A Geometrical Domain Decomposition Method in Computational Fluid Dynamics

G. Houzeaux
R. Codina

Monograph CIMNE Nº-70, December 2002

A Geometrical Domain Decomposition Method in Computational Fluid Dynamics

G. Houzeaux
R. Codina

Monograph CIMNE Nº-70, December 2002

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING
Edificio C1, Campus Norte UPC
Gran Capitán s/n
08034 Barcelona, Spain

Primera edición: Diciembre 2002

A GEOMETRICAL DOMAIN DECOMPOSITION METHOD IN COMPUTATIONAL FLUID DYNAMICS
Monografía CIMNE M70

ISBN: 84-95999-17-X

Depósito legal: B-51341-2002

Abstract

The domain decomposition (DD) method we present in this work aims at solving incompressible flows around objects in relative motion. The DD algorithm is based on a Dirichlet/Neumann(Robin) coupling applied to overlapping subdomains. Hence, it is an extension of the classical Dirichlet/Neumann(Robin) method which uses disjoint subdomains. Actually, the field of application of this work is wider as it proposes to set up a possible theoretical framework for studying the overlapping extensions of classical mixed methods: the Dirichlet/Robin, Dirichlet/Neumann, Robin/Neumann and Robin/Robin DD methods.

We observe that mixed DD methods inherit some properties of the Schwarz method while they keep the behavior of the classical mixed DD methods when the overlap tends to zero. As a main result, we show that the overlap makes the proposed methods more robust than disjoint mixed DD methods.

The DD method we propose is geometric and algorithmic. It is geometric because the partition of the computational domain is performed before the meshing, and in accordance to the DD coupling. It is also algorithmic because the solution on each subdomain is obtained on separate processes and the exchange of information between the subdomains is carried out by a Master code. This strategy is very flexible as it requires almost no modification to the original numerical code. Therefore, only the Master code has to be adapted to the numerical codes and strategies used on each subdomain.

We present a detailed description of the implementation of the DD methods in the numerical framework of finite elements. We present interpolation techniques for Dirichlet and Neumann data as well as conservation algorithms. Once the domain decomposition coupling and interpolation techniques are defined, we set up a Chimera method for the solution of the flow over objets in relative movements. Tensorial transformations are introduced to be able to express variables measures in one subdomain.

Finally, the DD algorithm is applied to an implicit finite element code for the solution of the Navier-Stokes equations and also of the Reynolds Averaged Navier-Stokes equations together with a one-equation turbulence model.

Resumen

El método de descomposición de dominios (DD) que se propone en esta tesis pretende resolver flujos incompresibles alrededor de objetos en movimiento relativo. El algoritmo de DD está basado en un acoplamiento del tipo Dirichlet/Neumann(Robin) aplicado a subdominios con solapamiento, y es, por tanto, una extensión del método Dirichlet/Neumann(Robin) clásico con subdominios disjuntos. En realidad, el campo de aplicación de este estudio es mucho más amplio puesto que en el se propone un posible marco teórico para abordar la extensión a subdominios solapados de los métodos mixtos clásicos: métodos Dirichlet/Robin, Dirichlet/Neumann, Robin/Neumann y Robin/Robin.

Se observa que los métodos mixtos propuestos heredan propiedades del método de Schwarz y al mismo tiempo conservan el comportamiento de sus equivalentes sin solapamiento cuando este tiende a cero. Se muestra como resultado principal que el solapamiento hace estos métodos más robustos que los métodos sin solapamiento.

El método de DD que se estudia es geométrico y algorítmico. Es geométrico en el sentido de que la partición del dominio computacional se lleva a cabo antes del proceso de mallado y de acuerdo con el acoplamiento de DD que se prevé usar. Es también algorítmico porque la solución en cada subdominio se obtiene en procesos diferentes y el intercambio de información entre subdominios se realiza mediante un código maestro. Tal estrategia es muy flexible puesto que requiere muy pocas modificaciones del código numérico original. Por consiguiente, sólo el código maestro necesita ser adaptado a los códigos y estrategias numéricos utilizados en cada subdominio.

Se presenta una descripción detallada de la implementación del método de DD propuesto en el contexto numérico de los elementos finitos. Presentamos técnicas de interpolación para los datos de tipo Dirichlet y Neumann y desarrollamos algoritmos de conservación. Una vez el acoplamiento de DD y las interpolaciones definidos, presentamos un método del tipo Chimera para la resolución de flujos alrededor de objetos en movimiento. En particular, definimos transformaciones tensoriales para transformar variables de un subdominio a otro.

Finalmente, el algoritmo de DD se aplica a un código implícito para la resolución de las ecuaciones de Navier-Stokes incompresibles y también a las ecuaciones de Navier-Stokes promediadas con un modelo de turbulencia de una ecuación.

Contents

Introduction

The domain decomposition (DD) method we present in this work aims at solving incompressible flows around objects in relative motion. The DD algorithm is based on a Dirichlet/Neumann(Robin) coupling applied to overlapping subdomains. Hence, it is an extension of the classical Dirichlet/Neumann(Robin) method which uses disjoint subdomains. Actually, the field of application of this work is wider as it proposes to set up a possible theoretical framework for studying the overlapping extensions of classical mixed methods: the Dirichlet/Robin, Dirichlet/Neumann, Robin/Neumann and Robin/Robin DD methods. As the proposed method are applied to overlapping subdomains, we expect them to inherit some properties of the Schwarz method and to conserve the behavior of the classical mixed DD methods when the overlap tends to zero. As noted by Lions [1]:

> [...] *the Schwarz algorithm* [...] *presents some properties (like "robustness", or indifference to the type of equations considered...) which do not seem to be enjoyed by other methods.*

As a main result, we show in fact that the overlap makes the proposed methods more robust than disjoint mixed DD methods.

The DD method we propose is geometric and algorithmic. It is geometric because the partition of the computational domain is performed before the meshing, and in accordance to the DD coupling. It is also algorithmic because the solution on each subdomain is obtained on separate processes and the exchange of information between the subdomains is carried out by a Master code. This strategy is very flexible as it requires almost no modification to the original numerical code. Therefore, only the Master code has to be adapted to the numerical codes and strategies used on each subdomain. Although the algorithm can be easily parallelized, this is not the objective of this work and we will only mention the possibility of using a multicoloring technique to parallelize the computation.

We present a detailed description of the implementation of the DD methods in the numerical framework of finite elements. We present interpolation techniques for Dirichlet and Neumann data as well as conservation algorithms. Once the domain decomposition coupling and interpolation techniques are defined, we set up a Chimera method for the solution of the flow over objets in relative movements. Tensorial transformations are introduced to be able to express variables measured in one subdomain in any other one.

Finally, the DD algorithm is going to be applied to an implicit finite element code for the solution of the Navier-Stokes equations and also of the Reynolds Averaged Navier-Stokes equations together with a one-equation turbulence model.

The monograph is organized as follows. Chapter 1 presents a finite element method to model advection-diffusion-reaction problems, as well as laminar and turbulent flows. The classical advection-diffusion-reaction (ADR) equation is considered as a model problem to introduce the finite element

formulation. This includes a brief description of the physical meaning of the ADR equation, an introduction to the theoretical context of the variational formulation, a review of the stabilization methods developed in the literature, and the presentation of a variational subgrid scale method as a stabilization technique. Next, all the concepts introduced in the study of the ADR equation are generalized to the Navier-Stokes equations. Then we introduce a one-equation turbulence model, namely the Spalart-Allmaras model, in an original form, and apply the numerical strategy developed for the general ADR equation. Finally, we study and validate the whole numerical model through the solution of three numerical examples.

Chapter 2 is a brief introduction to domain decomposition methods. We start by studying a one-dimensional example in a more intuitive than rigorous manner. Although very simple, this example is sufficient to present all the families of domain decomposition methods we will deal with. In particular, we contemplate the possibility of using mixed DD methods on non-overlapping as well as on overlapping subdomains. Then, we consider the more general advection-diffusion-reaction equation, and describe all the possible improvements that can be achieved to the classical DD methods. In particular, we will mention the adaptive methods. At this point, we will have introduced the necessary terminology to proceed with the variational approach. The weak formulation of the domain decomposed problem will enable to justify the choice of transmission conditions, involving the essential and natural conditions. Then we discuss some ways of applications to the finite element method and finally present the proposal of this work.

Chapter 3 proposes to apply some ideas developed in the preceding chapter to the solution of a one-dimensional problem. In particular, we consider the overlapping extensions of two families of disjoint DD methods, namely the Dirichlet/Neumann and Dirichlet/Robin methods. The results obtained with these methods are systematically compared to those of their non-overlapping counterparts and those of the Schwarz method. We study in detail the convergence of the unrelaxed sequential algorithm and its dependence upon the overlapping length. Apart from the general ADR equation, we study three limiting behaviors of the equation, i.e. the Poisson equation, the advection-diffusion equation, and the hyperbolic limit. We also study the relaxed sequential versions, as well as the unrelaxed and relaxed parallel methods. This one-dimensional example enables us to foresee the improvements in convergence and stability of the solution obtained by using mixed DD methods with overlapping subdomains.

In Chapter 4, we go on to the multi-dimensional ADR equation and study overlapping mixed methods within a variational framework. This chapter constitutes therefore a theoretical basis for the study of overlapping mixed methods. The model domain decomposition method is based on an overlapping Dirichlet/Robin coupling. The starting point is a two-domain variational formulation of the problem, originating from a geometrical decomposition of the original domain of study. We show how the formulation can be reformulated into an overlapping domain decomposition method based on a Dirichlet/Robin coupling. Next, the domain decomposition method for the subdomains is re-written in terms of a problem for the interface unknowns. An iterative and relaxed sequential scheme is then introduced in order to solve the DD problem. The convergence is studied through the interface equations. We present the generalization of the overlapping DD method introduced to other types of overlapping mixed couplings, in particular to an overlapping Dirichlet/Neumann method. Afterwards, we consider the discrete counterpart of the formulation. We then build an iterative strategy to solve the two-domain problem. This strategy is studied algebraically, using a finite element method and solving for the Schur complement system, and is illustrated by four numerical examples. Finally, we introduce the extension of the overlapping Dirichlet/Neumann method to the Stokes and Navier-Stokes equations.

In Chapter 5 we derive a possible finite element implementation of two overlapping domain decomposition methods, the classical Schwarz method and an overlapping Dirichlet/Neumann

method, with particular attention on the latter. We first identify the transmission conditions from the alternative formulations of the DD derived in the preceding chapter. Then we set an iteration-by-subdomain method applied to the solution of the ADR and Navier-Stokes based on a Master/Slave strategy. We briefly describe an element search algorithm, which consists in looking for the host elements (in the underlying mesh) of the nodes involved in the iterative process. At this stage we are ready to interpolate the transmission conditions. We present the interpolation of the Dirichlet data and two interpolation schemes for the Neumann (or Robin) data. We then explain the need for using a conservative interpolation and present two algorithms: an interface constraining and a conservative interpolator. In order to be able to treat complex geometries, we introduce a Chimera method, using all the ingredients presented previously. Finally, the domain decomposition method is applied to moving subdomains by the way of tensorial transformations and appropriate time integration.

The last chapter presents five examples of application of the Dirichlet/Neumann method. They aim at showing the robustness of the algorithm for solving stationary and transient flows in laminar or turbulent state.

