INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **34**: 651–684

Fictitious domain approach for numerical modelling of Navier–Stokes equations

Khodor Khadra[a,1], Philippe Angot[b,*,2], Sacha Parneix[c] and Jean-Paul Caltagirone[c,3]

[a] *Mathématiques Appliquées de Bordeaux, UMR 5466 C.N.R.S., Université Bordeaux I, 351 Cours de la Libération, 33405 Talence Cedex, France*
[b] *IRPHE, Equipe Mathématiques Numériques, UMR 6594 C.N.R.S., Universités Aix-Marseille I & II, La Jetée, Technopôle de Château-Gombert, 38 rue F. Joliot Curie, 13451 Marseille Cedex 20, France*
[c] *MASTER-ENSCPB, Université Bordeaux I, Avenue Pey-Berland, BP 108, 33402 Talence Cedex, France*

SUMMARY

This study investigates a fictitious domain model for the numerical solution of various incompressible viscous flows. It is based on the so-called Navier–Stokes/Brinkman and energy equations with discontinuous coefficients all over an auxiliary embedding domain. The solid obstacles or walls are taken into account by a penalty technique. Some volumic control terms are directly introduced in the governing equations in order to prescribe immersed boundary conditions. The implicit numerical scheme, which uses an upwind finite volume method on staggered Cartesian grids, is of second-order accuracy in time and space. A multigrid local mesh refinement is also implemented, using the multi-level Zoom Flux Interface Correction (FIC) method, in order to increase the precision where it is needed in the domain. At each time step, some iterations of the augmented Lagrangian method combined with a preconditioned Krylov algorithm allow the divergence-free velocity and pressure fields be solved for. The tested cases concern external steady or unsteady flows around a circular cylinder, heated or not, and the channel flow behind a backward-facing step. The numerical results are shown in good agreement with other published numerical or experimental data. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: backward-facing step flow; fictitious domain method; multi-grid local mesh refinement; multi-level Zoom FIC method; Navier–Stokes/Brinkman model; wake of a bluff obstacle

1. INTRODUCTION

Fictitious domain methods (sometimes called 'domain embedding' methods) are presenting interesting capabilities for solving complicated boundary value problems motivated by

* Correspondence to: IRPHE Château-Gombert, UMR 6594 C.N.R.S., Universités Aix-Marseille I & II, La Jetée, Technopôle de Château-Gombert, 38 rue Frédéric Joliot Curie, 13451 Marseille Cedex 20, France.
[1] E-mail: khodor@math.u-bordeaux.fr
[2] E-mail: angot@marius.univ-mrs.fr
[3] E-mail: calta@lmaster.u-bordeaux.fr

Received December 1997
Revised April 1998

applications from science and engineering. The main reason for their popularity is that they allow the use of fairly structured meshes on a simple shape auxiliary domain (called a fictitious domain) containing the actual one. This allows the use of fast solvers or preconditioners for the auxiliary problems on complicated geometries. Fictitious domain methods for the solution of elliptic partial differential equations were proposed earlier, towards the end of the 1960s (see References [43,54] and the references herein). Since then, several authors have used the same approach with different techniques related to the so-called capacitance or influence matrix [18,26,55]. More recently, different fictitious domain methods have been introduced for the numerical modelling of flow problems [19,24,35,37,59,62].

In this article, we discuss a new fictitious domain method for solving the Navier–Stokes equations modelling unsteady incompressible viscous flows. This was originally proposed and experimented numerically in References [1,3,4] by considering a general fictitious domain model for viscous flows inside fluid–porous–solid systems of complicated geometries. The whole domain under study is embedded inside a fictitious auxiliary domain, geometrically simpler. It is governed by a single set of the so-called Navier–Stokes/Brinkman equations, where the Darcy drag [22,23] is directly included in the momentum equations. First proposed in References [16,17], to compute the natural convection flow inside a fluid–porous cavity, the Navier–Stokes/Brinkman model was generalized in Reference [3] to study complex fluid–porous–solid systems. Inside the fictitious domain, the particular medium is then taken into account by its characteristic permeability, i.e. a finite value for porous, infinite for pure fluid and zero for pure solid. In addition, the viscosity may also vary from its specific value in the pure fluid to an effective value in the porous medium, and up to infinity in the solid body. By this way, the fictitious domain model avoids the explicit expression of the transmission conditions at the interface of the different media. The physical understanding of this model, using results of homogenization as well as the calculation of drag and lift forces, are addressed in Reference [27]. The mathematical analysis of this model is made in Reference [11], which proves the convergence and derives some error estimates. The computations can be made on a simple structured mesh, which does not need to be body-fitted, but possibly locally refined with multi-level zoom methods [8,28,49–51]. A parallel implementation of the code was developed in Reference [9] for distributed memory machines. This model was used for the development of several computational codes and software allowing, in particular, interactive numerical experimentations [2,5,49]. Various numerical results for internal or external flows illustrate the capabilities and validate the efficiency of this fictitious domain approach, both for academic configurations, e.g. References [4,49], or for industrial applications [6,56,57,61]. Another interesting feature of the fictitious domain approach is that it allows the solution of optimal shape problems without remeshing the domain when the geometry is modified during the computations [36]. In particular, we will show the efficient use of rectangular Cartesian meshes for approximating curved geometries. Our goal is to develop a flexible and robust methodology, which takes account easily and fairly well of any geometry shape and boundary condition, especially when they are modified during the computation time [5].

In this paper, the main applications of these methods concern the study of external flows around profiles or obstacles that have porous or solid physical properties. The classical approach consists in solving the Navier–Stokes equations only in the fluid domain by taking into account the interface conditions on the boundary of the obstacles. Here, we assume that

the obstacles are imbedded inside a heterogeneous fictitious domain in which we globally solve the physical problem. The corresponding momentum equations are based on the porous media theory, introducing the permeability relative to each medium. This technique enables us to compute the velocity and the associated pressure inside the obstacles, and thus to very easily determine the drag and lift coefficients [27,49]. The numerical methodology is that developed in Reference [49]. The governing Navier–Stokes and energy equations are written in primitive variables. They are discretized by a linearly implicit time scheme and by a finite volume method (FVM) on staggered grids. We solve for the divergence-free velocity field of the Navier–Stokes equations by using an augmented Lagrangian technique, and we compute a velocity–pressure saddle-point by the Uzawa algorithm. The resolution of the discrete linear systems is performed by a preconditioned conjugate gradient technique.

The fluid flow and associated heat transfer become more and more difficult to simulate when the stresses applied to the physical system become more important, i.e. at high Reynolds or Rayleigh numbers. In this case, strong gradients, instabilities or turbulent spots may occur in some local regions of the domain. In order to efficiently compute these physical phenomena, a local mesh refinement is required in the areas of the domain where small-scale structures appear. We describe a method based on a local multi-grid architecture, which uses a conservative coarse-grid correction calculated from the flux balance at the interface between coarse and fine-grid control volumes.

The applications described here deal with laminar unsteady incompressible flows of a viscous fluid in two-dimensional domains containing bluff obstacles. The first study concerns the numerical modelling of external flows around a circular cylinder in an unbounded domain, and the second one is dealing with the flow in a channel behind a backward-facing step.

2. FICTITIOUS DOMAIN APPROACH

2.1. Embedding technique

Let Ω be an open-bounded domain in $\mathbb{R}^d$ ($d = 2, 3$), its boundary $\Gamma = \partial\Omega$ being regular enough. When studying external flows around obstacles that can be porous Ω_p or solid Ω_s, we define Ω by $\Omega = \Omega_f \cup \Omega_p \cup \Omega_s$, where Ω_f represents the fluid domain and its boundary Γ_f includes the interfaces $\partial\Omega_p$ and $\partial\Omega_s$ between the different media: fluid–porous and fluid–solid respectively (Figure 1). When compared with the usual numerical methodology, the introduction of the fictitious domain technique is based on the following observations:

- In general, the geometry of the different boundaries or interfaces is curved and the shape of the sub-domains Ω_f, Ω_p or Ω_s cannot be easily modified during the computations, if we use a body-fitted mesh structured or not.
- When we solve for the specific physical problem inside each sub-domain, we have to take the interface conditions between the different media explicitly into account, e.g. the coupling between solid and fluid heat transfer cannot be solved implicitly.
- A relatively important part of the computation time is spent carrying out the discretization: mesh generation; interface or boundary conditions; building the discrete problems for complex geometries, especially when they are likely to change during the computations: variable shape, size or position, engine dragging . . .

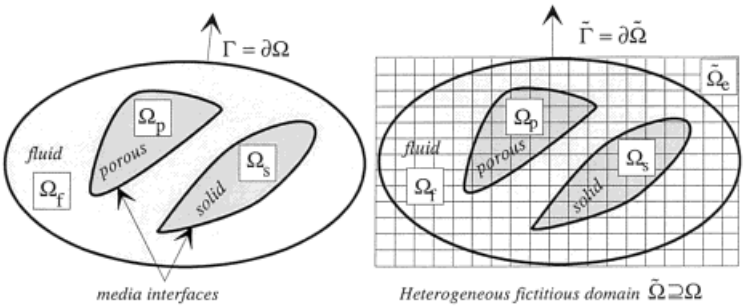

Figure 1. Embedding in a simple fictitious domain.

Because of these reasons, we develop a flexible and robust computational technique, allowing easy implementation and saving computation time, for solving relatively complex physical problems. We suppose that the domains Ω and thus Ω_f, Ω_p and Ω_s are embedded in the smallest rectangular (in two dimensions) or box (in three dimensions) domain, $\tilde{\Omega}$, that can contain them. The simple fictitious domain $\tilde{\Omega} \supseteq \Omega$ is then heterogeneous in the sense that it contains the obstacles with different shapes and physical characteristics. For example, as shown in Figure 1, the domain $\tilde{\Omega}$ is composed of different media: fluid, solid, porous. It is partitioned into the following way: $\tilde{\Omega} = \Omega_f \cup \Omega_p \cup \Omega_s \cup \tilde{\Omega}_e$, where $\tilde{\Omega}_e$ is the out-part of Ω in $\tilde{\Omega}$. Let $\tilde{\Gamma} = \partial\tilde{\Omega}$ be the boundary of the domain $\tilde{\Omega}$ (Figure 1).

The objective of the fictitious domain approach is to solve a single set of global governing equations over the whole domain $\tilde{\Omega}$ in order to determine the physical fields in each medium: velocity, pressure, temperature.... We will show how the interface conditions between the different media are implicitly taken into account in these equations. One crucial point is to solve correctly the problem in the domain $\tilde{\Omega}_e$: in particular, we have to take account of the original boundary conditions on Γ and to define the fictitious boundary conditions on $\tilde{\Gamma}$.

From the discretization point of view, we work with simple Cartesian meshes, disconnecting the mesh grid from the geometry of the sub-domains. We simply project a single rectangular Cartesian grid over the whole fictitious domain $\tilde{\Omega}$ independently of the shapes of the obstacles (Figure 1). Then, we account for the geometry of the different media by their physical characteristics assigned to the grid points: permeability, thermal conductivity, viscosity.... That allows easy implementation and fast computation, especially when we want to modify the physical characteristics of the flow or the geometry of the different media (shape, position, size) during the calculation time without remeshing the domain and rewriting the corresponding interface or boundary conditions.

2.2. Global model equations

We solve the unsteady, incompressible, dimensionless Navier–Stokes and energy equations, written in primitive variables formulation, velocity–pressure–temperature, and defined in the fluid domain Ω_f

Continuity equation

$$\nabla\cdot\mathbf{U} = 0 \quad \text{in } \Omega_f \tag{1}$$

Momentum (Navier–Stokes) equations

$$\frac{\partial \mathbf{U}}{\partial t} - \frac{1}{Re}\nabla\cdot[\mu(\nabla\mathbf{U} + \nabla\mathbf{U}^T)] + \mathbf{U}\cdot\nabla\mathbf{U} + \nabla p = \mathbf{f} \quad \text{in } \Omega_f \tag{2}$$

Energy equation

$$\frac{\partial T}{\partial t} - \frac{1}{RePr}\nabla\cdot[\lambda\nabla T] + \mathbf{U}\cdot\nabla T = q \quad \text{in } \Omega_f \tag{3}$$

where $\mathbf{U}(t, \mathbf{x})$ represents the velocity field [in the two-dimensional case, the velocity $\mathbf{U}$ has two scalar components $\mathbf{U}(t, \mathbf{x}) = (u(t, \mathbf{x}), v(t, \mathbf{x}))$], $p(t, \mathbf{x})$ is the pressure, $T(t, \mathbf{x})$ is the temperature, $\mu(t, \mathbf{x})$ and $\lambda(t, \mathbf{x})$ are the dimensionless dynamic viscosity and thermal conductivity respectively. The similitude parameters Re and Pr denote the Reynolds and Prandtl numbers respectively. The momentum equations here are independent of the energy equation as for forced convection problems.

