^n$, $\mathbf{g}^n$ *are known, compute* P^{n+1}, λ^{n+1}, $\mathbf{U}^{n+1}$, $\mathbf{w}^{n+1}$, r_1^{n+1}, $\mathbf{r}_2^{n+1}$, $\mathbf{g}^{n+1}$ *as follows:*

solve the intermediate Dirichlet problem:

$$\text{(3.19)}\quad \begin{aligned} &\int\limits_\Omega (\alpha\bar{\mathbf{U}}^n\cdot\mathbf{v} + \nu\nabla\bar{\mathbf{U}}^n\cdot\nabla\mathbf{v})\, d\mathbf{x} \\ &= \int\limits_\gamma \mathbf{w}_2^n\cdot\mathbf{v}\, d\gamma + \int\limits_\Omega w_1^n\nabla\cdot\mathbf{v}\, d\mathbf{x},\ \forall\mathbf{v}\in\mathbf{V}_0;\ \bar{\mathbf{U}}^n\in\mathbf{V}_0, \end{aligned}$$

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:*

$$\bar{g}_1^n = \alpha\bar{\phi}^n + \nu\bar{r}_1^n, \tag{3.20}$$

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

$$\begin{cases} -\Delta\bar{\phi}^n = \bar{r}_1^n \; in \, \Omega, \\ \dfrac{\partial\bar{\phi}^n}{\partial\mathbf{n}} = 0 \; on \, \Gamma_0; \; \bar{\phi}^n = 0 \; on \, \Gamma_1, \end{cases} \tag{3.21}$$

$$b(\bar{\mathbf{g}}_2^n, \mu) = \int_\gamma \bar{\mathbf{r}}_2^n \cdot \mu \, d\gamma \; \forall \mu \in \Lambda; \; \bar{\mathbf{g}}_2^n \in \Lambda. \tag{3.22}$$

We compute then $\rho_n = \int_\Omega r_1^n g_1^n \, d\mathbf{x} + \int_\gamma \mathbf{r}_2^n \cdot \mathbf{g}_2^n \, d\gamma / \int_\Omega \bar{r}_1^n w_1^n \, d\mathbf{x} + \int_\gamma \bar{\mathbf{r}}_2^n \cdot \mathbf{w}_2^n \, d\gamma$, *and set*

$$P^{n+1} = P^n - \rho_n w_1^n, \; \mathbf{U}^{n+1} = \mathbf{U}^n - \rho_n \bar{\mathbf{U}}^n, \tag{3.23}$$

$$\lambda^{n+1} = \lambda^n - \rho_n \mathbf{w}_2^n, \; \mathbf{g}^{n+1} = \mathbf{g}^n - \rho_n \bar{\mathbf{g}}^n, \tag{3.24}$$

$$r_1^{n+1} = r_1^n - \rho_n \bar{r}_1^n, \; \mathbf{r}_2^{n+1} = \mathbf{r}_2^n - \rho_n \bar{\mathbf{r}}_2^n. \tag{3.25}$$

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* $\lambda = \lambda^{n+1}$, $P = P^{n+1}$, $\mathbf{U} = \mathbf{U}^{n+1}$. *If not, compute*

$$\gamma_n = \int_\Omega r_1^{n+1} g_1^{n+1} \, d\mathbf{x} + \int_\gamma \mathbf{r}_2^{n+1} \cdot \mathbf{g}_2^{n+1} \, d\gamma / \int_\Omega r_1^n g_1^n \, d\mathbf{x} + \int_\gamma \mathbf{r}_2^n \cdot \mathbf{g}_2^n \, d\gamma, \tag{3.26}$$

and set $\mathbf{w}^{n+1} = \mathbf{g}^n + \gamma_n \mathbf{w}^n$.
Do $n = n + 1$ *and go back to* (3.19).

3.3. Numerical experiments

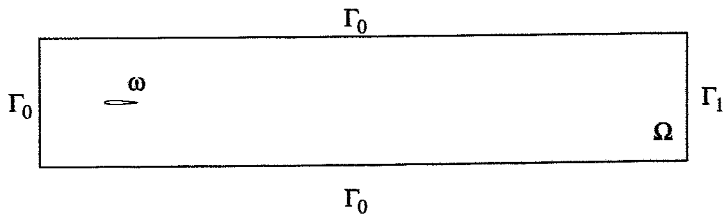

Figure 3.2.