Chapter 1

A Finite Element Method for Incompressible Flows

This chapter presents the finite element method used in this work to model advection-diffusion-reaction problems, as well as laminar and turbulent flows. In the first section, the classical advection-diffusion-reaction (ADR) equation is considered as a model problem to introduce the finite element formulation. This includes a brief description of the physical meaning of the ADR equation, an introduction to the theoretical context of the variational formulation, a review of the stabilization methods developed in the literature, and the presentation of a variational subgrid scale method as a stabilization technique. Next, all the concepts introduced in the study of the ADR equation are generalized to the Navier-Stokes equations. Then we introduce a one-equation turbulence model, namely the Spalart-Allmars (SA) model, in an original form, and apply the numerical strategy developed for the general ADR equation. Finally, we study and validate the whole numerical model through the solution of three numerical examples.

1.1 The Advection-Diffusion-Reaction Equation

This section studies a transport model equation known as the ADR equation. After a brief presentation of the physical meaning of the equation, we introduce the variational formulation of the stationary problem. Then we derive three weak formulations and study the properties of the associated bilinear forms. Particular attention is paid on the natural conditions, as they are of special interest when studying mixed domain decomposition methods. We introduce a stabilized finite element method based on a subgrid scale approach. Finally, we solve the transient ADR equation using the trapezoidal rule for the time discretization.

1.1.1 Description of the ADR equation

The advection-diffusion-reaction equation is a transport equation for a physical variable that models advection, diffusion and reaction effects. The physical quantity of interest can be the concentration of a passive chemical species, the temperature, a turbulence quantity, etc. In the following, it is assumed that this scalar quantity does not interact with its surroundings: it is a passive scalar. We propose to study the following ADR problem:

$$Lu := -\varepsilon \Delta u + \nabla \cdot (\boldsymbol{a} u) + s u = f \qquad \text{in } \Omega, \tag{1.1}$$

where Ω is a n_{d}-dimensional domain (n_{d}=1,2,3) with boundary $\partial\Omega$; ε is the diffusion constant of the medium; f is the force term; $\boldsymbol{a}$ is the advection field (not necessarily solenoidal) and s is a source term.

Apart from the transport phenomena, the partial differential equation (1.1) is able to model many other physical problems; for example, the Helmholtz equation

$$\Delta u + k^2 u = 0,$$

which governs the motion of time-harmonic waves where k is the frequency parameter (this corresponds to the case $s \leq 0$); the Poisson equation

$$-\Delta u = f,$$

which for example models the electric field in a region of charge density f. We understand the importance of our model equation in the modelling of physical problems, and the need for a robust and reliable numerical strategy to solve it.

The differential equation must be furnished with appropriate boundary conditions on $\partial\Omega$. These boundary conditions are a compromise between physical and mathematical considerations. From the definition of the physical problem, which for example tries to reproduce an experiment, we may know the values of the unknown on part of the boundary, the flux on another, and have no information on those remaining. By determining if the differential equation is elliptic, parabolic or hyperbolic in character, we may have to give up imposing some of the data inherited from the experiment; for example, it is well known that when the equation is hyperbolic, the unknown cannot be prescribed on outflows. Finally, we will see that the weak formulation of the problem will, in its turn, propose essential and natural boundary conditions, and the requirements for the existence and uniqueness of a weak solution may restrict the possible choices even more. To simplify, we will only consider here only three types of boundary conditions, namely of Dirichlet, Neumann or Robin types, prescribing the following quantities:

$$\begin{aligned} &\text{Dirichlet:} && u, \\ &\text{Neumann:} && \varepsilon \frac{\partial u}{\partial n} := \varepsilon \nabla u \cdot \boldsymbol{n}, \\ &\text{Robin:} && \varepsilon \frac{\partial u}{\partial n} + \alpha_{\mathrm{r}} u, \end{aligned} \tag{1.2}$$

where $\partial(\cdot)/\partial n = \boldsymbol{n} \cdot \nabla(\cdot)$, $\boldsymbol{n}$ being the outward unit vector normal to the boundary considered, and α_{r} is called the Robin factor, coming from physical information or from the natural boundary condition derived from the weak formulation. All along this work on domain decomposition methods, we will have to juggle with these three boundary conditions and we will have to face many compromises, as some useful physical boundary conditions are not necessarily mathematically well suited! Sometimes we will be lucky, sometimes less... Now let us go back to our differential equation itself.

In order to obtain a first insight into a new problem, it can be useful to derive the transport equation in a dimensionless form. This non-dimensionalization will enable us to measure the relative effects of the advection, diffusion and reaction processes and, in some limit cases, determine which of them will drive the transport of the scalar. We define D a characteristic measure of Ω, A a characteristic value of the advection field and $|s|_\infty$ a characteristic value of the reaction term. Then

we label the dimensionless variables and operators of the problem with a superscript asterisk. By introducing two dimensionless parameters, namely the Péclet number Pe and the reaction number R of the equation, defined as

$$\mathrm{Pe} := \frac{AD}{\varepsilon},$$
$$\mathrm{R} := \frac{|s|_\infty D^2}{\varepsilon},$$

we can rewrite Equation (1.1) as follows:

$$-\frac{1}{\mathrm{Pe}}\Delta^* u^* + \nabla^* \cdot (\boldsymbol{a}^* u^*) + \mathrm{R}\, s^* u^* = f^* \text{ in } \Omega.$$

For example, when dealing with the temperature equation in heat transfer problems, the Péclet number is the ratio of the bulk heat transfer to the conductive heat transfer, i.e.

$$\mathrm{Pe} = \frac{\rho c_p D A}{k},$$

where k is the thermal conductivity, c_p the specific heat, ρ the density of the fluid, and D and A are defined as before. Keeping in mind the dimensionless form of the ADR equation, we will go on with the original (dimensional) equation.

1.1.2 Properties of the work spaces

The first steps for deriving a weak formulation of the problem are classical. We multiply the differential equation (1.1) by a test function v belonging to a suitable space (to be defined), integrate the result over Ω, and integrate by parts some of the terms. We will present here three weak formulations, each one being derived from different integrations by parts of the convective term. Before considering the weak formulations, we need to introduce some notations. We split $\partial\Omega$ into two components denoted Γ_D and Γ_N, and such that $\partial\Omega = \overline{\Gamma_D \cup \Gamma_N}$. As usual, $L_2(\Omega)$ refers to the space of square integrable functions in Ω. We set the following definitions:

$$\begin{aligned}
(w, v) &:= \int_\Omega wv \, d\Omega, \\
H^1(\Omega) &:= \{v \in L^2(\Omega) \mid \frac{\partial v}{\partial x_j} \in L^2(\Omega), j = 1, \ldots, n_\mathrm{d}\}, \\
H_0^1(\Omega) &:= \{v \in H^1(\Omega) \mid v_{|\partial\Omega} = 0\}, \\
H_{\Gamma_D}^1(\Omega) &:= \{v \in H^1(\Omega) \mid v_{|\Gamma_D} = 0\}, \\
V^0 &:= H_0^1(\Omega), \\
V &:= H_{\Gamma_D}^1(\Omega),
\end{aligned}$$

Likewise, we use the notation

$$\begin{aligned}
\langle \cdot, \cdot \rangle_\omega &:= \langle \cdot, \cdot \rangle_{H^{-1/2}(\omega) \times H_{00}^{1/2}(\omega)} && \text{for } \omega \ (n_\mathrm{d} - 1)\text{-dimensional}, \\
\langle \cdot, \cdot \rangle_\omega &:= \langle \cdot, \cdot \rangle_{(H^1(\omega))' \times H^1(\omega)} && \text{for } \omega \ n_\mathrm{d}\text{-dimensional},
\end{aligned}$$

for the duality pairings to be used. For these spaces, the duality paring is simply the integral

$$\langle \cdot, \cdot \rangle_\omega = \int_\omega (\cdot)(\cdot)\, d\omega.$$

We endow $H^1(\Omega)$ with the following scalar product

$$(w, v)_1 = (w, v) + (\nabla w, \nabla v),$$

and the associated norm

$$\|w\|_1 = \left[(w, w) + (\nabla w, \nabla w)\right]^{1/2}.$$

This norm is not "physically appropriate" as the units of its terms are not compatible. A more adequate choice would be to choose the graph norm, which contains the physical coefficient of the problem. However, we are not going to develop estimates displaying their dependences on the physical coefficients of L, and we prefer to work along with the usual $H^1(\Omega)$-norm.

Finally, we introduce the *trace operator*. From the trace theorem (see e.g. [2, 3]), we know that there exists a unique linear continuous map γ_0, called the trace operator, defined as

$$\gamma_0 v = v_{|\partial\Omega} \in H^{1/2}(\partial\Omega) \qquad \forall\, v \in V. \tag{1.3}$$

This result applies equivalently to any Lipschitz continuous subset of $\partial\Omega$. The continuity of γ_0 means that there exists a positive constant C^* such that

$$\|v_{|\partial\Omega}\|_{1/2,\partial\Omega} \leq C^* \|v\|_1 \qquad \forall\, v \in H^1(\Omega).$$

1.1.3 Weak formulations

Let us consider our differential problem (1.1). We restrict ourselves to solutions in $H^1(\Omega)$, which obliges us to choose the force term such that

$$f \in (H^1(\Omega))'.$$

In order to be able to show the existence and uniqueness of the solution, we must assume that all the remaining terms present in last equation are bounded, i.e.

$$s \in L_\infty(\Omega), \nabla \cdot \boldsymbol{a} \in L_\infty(\Omega), \boldsymbol{a} \in L_\infty(\Omega)^{n_d}. \tag{1.4}$$

As we have just mentioned at the beginning of the section, we consider here three different weak formulations. We will denote them weak formulations 0, 1/2 and 1. Before starting with the first weak form, let us assume for the sake of clarity that we prescribe a homogeneous Dirichlet condition on Γ_D; the generalization to any Dirichlet data in $H^{1/2}(\Gamma_D)$ is straightforward.

0-weak formulation. The first weak formulation we study is derived by integrating by parts only the diffusion term. The weak formulation reads: find $u \in V$ such that

$$a^0(u,v) = \langle g^0, v\rangle_{\Gamma_N} + \langle f, v\rangle_\Omega \qquad \forall\, v \in V,$$

where

$$a^0(w,v) := \varepsilon(\nabla w, \nabla v) + (\boldsymbol{a}\cdot\nabla w, v) + ([s + \nabla\cdot\boldsymbol{a}]w, v), \tag{1.5}$$

and

$$g^0 := \varepsilon\frac{\partial u}{\partial n}$$

is known on Γ_N.

The prescription of g^0 is the natural boundary condition and stems only from the integration by parts of the diffusion term. At the differential level, this condition corresponds to a Neumann prescription, i.e. the prescription of the flux.