Since the computation is now performed over the whole heterogeneous fictitious domain $\tilde{\Omega}$, we are going to write a single set of global model equations defined in $\tilde{\Omega}$. We consider the whole domain $\tilde{\Omega}$ as a Brinkman porous medium, characterized by its permeability $K(t, \mathbf{x})$, which can be variable in time and space. The porous medium Ω_p must be assigned to its specific permeability value K_p. We model the fluid and solid phases, Ω_f and Ω_s, as particular porous media, which are the limits of the Brinkman model. The permeability value related to each medium is defined by

$$K(t, \mathbf{x}) = \begin{cases} K_f \to +\infty & \text{if } \mathbf{x} \in \Omega_f \\ K_p & \text{if } \mathbf{x} \in \Omega_p \\ K_s \to 0^+ & \text{if } \mathbf{x} \in \Omega_s \end{cases}$$

Generally, the exterior domain $\tilde{\Omega}_e$ can be represented by a porous medium or, more often, by a solid one, i.e. $\tilde{K}_e = K_s$. The methodology consists in adding to the Navier–Stokes equations a term of volumic drag, called the Darcy drag, which represents the action of the fictitious porous medium over the flow. We then define one single global set of Navier–Stokes/Brinkman equations over the whole domain $\tilde{\Omega}$

$$\frac{\partial \mathbf{U}}{\partial t} - \frac{1}{Re}\nabla\cdot[\mu(\nabla\mathbf{U} + \nabla\mathbf{U}^T)] + \mathbf{U}\cdot\nabla\mathbf{U} + \frac{\mu\mathbf{U}}{Re\,Da\,K} + \nabla p = \mathbf{f} \quad \text{in } \tilde{\Omega} \tag{4}$$

where $Da = K_0/L_0^2$, the ratio of a reference permeability over a length scale squared, denotes the Darcy number. From this global model Equation (4), we can asymptotically find by penalty the local equations related to the fluid, porous and solid media [3]. Indeed:

- When $K = K_f$, the term $\mu \mathbf{U}/ReDaK$ tends towards zero and is negligible compared with the other terms of the equation. Then, in the fluid domain Ω_f, we nearly solve the classical Navier–Stokes Equation (2).
- When $K = K_s$, the solid is considered as a particular porous medium with a porosity $\phi \approx 1$, but with a permeability nearly equal to zero (Figure 2) [27]. The term $\mu \mathbf{U}/ReDaK$ appears as a penalization term, which imposes the velocity to tend towards zero in the solid. Thus, the terms $\mathbf{U} \cdot \nabla \mathbf{U}$ and $\partial \mathbf{U}/\partial t$ become negligible. We notice that if the value of the velocity is nearly equal to zero in Ω_s, the ratio $\mu \mathbf{U}/ReDaK$ is perfectly well defined and of order of magnitude of unity. Then, in the solid domain Ω_s, we solve the classical Brinkman equation, well known in porous media theory

$$-\frac{1}{Re} \nabla \cdot [\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T)] + \frac{\mu \mathbf{U}}{ReDaK} + \nabla p = \mathbf{f} \tag{5}$$

Let us note that the no-slip condition of the viscous fluid at the surface of the solid (i.e. on the interface $\partial\Omega_s$ between the two media Ω_f and Ω_s) is then asymptotically satisfied; indeed, if $K = K_s \to 0^+$ in Ω_s, which implies that $\mathbf{U}_s \approx 0$ on $\partial\Omega_s$, and because of continuity, $\mathbf{U}_f \approx 0$ on $\partial\Omega_s$.

A mathematical justification of this fictitious domain model based on the L^2-penalized Navier–Stokes/Brinkman equations is derived in Reference [11]. In particular, it is proved in Reference [11] for the linear steady case and in Reference [12] for the non-linear unsteady case, that the error estimate of the velocity field in the H^1-norm over the whole domain $\tilde{\Omega}$ is at least of the order of $\mathcal{O}(K_f^{-1} + K_s^{1/4})$, whereas the L^2-norm of the velocity inside the solid body is $\mathcal{O}(K_s^{3/4})$. Moreover, some numerical experiments are carried out in Reference [12] that compare our fictitious domain model with the standard Navier–Stokes model, including the application of the explicit no-slip boundary condition on a square cylinder. This shows the influence of the value of the penalty parameter K_s on the velocity inside the solid body and on the wake behind the bluff body when K_s is not chosen small enough.

At this stage and in the following, we shall consider that the viscosity of the fluid is constant, i.e. $\mu = 1$. Since we use only solid obstacles here, we do not need the viscosity of the equivalent porous medium. Thus, thanks to the incompressibility constraint, we can simplify Equation (4) as

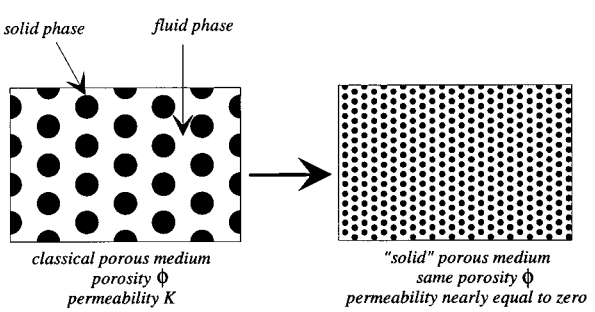

Figure 2. Representation of the solid as a porous medium.

$$\frac{\partial \mathbf{U}}{\partial t} - \frac{1}{Re}\Delta \mathbf{U} + \mathbf{U}\cdot\nabla\mathbf{U} + \frac{\mathbf{U}}{Re\,Da\,K} + \nabla p = \mathbf{f} \quad \text{in } \tilde{\Omega} \tag{6}$$

The global energy equation defined in the domain $\tilde{\Omega}$ is finally written as

$$\frac{\partial T}{\partial t} - \frac{1}{Re\,Pr}\nabla\cdot[\lambda\nabla T] + \mathrm{U}\cdot\nabla T = q \quad \text{in } \tilde{\Omega} \tag{7}$$

where $\lambda(t, \mathbf{x})$ is the thermal conductivity of the fictitious medium:

$$\lambda(t, \mathbf{x}) = \begin{cases} \lambda_{\mathrm{f}} & \text{if } \mathbf{x}\in\Omega_{\mathrm{f}} \\ \lambda_{\mathrm{p}} & \text{if } \mathbf{x}\in\Omega_{\mathrm{p}} \\ \lambda_{\mathrm{s}} & \text{if } \mathbf{x}\in\Omega_{\mathrm{s}} \end{cases}$$

In the exterior domain $\tilde{\Omega}_{\mathrm{e}}$, depending on the thermal boundary conditions (Dirichlet, Neumann or Fourier) required on Γ, we generally assign $\tilde{\lambda}_{\mathrm{e}} = \lambda_{\mathrm{s}}$ or $\tilde{\lambda}_{\mathrm{e}} \to \infty$ or also $\tilde{\lambda}_{\mathrm{e}} \to 0^{+}$ associated with adequate boundary conditions on $\tilde{\Gamma}$, see [1,3].

Therefore, the global model equations with discontinuous coefficients in $\tilde{\Omega}$ that are considered in this study are the following:

Continuity equation

$$\nabla\cdot\mathbf{U} = 0 \quad \text{in } \tilde{\Omega} \tag{8}$$

Momentum (Navier-Stokes/Brinkman) equations

$$\frac{\partial \mathbf{U}}{\partial t} - \frac{1}{Re}\Delta \mathbf{U} + \mathbf{U}\cdot\nabla\mathbf{U} + \frac{\mathbf{U}}{Re\,Da\,K} + \nabla p = \mathbf{f} \quad \text{in } \tilde{\Omega} \tag{9}$$

Energy equation

$$\frac{\partial T}{\partial t} + \frac{1}{Re\,Pr}\nabla\cdot[\lambda\nabla T] + \mathbf{U}\cdot\nabla T = q \quad \text{in } \tilde{\Omega} \tag{10}$$

Let us notice also that such a methodology allows the prolongation of both the velocity and pressure fields inside the solid medium [27,11]. The velocity profile normally to the boundary of a bluff cylinder (see section 5.1) represented in Figure 14 shows how the no-slip condition is satisfied.

2.3. Immersed boundary conditions

All the boundary conditions are written into the general following form, where the variable Φ ($\Phi = \mathbf{U}$ or T) is linked with the flux:

$$-\frac{\partial \Phi}{\partial \mathbf{n}} = \alpha_\Phi(\Phi - \Phi_\infty) + g_\Phi \quad \text{on} \quad \Sigma \tag{11}$$

These boundary conditions are defined on Σ which can be $\tilde{\Gamma}$, Γ or, if needed, the interfaces $\partial\Omega_p$, $\partial\Omega_s$, as well; $\mathbf{n}$ is the outward normal unit vector on Σ. The parameter α_Φ represents a surfacic transfer coefficient, and g_Φ a surfacic flux. This form enables us to prescribe classical Fourier, Neumann, or Dirichlet (by penalization $\alpha_\Phi \to +\infty$) boundary conditions on Σ.

For a better flexibility of the methodology, allowing to modify easily the boundary conditions at any part of Σ during the computation time, we implicitly introduce the above boundary conditions (11) into the Navier–Stokes/Brinkman and energy equations. This is made through the introduction of volumic control terms directly in the governing equations, as proposed in [1,3]; then, we obtain the following fictitious domain equations:

Momentum (Navier–Stokes/Brinkman) equations

$$\frac{\partial \mathbf{U}}{\partial t} - \frac{1}{Re}\Delta \mathbf{U} + \mathbf{U}\cdot\nabla\mathbf{U} + \frac{U}{Re\,Da\,K} + \nabla p + \boxed{\beta_\mathbf{U}(\mathbf{U} - \mathbf{U}_\infty) + \varphi_\mathbf{U}} = \mathbf{f} \quad \text{in} \quad \tilde{\Omega} \tag{12}$$

Energy equation

$$\frac{\partial T}{\partial t} - \frac{1}{Re\,Pr}\nabla\cdot[\lambda\nabla T] + \mathbf{U}\cdot\nabla T + \boxed{\beta_T(T - T_\infty) + \varphi_T} = q \quad \text{in} \quad \tilde{\Omega} \tag{13}$$

where the volumic terms are defined by $\beta_\Phi = \alpha_\Phi/\epsilon$ and $\varphi_\Phi = g_\Phi/\epsilon$ on Σ and 0 elsewhere in $\tilde{\Omega}$. The parameter ϵ represents a characteristic length which was shown in [1] to be $\mathcal{O}(h)$ if h is the local mesh step at the current grid point on Σ. At this stage, by default for the variables $\mathbf{U}$ and T, we take into account the homogeneous Neumann condition:

$$-\frac{\partial \Phi}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{14}$$

Then by using the control terms β_Φ, Φ_∞ and φ_Φ, which can be possibly re-calculated at each time step, we notice that if α_Φ and g_Φ are given functions, variable in time and space, we are able to apply any kind of usual or immersed boundary conditions at any part of Σ, even if Σ is moving:

- $\alpha_\mathbf{U}$, $\alpha_T = 0$: Neumann condition $\left(-\dfrac{\partial \Phi}{\partial \mathbf{n}} = g_\Phi\right)$,
- $\alpha_\mathbf{U}$, $\alpha_T \to +\infty$: Dirichlet condition $(\Phi = \Phi_\infty)$,
- $0 < \alpha_\mathbf{U}$, $\alpha_T < +\infty$: Fourier or Robin condition.

In particular, when we solve the global problem in $\tilde{\Omega}$, we can impose, if necessary, any interface boundary conditions on the surfaces of the obstacles according to their permeability. Let us notice that the problem we solve in $\tilde{\Omega}_e$, depends on the original boundary condition required on Γ.

If necessary, a Dirichlet boundary condition for the pressure could be applied in the same way by penalization on Σ, with the following modified continuity equation:

$$\nabla \cdot \mathbf{U} + \boxed{\beta_p(p - p_\infty)} = 0 \quad in \quad \tilde{\Omega} \tag{15}$$

Here, we do not need pressure boundary conditions, as the pressure is computed explicitly by an Uzawa algorithm, as we will see further.

Let us mention that another method to apply immersed boundary conditions is described in [1] and theoretically analysed in [13,14].

2.4. Computation of drag and lift coefficients

The flow of a viscous fluid around a bluff profile is, in general, calculated by solving the Navier–Stokes equations only in the fluid domain associated with the no-slip boundary condition, corresponding to the adherence of the fluid at the surface of the obstacle, without necessarily requiring the physical properties of the profile. The knowledge of the velocity and pressure fields around the obstacle allows us to calculate the resulting forces applied on its surface by the viscous fluid. However, these computations are somehow difficult, as they require accurate interpolations and surfacic integrations of quantities based on the pressure and velocity gradients along the profile. It becomes harder also if we use staggered grids to avoid checkerboard pressure modes. Indeed, we have to compute: $\mathbf{F} = \int_{\partial\Omega s} \mathbf{T}_{fs}\, d\gamma$, where $\mathbf{T}_{fs}$, is the total stress vector defined by:

$$\mathbf{T}_{fs} = \sigma \cdot \mathbf{n} = \left[-pI + \frac{\mu}{Re}(\nabla \mathbf{U} + \nabla \mathbf{U}^T) \right] \cdot \mathbf{n}$$

and $\mathbf{n}$ is the outside normal unit vector on the surface of the obstacle.