We consider the test problem where ω is a NACA0012 airfoil with zero degree angle of attack centered at $(0, 0)$ and Ω is $(-0.625, 4.375) \times (-0.5, 0.5)$ (see Fig. 3.2). The boundary conditions are defined as follows:

$$\mathbf{u} = \begin{cases} (1 - e^{-ct}) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \; on \, \Gamma_0, \\ \mathbf{0} \; on \, \gamma, \end{cases} \tag{3.27}$$

where c is a positive constant and $\nu\frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{n}p = \mathbf{0}$ on Γ_1.

As a finite dimensional subspace of $\mathbf{V}$, we choose $\mathbf{V}_h = \{\mathbf{v}_h | \mathbf{v}_h \in H^1_{0h} \times H^1_{0h}\}$ where

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

where $\mathcal{T}_h$ is a triangulation of Ω (see, e.g, Fig. 3.3), P_1 being the space of the polynomials in x_1, x_2 of degree ≤ 1. A traditional way of approximating the pressure is to take it in the space

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

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

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

A particular choice for mesh points on γ is visualized on Fig. 3.3.

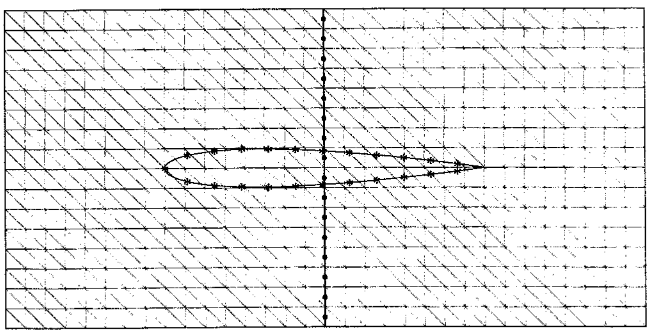

FIGURE 3.3. MESH POINTS MARKED BY "*" ON γ, PART OF MESH POINTS MARKED BY "•" ON THE INTERFACE BETWEEN Ω_1 AND Ω_2 AND PART OF THE TRIANGULATION OF Ω WITH $h = 1/64$.

Using the θ–scheme, we solve at each time step two quasi–Stokes subproblems by the one shot method (3.14)–(3.26) and one advection–diffusion subproblem in a least–squares formulation by a conjugate gradient algorithm. We divide Ω into two subdomains $\Omega_1 = (-0.625, 0.0) \times (-0.5, 0.5)$ and $\Omega_2 = (0.0, 4.375) \times (-0.5, 0.5)$ (see Fig. 3.4) and use domain decomposition methods introduced in Section 2 to solve the elliptic problems arising in the one shot method and in the conjugate gradient algorithm for the least–square problems. The mesh points on the interface between Ω_1 and Ω_2 are shown in Fig. 3.3.

Here we have chosen meshsizes $h_v = 1/64$ for velocity and $h_p = 1/32$ for pressure, time step $\triangle t = 0.01$ and $c = 20$ in (3.29). The number of iterations for the one shot method is from 40 to 60 except the first several time steps.

The number of iterations of the conjugate gradient method for the least–squares method is from 1 to 2. In Fig. 3.5, we observe a *Kàrmàn vortex shedding* (here, the Reynolds number is 1000).

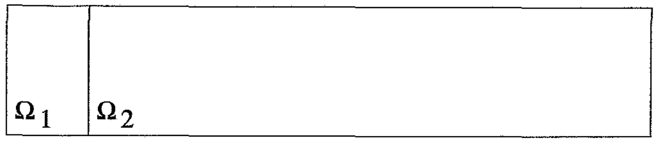

Figure 3.4

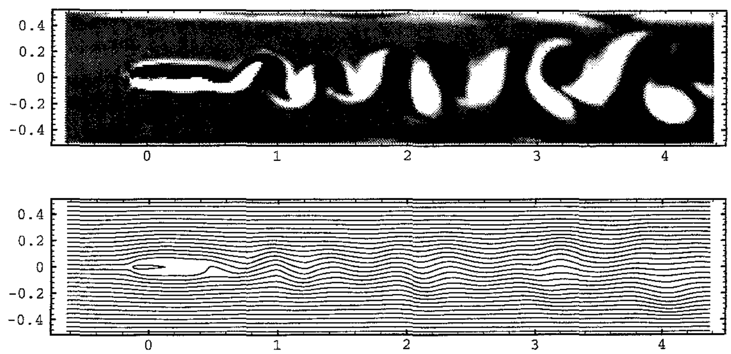

FIGURE 3.5. VORTICITY DENSITY (TOP) AND STREAM LINES (BOTTOM) FOR THE FLOW PASSING AROUND NACA0012 WITH ZERO DEGREE ANGLE OF ATTACK. FLOW DIRECTION IS FROM THE LEFT TO THE RIGHT, THE REYNOLDS NUMBER IS 1000, DIMENSIONLESS TIME IS 6.