1/2-weak formulation. The derivation of the second weak form is more subtle. We first note that according to the Gauss theorem (also known as divergence theorem), we have that for any $w, v \in V$,

$$\begin{aligned}\int_\Omega \nabla\cdot(\boldsymbol{a}wv)\,d\Omega &= \int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})wv\,d\Gamma \\ &= \int_\Omega(\nabla\cdot\boldsymbol{a})wv\,d\Omega + \int_\Omega(\boldsymbol{a}\cdot\nabla v)w\,d\Omega + \int_\Omega(\boldsymbol{a}\cdot\nabla w)v\,d\Omega.\end{aligned} \tag{1.6}$$

Therefore, the convective term can be rewritten as

$$\begin{aligned}\int_\Omega(\boldsymbol{a}\cdot\nabla w)v\,d\Omega &= \frac{1}{2}\int_\Omega(\boldsymbol{a}\cdot\nabla w)v\,d\Omega + \frac{1}{2}\int_\Omega(\boldsymbol{a}\cdot\nabla w)v\,d\Omega \\ &= \frac{1}{2}\int_\Omega(\boldsymbol{a}\cdot\nabla w)v\,d\Omega + \frac{1}{2}\int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})wv\,d\Gamma \\ &\quad -\frac{1}{2}\int_\Omega(\boldsymbol{a}\cdot\nabla v)w\,d\Omega - \frac{1}{2}\int_\Omega(\nabla\cdot\boldsymbol{a})wv\,d\Omega,\end{aligned}$$

and integrating by parts the diffusion term and only half of the convective term as shown by the latter identity, the 1/2-weak formulation reads: find $u \in V$ such that

$$a^{1/2}(u,v) = \langle g^{1/2}, v\rangle_{\Gamma_N} + \langle f, v\rangle_\Omega \qquad \forall\, v \in V,$$

where

$$a^{1/2}(w,v) := \varepsilon(\nabla w, \nabla v) + \frac{1}{2}(\boldsymbol{a}\cdot\nabla w, v) - \frac{1}{2}(w, \boldsymbol{a}\cdot\nabla v) + ([s + \frac{1}{2}\nabla\cdot\boldsymbol{a}]w, v), \tag{1.7}$$

and

$$g^{1/2} := \varepsilon\frac{\partial u}{\partial n} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n})u$$

is known on Γ_N. At the differential level, the prescription of $g^{1/2}$ is a Robin condition with Robin coefficient $\alpha_r = -\frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n})$.

1-weak formulation. The last weak form is derived by integrating by parts both the diffusion term and the convective term, using Equation (1.6). We can easily show that the weak formulation of the problem thus reads: find $u \in V$ such that

$$a^1(u,v) = \langle g^1, v\rangle_{\Gamma_N} + \langle f, v\rangle_\Omega \qquad \forall\, v \in V,$$

where

$$a^1(w,v) := \varepsilon(\nabla w, \nabla v) - (w, \boldsymbol{a}\cdot\nabla v) + (sw, v), \tag{1.8}$$

and

$$g^1 := \varepsilon\frac{\partial u}{\partial n} - (\boldsymbol{a}\cdot\boldsymbol{n})u$$

is known on Γ_N. In this case, the natural condition corresponds to a Robin condition at the differential level, with Robin factor $\alpha_r = -(\boldsymbol{a}\cdot\boldsymbol{n})$, as defined in Equation (1.2).

We notice that we can write all the weak formulations derived earlier in the same way, introducing a constant $\flat$ which can take the following values:

$$\begin{aligned} &\text{0-Weak formulation:} && \flat = 0,\\ &\text{1/2-Weak formulation:} && \flat = 1/2,\\ &\text{1-Weak formulation:} && \flat = 1. \end{aligned}$$

Owing to this definition, all the weak formulations read: find $u \in V$ such that

$$a^\flat(u,v) = \langle g^\flat, v\rangle_{\Gamma_N} + \langle f, v\rangle_\Omega \qquad \forall\, v \in V, \tag{1.9}$$

where

$$a^\flat(w,v) := \varepsilon(\nabla w, \nabla v) + (1-\flat)(\boldsymbol{a}\cdot\nabla w, v) - \flat(w, \boldsymbol{a}\cdot\nabla v) + ([s + (1-\flat)\nabla\cdot\boldsymbol{a}]w, v), \tag{1.10}$$

and

$$g^\flat := \varepsilon\frac{\partial u}{\partial n} - \flat(\boldsymbol{a}\cdot\boldsymbol{n})u \tag{1.11}$$

is known on Γ_N. Although strictly algebraically equivalent, these formulations have two major differences: their respective natural conditions and their variational properties. The former has already appeared explicitly, as the natural boundary conditions are given by Equation (1.11). The latter becomes evident when we study the existence and uniqueness of the solutions. That is precisely what we are going to do next.

From Lax-Milgram lemma, problem (1.9) has a unique solution if $a(w,v)$ is both continuous and coercive for any $w, v \in V$. Applying Cauchy -Schwartz inequality, we have for $\flat = 0, 1/2, 1$

$$|a^\flat(w,v)| \le M\|w\|_1\|v\|_1 \qquad \forall\, w, v \in V,$$

with

$$M = \varepsilon + \|\boldsymbol{a}\|_{\infty,\Omega} + \|s + (1-\flat)\nabla\cdot\boldsymbol{a}\|_{\infty,\Omega}.$$

According to hypothesis (1.4), M is bounded which implies that $a^\flat$ is continuous for $\flat = 0, 1/2, 1$. We now study the coercivity of $a^\flat$. Owing to the definition of $a^\flat$ given by Equation (1.10),

$$a^\flat(v,v) = \varepsilon\|\nabla v\|^2_{0,\Omega} + (1-2\flat)\int_\Omega (\boldsymbol{a}\cdot\nabla v)v\,d\Omega + \int_\Omega (s + (1-\flat)\nabla\cdot\boldsymbol{a})v^2\,d\Omega \qquad \forall\, v \in V. \tag{1.12}$$

Using Equation (1.6), we have

$$\int_\Omega (\boldsymbol{a}\cdot\nabla v)v\,d\Omega = \frac{1}{2}\int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})v^2\,d\Gamma - \frac{1}{2}\int_\Omega(\nabla\cdot\boldsymbol{a})v^2\,d\Omega.$$

Substituting this result into Equation (1.12), we obtain

$$a^\flat(v,v) = \varepsilon\|\nabla v\|^2_{0,\Omega} + \frac{1-2\flat}{2}\int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})v^2\,d\Gamma + \int_\Omega (s + \frac{1}{2}\nabla\cdot\boldsymbol{a})v^2\,d\Omega \qquad \forall\, v \in V.$$

Assuming that

$$s + \frac{1}{2}\nabla\cdot\boldsymbol{a} \geq 0 \qquad \text{almost everywhere,} \tag{1.13}$$

we find

$$a^\flat(v,v) \geq \varepsilon\|\nabla v\|^2_{0,\Omega} + \frac{1-2\flat}{2}\int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})v^2\,d\Gamma \qquad \forall\, v \in V.$$

Using the Poincaré -Friedrichs inequality, we have

$$\|v\|^2_{0,\Omega} \leq C_\Omega\|\nabla v\|^2_{0,\Omega} \qquad \forall\, v \in V,$$

and we find

$$a^\flat(v,v) \geq \frac{\varepsilon}{1+C_\Omega}\|v\|^2_1 + \frac{1-2\flat}{2}\int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})v^2\,d\Gamma \qquad \forall\, v \in V. \tag{1.14}$$

We are now left with an annoying term involving the advection. Note that for pure Dirichlet problems, this term disappears and the coercivity follows. If we consider the 1/2-weak formulation, this term cancels and the coercivity is shown. However for the 0 and 1-weak formulations, we must assume further hypothesis on the data, i.e. on the relative size of the advection vector $\boldsymbol{a}$ with respect to the diffusion coefficient and/or on its direction. Let us study the 0-weak formulation; letting $\flat = 0$, we have that

$$a^0(v,v) \geq \frac{\varepsilon}{1+C_\Omega}\|v\|^2_1 + \frac{1}{2}\int_{\Gamma_N}(\boldsymbol{a}\cdot\boldsymbol{n})v^2\,d\Gamma \qquad \forall\, v \in V.$$

For the second term to be positive, we must assume that $\boldsymbol{a}\cdot\boldsymbol{n} \geq 0$ on Γ_N, i.e. that Γ_N is an outflow. Conversely, for the 1-weak formulation, we would require Γ_N to be an inflow. This is a

condition on the direction of the flow on Γ_N. On the other hand, if the direction of the flow is not as required to show the coercivity, or if it changes along Γ_N, we can still derive a very simple estimate by noting that the last equation gives

$$a^{\flat}(v,v) \geq \frac{\varepsilon}{1+C_\Omega}\|v\|_1^2 - \left|\frac{1-2\flat}{2}\right| \|\boldsymbol{a}\cdot\boldsymbol{n}\|_{\infty,\Gamma_N}\|v_{|\Gamma_N}\|_{0,\Gamma_N}^2 \qquad \forall\, v \in V. \tag{1.15}$$

In addition, knowing that $H^{1/2}(\Gamma_N) \hookrightarrow L^2(\Gamma_N)$, we have that

$$\|v_{|\Gamma_N}\|_{0,\Gamma_N}^2 \leq C\,\|v_{|\Gamma_N}\|_{1/2,\Gamma_N}^2$$

where C is a constant. Using the trace inequality (1.3), last equation gives

$$\|v_{|\Gamma_N}\|_{0,\Gamma_N}^2 \leq C\,C^*\|v\|_1^2.$$

Substituting this last result into Equation (1.15), we finally find

$$a^{\flat}(v,v) \geq N\|v\|_1^2 \qquad \forall\, v \in V,$$

with

$$N = \frac{\varepsilon}{1+C_\Omega} - C\,C^*\frac{|1-2\flat|}{2}\|\boldsymbol{a}\cdot\boldsymbol{n}\|_{\infty,\Gamma_N}.$$

The condition for coercivity is $N > 0$; by introducing the constant C' to absorb all the constants of the latter equation, we then require that

$$\varepsilon > C'|1-2\flat|\,\|\boldsymbol{a}\cdot\boldsymbol{n}\|_{\infty,\Gamma_N} \tag{1.16}$$

which implies that $\|\boldsymbol{a}\cdot\boldsymbol{n}\|_{\infty,\Gamma_N}$ should not be too high with respect to the diffusion ε. If the condition $N > 0$ is satisfied, Problem (1.9) has a unique solution.

Let us sum up the assumptions we have stated up to now in order to prove the uniqueness of the solution.

1. The data of the problem are such that $s \in L_\infty(\Omega), \boldsymbol{a} \in L_\infty(\Omega)^{n_d}$;
2. the source term and advection satisfy $s + \frac{1}{2}\nabla\cdot\boldsymbol{a} \geq 0$ almost everywhere;
3. Finally, we require that
   - 0-weak formulation: Γ_N is an outflow or the advection is not too high in the sense of Equation (1.16);
   - 1/2-weak formulation: no additional condition;
   - 1-weak formulation: Γ_N is an inflow or the advection is not too high in the sense of Equation (1.16).

We can easily see the great advantage that the weak 1/2-formulation has over the other two: apart from condition (1.13), no condition on the magnitude and direction of $\boldsymbol{a}$ is required on Γ_N. In Chapter 4, we will see how this property can be used for designing efficient mixed domain decomposition methods.