When using the fictitious domain approach, the determination of both the velocity and pressure fields inside the solid medium through the Brinkman model allows to compute very easily the drag and lift forces acting on the solid [27]. After transforming the surfacic integral into a volumic integral by using the Green–Ostrogradsky formula, the Brinkman Equation (5) can be written for $\mu = 1$ and $\mathbf{f}|_{\Omega_s} = 0$, as:

$$\mathbf{F} = \int_{\partial\Omega s} \mathbf{T}_{fs}\, d\gamma = -\int_{\Omega s} \nabla p\, d\omega + \int_{\Omega s} \frac{1}{Re} \nabla U\, d\omega = \int_{\Omega s} \frac{\mathbf{U}}{Re\, Da\, K} d\omega \tag{16}$$

These volumic integrals represent respectively the global force applied on the obstacle, and its pressure and viscous contributions. They can be easily computed using a classical finite volume discretization. The components of these forces respectively parallel and orthogonal to the direction of the mean flow, correspond to the so-called drag and lift forces. Proposed in [27],

this methodology to compute the drag and lift coefficients was numerically experimented in [49] and mathematically justified in [11]. Moreover, it is proved in [11] that the error estimate for the global resulting force applied by the fluid onto the solid body is at least of the same order as the error of the velocity field, i.e. $\mathcal{O}(K_s^{1/4})$.

2.5. Application to external flows around a profile

The main applications that we study concern unsteady external (or internal) flows around profiles or obstacles that can be variable during the time (change of geometry, size, position, permeability, . . .). Theoretically, the problem is formulated in an unbounded fluid domain Ω_f. However, the numerical resolution will impose a fictitious boundary set far enough from the obstacles on which we shall define specific boundary conditions in such a way that the different parameters of the flow are not affected by the existence of this artificial boundary. In particular, we have to define adequate conditions at the downstream flow that allow the eddies to go out from the computational domain. Although other kinds of outflow boundary conditions can be used [34,38,25], we have experimented heuristically in [49,56] that a Fourier-type condition for the velocity yields satisfying results for the computation of the wake behind bluff bodies, provided that the domain is large enough, i.e. with an aspect ratio equal to four, in our case. Next, we perform a fictitious domain technique by solving the global Navier–Stokes/Brinkman equations over the whole auxiliary domain $\tilde{\Omega}$.

When the constraints applied to the physical system become important, i.e. for high Reynolds numbers, strong gradients within thin boundary layers appear in some local regions of the domain, and the flow may become turbulent. In order to take account of all the scale variations of the problem, and to capture the very small structures of the flow, a local mesh refinement is needed. We use here local multi-level mesh refinement, e.g. [50,52].

For example in Figure 3, $\tilde{\Omega}$ is partitioned as $\tilde{\Omega} = \Omega_f \cup \Omega_o$, where Ω_f and Ω_o represent respectively the fluid and the obstacle domain that can be porous or solid, depending on its permeability value. It is shown in [4] that a ratio $K_s/K_p = 10^{-5}$ is generally small enough. For the computations presented in the following, we used fluid and solid permeability values, such

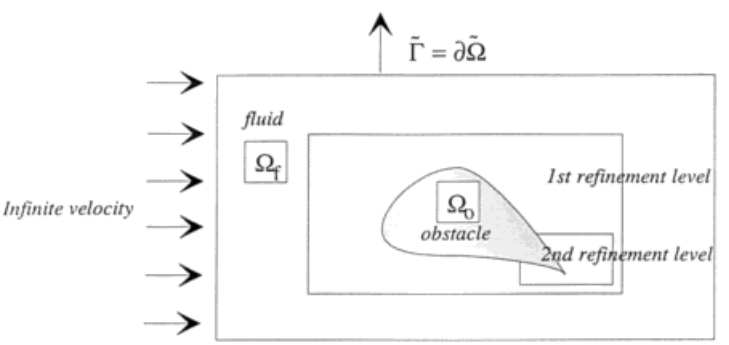

Figure 3. External flow around an obstacle.

that $Da K_f = 10^6$ and $Da K_s = 10^{-14}$. We will also show how mesh refinement zones can be efficient for solving local phenomena, without overly increasing the number of discretization points.

3. NUMERICAL MODEL

3.1. Time discretization

The equations (8, 12, 13, 14) are first discretized in time by using a finite difference implicit scheme. We denote by Δt the time discretization step, and the index n characterizes the time iteration at $t_n = n\Delta t$. We use a Gear scheme, i.e. a second order backward scheme in time, and we obtain the following discrete equations for the computation of $(\mathbf{U}^n, p^n, T^n)$ at the time iteration $n \geq 1$:

$$\nabla \cdot \mathbf{U}^n = 0 \quad \text{in} \quad \tilde{\Omega} \tag{17}$$

$$\frac{3\mathbf{U}^n}{2\nabla t} - \frac{1}{Re}\Delta \mathbf{U}^n + \mathbf{U}^n \cdot \nabla \mathbf{U}^n + \frac{\mathbf{U}^n}{Re\, Da\, K^n} + \nabla p^n + \beta_{\mathbf{U}}^n \mathbf{U}^n$$

$$= \mathbf{f}^n + \beta_{\mathbf{U}}^n \mathbf{U}_\infty^n - \varphi_{\mathbf{U}}^n + \frac{4\mathbf{U}^{n-1} - \mathbf{U}^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{18}$$

$$\frac{\partial \mathbf{U}^n}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{19}$$

$$\frac{3T^n}{2\Delta t} - \frac{1}{Re\, Pr}\nabla \cdot (\lambda^n \nabla T^n) + \mathbf{U}^n \cdot \nabla T^n + \beta_T^n T^n = q^n + \beta_T^n T_\infty^n - \varphi_T^n + \frac{4T^{n-1} - T^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{20}$$

$$\frac{\partial T^n}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{21}$$

The set of equations (18, 20) is non-linear as a result of the advection terms. A linearization process is then applied using Picard fixed point iterations. Thus, the determination of each iterate $(\mathbf{U}^n, p^n, T^n)$ requires the computation of several linearization iterates $(\mathbf{U}^l, p^l, T^l)$ solution of the following equations:

$$\nabla \cdot \mathbf{U}^l = 0 \quad \text{in} \quad \tilde{\Omega} \tag{22}$$

$$\frac{3\mathbf{U}^l}{2\Delta t} - \frac{1}{Re}\Delta \mathbf{U}^l + \mathbf{U}^{l-1} \cdot \nabla \mathbf{U}^l + \frac{\mathbf{U}^l}{Re\, Da\, K^n} + \nabla_{\mathrm{p}}^l + \beta_{\mathbf{U}}^n \mathbf{U}^l$$

$$= \mathbf{f}^n + \beta_{\mathbf{U}}^n \mathbf{U}_\infty^n - \varphi_{\mathbf{U}}^n + \frac{4T^{n-1} - T^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{23}$$

$$\frac{\partial \mathbf{U}^l}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{24}$$

$$\frac{3T^l}{2\Delta t} - \frac{1}{Re\,Pr}\nabla\cdot(\lambda^n \nabla T^l) + \mathbf{U}^l\cdot\nabla T^l + \beta_T^n T^l = q^n + \beta_T^n T_\infty^n - \varphi_T^n + \frac{4T^{n-1} - T^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{25}$$

$$\frac{\partial T^l}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{26}$$

The momentum equations do not depend on the temperature if we neglect the buoyancy forces, as it is usually assumed for forced convection problems. We first solve equations (22, 23, 24) for the computation of the velocity and pressure $(\mathbf{U}^l, p^l)$. The resulting velocity $\mathbf{U}^l$ is replaced into the Equation (25), and we solve equations (25, 26) for the computation of the temperature T^l.

In order to deal with the constraint of divergence-free velocity, we use the augmented Lagrangian method for solving the system of equations (22, 23, 24), see [33,34]. Some efficiently preconditioned fully-coupled Navier–Stokes solvers could be used as well, e.g. [29]. We formulate this system as a velocity-pressure minimization–maximization problem for the computation of a saddle-point $(\mathbf{U}^l, p^l)$ associated to the augmented Lagrangian of the problem. The set of equations (22, 23, 24) is then equivalent to the following one [49]:

$$\frac{3\mathbf{U}^l}{2\Delta t} - \frac{1}{Re}\Delta\mathbf{U}^l + \mathbf{U}^{l-1}\cdot\nabla\mathbf{U}^l + \frac{\mathbf{U}^l}{Re\,Da\,K^n} - \boxed{r\nabla(\nabla\cdot\mathbf{U}^l)} + \nabla_{\mathrm{p}}^l + \beta_{\mathbf{U}}^n\mathbf{U}^l$$
$$= \mathbf{f}^n + \beta_{\mathbf{U}}^n\mathbf{U}_\infty^n - \varphi_{\mathbf{U}}^n + \frac{4\mathbf{U}^{n-1} - \mathbf{U}^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{27}$$

$$\frac{\partial \mathbf{U}^l}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{28}$$

We notice that the incompressibility constraint has been introduced implicitly into the momentum equations, and the pressure components are considered as Lagrange multipliers. Then, we compute each saddle-point $(\mathbf{U}^l, p^l)$ by using an iterative Uzawa algorithm to solve the above augmented Lagrangian problem, which generates the following k-iterations $(\mathbf{U}^k, p^k)$:

$$\frac{3\mathbf{U}^k}{2\Delta t} - \frac{1}{Re}\Delta\mathbf{U}^k + \mathbf{U}^{l-1}\cdot\nabla\mathbf{U}^k + \frac{\mathbf{U}^k}{Re\,Da\,K^n} - r\nabla(\nabla\cdot\mathbf{U}^k) + \beta_{\mathbf{U}}^n\mathbf{U}^k$$
$$= -\nabla p^{k-1} + \mathbf{f}^n + \beta_{\mathbf{U}}^n\mathbf{U}_\infty^n - \varphi_{\mathbf{U}}^n + \frac{4\mathbf{U}^{n-1} - \mathbf{U}^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{29}$$

$$\frac{\partial \mathbf{U}^k}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \tilde{\Gamma} = \partial\tilde{\Omega} \tag{30}$$

$$p^k = p^{k-1} - \rho\nabla\cdot\mathbf{U}^k \quad \text{in} \quad \tilde{\Omega} \tag{31}$$

We do not need boundary conditions for the pressure: this latter is calculated explicitly in the whole domain $\tilde{\Omega}$ with Equation (31). We stop the Uzawa iterations by setting a criterion based on the divergence of the velocity field, i.e. $\|\nabla \cdot \mathbf{U}\| \leq \varepsilon$, for ε small enough. The parameters $r > 0$ and $\rho > 0$ are estimated according to the convergence conditions derived in the literature, e.g. [33,34]. For example, it is proved that for the solution of the incompressible Navier–Stokes equations with homogeneous Dirichlet boundary conditions, the Uzawa algorithm for the augmented Lagrangian problem converges for $0 < \rho < 2(r + (1/Re))$. Note that in the case $r = 0$, the Uzawa algorithm converges very slowly, or may not converge at all. Practically, when the velocity and pressure variables are of order of magnitude equal to one, the values of r and ρ are taken of order of one too, and we often choose $\rho = r$.

We summarize all the successive inner–outer iterations in the following algorithm:

* *Initialization of the time iterations:*
$(\mathbf{U}^{n=0}, p^{n=0}, T^{n=0})$ is given as an initial time solution
* *Computation of the time iterates* $(\mathbf{U}^n, p^n, T^n)$:
for $n = 1$ to N do
  * *Initialization of the linearization iterations:*
  $(\mathbf{U}^{l=0}, p^{l=0}, T^{l=0}) = (\mathbf{U}^{n-1}, p^{n-1}, T^{n-1})$
  * *Computation of the linearization iterates* $(\mathbf{U}^l, p^l, T^l)$:
  for $l = 1$ to L do
    * *Initialization of the Uzawa iterations:*
    $(\mathbf{U}^{k=0}, p^{k=0}) = (\mathbf{U}^{l-1}, p^{l-1})$
    * *Computation of the Uzawa iterates* $(\mathbf{U}^k, p^k)$:
    for $k = 1$ to K do
    {*Solve the linear set of equations (29, 30, 31)*
    end for: $(\mathbf{U}^l, p^l) = (\mathbf{U}^K, p^K)$
    * *Compute* T^l *solution of the linear set of equations (25, 26)*
  end for: $(\mathbf{U}^n, p^n, T^n) = (\mathbf{U}^L, p^L, T^L)$
end for

Practically, we combine very often the linearization iterations with the augmented Lagrangian ones by taking $K = 1$.

3.2. Finite volume discretization

The equations (29, 30, 31), and (25, 26) are discretized in space on a rectangular Cartesian staggered grid using a finite volume method for a mesh of MAC type, e.g. [60]. The domain $\tilde{\Omega}$ is partitioned into cell-vertex control volumes associated to the scalar grid nodes. The scalar quantities, like pressure and temperature, are stored at these grid points. In contrast, the two velocity components are respectively stored at nodes that are staggered with respect to the scalar grid nodes. A schematic drawing of this two-dimensional grid arrangement is shown on Figures 4 and 5.