4. Conclusion

Domain decomposition methods combined to fictitious domain methods seem to provide an efficient alternative to conventional solution methods for the solution of Poisson and Navier–Stokes equations on parallel MIMD computer.

This new methodology looks also promising for the simulation of time dependent solution of viscous flow problems around moving rigid bodies. However, further experiments are needed for very large problems to explore parallelization properties of one shot algorithm for 3-D flows, turbulent flow with one point (Baldwin–Lomax) or two point (k–ϵ) closure models and also local higher order approximations for higher values of the Reynolds number.

Acknowledgements

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

The support of the following corporations and institutions is acknowledged: AWARE, Dassault Aviation, INRIA, Texas Center for Advanced Molecular Computation, University of Houston, Université P. et M. Curie. We also benefited from the support of NSF (Grants DMS 8822522, DMS 9112847, INT 8612680 and DMS 9217374), the Texas Board of Higher Education (Grants 003652156ARP and 003652146ATP), DRET (Grant 89424) and DARPA (Contracts AFOSR F49620-89-C-0125 and AFOSR -90-0334).

REFERENCES

1. Young, D. P., Melvin, R. G., Bieterman, M. B., Johnson, F. T., Samanth, S. S., Bussoletti, J. E., 1991, *A locally refined finite rectangular grid finite element method. Application to Computational Physics*, J. Comp. Physics **92** (1991), pp. 1–66.
2. Bussoletti, J. E., Johnson, F. T., Samanth, S. S., Young, D. P., Burkhart, R. H.,, *EM-TRANAIR: Steps toward solution of general 3D Maxwell's equations*, in Computer Methods in Applied Sciences and Engineering, R. Glowinski ed., Nova Science, Commack, NY, 1992, pp. 49–72.
3. Buzbee, B. L., Dorr, F. W., George, J. A., Golub, G. H., 1971, *The direct solution of the discrete Poisson equation on irregular regions*, SIAM J. Num. Anal. **8** (1971), pp. 722–736.
4. Glowinski, R., Pan, T. W., Periaux, J., *A fictitious domain method for Dirichlet problem and applications*, Comp. Meth. Appl. Mech. Eng. **111** (1994), pp. 283–303.
5. Glowinski, R., Pan, T. W., Periaux, J., *A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations*, Comp. Meth. Appl. Mech. Eng. **112** (1994), pp. 133–148.
6. R. Glowinski, M. F. Wheeler, *Domain decomposition and mixed finite element methods for elliptic problems*, in Domain decomposition methods for partial differential equations, R. Glowinski, G. H. Golub, G. Meurant, J. Periaux eds., SIAM, Philadelphia, 1988, 144–172.
7. Glowinski, R., Pan, T. W., Periaux, J., *A one shot domain decomposition/fictitious domain method for the solution of elliptic equations*, in the proceeding of Parallel CFD'93, Paris, France, May, 1993 (to appear).
8. O. Buneman, *A compact non-iterative Poisson solver*, Report 294, Standford University Institute for Plasma Research (1969), Stanford, Cal., 1969.
9. G. H. Golub, C. F. Van Loan, *Matrix computations*, Johns Hopkin University Press, Baltimore, 1983.
10. R. A. Sweet, *a cyclic reduction algorithm for solving block tridiagonal systems of arbitrary dimension*, SIAM J. Num. Anal. **14** (1977), 706–720.
11. R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, 1984.
12. R. Glowinski, *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, 1985, 173–210.
13. M. O. Bristeau, R. Glowinski and J. Periaux, *Numerical methods for the Navier-Stokes equations*, Comp. Phys. Rep. **6** (1987), 73–187.
14. R. Glowinski, *Finite element methods for the numerical simulation of incompressible viscous flow. Introduction to the control of the Navier-Stokes equations*, Lectures in Applied Mathematics **28**, AMS, Providence, R. I., 1991, 219–301.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HOUSTON, HOUSTON, TEXAS 77204, UNIVERSITÉ P. ET M. CURIE, PARIS, AND CERFACS, TOULOUSE, FRANCE

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HOUSTON, HOUSTON, TEXAS 77204
E-mail address: : pan@math.uh.edu

DASSAULT AVIATION, 92214 SAINT-CLOUD, FRANCE
E-mail address: : periaux@menusin.inria.fr