Remark 1.1. Although up to now we have considered only the natural conditions of the weak form as possible boundary conditions, we can always transform the bilinear form in order to accommodate the formulation to our need. For example, considering the 1/2-weak formulation, part of the contour integral on the right-hand-side can be recast to the left-hand side. Let us divide Γ_N into two parts Γ_{N_1} and Γ_{N_2} such that $\Gamma_N = \Gamma_{N_1} \cup \Gamma_{N_2}$, and find $u \in V$ such that

$$a^{1/2}(u,v) + \langle \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n})u, v\rangle_{\Gamma_{N_2}} = \langle g_1^{1/2}, v\rangle_{\Gamma_{N_1}} + \langle g_2^{1/2}, v\rangle_{\Gamma_{N_2}} + \langle f, v\rangle_\Omega \qquad \forall\, v \in V,$$

where $a^{1/2}$ is defined as in Equation (1.7) and

$$g_1^{1/2} := \varepsilon\frac{\partial u}{\partial n} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n})u,$$
$$g_2^{1/2} := \varepsilon\frac{\partial u}{\partial n},$$

are known on Γ_{N_1} and Γ_{N_2}, respectively. Therefore, we have to prescribe a Robin condition on Γ_{N_1} and a classical Neumann condition on Γ_{N_2}. If in addition Γ_{N_1} is an outflow $\boldsymbol{a}\cdot\boldsymbol{n} \geq 0$ on Γ_{N_1} and the bilinear form is coercive. In the same way, we could create a Robin condition from the 0-weak formulation by adding both to the left and right hand side a Robin like term $\alpha_r u$ where α_r could be a constant; for example, we propose to find $u \in V$ such that

$$a^0(u,v) + \langle \alpha_r u, v\rangle_{\Gamma_N} = \langle g^0, v\rangle_{\Gamma_N} + \langle f, v\rangle_\Omega \qquad \forall\, v \in V,$$

where

$$g^0 := \varepsilon\frac{\partial u}{\partial n} + \alpha_r u,$$

is the Robin condition, known on Γ_N.

1.1.4 Finite Element formulation

Galerkin formulation

For the sake of clarity, we will consider here the following pure Dirichlet problem:

$$\begin{cases} Lu := -\varepsilon\Delta u + \boldsymbol{a}\cdot\nabla u + su &= f & \text{in } \Omega, \\ u &= 0 & \text{on } \partial\Omega, \end{cases} \tag{1.17}$$

where we have taken $\nabla\cdot\boldsymbol{a} = 0$, and $s > 0$. We now study the existence and uniqueness of a weak solution to the latter problem, using the Galerkin method, and present a very simple error estimate. This estimate will justify the need for a stabilization technique under well-known circumstances and this is the subject of the following section. Then we introduce a discontinuity capturing technique. Finally, the problem is generalized to transient situations. For the study of the same problem with more general boundary conditions, see for example [3].

Let $\{\Omega^e\}$ be a regular finite element partition of the domain Ω, with index e ranging from 1 to the number of elements n_e. The diameter of $\{\Omega^e\}$ will be denoted by h as usual. Let us construct the functional linear subspace $V_h^0 \subset V^0$ from the previous partition: the resulting finite element

approximation is said to be *conforming*. The discrete Galerkin formulation of the problem consists in finding $u_h \in V_h^0$ such that

$$a(u_h, v_h) = \langle f, v_h \rangle_\Omega \qquad \forall\, v_h \in V_h^0, \tag{1.18}$$

where

$$a(w, v) := \varepsilon(\nabla w, \nabla v) + (\boldsymbol{a} \cdot \nabla w, v) + (sw, v). \tag{1.19}$$

Note that in this particular case for which we only impose Dirichlet conditions ($\Gamma_N = \emptyset$), the 0, 1/2 and 1-weak formulations are identical is the sense that they all have the same continuity and coercivity constants. We have:

$$\begin{aligned} &a)\ |a(w_h, v_h)| \le M \|w_h\|_1 \|v_h\|_1 && \forall\, w_h, v_h \in V_h, \\ &b)\ a(v_h, v_h) \ge N \|v_h\|_1^2 && \forall\, v_h \in V_h, \end{aligned}$$

with

$$\begin{aligned} M &= \varepsilon + \|\boldsymbol{a}\|_{\infty,\Omega} + \|s\|_{\infty,\Omega}, \\ N &= \frac{\varepsilon}{1 + C_\Omega}. \end{aligned}$$

According to Lax-Milgram lemma, Equation (1.18) has a unique solution $u_h \in V_h^0$. In addition, we can derive the following error estimate

$$\|u - u_h\|_1 \le \frac{M}{N} \inf_{v_h \in V_{h,0}} \|u - v_h\|_1$$

where u is the solution of the problem

$$a(u, v) = \langle f, v \rangle_\Omega \qquad \forall\, v \in V_0.$$

The latter error estimate simply states that the finite element solution is the best approximation over all possible $v_h \in V_h$. Now let us introduce m, the degree of the polynomials used in the finite element discretization. Under regularity assumption on the domain Ω, its boundary, and assuming the solution u is smooth enough (see for example [3] for the details), we can show in addition that

$$\|u - u_h\|_1 \le c\, \frac{M}{N} h^m \|u\|_{m+1}, \tag{1.20}$$

where h is the maximum diameter of the polyhedron of the triangulation, and c is a constant depending on the geometry and triangulation of Ω, but not on h. Error estimate (1.20) is optimal in the H^1 norm, so we conclude that the Galerkin method can lack stability when $M \gg N$, that is, when the diffusion ε is small compared to $\|\boldsymbol{a}\|_{\infty,\Omega}$ and $\|s\|_{\infty,\Omega}$ and if h is not sufficiently small. In fact, taking $u = v = u_h$ in Equation (1.19), we have

$$a(u_h, u_h) = \varepsilon \|\nabla u_h\|_0^2 + \|s^{1/2} u_h\|_0^2,$$

as the convective term disappears when it is integrated by parts. We observe that we have no control on the advective term of the equation. In addition, when $s^{1/2}$ is high, we gain control on the L_2 norm of the unknown at the expense of loosing control on its gradient. This is why a stabilization method is necessary.

Stabilization strategy

We now present briefly the historical context of the stabilization technique used in this work, the variational subgrid scale model. For a detailed review of these methods, see for example [4] and the complete volume [5]. Before starting, let us mention that we will not mention techniques such as the virtual bubbles [6], as they do not explicitly belong to the stabilization family considered here (although they are closely related). We will neither mention the Characteristic-Galerkin formulation [7] and the Taylor-Galerkin [8] method, as they are devised starting from a transient equation.

The first stabilization methods that were developed were called *artificial viscosity* methods [9]. They consist in adding a viscosity-like term to the equation, a numerical viscosity, as follows:

$$a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_\Omega \tau(\nabla u_h \cdot \nabla v_h)\, d\Omega = 0,$$

where τ is the stability parameter. It must be proportional to h and the norm of the advection field for two reasons: the stability term must vanish when h goes to zero, and it must be higher and higher as the advection increases. These methods are not consistent, i.e. the exact solution to the original problem does not satisfy the equation of the perturbed one. The diffusion is added isotropically so the results can be over-diffusive, particularly in the crosswind direction. They deteriorate the rate of convergence, as they were originally first order methods, although more precise artificial viscosity methods have been developed since then to increase the order of convergence (see for example [10]).

To correct the indiscrimination of the artificial viscosity, a streamline upwind was introduced, first in the finite difference context and next introduced in the finite element context in [11]. They lead to the following formulation:

$$a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_\Omega \tau(\boldsymbol{a} \cdot \nabla u_h)(\boldsymbol{a} \cdot \nabla v_h)\, d\Omega = 0,$$

where τ is a stability parameter to be determined. Although less diffusive, this method shared the non-consistency of artificial viscosity methods. The consistent version which was called SUPG (standing for streamline-upwind Petrov-Galerkin) was presented in [12]; this time the stabilization term was added at the element level. Let us define

$$\int_{\Omega'} := \sum_{e=1}^{n_e} \int_{\Omega_e},$$

where Ω_e is the interior of element e of the partition. The weak form of the consistent SUPG method is

$$a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_{\Omega'} \tau_e(\boldsymbol{a} \cdot \nabla v_h)(Lu_h - f)\, d\Omega = 0,$$

where τ_e is an element-wise stability parameter. Although it was clear where and how the SUPG method acted as a stability method, no intuitive and perceptible interpretation had been found. It was even thought that the upwind was necessary in order to be consistent with the hyperbolic character of the equation when the advection was dominant, as information can only move downstream. Actually, the concrete interpretation of stabilization methods would only come 15 years after... as we will soon see.

Simultaneously with their work on the advection-diffusion systems of equations, the Stanford team [13] devised a Petrov-Galerkin formulation for solving the Stokes problem which avoided the need to satisfy the Babuška-Brezzi (BB) stability condition, by adding a new perturbation to the continuity equation proportional to the pressure gradient test function. Following the same ideas, the Galerkin/Least-square (GLS) method was presented in [14] as a "conceptual simplification" and generalization of the SUPG method for advection-diffusion equations. The GLS formulation reads:

$$a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_{\Omega'} \tau_e L v_h (L u_h - f)\, d\Omega = 0,$$

where τ_e is still the stability parameter. By the same time, Douglas and Wang [15] developed a stabilization technique (known as Douglas-Wang method) for the Stokes problem, similar to that developed in [13] with the only difference being the sign of the Laplacian of the perturbation function. The method was soon applied to the advection-diffusion reaction in [16] and showed better stability than the GLS method; the authors found that the perturbation function should no longer be the differential operator L but minus its adjoint L^*. The new formulation reads:

$$a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_{\Omega'} \tau_e (-L^* v_h)(L u_h - f)\, d\Omega = 0,$$

where τ_e is the stability parameter. In the case of the ADR equation under study the adjoint operator is given by

$$L^* u = -\varepsilon \Delta u - \boldsymbol{a} \cdot \nabla u + s u.$$

All the methods presented up to now, i.e. SUPG, GLS and the Douglas-Wang method, involve a stabilization parameter τ_e, and none of these methods introduces its value naturally. Actually, τ_e is calculated using convergence analysis, and/or adjusting its value to obtain exact nodal solution for some simple problems. τ_e having the units of time, it is generally called *the intrinsic time*. Referring to the stabilization parameters τ_i of a system of advection-diffusion, Hughes and Mallet[17] wrote:

> *The τ_i are intrinsic time scales of the various components of the solution. In the advection-dominated limit, the τ_i represent the transit times for information to be advected over a distance equal to one-half the element length. These times are reduced by the presence of diffusion. The τ_i approach zero as diffusion begins to dominate in keeping with the instantaneous propagation time of diffusive phenomena (...)*

Recently, Hughes [18] finally clarified the matter by introducing the idea of multiscales. The numerical instabilities of the Galerkin method are due to the unresolved space scales, i.e. the scales that are"smaller" than the element size. Therefore in some way the effects of the unresolved scales must be modelled at the resolved level. This method explains not only the instabilities; it also identifies clearly the intrinsic time. The unknown u is decomposed into a resolved scale $\overline{u}$ and an unresolved scale u' such that $u = \overline{u} + u'$. The first approximation of the unresolved scales [18] consists in solving a Green's function problem for these subgrid scales; in addition, his study leads to an analytical expression for τ_e. The resulting method is called algebraic subgrid scale model (ASGS). In order to avoid confusion with large eddy simulation (LES) turbulence model, this approach is often referred to as variational subgrid scale model. In [19], Codina pointed out that the subgrid scales should be logically sought in the space orthogonal to that of the resolved scales; his formulation leads to a more complicated formulation, but less diffusive, and "more precise"

in some sense. This method is referred to as orthogonal subgrid scale method (OSS). In [20], he also developed a first physically based expression for τ_e, basing its derivation on a Fourier space analysis. This is the expression for τ_e used in this work. The final formulation reads: find $u \in V_h$ such that

$$a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_{\Omega'} \tau_e(-L^* v_h)(Lu_h - f)\, d\Omega = 0 \qquad \forall\, v_h \in V_h,$$

where τ_e is given by

$$\tau_e = \left(c_1 \frac{\varepsilon}{h_e^2} + c_2 \frac{|\boldsymbol{a}|_\infty}{h_e} + s \right)^{-1}, \tag{1.21}$$

where h_e is the characteristic length of the element e, $|\boldsymbol{a}|_\infty$ is the maximum of the Euclidean norm of $\boldsymbol{a}$ in element e, and the constants are $c_1 = 4$ and $c_2 = 2$. For quadratic elements, h is taken as half of the element size. Codina [19] obtained the following error estimate for the method

$$|||u - u_h||| \leq C(\varepsilon^{1/2} h^m + s^{1/2} h^{m+1} + |\boldsymbol{a}|_\infty^{1/2} h^{m+1/2}),$$

where C is a constant independent of h and $|||\cdot|||$ is defined as

$$|||v||| := \varepsilon^{1/2} \|\nabla v\|_0 + \|\tilde{s}^{1/2} v\| + \|\tau_e^{1/2} \boldsymbol{a} \cdot \nabla v\| \qquad \forall\, v_h \in V_h,$$

where $\tilde{s}$ is a modified reaction term defined as

$$\tilde{s} = s - \tau_e s^2,$$

which, form the expression of τ_e given by Equation (1.21), cannot be negative. We notice also that the asymptotic behavior of $\tilde{s}$ as s tends to infinity enables not to loose control over the gradient. In addition, the third term of the equation is the stabilization term that enables us to gain control over the gradient. However, we see that this control is only obtained in the streamline direction, this is a reason why a discontinuity capturing could be needed.