The geometry of a porous or solid obstacle Ω_o ($\Omega_o = \Omega_p$ or Ω_s) is approximated by rectangular Cartesian finite volumes, according to the permeability field K ($K = K_f$ in Ω_f, and $K = K_o$ in Ω_o) in the following way: if a scalar node p_{ij} or T_{ij} belongs to Ω_f (respectively to Ω_o),

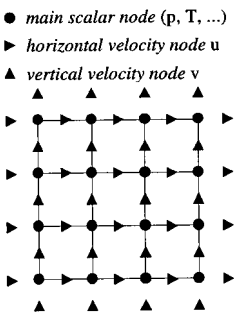

Figure 4. The staggered grid.

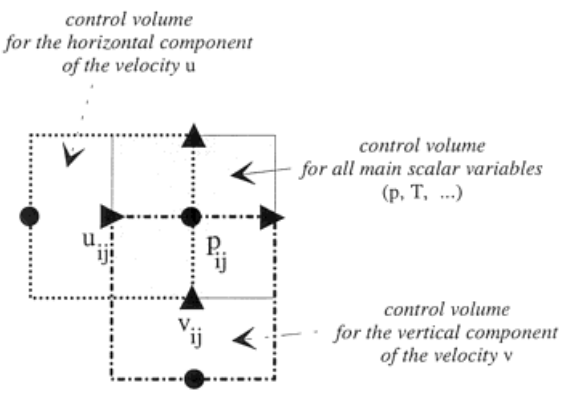

Figure 5. Control volume associated to each node.

then the whole control volume associated to this node has the permeability K_f (respectively K_o; cf. Figure 6). Thus, the approximated interfaces between the two media Ω_f and Ω_o coincide with control volume interfaces. Let us note that if a scalar node is located on the interface $\partial\Omega_o$ it is considered as belonging to the obstacle. In this way, we have:

$$K_p^{ij} = K_u^{ij} = K_u^{i+1,j} = K_v^{ij} = K_v^{i,j} = \begin{cases} K_f & \text{if} \quad p_{ij} \in \Omega_f \\ K_o & \text{if} \quad p_{ij} \in \bar{\Omega}_o \end{cases}$$

As we use a finite volume technique, before discretizing equations (29) and (25), we re-write the convective terms in a conservative form:

$$(\mathbf{U}^{l-1} \cdot \nabla)\mathbf{U}^k = \nabla \cdot (\mathbf{U}^{l-1} \otimes \mathbf{U}^k) - (\nabla \cdot \mathbf{U}^{l-1})\mathbf{U}^k \tag{32}$$

$$\mathbf{U}^l \cdot \nabla T^l = \nabla \cdot (T^l \mathbf{U}^l) - (\nabla \cdot \mathbf{U}^l) T^l \tag{33}$$

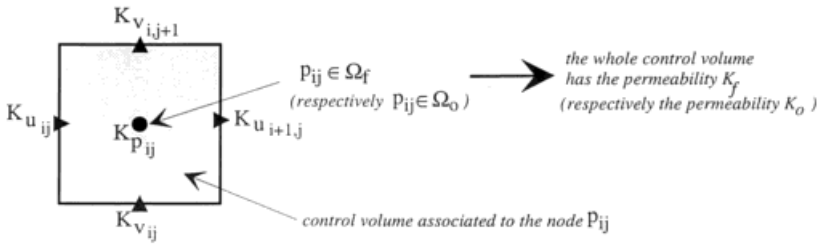

Figure 6. Determination of the permeability on each finite volume.

Then, the conservative equations are:

$$\frac{3\mathbf{U}^k}{2\Delta t} - \frac{1}{Re}\Delta\mathbf{U}^k + \nabla\cdot(\mathbf{U}^{l-1}\otimes\mathbf{U}^k) - (\nabla\cdot\mathbf{U}^{l-1})\mathbf{U}^k + \frac{\mathbf{U}^k}{Re\,Da\,K^n} - r\nabla(\nabla\cdot\mathbf{U}^k) + \beta_{\mathbf{U}}^n\mathbf{U}^k$$
$$= -\nabla p^{k-1} + \mathbf{f}^n + \beta_{\mathbf{U}}^n\mathbf{U}_\infty^n - \varphi_{\mathbf{U}}^n + \frac{4\mathbf{U}^{n-1} - \mathbf{U}^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{34}$$

$$\frac{3T^l}{2\Delta t} - \frac{1}{Re\,Pr}\nabla\cdot(\lambda^n\nabla T^l) + \nabla\cdot(T^l\mathbf{U}^l) - (\nabla\cdot\mathbf{U}^l)T^l + \beta_T^n T^l$$
$$= q^n + \beta_T^n T_\infty^n - \varphi_T^n + \frac{4T^{n-1} - T^{n-2}}{2\Delta t} \quad \text{in} \quad \tilde{\Omega} \tag{35}$$

The discrete equations are obtained by an approximate integration of equations (34, 30), (31), and (35, 26) over the appropriate finite volumes associated to the scalar variables u, v, p, T, following the finite volume procedure outlined by Patankar [58]. We choose piecewise constant approximation functions on each control volume for the computation of the integrals of fluxes or volumic terms. For the approximation of the first order derivatives associated to the diffusion fluxes, we use a second-order centred scheme at the interface of the control volumes. We then obtain the classical five points scheme in 2D. For the transported quantities (convective terms), the accuracy of the interpolation scheme and its stability are primarily concerned at high Reynolds numbers. The quadratic upwind scheme QUICK, proposed by Leonard [53], has both good stability and accuracy properties. The scheme is formally third-order accurate, and significantly weighted towards upstream values, which accounts for its good stabilizing properties. In the QUICK scheme, the evaluation of a variable Ψ ($\Psi = u$, $\Psi = v$, or $\Psi = T$) at any control–volume interface is based on a quadratic interpolation involving two nodes located upstream from the control volume, and one located downstream. Obviously, this expression is set according to the sign of the velocity at the interface of the control volume. For example, as shown in Figure 7, we have for the horizontal direction:

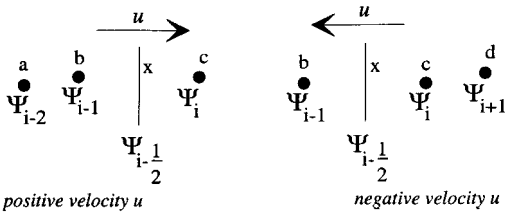

Figure 7. The QUICK scheme.

$$\begin{cases} \text{if} \quad u_{i-1/2,j} > 0 \quad \text{then} \quad \Psi_{i-1/2,j} = \left[\dfrac{(x-b)(x-c)}{(a-b)(a-c)}\right]\Psi_{i-2} + \left[\dfrac{(x-a)(x-c)}{(b-a)(b-c)}\right]\Psi_{i-1} + \left[\dfrac{(x-a)(x-b)}{(c-a)(c-b)}\right]\Psi_i \\ \text{if} \quad u_{i-1/2,j} < 0 \quad \text{then} \quad \Psi_{i-1/2,j} = \left[\dfrac{(x-b)(x-c)}{(d-b)(d-c)}\right]\Psi_{i+1} + \left[\dfrac{(x-b)(x-d)}{(c-b)(c-d)}\right]\Psi_i + \left[\dfrac{(x-c)(x-d)}{(b-c)(b-d)}\right]\Psi_{i-1} \end{cases}$$

where a, b, c, d, and x are respectively the abscisses of nodes number $i-2$, $i-1$, $i-\frac{1}{2}$, i and $i+1$.

For uniform meshes, we obtain:

$$\begin{cases} \text{if} \quad u_{i-1/2,j} > 0 \quad \text{then} \quad \Psi_{i-1/2,j} = -\frac{1}{8}\Psi_{i-2,j} + \frac{6}{8}\Psi_{i-1,j} + \frac{3}{8}\Psi_{i,j} \\ \text{if} \quad u_{i-1/2,j} < 0 \quad \text{then} \quad \Psi_{i-1/2,j} = -\frac{1}{8}\Psi_{i+1,j} + \frac{6}{8}\Psi_{i,j} + \frac{3}{8}\Psi_{i-1,j} \end{cases}$$

3.3. Linear solver

The discrete linear systems associated to the discretization of equations (34, 30) and (35, 26) are non-symmetric and of indefinite type. Let us notice that the introduction of the incompressibility constraint into the momentum equations with the term $-r\nabla(\nabla\cdot\mathbf{U})$ leads to a coupling between the different components of the velocity. By using the discretization described above, the associated matrices (of which the coefficients depending on the velocity $\mathbf{U}$ vary at each iteration k) have the sparse structure of Figure 8.

The resolution is carried out by the Krylov method BI-CGSTAB [67], with diagonal preconditioning, at least, to cope with the discontinuous coefficients of the operators. We also developed a modified incomplete Gauss factorization MILU(0), which proved to be an efficient preconditioner for this kind of problem [49]. However, to obtain better vector accelerations, we generally use only the Jacobi preconditioner. In that case, we obtain typical

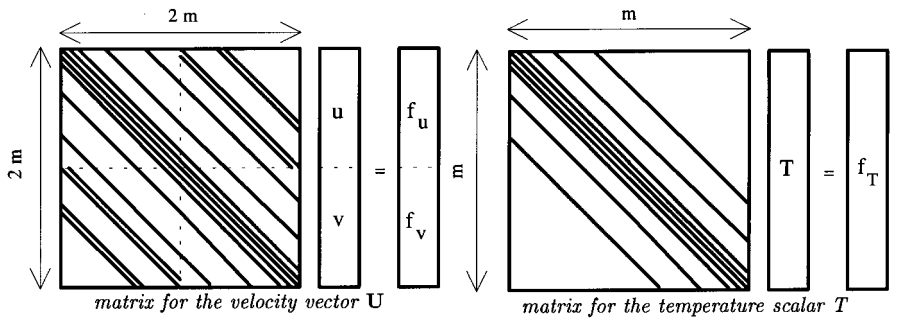

Figure 8. General structure of the discrete operators.

execution speeds of 233 *M flops* on Cray YMP-EL and 610 *M flops* on Cray C98. A parallel version of the code was also developed, based on partitioned iterative solvers on Single Program and Multiple Data (SPMD) distributed memory machines, like Cray T3D/T3E or IBM SP2 [9].

When the parameter r increases, the Uzawa algorithm converges faster, but the conditioning of the matrix becomes worse. Thus, the conjugate gradient algorithm converges slowly, or it becomes more unstable and may not converge at all. There is a compromise to set between the number of outer Uzawa iterations and inner BI-CGSTAB iterations. In general, we prefer performing more outer iterations and a few inner iterations.

4. ADAPTIVE MULTIGRID LOCAL MESH REFINEMENT

Large-scale physical models exhibit significant variations of scale in the solution of the model equations. Scale variations can come from the strong constraints applied to the physical system, leading to strong gradients or small-scale structures in some local regions of the domain. In order to model well locally the physical phenomena, a local mesh refinement is necessary in some areas of the domain. We choose to avoid working with a single non-uniform and unstructured mesh [50,51]. The adaptive local mesh refinement procedure that we use here is based on a hierarchical multigrid architecture for a set of embedded subdomains. We consider a basic global grid, G_0, that discretizes the whole computational domain $\tilde{\Omega}$, from which we generate a set of local nested zoom grids G_ℓ, $1 \leq \ell \leq \ell^*$, with decreasing mesh steps (see Figure 9). In that way, all the grids are structured and uniform. As the size of the subdomains decreases too, on each grid we have a moderate number of nodes. It yields a set of linear systems of moderate dimension, which can be solved very fast, with good convergence rates, as they exhibit only moderate condition numbers. The connection between the grid solutions is carried out by using projection operators [7,48,8,49,50].

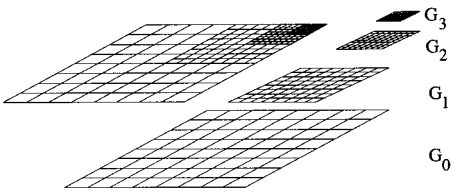

Figure 9. Local multigrid architecture with $\ell^* = 3$.