Discontinuity capturing

The use of the subgrid scale model does not avoid the local oscillations present near sharp layers. In the finite element context, Hughes *et al.* [21] developed a first discontinuity capturing technique (also called shock capturing technique). Noting that the oscillations appeared in the direction normal to the gradient of the transported quantity, precisely where the ASGS stabilization does not act, they introduced an additional dissipation in this *crosswind* direction. Codina [22] designed a crosswind diffusion based on the study of the discrete maximum principle. The method is an anisotropic discontinuity capturing technique, referred to as ADC. For simple cases, the method leads to a monotonicity preserving scheme; the resulting algorithm is non-linear and reads: find $u_h \in V_h$ such that

$$\begin{aligned} &a(u_h, v_h) - \langle f, v_h \rangle_\Omega + \int_{\Omega'} \tau_e(-L^* v_h)(Lu_h - f)\, d\Omega \\ &\quad + \int_{\Omega'} \left[k_e \nabla v_h \cdot \nabla u_h + (H(k_e - k_e') - k_e) \nabla v_h \cdot \left(\frac{\boldsymbol{a} \otimes \boldsymbol{a}}{|\boldsymbol{a}|^2} \right) \cdot \nabla u_h \right] d\Omega = 0 \qquad \forall\, v_h \in V_h, \end{aligned}$$

where $H(x)$ is the Heaviside function defined as

$$H(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{elsewhere,} \end{cases} ;$$

k_e' is the additional diffusion of the SUPG-like term of the SGS method, given by

$$k_e' = \tau_e |\boldsymbol{a}|^2$$

k_e is the diffusion of the shock capturing method defined as

$$k_e = \begin{cases} \dfrac{1}{2} \alpha_e h_e \dfrac{|Lu_h - f|}{|\nabla u_h|} & \text{if } |\nabla u_h| \neq 0, \\ 0 & \text{elsewhere.} \end{cases} .$$

In the latter expression, h_e is the element characteristic length, and α_e is given by

$$\alpha_e = \max\left(0, C - \frac{2\varepsilon}{|\boldsymbol{a}_\parallel| h_e}\right) \tag{1.22}$$

where C depends on the type of element used ($C = 0.7$ for linear elements and $C = 0.35$ for quadratic elements). Finally, $\boldsymbol{a}_\parallel$ is obtained by:

$$\boldsymbol{a}_\parallel = \frac{Lu_h - f}{|\nabla u_h|^2} \nabla u_h .$$

Let us take a close look at the expression of the discontinuity capturing. First of all, the term proportional to $\boldsymbol{a} \otimes \boldsymbol{a}$ avoids adding twice the numerical diffusion in the streamline direction, as it is already provided by the SUPG term of the ASGS model. This justifies the name anisotropic shock-capturing. The diffusion introduced by the ADC method is logically proportional to the residual and this makes the method consistent. Finally, the expression for α_e given by Equation (1.22) was obtained in order to satisfy the maximum discrete principle for some particular examples.

1.1.5 Transient problem

We now want to solve the following transient problem in Ω, from time 0 to time T:

$$\begin{cases} \partial_t u + L(t)u = f(t) & \text{in } \Omega \times (0,T), \\ u = 0 & \text{on } \partial\Omega \times (0,T), \\ u = u_0 & \text{in } \Omega \times \{0\}, \end{cases} \tag{1.23}$$

where operator L is defined by Equation $(1.17)_1$, and could depend explicitly on time, as well as the force term f.

The generalized trapezoidal rule

The time discretization is carried out using the generalized trapezoidal rule, i.e. a finite difference scheme. Using such a rule, the weak formulation of the transient problem can be equivalently obtained by first deriving the weak formulation of Problem (1.23) and then discretizing it in time,

or first by performing the time discretization directly on Problem (1.23) before deriving the weak formulation. For the sake of clarity, we choose the last option. Let us introduce a uniform partition of the time interval $[0, T]$ and define

$$\begin{aligned} u^{n+\theta} &:= \theta u^{n+1} + (1-\theta)u^n, \\ \delta t &:= t^n - t^{n-1}, \\ \delta_t u^{n+\theta} &:= \frac{u^{n+\theta} - u^n}{\theta \delta t}. \end{aligned}$$

where δt is the time step size and superscript n denotes the approximated solution at time $n\delta t$. According to this integration rule, we have to solve the following equation for the unknown $u^{n+\theta}$:

$$\delta_t u^{n+\theta} + L^{n+\theta} u^{n+\theta} = f^{n+\theta}, \tag{1.24}$$

from which we compute the solution at time step $n+1$ as

$$u^{n+1} = u^n + \frac{u^{n+\theta} - u^n}{\theta}. \tag{1.25}$$

In Equation (1.24), L and f have to be calculated at time $(n+\theta)\delta t$, as indicated by the superscript notation. The stabilization techniques, ASGS as well as ADC, can be applied directly to the latter system by replacing the residual $(Lu_h - f)$ by $(\delta_t u_h^{n+\theta} + L^{n+\theta} u_h^{n+\theta} - f^{n+\theta})$ in their respective expressions. We now describe two practical integrations, namely the backward Euler and Crank-Nicolson scheme. The details of their implementations can be found in [23].

The backward Euler approximation

The backward Euler approximation is obtained by choosing $\theta = 1$. This approximation is of first order in time and is unconditionally stable [24]. The discrete Galerkin formulation form of the problem consists in finding, for each $n \geq 0$, $u_h^{n+1} \in V_h$ such that

$$\begin{aligned} (\delta_t u_h^{n+1}, v_h) + a^{n+1}(u_h^{n+1}, v_h) - \langle f^{n+1}, v_h \rangle_\Omega &= 0, && \forall\, v_h \in V_h, \\ (u_h^0, v_h) &= (u_0, v_h) && \forall\, v_h \in V_h, \end{aligned}$$

where the bilinear form a^{n+1} is given by Equation (1.19) where all its terms are evaluated at time $n+1$.

The Crank-Nicolson approximation

When precise time integration is needed, it may be useful to use a second order scheme. Crank-Nicolson approximation is of second order and corresponds to the choice $\theta = 1/2$. The discrete Galerkin formulation form of the problem consists in finding, for each $n \geq 0$, $u_h^{n+1/2} \in V_h$ such that

$$\begin{aligned} (\delta_t u_h^{n+1/2}, v_h) + a^{n+1/2}(u_h^{n+1/2}, v_h) - \langle f^{n+1/2}, v_h \rangle_\Omega &= 0, && \forall\, v_h \in V_h \times (0, T), \\ (u_h^0, v_h) &= (u_0, v_h) && \forall\, v_h \in V_h. \end{aligned}$$

Contrary to the Euler scheme, the solution at time $n+1$ must be explicitly calculated using Equation (1.25). The Crank-Nicolson scheme presents a singularity at $t \to 0$. It is therefore recommended to use the more dissipative backward Euler scheme during the first time iterations.

1.2 Incompressible flow Equations

This section studies the equations of motion of an incompressible fluid, namely the Navier-Stokes equations. We first present the set of equations and introduce each one of the variables and physical properties in play, which are the viscosity, the density, the pressure and the velocity of the fluid. We briefly comment on some results on the existence and uniqueness of a solution to the equations. Then we derive the form of the Navier-Stokes equations in a non-inertial frame of reference which may be required when considering flows including moving objects. At this point, we are ready to present a finite element formulation to solve general incompressible fluid problems.

1.2.1 Navier-Stokes Equations

Newton's second law of motion is the starting point of the derivation of the governing equation for a fluid as a continuum, the Navier-Stokes equations. A detailed derivation of these equations can be found in [25]. They are:

$$\begin{aligned} \rho\,\partial_t \boldsymbol{u} + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - 2\mu\nabla\cdot\boldsymbol{\varepsilon}(\boldsymbol{u}) + \nabla p &= \rho\boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} &= 0, \end{aligned}$$

together with appropriate boundary and initial conditions. $\boldsymbol{u}$ is the velocity of the fluid and p its pressure; μ is the dynamic viscosity and ρ the density; $\boldsymbol{\varepsilon}(\boldsymbol{u})$ is the rate of deformation tensor given by

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^t);$$

$\boldsymbol{f}$ is the vector of body forces (for example, gravity). These equations describe the motion of an incompressible fluid, i.e. a fluid for which the density does not change significantly with the pressure gradients in play. We expect the effects of compression to become important when the velocity of the fluid approach the sound velocity, i.e. the velocity of propagation of the pressure waves. In addition, we assume that the density is constant over the computational domain and is insensitive to pressure variations.

The viscosity μ is a measure of the internal friction of the fluid and consequently depends on the temperature of the fluid. Its mechanical role is to eliminate any local deformation. The value of the viscosity for a specific fluid (gas or liquid) is usually measured experimentally as there do not exist suitable theoretical arguments to derive an expression for general fluids. The expression for the viscous term was first considered by Newton who recognized that in a parallel two dimensional flow, the shear stress should be proportional to the rate of deformation. The interpretation of the velocity $\boldsymbol{u}$ in an Eulerian context is simply the velocity of the fluid point measured at a given position and at a given time. The variable p is the mechanical pressure. If we denote $\boldsymbol{\sigma}$ as the stress tensor, then we have by definition

$$p := -\frac{1}{3}\sigma_{ii}.$$

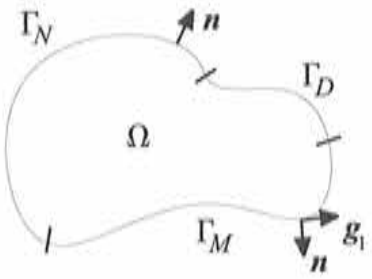


Figure 1.1: Partition of the contour of Ω.

It can be shown that "minus the average of the normal component of the stress on a surface element at a point over all directions of the normal $\boldsymbol{n}$ to the element" is precisely the pressure p, that is

$$-\frac{1}{4\pi}\int \sigma_{ij} n_i n_j d\Omega(\boldsymbol{n}) = -\frac{1}{3}\sigma_{ii} = p,$$

where $d\Omega(\boldsymbol{n})$ is an element of solid angle about $\boldsymbol{n}$. This equation constitutes an intuitive definition of the pressure. It should be pointed out that p is *not* a thermodynamic variable, although it can be related to the thermodynamic pressure in some way (see the discussion in [25]).

The Navier-Stokes equations are solved in a domain Ω of dimension n_{d}, together with appropriate boundary conditions on the contour $\Gamma := \partial\Omega$. For example,

$$\begin{aligned}
\boldsymbol{u} &= \boldsymbol{u}_g && \text{on } \Gamma_D \times (0,T),\\
\boldsymbol{\sigma}\cdot\boldsymbol{n} &= \boldsymbol{t}_n && \text{on } \Gamma_N \times (0,T),\\
\boldsymbol{u}\cdot\boldsymbol{n} = 0,\ \boldsymbol{g}_1\cdot\boldsymbol{\sigma}\cdot\boldsymbol{n} &= t_1, && \\
\boldsymbol{g}_2\cdot\boldsymbol{\sigma}\cdot\boldsymbol{n} &= t_2 && \text{on } \Gamma_M \times (0,T),\\
\boldsymbol{u} &= \boldsymbol{u}_0 && \text{on } \Omega \times \{0\},
\end{aligned} \tag{1.26}$$

where $\Gamma = \Gamma_N \cup \Gamma_D \cup \Gamma_M$, $\boldsymbol{n}$ is the outward unit normal (see Figure (1.1)), $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are the unit vectors spanning the space tangent to Γ_M, $t_1 = \boldsymbol{t}_t \cdot \boldsymbol{g}_1$ and $t_2 = \boldsymbol{t}_t \cdot \boldsymbol{g}_2$ are the components of the tangential traction $\boldsymbol{t}_t$ and $\boldsymbol{\sigma}$ is the stress tensor

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\nu\boldsymbol{\varepsilon}(\boldsymbol{u}).$$

The prescription of $\boldsymbol{t}_t$ can be known, for example, from a wall function law if the turbulence equations are to be solved. We assume that all the boundary conditions belong to the appropriate trace spaces.

We now briefly introduce the problem of existence and uniqueness of a solution $(\boldsymbol{u}, p)$ to the Navier-Stokes equations [26]. The first studies were carried out by Jean Leray, a French mathematician, in the 1930's... but the proof of uniqueness of the solution in three dimensions still remains an open problem...

Let us look for a solution in a domain Ω, during a time interval $(0,T)$. We assume that appropriate boundary conditions are imposed, for example $\boldsymbol{u} = 0$ on $\partial\Omega$; initial conditions are

prescribed with some required regularity and assumed to be solenoidal (divergence free); the domain boundary is considered to be sufficiently smooth.

The general framework of study of the existence of a solution is the variational form of the Navier-Stokes equations. The solution we seek is therefore weak, and this enables us to naturally define its spatial and time regularity. In our framework, the solutions $\boldsymbol{u}$ and p we seek are in $H^1(\Omega)$ and $L^2(\Omega)$ respectively, for all time t, and their energy are bounded in the time interval of study in the sense that

$$\int_0^T \|p\|_{0,\Omega}^2 + \int_0^T \|\boldsymbol{u}\|_{1,\Omega}^2 < \infty.$$

For the transient problem, we have the following results:

- for 2D and 3D flows, there exists a solution;
- for 2D flows, this solution is unique;
- for 3D flows, uniqueness is an open problem!

In addition, it can be shown that for 3D flows, if a more-regular-than-necessary solution exists, then it is unique. For the steady problem, we have to invoke the Reynolds number Re:

- for small Re, there exists a unique solution;
- for high Re, there exists at least one solution.

A good example of the existence of multiple stationary laminar solutions is the Couette-Taylor flow [27]. The Couette-Taylor problem studies the flow between two cylinders in relative rotation. It admits a laminar Couette solution at any Reynolds number, and when the Reynolds number is sufficiently large, the flow becomes unstable and admits other stationary solutions. The way that the flow bounces from one flow state to the other depends on the history of the flow. This is a characteristic of non-linear problems.

1.2.2 Non-inertial frame of reference

Before proceeding, we lighten the Navier-Stokes equations by dividing the momentum equation by ρ. We have:

$$\begin{aligned} \partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - 2\nu \nabla \cdot \varepsilon(\boldsymbol{u}) + \nabla p &= \boldsymbol{f}, \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned}$$

where ν is called the kinematic viscosity and p is now the kinematic pressure, i.e. the dynamic pressure divided by the density.

The classical Navier-Stokes equations model flows in inertial (Galilean) frames of reference. When the boundaries of the fluid are accelerated, it may be convenient to solve the Navier-Stokes equations in the frame of reference in which these boundaries remain at rest. We know that the Navier-Stokes equations express Newton's second law, i.e. that the rate of change of momentum of an element of fluid is equal to the sum of the forces acting on it. If the element of fluid is considered in a non-inertial frame of reference, we must therefore add to the original list of forces the ones due to the acceleration of the frame. We are now going to measure the effective acceleration a particle of fluid P is undergoing when its referential is accelerated. Let $\boldsymbol{e}$ be an orthonormal basis of the

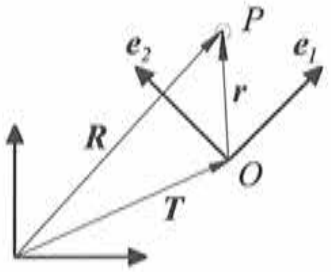


Figure 1.2: Inertial frame to non-inertial frame transformation.

non-inertial frame, and let x the position vector of P, equivalently $\boldsymbol{X}$ in the inertial frame, such that

$$\boldsymbol{x} = x_i \boldsymbol{e}_i,$$

where the x_i's are the coordinates of P. We consider the following transformation:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{T} + \boldsymbol{x} \\ &= \boldsymbol{T} + x_i \boldsymbol{e}_i, \end{aligned}$$

where $\boldsymbol{T}$ is the position of the origin of $\boldsymbol{e}$, as illustrated by Figure (1.2).

As mentioned earlier, we want to calculate the effective acceleration of P. Differentiating twice the latter equation, we obtain

$$\frac{d^2\boldsymbol{X}}{dt^2} = \frac{d^2\boldsymbol{T}}{dt^2} + \frac{d^2 x_i}{dt^2}\boldsymbol{e}_i + 2\frac{dx_i}{dt}\frac{d\boldsymbol{e}_i}{dt} + x_i \frac{d}{dt}\left(\frac{d\boldsymbol{e}_i}{dt}\right). \tag{1.27}$$

The first term of the right-hand side is obviously the acceleration due to the movement of the origin of the non-inertial basis $\boldsymbol{e}$. The second term is the acceleration of P measured in $\boldsymbol{e}$. Let us now investigate the last two terms, involving movements of the non-inertial basis. Given an infinitesimal rotation vector $\delta\boldsymbol{\theta}$, we can express an infinitesimal variation $\delta\boldsymbol{e}_i$ of $\boldsymbol{e}_i$, as

$$\delta\boldsymbol{e}_i = \delta\boldsymbol{\theta} \times \boldsymbol{e}_i,$$

for each i, as shown in Figure (1.3). Dividing last equation by δt, an infinitesimal increment of time, and taking the limit to zero, we have that

$$\frac{d\boldsymbol{e}_i}{dt} = \frac{d\boldsymbol{\theta}}{dt} \times \boldsymbol{e}_i.$$

We now define the angular velocity $\boldsymbol{\omega}$ and linear acceleration $\boldsymbol{a}_{\text{ref}}$ vectors such that

$$\begin{aligned} \boldsymbol{\omega} &= \frac{d\boldsymbol{\theta}}{dt}, \\ \boldsymbol{a}_{\text{ref}} &= \frac{d^2\boldsymbol{T}}{dt^2}. \end{aligned}$$

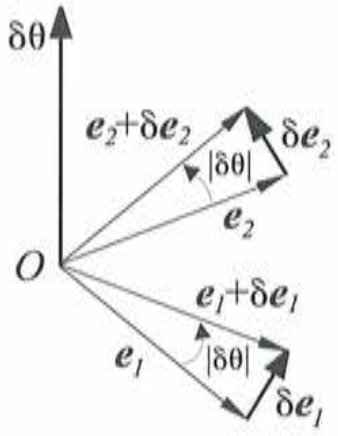


Figure 1.3: Infinitesimal rotation of the non-inertial basis.

Owing to the latter three equalities, Equation (1.27) gives

$$\begin{aligned}\frac{d^2\boldsymbol{X}}{dt^2} &= \boldsymbol{a}_{\text{ref}} + \frac{d^2x_i}{dt^2}\boldsymbol{e}_i + 2\frac{dx_i}{dt}\frac{d\boldsymbol{e}_i}{dt} + x_i\frac{d}{dt}(\boldsymbol{\omega}\times\boldsymbol{e}_i)\\ &= \boldsymbol{a}_{\text{ref}} + \frac{d^2x_i}{dt^2}\boldsymbol{e}_i + 2\boldsymbol{\omega}\times\frac{dx_i}{dt}\boldsymbol{e}_i + \frac{d\boldsymbol{\omega}}{dt}\times\boldsymbol{e}_ix_i + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{e}_ix_i).\end{aligned}$$

Introducing the velocity $\boldsymbol{u}$ measured in the non-inertial frame

$$\boldsymbol{u} = \frac{dx_i}{dt}\boldsymbol{e}_i,$$

and remembering that $\boldsymbol{x} = x_i\boldsymbol{e}_i$, we finally find

$$\frac{d^2\boldsymbol{X}}{dt^2} = \boldsymbol{a}_{\text{ref}} + \frac{d\boldsymbol{u}}{dt} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \frac{d\boldsymbol{\omega}}{dt}\times\boldsymbol{x} + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{x}).$$

The effective acceleration of the particle now has the required form. As already stated, the first term is the linear acceleration of the non-inertial basis. The second term is the acceleration measured in the non-inertial frame. The last three terms are due to the rotation of the non-inertial axes: the first one of these rotation terms is the Coriolis force, the second one is the term due to the acceleration of the angular velocity and the last one is the centrifugal force.

We are now ready to write down the Navier-Stokes equations in a non-inertial frame of reference. We must add the relative acceleration to the acceleration of the particle measured in the non-inertial frame of reference. The transient Navier-Stokes equations are

$$\partial_t\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} - 2\nu\nabla\cdot\varepsilon(\boldsymbol{u}) + \nabla p = \boldsymbol{f}, \tag{1.28}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{1.29}$$

where $\boldsymbol{f}$ is the vector of body forces, including the gravitational force, and the non-inertial terms:

$$\boldsymbol{f} = \boldsymbol{g} - \boldsymbol{a}_{\text{ref}} - \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{x}) - \frac{d\boldsymbol{\omega}}{dt}\times\boldsymbol{x}.$$

1.2.3 Dimensionless form

As was done for the ADR equation, the Navier-Stokes equations can be non-dimensionalized. We assume $\boldsymbol{\omega}$ is constant and introduce the Reynolds number Re and the Ekman number Ek as

$$\begin{aligned} \mathrm{Re} &:= \frac{UD}{\nu}, \\ \mathrm{Ek} &:= \frac{\nu}{2|\boldsymbol{\omega}|D^2}, \end{aligned}$$

where D is a characteristic length and U a characteristic velocity. Then, the dimensionless Navier-Stokes equations are:

$$\begin{aligned} \partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \frac{1}{\mathrm{Re}\,\mathrm{Ek}}\,\boldsymbol{\omega}\times\boldsymbol{u} - \frac{2}{\mathrm{Re}}\nabla\cdot\boldsymbol{\varepsilon}(\boldsymbol{u}) + \nabla p &= \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} &= 0, \end{aligned}$$

where $\boldsymbol{u}$, p and $\boldsymbol{f}$ are now dimensionless velocity, pressure and force, respectively. The Reynolds number is a measure of the relative importance of the convective effects and viscous effects, while the Ekman number measures the relative importance of the viscous term and non-inertial term due to rotation. The product (Re Ek) is known as the Rossby number.