At each time step, we perform local multigrid iterations as follows:

- *Step 1—recursive interpolation*: we start from the resolution of equations (34, 30), (31) and (35, 26) on the coarsest grid G_0. Next, we define, in an adaptive way, a first local fine-grid, on which we perform the same computations after determining the interface boundary conditions. We choose a Dirichlet interface condition, calculated by interpolation of the coarse-grid solution on the fine-grid interface nodes. We repeat recursively this procedure, which consists in defining adaptively the mesh refinement grid regions, computing interface boundary conditions between two successive subdomains, and computing the fine-grid solutions. We stop the mesh refinement process until we reach the finest grid level G_{ℓ^*}, on which the computed solution is likely to be accurate enough.
- *Step 2—recursive coarse-grid correction*: we transfer all the local fine-grid informations to the coarser grids, in order to make a correction of the previous coarse-grid solutions. For that, we restrict the fine-grid solution on the coarse-grid nodes, which are included in the refinement region, by using the canonical restriction operator. In this case, the correction of the coarse-grid solution is only effective locally, in the region where the local mesh refinement has been applied. To better improve the coarse-grid solution, the main idea is to make a global correction on the coarse-grid, i.e. a propagation of the correction beyond the local mesh refinement region. For that, we first compute a local residual based on a restriction of the fine-grid solution at each local coarse node included in the refinement zone. Afterwards, we solve globally on the whole coarse-grid the associated error equation, and we add this error to the previous coarse-grid solution. The error equations are of the same kind as the previous Navier–Stokes and energy equations (34, 30), (31) and (35, 26). The partial differential operators are the same, the right-hand side members are now defined with the local residuals, and the boundary conditions on a real physical boundary are homogeneous. Many methods are proposed and differ in the way to define the local residuals, e.g. [50]. We propose the flux interface correction (FIC) method, where the local residual is computed by balancing, in a conservative way, the coarse and fine fluxes, i.e. the fluxes associated to each divergential term of equations (34), (31), (35), through the interface of each coarse-grid finite volume included inside the refinement region, see [8,50,10] for more details. As proposed in [28] or [52], we choose a ratio equal to three between the space discretization steps of two successive coarse and fine-grids, which yields a partitioning of a coarse-grid scalar control volume into nine fine-grid control volumes (cf. Figure 10). In that way, the interface of coarse control volumes coincides with interfaces of fine control volumes. This enables us to simplify the expression of the restriction operators, ensuring the conservative balance of the fluxes from both coarse and fine-grids [28].

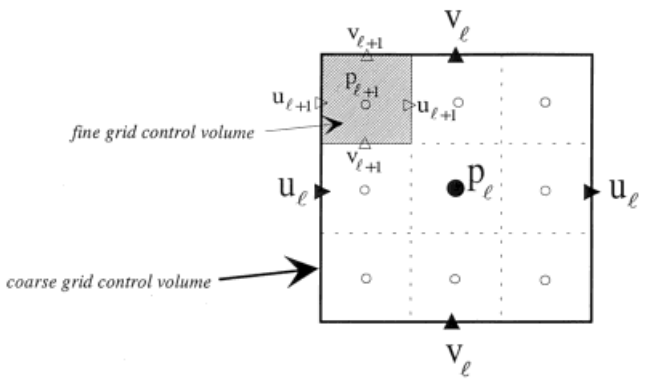

Figure 10. Partitioning of a coarse-grid control volume into fine-grid control volumes.

5. NUMERICAL APPLICATIONS

5.1. External flow around a cylinder

5.1.1. Description of the problem. This application concerns the two-dimensional, incompressible, laminar and unsteady flow of a viscous fluid around a solid circular cylinder in an open infinite domain. The numerical resolution requires a 'fictitious boundary' $\tilde{\Gamma}$ set far enough from the obstacle, on which suitable boundary conditions must be written. We choose a Dirichlet condition upstream, imposing the velocity and temperature of the fluid at infinity, and a Fourier or Robin condition on the other parts of the boundary. We study the following configuration, where the fictitious domain is $\tilde{\Omega} = \Omega_f \cup \Omega_s$ (Ω_f and Ω_s represent respectively the fluid and the solid domain). The computational domain is $\tilde{\Omega} = [0, 20] \times [0, 5]$, and the centre of the cylinder is located at (5, 2.5) (cf. Figure 11). The Reynolds number is defined by $Re = U_\infty D/v$, where $U_\infty = 1$ is the fluid velocity at upstream infinity, $D = 1$ the diameter of the cylinder, taken as the reference dimension, and v the kinematic viscosity of the fluid.

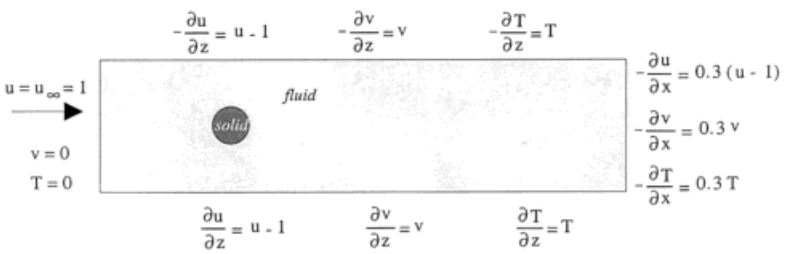

Figure 11. Boundary conditions defined on the fictitious boundary $\tilde{\Gamma}$.

We have made different numerical tests for the choice of the coefficients α_u, α_v, α_T, in the Fourier boundary conditions (see section 2.3). From a semi-empiric point of view, the best values seem to be located between 0 and 1.

We approximate the solid obstacle by a rectangular Cartesian mesh (cf. Figure 12). This kind of approximation is, surely, less accurate than the choice of curved elements with a body-fitted mesh, but it is much cheaper and gives satisfactory results, as we will see further. In fact, whatever the kind of mesh, the main difficulty is the good representation of the boundary layer along the obstacle. It requires a sufficient number of nodes in this region of thickness, varying like $\mathcal{O}(Re^{-1/2})$, in order to efficiently model the boundary layer; *a priori*, this can be achieved using a local mesh refinement.

5.1.2. Application to steady flows. For a first validation, we study the case of the steady flow. The computations are performed on a 256×64 uniform mesh grid (about 16400 nodes). On Figure 13, we observe the good accuracy of the results, comparing them with some experimental results, and most of the numerical ones, precisely referred in [20]. This first study shows the good behaviour and stability of our methodology; see also Figure 14 which shows how the no-slip boundary condition is satisfied on the solid cylinder. It also validates the calculation of drag and lift forces with Equation (16). In particular, although the flow is supposed to become unsteady and periodic for $45 < Re \leq 100$, e.g. [32], if no numerical instability appears, we can get the steady symmetric solution, whereas the configuration remains symmetric. Let us note that for the steady symmetric case, the lift coefficient remains equal to zero.

5.1.3. Application to unsteady flows. The next study concerns the unsteady case with larger Reynolds numbers $50 \leq Re \leq 200$. Note, that the flow becomes physically three-dimensional when Re is about 200, see [47]. The computations are performed with a dimensionless time step equal to 0.1 on a 256×64 uniform mesh grid. We perform 1000 time iterations, and at each time step, we stop the augmented Lagrangian iterations when $\|\nabla \cdot \mathbf{U}\|_{L^1} < 10^{-5}$. The computation time is about 1 h 30 min on a Cray C98. Thanks to the flexibility of our methodology, we introduce a physical perturbation in the flow to make it unsteady. It is done, for example, by slightly moving the cylinder to the bottom during a short period of computation after the initial time, typically during 10 or 20 time steps, we replace it afterwards at its original position. We first observe, on Figure 15, the good qualitative representation of the resulting flow where the Von Karman alternating eddies occur in the wake of the obstacle.

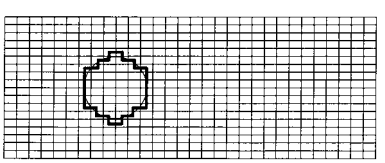

Figure 12. Approximation of the obstacle using a rectangular Cartesian mesh.

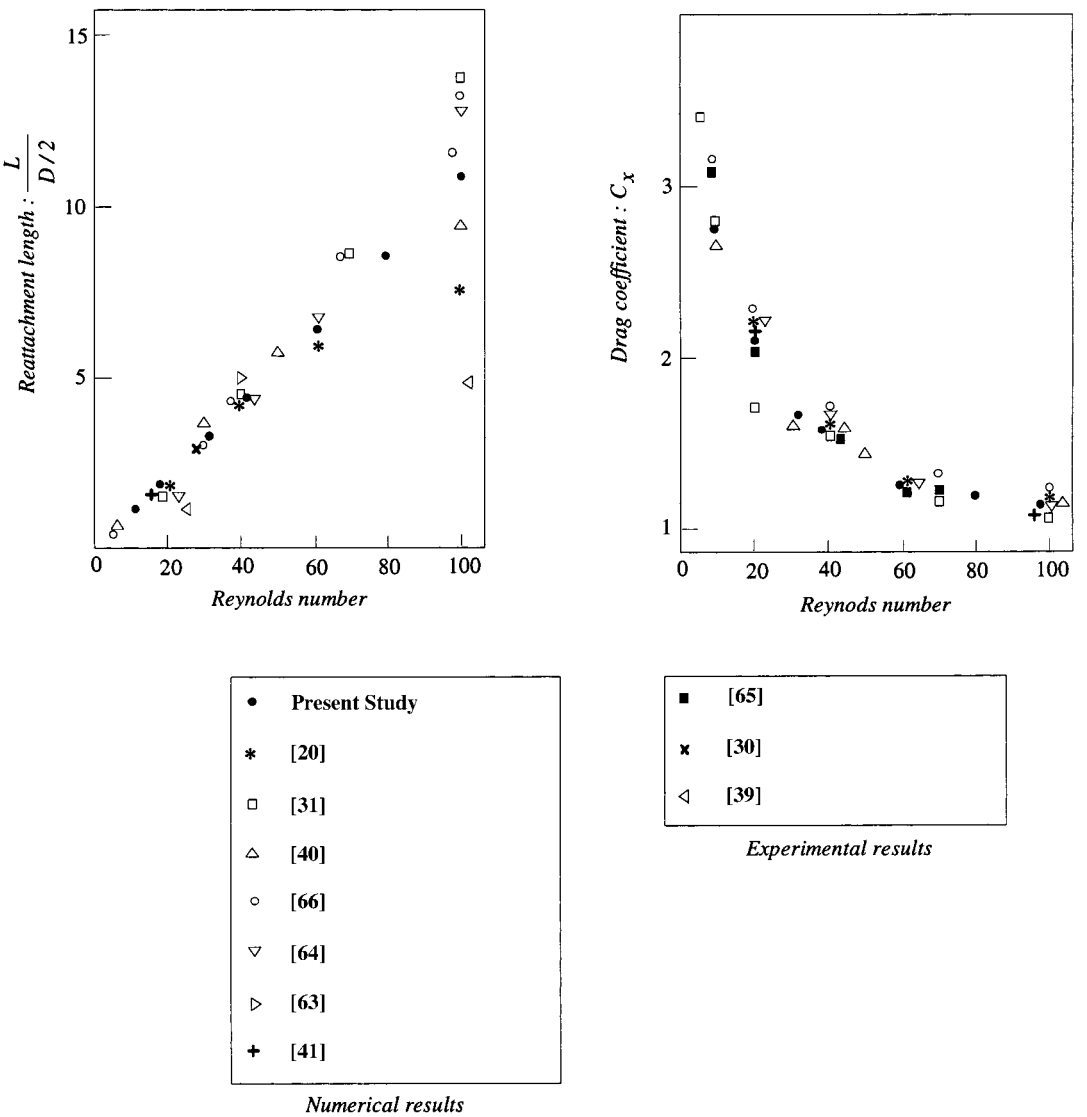

Figure 13. Reattachment length and drag coefficient versus Reynolds number for $\tilde{\Omega} = [0, 20] \times [0, 5]$—uniform mesh 256×64.

Using fast Fourier transform (FFT) computations, we make a frequency analysis of the signals shown in Figure 16, and associated to the drag and lift coefficients calculated from Equation (16) by:

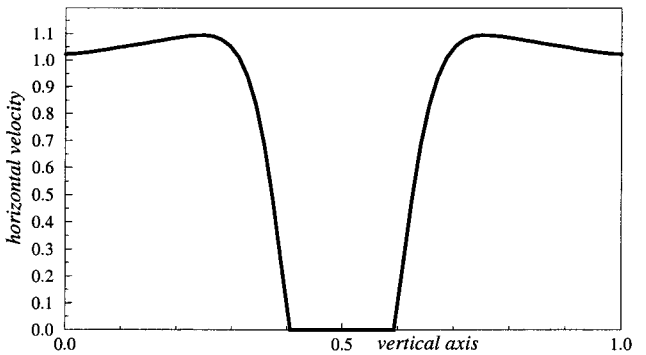

Figure 14. Horizontal velocity profile for $Re = 30$, $\tilde{\Omega} = [0, 4] \times [0, 1] - D = 0.2$—uniform mesh 256×64.

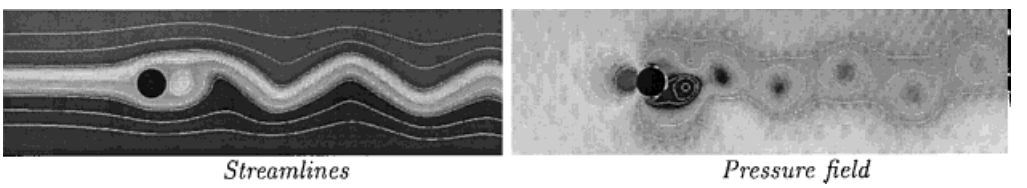

Figure 15. Representation of the unsteady flow. $Re = 200 - \Delta t = 0.1 - \tilde{\Omega} = [0, 20] \times [0, 5]$—uniform mesh 256×64.

$$C_x = \frac{F_x}{\frac{1}{2}\rho U_\infty^2 D}, \quad C_z = \frac{F_z}{\frac{1}{2}\rho U_\infty^2 D} \tag{36}$$

We next show, on Figure 17, the variation of the Strouhal number, i.e. the dimensionless frequency $St = fD/U_\infty$ versus the Reynolds number. In spite of the Cartesian mesh approximation of the cylinder and the use of a single uniform grid without any local mesh refinement, the results are very close to the experimental ones [68,42], and even better than some other numerical results of the literature [21,46,47]. In this range of small Reynolds numbers, the mesh used here is fine enough and no mesh refinement seems necessary.

For a Reynolds number equal to 400, we investigate on Figure 18 both the influence of the size of the computational domain $\tilde{\Omega}$, and the influence of the mesh size for the approximation of the cylinder. For two different domains chosen with the same discretization step: $\tilde{\Omega} = [0, 20] \times [0, 5]$, with uniform mesh 256×64, and $\tilde{\Omega} = [0, 40] \times [0, 10]$, with uniform mesh 512×128, we obtain similar results, except for the phase error resulting from different initial conditions when forcing the unsteady state; it proves the efficiency of the outflow boundary conditions used here. However, at this Reynolds number, these grids are too coarse, and the cylinder is roughly approximated with respect to the effective representation of the boundary layer. The use of a twice smaller mesh step: $\tilde{\Omega} = [0, 20] \times [0, 5]$ with a uniform mesh 512×128,

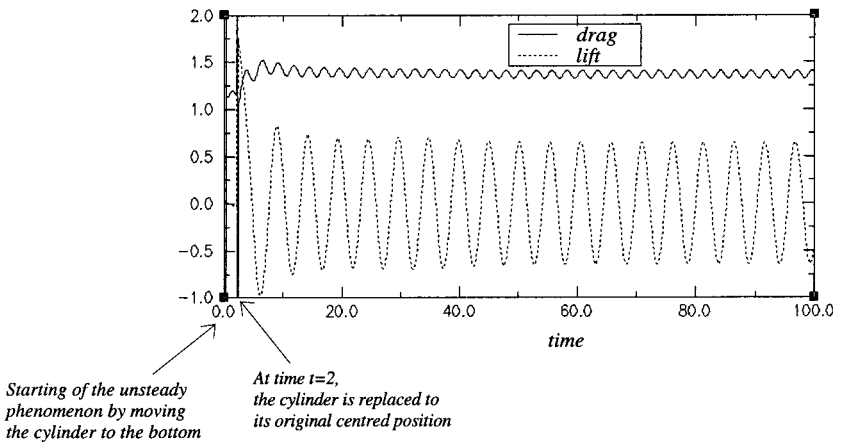

Figure 16. Drag and lift coefficients versus time. $Re = 200 - \Delta t = 0.1 - \tilde{\Omega} = [0, 20] \times [0, 5]$—uniform mesh 256×64.

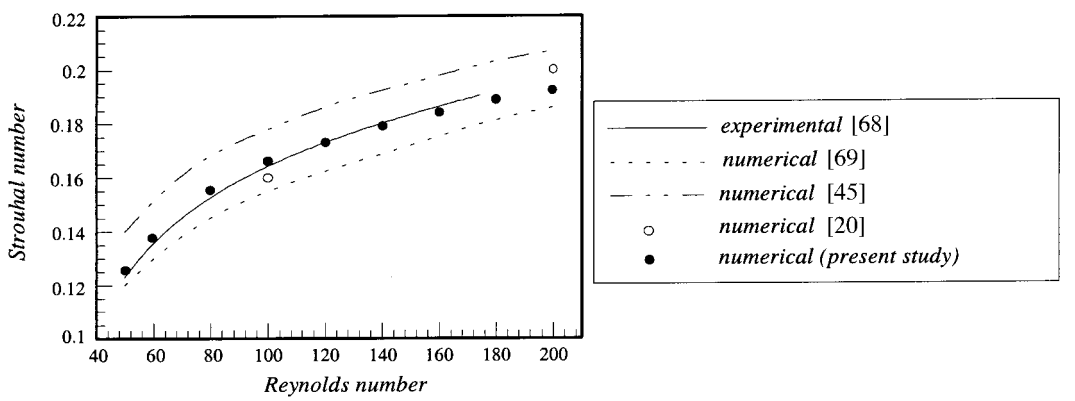

Figure 17. Strouhal number $St = fD/\mathbf{U}_\infty$ versus Reynolds number $\mathrm{Re} = \mathbf{U}_\infty D/v$. $\Delta t = 0.1 - \tilde{\Omega} = [0, 20] \times [0, 5]$—uniform mesh 256×64.

gives more accurate results. The best results are obtained for a domain of computation $\tilde{\Omega} = [0, 20] \times [0, 5]$, with the use of a 512×128 non-uniform mesh grid, where more nodes are concentrated close to the surface of obstacle (cf. Figure 18). In that case, the geometry of the cylinder is better approximated, and the boundary layer is also better resolved because it contains more nodes, see Figures 19 and 20. As the latter two curves with 512×128 uniform and non-uniform meshes are relatively close, the mesh convergence is surely reached in that case.

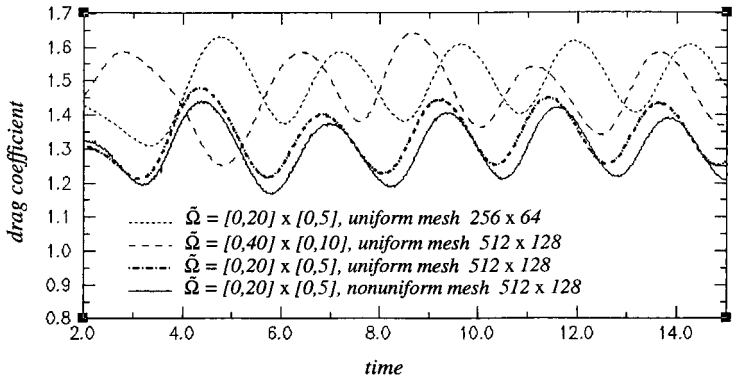

Figure 18. Drag coefficients versus time. $Re = 400 - \Delta t = 0.1$.

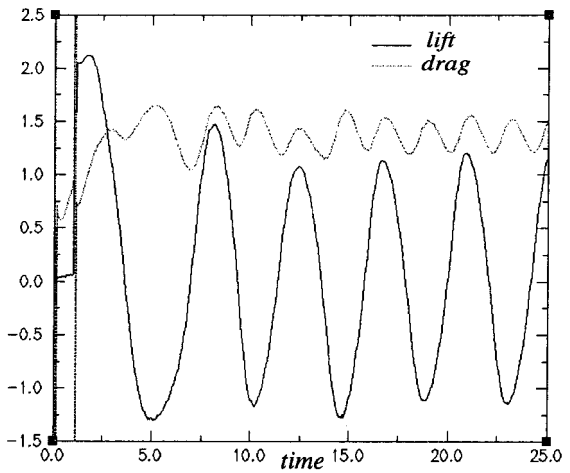

Figure 19. Drag and lift coefficients versus time. $Re = 400 - \tilde{\Omega} = [0, 20] \times [0, 5]$—non-uniform mesh 512×128.

For a domain of computation $\tilde{\Omega} = [0, 20] \times [0, 5]$, with a 512×128 non-uniform mesh grid, we show in Figure 21 the evolution of the drag coefficient, during the time, for three different time steps. The initial conditions when forcing the unsteady state are different in the three cases, which explains the relatively high phase shift, but both the amplitude and the frequency of the signals are the same. Thus, the results obtained in these three cases are similar, but the computation times are shorter for larger time steps: $CPU_{\Delta t = 0.015} = 8$ h, $CPU_{\Delta t = 0.03} = 6$ h 15 min, and $CPU_{\Delta t = 0.04} = 5$ h 45 min, as less time iterations are needed. In fact, we have to find a compromise between the required time accuracy, the stability of the computation which

Figure 20. Vorticity field. $Re = 400 - \tilde{\Omega} = [0, 20] \times [0, 5]$—non-uniform mesh 512×128.

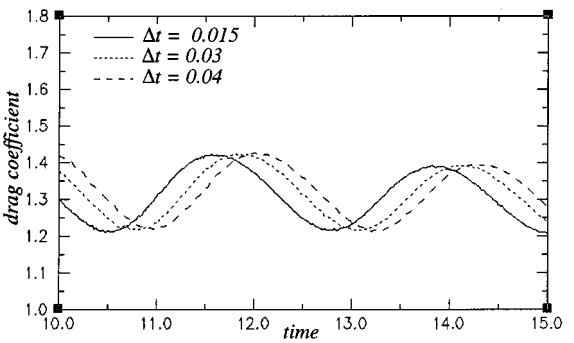

Figure 21. Drag coefficient versus time. $Re = 400 - \tilde{\Omega} = [0, 20] \times [0, 5]$—non-uniform mesh 512×128.

deteriorates when the time step increases, and the time of computation. Indeed, the more the time step Δt increases, the less the main diagonal of the linear systems becomes dominant, yielding a slower convergence of the solvers.

We next investigate, in Figure 22, the influence of ε in the stopping test, based on the velocity divergence: $\|\nabla \cdot \mathbf{U}\|_{L^1} \leq \varepsilon$ of the augmented Lagrangian–Uzawa iterations at each time step. The curve with $\varepsilon = 10^{-4}$ is the same as the one obtained with $\varepsilon = 10^{-5}$, but it requires only 0 h 54 min of computation time, instead of 1 h 24 min, for a total dimensionless time of simulation equal to 100. For $\varepsilon = 10^{-3}$ (*CPU* time equal to 0 h 32 min), we observe important numerical errors, especially on the signal of the drag coefficient, whereas they remain weak for the lift coefficient. For $\varepsilon = 10^{-2}$, the computation is entirely false. Here again, we have to find a compromise between the stopping criterion and the computation time for using inexact solves.

For a Reynolds number equal to 1000, the flow represented in Figure 23 seems to be qualitatively coherent compared with [21], but the mean drag coefficient of about 1.3 is a little overestimated, compared with the experimental reference values, about 1.2. In fact, more nodes would be required near the wall of the obstacle: the cylinder here is roughly

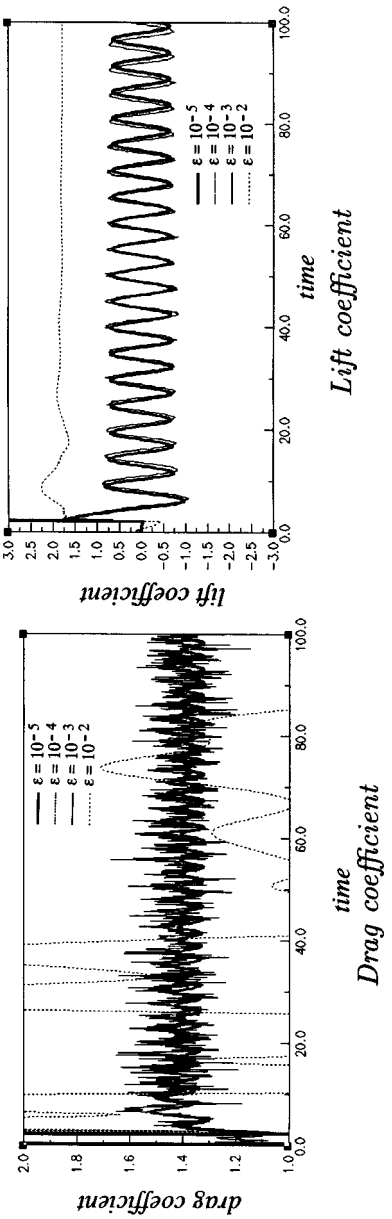

Figure 22. Representation of the drag and lift coefficients. $Re = 400 - \Delta t = 0.1 - \tilde{\Omega} = [0, 20] \times [0, 5]$—non-uniform mesh 512×128.