1.2.4 Linearization and time discretization

Linearization

Equation (1.30) is non-linear because of the convective term. The linearization of this term can be performed at the continuous level or at the variational level, they are both equivalent. We introduce an iterative scheme, and denote by m the iteration number. We propose the following linearization strategy:

$$[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}]^{m+1} \approx (\boldsymbol{u}^m\cdot\nabla)\boldsymbol{u}^{m+1} + \beta(\boldsymbol{u}^{m+1}\cdot\nabla)\boldsymbol{u}^m - \beta(\boldsymbol{u}^m\cdot\nabla)\boldsymbol{u}^m$$

Taking $\beta = 0$, we obtain the so-called Picard method; taking $\beta = 1$ we obtain the Newton-Raphson method. If convection is not too high, it can be shown that the Picard method converges linearly. If, in addition, the initial solution is not too far from the exact solution, the Newton-Raphson method converges quadratically. See for example [28] for the proofs. An efficient numerical strategy to obtain a converged solution for a Navier-Stokes problem would consist in the following:

1. Solve the Stokes problem: this provides a unique initial solution.

2. Solve a few Picard iterations and take advantage of its robustness.

3. Switch to the Newton-Raphson method to accelerate the rate of convergence.

Time discretization

Let us introduce a uniform partition of the time interval $[0,T]$ and define

$$\begin{aligned} \boldsymbol{u}^{n+\theta} &:= \theta \boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{u}^n, \\ \delta t &:= t^n - t^{n-1}, \\ \delta_t \boldsymbol{u}^{n+\theta} &:= \frac{\boldsymbol{u}^{n+\theta} - \boldsymbol{u}^n}{\theta \delta t}. \end{aligned}$$

where we use the same notation as in the case of the ADR equation. According to this integration rule, the time-discretized Navier-Stokes equations are solved as follows. Given an initial condition $\boldsymbol{u}^0$, find $\boldsymbol{u}^{n+1}$ and p^{n+1} for each $n \geq 0$ such that

$$\begin{aligned} \delta_t \boldsymbol{u}^{n+\theta} + (\boldsymbol{u}^{n+\theta} \cdot \nabla)\boldsymbol{u}^{n+\theta} + 2\boldsymbol{\omega}^{n+\theta} \times \boldsymbol{u}^{n+\theta} - 2\nu \nabla \cdot \boldsymbol{\varepsilon}(\boldsymbol{u}^{n+\theta}) + \nabla p^{n+\theta} &= \boldsymbol{f}^{n+\theta} && \text{in } \Omega, \\ \nabla \cdot \boldsymbol{u}^{n+\theta} &= 0 && \text{in } \Omega, \end{aligned}$$

with the following boundary conditions

$$\begin{aligned} \boldsymbol{u}^{n+\theta} &= \boldsymbol{u}_g && \text{on } \Gamma_D, \\ \boldsymbol{\sigma}^{n+\theta} \cdot \boldsymbol{n} &= \boldsymbol{t}_n && \text{on } \Gamma_N, \\ \boldsymbol{u}^{n+\theta} \cdot \boldsymbol{n} = 0, \ \boldsymbol{g}_1 \cdot \boldsymbol{\sigma}^{n+\theta} \cdot \boldsymbol{n} &= t_1, && \\ \boldsymbol{g}_2 \cdot \boldsymbol{\sigma}^{n+\theta} \cdot \boldsymbol{n} &= t_2 && \text{on } \Gamma_M, \end{aligned}$$

where tha data could depend on time and where each time dependent variable x satisfies

$$x^{n+1} = x^n + \frac{x^{n+\theta} - x^n}{\theta}.$$

1.2.5 Finite element formulation

Let us introduce the following functional spaces:

$$\begin{aligned} V &= \{\boldsymbol{v} \in H^1(\Omega)^{n_d} \,|\, \boldsymbol{v}_{|\Gamma_D} = \boldsymbol{0}, (\boldsymbol{v} \cdot \boldsymbol{n})_{|\Gamma_M} = 0\}, \\ Q &= L^2(\Omega), \\ U &= \{\boldsymbol{v} \in H^1(\Omega)^{n_d} \,|\, \boldsymbol{v}_{|\Gamma_D} = \boldsymbol{u}_g, (\boldsymbol{v} \cdot \boldsymbol{n})_{|\Gamma_M} = 0, \ t \in (0,T)\}, \\ P &= \{p \in L^2(\Omega) \,|\, \int_\Omega p \, d\Omega = 0 \text{ if } \Gamma_N = \emptyset, \ t \in (0,T)\}. \end{aligned}$$

The last space is the functional space for the pressure unknown. If the normal component of the traction is not prescribed anywhere on the contour, then the pressure is only defined up to any additive constant. This is why we require explicitly its average over Ω to be zero. Let us construct the discrete linear subspaces $U_h \subset U$, $P_h \subset P$, $V_h \subset V$ and $Q_h \subset Q$ from a partition of Ω in n_e elements. Let $\boldsymbol{U}_h = [\boldsymbol{u}_h, p_h]^t$ be the vector of nodal unknowns and $\boldsymbol{V}_h = [\boldsymbol{v}_h, q_h]^t$ the associated weight functions vector. We define

$$\delta_t \boldsymbol{U}_h^{n+\theta} = [\delta_t \boldsymbol{u}_h^{n+\theta}, 0]^t.$$

We now present the linearized and time-discretized Galerkin formulation. Thus we introduce the two iteration indices presented earlier, the linearization index m and the time index n, and denote $x^{n+1,m+1}$ any variable considered at linearization step $m+1$ and time step $n+1$. The linearization and time iterations are nested, the linearization obviously being the inner iterative process; for the sake of clarity, we assume we use the Picard linearization. The discrete Galerkin formulation of the problem reads as follows. Given $\boldsymbol{u}_h^{n+\theta,0} = \boldsymbol{u}^n \in U_h$, for each time step $n \geq 0$, find for $m = 0, 1, \ldots$ until convergence, $\boldsymbol{U}_h^{n+1,m+1} \in U_h \times P_h$ such that

$$(\delta_t \boldsymbol{U}_h^{n+\theta,m+1}, \boldsymbol{V}_h) + a^{n+\theta,m}(\boldsymbol{U}_h^{n+\theta,m+1}, \boldsymbol{V}_h) = l^{n+\theta}(\boldsymbol{V}_h), \qquad \forall\, \boldsymbol{V}_h \in V_h \times Q_h, \quad (1.30)$$

where the bilinear form and the force term are evaluated at time $n+\theta$ as

$$\begin{aligned} a^{n+\theta,m}(\boldsymbol{U},\boldsymbol{V}) &= 2\int_\Omega \nu^{n+\theta,m}\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega + \int_\Omega [(\boldsymbol{u}^{n+\theta,m}\cdot\nabla)\boldsymbol{u}]\cdot\boldsymbol{v}\, d\Omega \\ &\quad + \int_\Omega q\,\nabla\cdot\boldsymbol{u}\, d\Omega - \int_\Omega p\,\nabla\cdot\boldsymbol{v}\, d\Omega + 2\int_\Omega (\boldsymbol{\omega}^{n+\theta}\times\boldsymbol{u})\cdot\boldsymbol{v} d\Omega, \\ l^{n+\theta}(\boldsymbol{V}) &= \int_\Omega \boldsymbol{f}^{n+\theta}\cdot\boldsymbol{v}\, d\Omega + \int_{\Gamma_N} \boldsymbol{t}_n\cdot\boldsymbol{v}\, d\Gamma + \int_{\Gamma_M} (t_1\boldsymbol{g}_1 + t_2\boldsymbol{g}_2)\cdot\boldsymbol{v}\, d\Gamma. \end{aligned}$$

The viscosity remains in the integral as it can be directly substituted by the molecular viscosity plus the eddy-viscosity when using a turbulence model.

It is well-known that the latter formulation can lack stability for three major reasons. The first reason is related to the compatibility of the finite element spaces for the velocity and the pressure which have to satisfy the so-called Ladyzhenskaya-Brezzi-Babuška condition [29]. This condition is necessary to obtain a stability estimate for the pressure; without requiring this condition, the pressure would be out of control. The second reason is attributed to the relative importance of the viscous and convective effects. It can be directly related to the instabilities caused by high advection in the case of the ADR equation, as studied in Section 1.1.4. Finally, the third one appears when the Coriolis force becomes important with respect to viscous effects. We will now present a stabilized formulation, based on the ASGS model described in [18]. The method is extensively described in [30] and [31].

The original Navier-Stokes system (1.28),(1.29) can be re-written in a compact form as

$$\delta_t \boldsymbol{U}^{n+\theta} + \boldsymbol{L}^{n+\theta}\boldsymbol{U}^{n+\theta} = \boldsymbol{F}^{n+\theta} \qquad \text{in } \Omega,$$

where $\boldsymbol{L}^{n+\theta}\boldsymbol{U}$ is defined as

$$\boldsymbol{L}^{n+\theta}\boldsymbol{U} := \begin{bmatrix} (\boldsymbol{a}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}^{n+\theta}\times\boldsymbol{u} - 2\nu\nabla\cdot\boldsymbol{\varepsilon}(\boldsymbol{u}) + \nabla p \\ \nabla\cdot\boldsymbol{u} \end{bmatrix},$$

where $\boldsymbol{a} = \boldsymbol{u}$ before linearization, and the force term is defined as

$$\boldsymbol{F}^{n+\theta} = \begin{bmatrix} \boldsymbol{f}^{n+\theta} \\ 0 \end{bmatrix}.$$

The stabilized weak form reads: given $\boldsymbol{u}_h^{n+1,0} \in U_h$, for each time step $n \geq 0$, find for $m = 0, 1, \ldots$

until convergence, $\boldsymbol{U}_h^{n+1,m+1} \in U_h \times P_h$ such that

$$(\delta_t \boldsymbol{U}_h^{n+\theta,m+1}, \boldsymbol{V}_h) + a^{n+\theta,m}(\boldsymbol{U}_h^{n+\theta,m+1}, \boldsymbol{V}_h) + \sum_{e=1}^{n_e} \int_{\Omega_e} (-\boldsymbol{L}^{*n+\theta}\boldsymbol{V}_h)^t \boldsymbol{\tau}_e [\delta_t \boldsymbol{U}_h^{n+\theta,m+1} + \boldsymbol{L}^{n+\theta}\boldsymbol{U}_h^{n+\theta,m+1} - \boldsymbol{F}^{n+\theta}] = l^{n+\theta}(\boldsymbol{V}_h), \quad (1.31)$$

$\forall\, \boldsymbol{V}_h \in V_h \times Q_h$, where the convection is taken from the previous linearization step, i.e.

$$\boldsymbol{a} = \boldsymbol{u}^{n+\theta,m},$$

and where $\boldsymbol{L}^{*n+\theta}$ is the adjoint of $\boldsymbol{L}^{n+\theta}$ given by

$$\boldsymbol{L}^{*n+\theta}\boldsymbol{V} := \begin{bmatrix} -(\boldsymbol{a}\cdot\nabla)\boldsymbol{v} - 2\boldsymbol{\omega}^{n+\theta} \times \boldsymbol{v} - 2\nu\nabla\cdot\boldsymbol{\varepsilon}(\boldsymbol{v}) - \nabla q \\ -\nabla\cdot\boldsymbol{v} \end{bmatrix}.$$

$\boldsymbol{\tau}$ is the matrix of stabilization parameters and is defined in each element as

$$\boldsymbol{\tau}_e = \operatorname{diag}(\tau_1 \boldsymbol{I}, \tau_2), \qquad \text{where}$$

$$\tau_1 = \left(\frac{c_1 \nu}{h_e^2} + \frac{c_2 |\boldsymbol{a}|}{h_e} + c_3 |\boldsymbol{\omega}^{n+\theta}| \right)^{-1}, \quad (1.32)$$

$$\tau_2 = c_4 \frac{h_e^2}{\tau_1}. \quad (1.33)$$

$\boldsymbol{I}$ is the n_d-dimensional identity. τ_2 contributes to enforcing the incompressibility of the flow, which is excessively relaxed by the term multiplied by τ_1. The values of the constants we use are $c_1 = 4$, $c_2 = 2$, $c_3 = 1$, $c_4 = 1$ and h_e is the characteristic element length. For quadratic elements, h_e is taken as half of the element size.