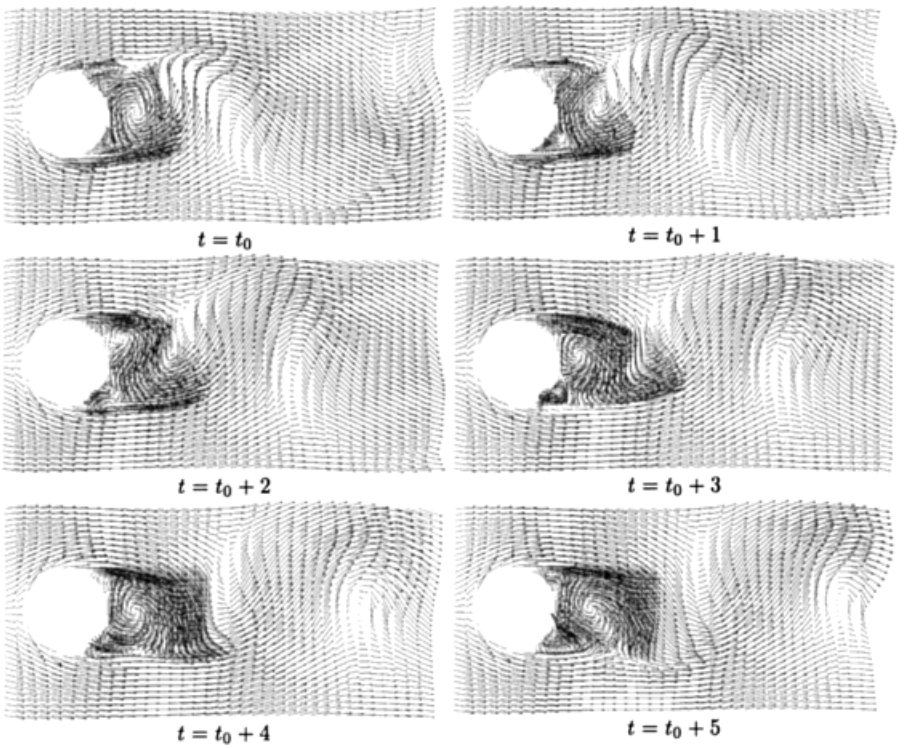

Figure 23. Velocity vector field. $Re = 1000 - \Delta t = 0.1 - \tilde{\Omega} = [0, 20] \times [0, 5]$—non-uniform mesh 512×128.

approximated, compared with the thickness of the boundary layer, and, hence, it influences the numerical results. In any case, apart from the approximation of the obstacle geometry, strong gradients and small scales structures appear in local regions of the domain at this level of Reynolds, which also requires a larger number of nodes in the boundary layers, as well as in the advected eddies.

5.1.4. The heated cylinder. The above results correspond to isothermal cases. As in [45], we now study the evolution of the temperature field during the time when the cylinder is heated. For that, we impose a volumic heat power proportional to the temperature of the obstacle in the domain Ω_s, This is made by taking $\beta_T = 0$ in Ω_f and $\beta_T = 0.3$ in Ω_s, $T_\infty = 1$ in Ω_s, whereas the initial temperature is zero everywhere, associated with the boundary conditions defined in Figure 11. At different time steps at $Re = 200$, we present, in Figure 24, the temperature fields showing the Von Karman vortices in the wake of the heated cylinder.

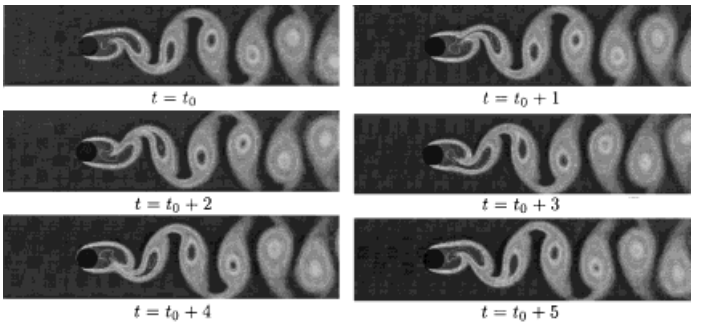

Figure 24. Temperature field. $Re = 200 - \Delta t = 0.1 - \tilde{\Omega} = [0, 20] \times [0, 5]$—uniform mesh 256×64.

5.2. Channel flow behind a backward facing step

5.2.1. Description of the problem. This application concerns the two-dimensional, incompressible, laminar and unsteady flow of a viscous fluid behind a backward facing step in a channel. We choose a Dirichlet condition upstream, imposing the velocity and temperature of the fluid at infinity. In this case, as we compute only the steady flow, we have numerically experimented that both Fourier or homogeneous Neumann outflow boundary conditions provide the same results, if the aspect ratio is large enough. We choose a Neumann condition downstream. We impose at the top and the bottom the no-slip condition of the viscous fluid on the walls of the channel. We study the configuration shown in Figure 25, which was chosen in [15] or [44]. The computational fictitious domain is $\tilde{\Omega} = \Omega_f \cup \Omega_s = [0, 30] \times [0, 2]$. The Reynolds number is defined by $Re = U_\infty H/v$ where U_∞ is the fluid velocity at upstream infinity, $H = 1$ the height of the step, and v the kinematic viscosity of the fluid.

5.2.2. Application to steady flows. We present results concerning the reattachment lengths that are of three kinds (see Figure 26).

First, we perform the computations on a single global uniform grid without any local mesh refinement, for different Reynolds numbers and mesh grids. We investigate the mesh independence for the three reattachment lengths, as shown at $\mathrm{Re} = 500$ in Figure 27. Moreover, we see also in Figure 27 that the results of $L1$, for example, are in perfect agreement with some experimental [15] and numerical data [44] found in the literature. Next, we study the evolution

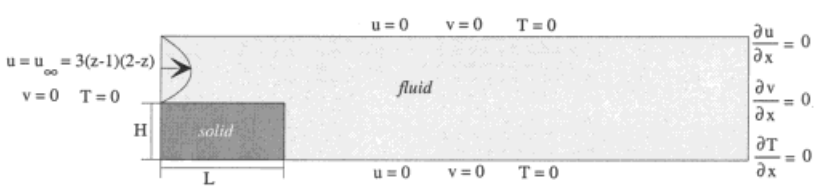

Figure 25. Computational domain $\tilde{\Omega} = [0, 30] \times [0, 2]$, $L = 3$, $H = 1$.

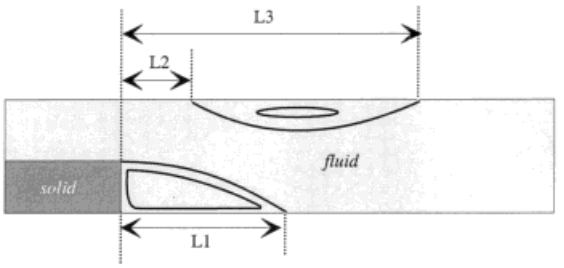

Figure 26. Different reattachment lengths.

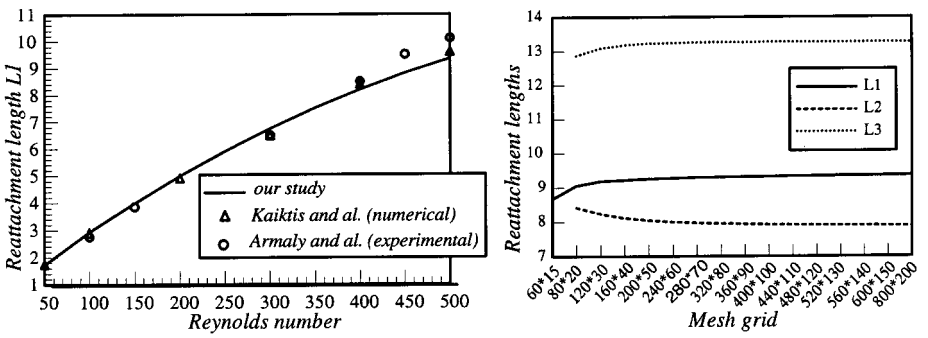

Figure 27. Reattachment lengths for a steady flow.

of the *CPU* time needed for the convergence of the problem to the steady solution, according to the number of nodes in the discrete domain. We observe in Figure 28 that the computation time asymptotically increases a little bit more than linearly with the number of discretization points. This justifies the interest for choosing carefully the regions that need a refinement.

In a second step, we compute a reference solution at $Re = 500$ on a single very fine uniform grid with 900×450 cells (about 406000 nodes), with 1200 time iterations, and a divergence criterion, such that $\|\nabla \cdot \mathbf{U}\|_{L^1} < 10^{-5}$ for the Uzawa algorithm. The solution is computed within 5 h 15 min of *CPU* time on a Cray C98. The corresponding reattachment lengths are respectively $L1 = 9.142$, $L2 = 7.660$ and $L3 = 13.497$. This reference solution allows us to compute the discrete errors associated to coarser mesh grids. In particular, we can compare the accuracy of the solution obtained on a single coarse-grid without any mesh refinement with the one obtained on this same grid by using a local multigrid method.

We now perform a local multigrid mesh refinement around the step. After solving the problem on a coarse-grid, we have to define a boundary condition at the interface γ between two successive subdomains for the computation of the fine-grid solution. This boundary condition is based on an interpolation of the coarse-grid solution at the fine-grid nodes on the interface. On the boundary interface γ between a coarse and a fine-grid, we write:

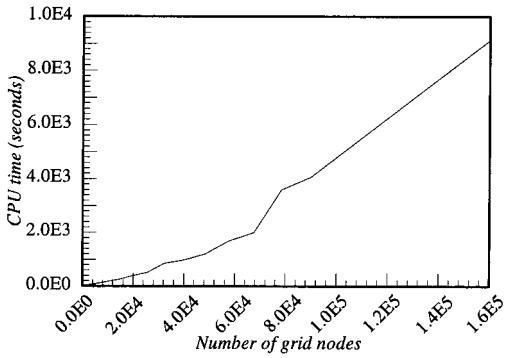

Figure 28. Computation time versus the number of nodes at $Re = 500$.

$$-\frac{\partial \mathbf{U}_{\ell+1}}{\partial n} = \beta_{\mathbf{U}_{\ell+1}}(\mathbf{U}_{\ell+1} - \mathbf{U}_{\infty,\ell+1}) \quad \text{on} \quad \gamma \tag{37}$$

with $\mathbf{U}_{\infty,\ell+1} = P\mathbf{U}_{\ell}$, where P is a projection (prolongation) operator on γ. The choice of the order of the interpolation operator P generally depends on the order of precision of the discretization scheme. Moreover, the interpolation must conserve on γ the mass flow between the subdomains; thus we impose that:

$$\int_{\gamma} \mathbf{U}_{\infty,\ell+1} \cdot \mathbf{dS}_{\ell+1} = \int_{\gamma} \mathbf{U}_{\ell} \cdot \mathbf{dS}_{\ell} \tag{38}$$

As the numerical spatial schemes used here are globally second-order accurate, we use a first-order polynomial interpolation, and a Dirichlet condition, i.e. $\beta_{\mathbf{U}_{\ell+1}} + \infty$, on the normal components of the velocity.

We compare the results obtained with a single uniform grid G_0 with 20×10 cells, to the one obtained with a two-grid mesh refinement method, where we use the same global coarse-grid associated with a local fine-grid G_1, with 36×18 cells, which overlaps the step. The fine-grid step is equal to the coarse one divided by three. We compute the relative errors on the coarse-grid G_0 associated to the longitudinal component u of the velocity and the reattachment lengths $L1$ and $L2$ by comparison with the reference solution. For the correction on the coarse-grid, we use either the canonical injection for the restriction of the fine-grid solution on the coarse nodes included in the refinement region, or the Zoom FIC(ω) method, which provides a global correction over the whole heterogeneous domain $\tilde{\Omega}$. On each grid, we use a 'quasi-exact' Uzawa solver, such that $\|\nabla \cdot \mathbf{U}\|_{L^1} < 10^{-8}$ at each multigrid iteration, and the convergence is typically obtained within 10 multigrid iterations. In Table I, we observe that the local canonical restriction of the solution used alone does not almost improve the general solution (see u), but gives the worst local results (see $L1$). Moreover, the external field out of the zoom area is not modified (see $L2$). On the contrary, with a FIC global correction, the improvement is provided in the whole computational domain, not only in the zoom area (see

Table I. Relative errors (%) computed on the global 20×10 coarse-grid at $Re = 500$

	Mean L^1-error on u over $\tilde{\Omega}$	Error on $L1$	Error on $L2$	*CPU* time on Cray C98
Single uniform grid, no mesh refinement	3.14	4.1	17.4	26 s
Two-grid method with local canonical restriction	3.05	14.8	17.4	30 s
Two-grid method with zoom FIC ($\omega = 0.2$)	2.07	2.2	3.9	54 s

$L1$), but also beyond the refinement zone (see u and $L2$). The errors on $L1$ and $L2$ are respectively divided by 6.5 and 4.5, whereas the *CPU* time is only twice more. As a simple comparison, we can deduce from the results of Figure 28 that the computation on a single global finer grid having the same mesh step as G_1, i.e. with nine times more nodes than the present grid G_0, requires a *CPU* time ten times higher than for G_0.

6. CONCLUSION AND PERSPECTIVES

This study shows the good efficiency of the fictitious domain methodology proposed here for the numerical modelling of fluid flows and heat transfer at moderate Reynolds numbers. The results are in perfect agreement with the experimental ones. This approach provides effective robustness and flexibility in what concerns a mutual independence of both geometry, mesh grids and boundary conditions, allowing an easy modification of these three parameters during the computation time. The fictitious domain approach, combined with a finite volume method using a Cartesian mesh, yields accurate results up to moderate Reynolds numbers.

At high Reynolds numbers, it is necessary to increase the number of nodes in some local regions of the domain. For that, we have shown how a multigrid local mesh refinement with the multilevel zoom FIC method appears to be very attractive and efficient, when combined with the fictitious domain modelling. Moreover, the local multigrid approach is very interesting, as it requires to solve a set of discrete problems of moderate size only which is attractive in terms of convergence rate of the solvers. The generalization to 3D is also straightforward because we use the primitive variables formulation. The efficiency of this numerical strategy, together with its possible parallelization, opens the way for some interesting evolution of such techniques.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the use of Gray C98 at I.D.R.I.S.

REFERENCES

1. Angot Ph. Contribution à l'étude des transferts thermiques dans des systèmes complexes; application aux composants électroniques, Thèse de Doctorat de l'Université Bordeaux I. Spécialité Mécanique, Avril, 1989.
2. Angot Ph, Caltagirone JP. Logiciel d'Ingéniérie Scientifique THANES, Manuel Technique, Commercialisé sous Licence no. L89043 CNRS/ENSAM/Université Bordeaux I, Société G5G, PA Château Rouquey—BP 168–33708 Mérignac, 1989.
3. Angot Ph, Caltagirone JP. New graphical and computational architecture concept for numerical simulation on supercomputers. In *Proceedings 2nd World Congress on Computational Mechanics*, vol. 1. Stuttgart, 1990; 973–976.
4. Angot Ph, Caltagirone JP. Natural convection through periodic porous media (Proceedings 9th International Heat Transfer Conference, Jerusalem). *Hemisphere* 1990; **5**: 219–224.
5. Angot Ph. Le Projet AQUILON, Développement d'un environnement informatique et graphique adapté à l'expérimentation numérique interactive, CERFACS Technical Report, TR/VI/90/41, October, 1990.
6. Angot Ph, Darnaudéry JP. Thermal Modelling of IBM Power Single Chip Modules, IBM Technical Report, TR-47140, January, 1991.
7. Angot Ph, Caltagirone JP, Khadra K. Une méthode adaptative de raffinement local: la Correction du Flux à l'Interface. *CR Académie Science de Paris* 1992; **315**(Série I): 739–745.
8. Angot Ph, Laugier M. La méthode FIC de raccordement conservatif de sous-domaines emboîtes pour un modèle de circulation océanique. *CR Académie Science de Paris* 1994; **319**(Série II): 993–1000.
9. Angot Ph, Kortas S, Fürst J. Parallel and distributed multi-domain methods for numerical fluid dynamics. In *High Performance Computing in Europe on IBM Platforms*, *Sup'Eur 96 Conference Proceedings*, Bubak M, Mościński J (eds). ACC CYFRONET: Kraków, 1996; 111–120.
10. Ph Angot, M Laugier. Conservative matching of non-conforming grids with the nested Multi-level Zoom FIC method; application to an ocean circulation model, *East–West Journal of Numerical Mathematics* 1996; Preprint (accepted).
11. Angot Ph. Analysis of singular perturbations on the Brinkman problem for fictitious domain models of viscous flows, M^2 AS. *Mathematical Methods in the Applied Sciences* 1999; **22**: 1395–1412.
12. Angot Ph, Bruneau CH, Fabrie P. A penalization method to take into account obstacles in incompressible viscous flows. *Nümerische Mathematik* 1999; **81**(4): 497–520.
13. Angot Ph. Finite volume methods for non-smooth solution of diffusion models; application to imperfect contact problems. In *Recent Advances in Numerical Methods and Applications*, *Proceedings 4th International Conference of NMA 1998*, Iliev OP, Kaschiev MS, Margenov SD, Sendov BlH, Vassilevski PS (eds). World Scientific Publishing: Sofia (Bulgaria), 1999; 621–629.
14. Angot Ph. Mathematical and numerical modelling for a fictitious domain method with penalized immersed boundary conditions. Preprint HDR Thesis, Université Méditerranée Aix-Marseille, 1998.
15. Armaly BF, Durst F, Pereira JCF, Sch6nung B. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics* 1983; **127**: 473–496.
16. Arquis E, Caltagirone JP. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: application 4 la convection naturelle. *CR Acadé Science de Paris* 1984; **299**(1 Série II): 1–4.
17. Arquis E, Caltagirone JP. Interacting convection between fluid and open porous layers, ASME WAM, Boston, Paper 87 WA/HT-24, 1987.
18. Börgers C, Widlund OB. On finite element domain imbedding methods. *SIAM Journal of Numerical Analysis* 1990; **27**(4): 963–978.
19. Börgers C. Domain embedding methods for the Stokes equations. *Numeric Mathematics* 1990; **57**: 435–451.
20. Braza M. Simulation numérique du décollement instationnaire externe par une formulation vitesse-pression application à l'écoulement autour d'un cylindre. Thèse Doctorat de I'Institut National Polytechnique de Toulouse. Spécialité Mécanique, 1981.
21. Braza M, Chassaing P, Ha Minh H. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics* 1986; **165**: 79–130.
22. Brinkman HC. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Applied Scientific Research* 1947; **A1**: 27–34.
23. Brinkman HC. On the permeability of media consisting of closely packed porous particles. *Applied Scientific Research* 1947; **A1**: 81–86.
24. Briscolini M, Santangelo P. Development of the Mask method for incompressible unsteady flows. *Journal of Computer Physics* 1989; **84**: 57–75.
25. Bruneau ChH, Fabrie P. Effective downstream boundary conditions for incompressible Navier–Stokes equations. *International Journal of Numerical Methods in Fluids* 1994; **19**: 693–705.

26. Buzbee BL, Dorr FW, George JA, Golub GH. The direct solution of the discrete Poisson equation on irregular regions. *SIAM Journal of Numerical Analysis* 1971; **8**: 722–736.
27. Caltagirone JP. Sur l'interaction fluide-milieu poreux; application au calcul des efforts exercés sur un obstacle par un fluide visqueux. *CR Académie Science de Paris* 1994; **318**(Série II): 571–577.
28. Caltagirone JP, Khadra K, Angot Ph. Sur une méthode de raffinement local multigrille pour la résolution des équations de Navier–Stokes. *CR Academie Science de Paris* 1995; **320**(Série IIb): 295–302.
29. Cihlar J, Angot Ph. Numerical solution of Navier–Stokes systems. *Numerical Linear Algebra with Applications* 1999; **6**(1): 17–27.
30. Coutanceau M, Bouard R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation—Part 1—Steady flow. *Journal of Fluid Mechanics* 1977; **79**(1): 231–256.
31. Dennis SC, Chang G-Z. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *Journal of Fluid Mechanics* 1970; **42**(3): 471–489.
32. Dušek J, Le Gal P, Fraunié Ph. A numerical and theoretical study of the first Hopf bifurcation in a cylinder wake. *Journal of Fluid Mechanics* 1994; **264**: 59–80.
33. Fortin M, Glowinski R. Méthodes de Lagrangien Augmenté; Applications à la Résolution Numérique de Problèmes aux Limites. In *Méthodes Mathematique de l'Informatique*, vol. 9. Dunod-Bordas: Paris, 1982.
34. Glowinski R. Numerical methods for non-linear variational problems. In *Springer Series in Computational Physics*. Springer-Verlag: New York, 1984.
35. Glowinski R, Pan TW, Periaux J. A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations. *Computing Methods in Applied Mechanical Engineering* 1994; **112**: 133–148.
36. Glowinski R, Pan TW, Kearsley AJ, Periaux J. Numerical simulation and optimal shape for viscous flow by a fictitious domain method. *International Journal of Numerical Methods in Fluids* 1995; **20**: 695–711.
37. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *Journal of Computer Physics* 1993; **105**: 354–366.
38. Gresho PM. Incompressible fluid dynamics: some fundamental formulation issues. *Annual Review of Fluid Mechanics* 1991; **23**: 413–453.
39. Grove AS, Shair FH, Petersen EE, Acrivos A. An experimental investigation of the steady separated flow past a circular cylinder. *Journal of Fluid Mechanics* 1965; **19**(1): 60–80.
40. Hamielec A, Raal J. Numerical studies of viscous flow around circular cylinders. *Physical Fluids* 1969; **12**(1): 11–17.
41. Ha Minh H. Décollement provoqué d'un écoulement turbulent incompressible, Thèse de Doctorat d'Etat. I.N.P. Toulouse, 1976.
42. Hammache M, Gharib M. An experimental study of the parallel and oblique vortex shedding from circular cylinders. *Journal of Fluid Mechanics* 1991; **232**: 567.
43. Hockney RW. The potential calculation and some applications. In *Methods in Computational Physics*, vol. 9, Adler, Fernbach, Rotenberg L (eds). Academic Press: London, 1969; 136–211.
44. Kaiktsis L, Karniadakis GEM, Orszag SA. Onset of three-dimensionality, equilibria and early transition in flow over a backward-facing step. *Journal of Fluid Mechanics* 1991; **231**: 501–528.
45. Karniadakis GEM. Numerical simulation of forced convection heat transfer from a cylinder in crossflow. *International Journal of Heat Mass Transfer* 1988; **31**(1): 107–118.
46. Karniadakis GEM, Triantafyllou GS. Frequency selection and asymptotic states in laminar wakes. *Journal of Fluid Mechanics* 1989; **199**: 441.
47. Karniadakis GEM, Triantafyllou GS. The three-dimensional dynamics and transition to turbulence in the wake of bluff objects. *Journal of Fluid Mechanics* 1992; **238**: 1–30.
48. Khadra K, Angot Ph, Caltagirone JP. A comparison of locally adaptative multigrid methods: L.D.C., F.A.C. and F.L.C. In *Sixth Copper Mountain Conference on Multigrid Methods*, vol. 1, Melson ND, Manteuffel TA, McCormick SF (eds). NASA Conference Publication CP-3224: Hampton, VA, 1993; 275–292.
49. Khadra K. Méthodes adaptatives de raffinement local multigrille; applications aux quations de Navier–Stokes et de l'énergie. Thèse de Doctorat de l'Université Bordeaux I, Spécialité Mathématiques Appliquées, Mars, 1994.
50. Khadra K, Angot Ph, Caltagirone JP, Morel P. Concept de Zoom adaptatif en architecture multigrille locale; étude comparative des méthodes L.D.C., F.A.C. et F.I.C., M^2 AN. *Mathematical Modelling and Numerical Analysis* 1996; **30**(1): 39–82.
51. Khadra K, Angot Ph, Parneix S, Caltagirone JP. Computation of the Navier–Stokes and thermal equations by multigfid local mesh refinement methods. In *Advanced Concepts and Techniques in Thermal Modelling*, Lemonnier D, Saulnier JB, Fiebig M (eds). Elsevier: New York/Amsterdam, 1996; 83–89.
52. Laugier M, Angot Ph, Mortier L. Nested-grid methods for an ocean model: a comparative study. *International Journal Numerical Methods in Fluids* 1996; **23**(11): 1163–1195.

53. Leonard BP. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computing Methods in Applied Mechanical Engineering* 1979; **19**: 59–98 (1rst ed. 1975).
54. Marchuk GI. Methods of numerical mathematics. In *Application of Maths*, vol. 2. Springer-Verlag: New York, 1982.
55. Marchuk GI, Kuznetsov YA, Matsokin AM. Fictitious domain and domain decomposition methods. *Soviet Journal of Numerical Analysis in Mathematic Modelling* 1986; **1**: 3–35.
56. Parneix S. Simulation des écoulements turbulents et des transferts thermiques en vue de l'optimisation du refroidissement des pales de turbine. Thèse de Doctorat de l'Université Bordeaux I, Spécialité Mécanique, Juillet, 1995.
57. Parneix S, Khadra K, Caltagirone JP. Simulation numérique directe de la turbulence par une méthode de raffinement local multigrille, 31ème colloque d'Aérodynamique Appliquée, AAAF, 1995.
58. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington DC, 1980.
59. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computer Physics* 1977; **25**: 220–252.
60. Peyret R, Taylor TD. Computational methods for fluid flow. In *Springer Series in Computational Physics*. Springer-Verlag: New York, 1983.
61. Richard N. Modélisation des écoulements tridimensionnels turbulents dans une pale de turbine. Thèse de Doctorat de l'Université Bordeaux I, Spécialité Mathématiques Appliquées, Decembre, 1992.
62. Saiki EM, Biringen S. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computer Physics* 1996; **123**: 450–465.
63. Son JS, Hanratty TJ. Numerical solution for the flow around a cylinder at Reynolds numbers of 40, 200 and 500. *Journal of Fluid Mechanics* 1969; **35**(2): 369–386.
64. Phuoc Loc Ta. Etude numérique de l'écoulement d'un fluide visqueux incompressible autour d'un cylindre fixe ou en rotation—Effet Magnus. *Journal Mécanique* 1975; **14**(1): 109–134.
65. Tritton DJ. A note on vortex streets behind circular cylinders at low Reynolds numbers. *Journal of Fluid Mechanics* 1971; **45**(1): 203–208.
66. Tuann SY, Olson M. Numerical studies of the flow around a circular cylinder by a finite element method. *Computers and Fluids* 1978; **6**: 219–240.
67. Van Der Vorst HA. BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems. *SIAM Journal of Scientific Statistics in Computing* 1992; **13**: 631–644.
68. Williamson CHK. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics* 1989; **206**: 579–627.
69. Chang K-S, Sa J-Y. Numerical study of the unsteady mixed convection heat transfer from a circular cylinder. *International Communications in Heat and Mass Transfer* 1989; **16**(3): 427.