1.2.6 Some finite elements

During this work we will consider two types of element using both equal order interpolation for the velocity and the pressure. The Q1/Q1 element is continuous and bilinear (trilinear in three dimensions) in both velocity and pressure. We will also work with the P1/P1 element, continuous and linear in velocity and pressure. These elements do not satisfy the BB condition and therefore require the use of stabilization.

1.2.7 Examples

Example 1: cavity flow

Through this example, we want to show the importance of the stabilization term involving the incompressibility constraint, i.e. the term multiplied by τ_2. We solve a square cavity flow at a relatively high Reynolds number

$$\mathrm{Re} = \frac{UH}{\nu} = 5\,000,$$

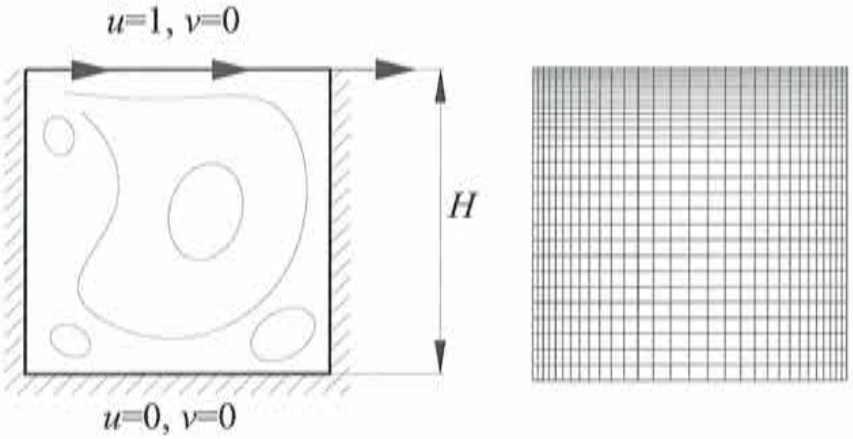


Figure 1.4: Cavity flow. (Left) Geometry and boundary conditions. (Right) Mesh.

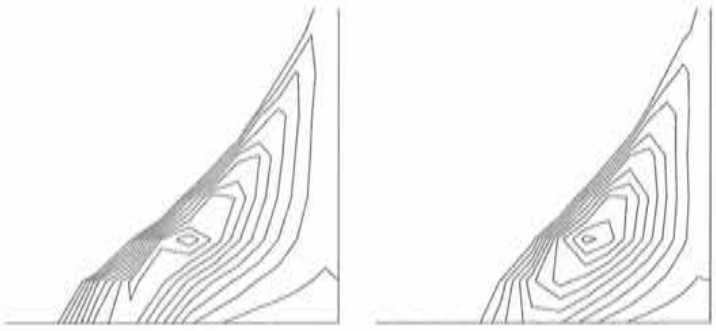


Figure 1.5: Cavity flow. (Left) $c_4 = 0$ in Equation (1.33). (Right) $c_4 = 1$ in Equation (1.33).

where U is the velocity at the top of the cavity and H is the height of the cavity. The geometry and boundary conditions are shown in Figure 1.4 (Left). This problem is solved using 900 Q1/Q1 elements, and the mesh is refined in the upper part of the cavity (see Figure 1.4 (Right)). As sketched in Figure 1.4 (Left), we expect four main vortices at the center of the cavity and at the top left, bottom left and bottom right corners. On way of verifying the fulfillment of the incompressibility is to look at the streamlines. Figure 1.5 (Left) shows some streamlines in the bottom right corner of the cavity taking $\tau_2 = 0$; we observe that some streamlines enter the bottom wall. Figure 1.5 (Right) shows the same streamlines and confirms the improvement achieved on the incompressibility constraint.

Example 2: stirred tank

We now study the importance of the stabilization of the rotation, i.e. the term involving the magnitude of the angular velocity in Equation 1.32. We solve the Stokes problem on the sixth part of a section of a stirred tank, shown in Figure 1.6 (Left), rotating at an angular velocity $\omega = [0, 0, 1]^t$. The radius of the tank and the shaft are 1 and 0.4, respectively. The widths of the

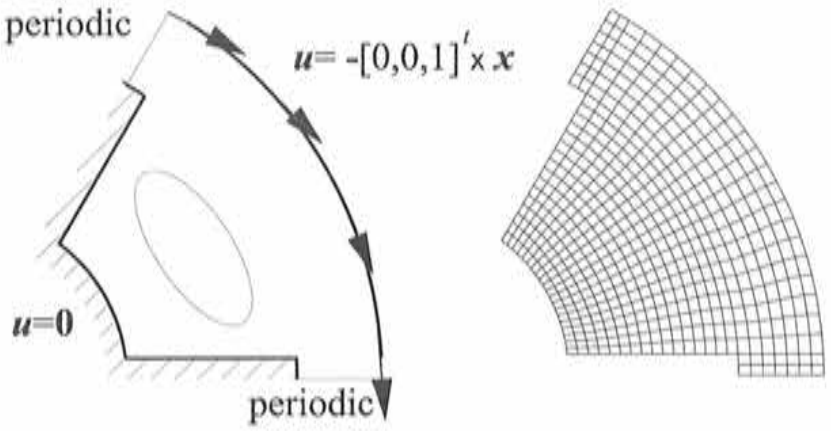


Figure 1.6: Stirred tank. (Left) Geometry and boundary conditions. (Right) Mesh.

blades are 0.1 and their radii are 0.8. The problem is solved in the rotating frame of reference so the velocity is prescribed to $\boldsymbol{u} = -\boldsymbol{\omega} \times \boldsymbol{x}$ on the outer wall and to zero on the blades and shaft. Periodic boundary conditions are imposed on the inflow and outflow of the domain (see [32] for the details on the implementation of periodic boundary conditions). The reactor is meshed with 488 Q1/Q1 elements, as shown in Figure 1.6 (Right). For a rotating two-dimensional flow, the centrifugal force can be absorbed by the pressure term. In fact, we have that

$$\boldsymbol{\omega} \times \boldsymbol{\omega} \times \boldsymbol{x} = -\frac{1}{2}\nabla(|\boldsymbol{\omega} \times \boldsymbol{x}|^2).$$

In addition, we notice that

$$\nabla \times (\boldsymbol{\omega} \times \boldsymbol{u}) = |\boldsymbol{\omega}|\nabla \cdot \boldsymbol{u} = 0,$$

which means that the Coriolis term is the gradient of a function. Therefore, both the centrifugal force and the Coriolis term can be included in the pressure term. This is possible because the flow is confined; if it were not the case, then we could not impose a physical traction on the Neumann contour. Hence, we expect the solution in velocity to be the same whatever the rotation is. This enables us to use the velocity field for $\boldsymbol{\omega} = 0$ as a reference solution.

In order to test the stabilization in a critical situation, we choose $\nu = 10^{-9}$ which gives an Eckman number based on the outer wall radius

$$\mathrm{Ek} = 0.5 \times 10^{-9}.$$

We only present here the results of the velocity module along the symmetry line of the domain. Figure 1.7 shows the improvement in the solution using the stabilization of the rotation. Note that the formulation with $c_3 = 0$ already includes some stabilization of the rotation. However, when using the Galerkin method, the inestability when the Eckman number is very small appears much clearer [33].

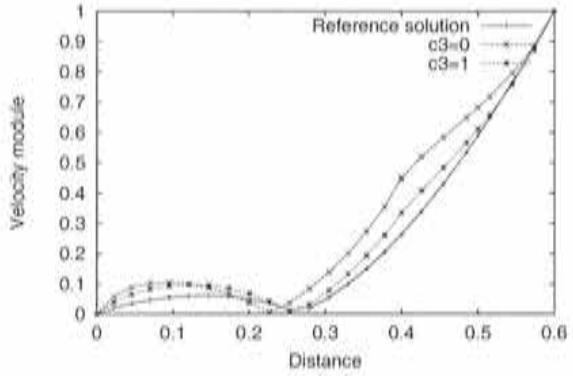


Figure 1.7: Stirred tank. Velocity module on symmetry line for different values of c_3 in Equation (1.32).

1.3 Turbulence Modelling

This section studies a simple approach to turbulence modelling. First, the need for modelling turbulence is justified by physical considerations. The classical statistical method is introduced to simplify the Navier-Stokes equations and prepare them to the introduction of turbulence models known as Reynolds-stress turbulence models. Then we introduce the Boussinesq approximation to model the Reynolds shear stress and present a one-equation turbulence model, namely, the Spalart-Allmaras model, which provides a closure equation to compute for the eddy viscosity. Special attention is paid to the boundary conditions to be imposed on the walls of the computational domain. In particular, we derive a wall function approach to avoid solving the whole boundary layer.

1.3.1 Why model turbulence?

The motion of a fluid is obtained from the principles of mass and energy conservation and the fundamental principle of mechanics, namely Newton's second law. As the Navier-Stokes equations are the mathematical description of such a motion, it is expected that they can describe deterministically the evolution of any fluid, provided its initial characteristics are prescribed. Hence they are able to predict turbulence. This is a simple reason for hoping that the three-dimensional transient solutions of the Navier-Stokes equations are unique.

Direct numerical simulations (DNS) solve the three dimensional and transient Navier-Stokes equations. Obviously, the mesh must be fine enough to capture all the participating scales of the flow, that is from the macroscopic scale (determined by the dimension of the domain) down to the characteristic length scale of viscous dissipation. By performing simple dimensional analysis [34], it can be estimated that the total number of degrees of freedom should be proportional to $\mathrm{Re}^{3/4}$ in each direction, i.e. $\mathrm{Re}^{9/4}$. Knowing that the time step size should in its turn be proportional to the mesh size, we obtain that the total computational work to integrate the transient Navier-Stokes equations grows like $\mathrm{Re}^{3/4} \times \mathrm{Re}^{9/4}$, i.e. Re^3. Hence, the number of degrees of freedom increases so drastically with the Reynolds number that DNS are far beyond our current computer possibilities... One way of achieving a numerical solution of turbulent flows is to take advantage of the random character of turbulence by using a statistical method. An instantaneous flow variable f is decomposed into a mean part $\overline{f}$ and a fluctuating part f' as follows

$$f = \overline{f} + f',$$

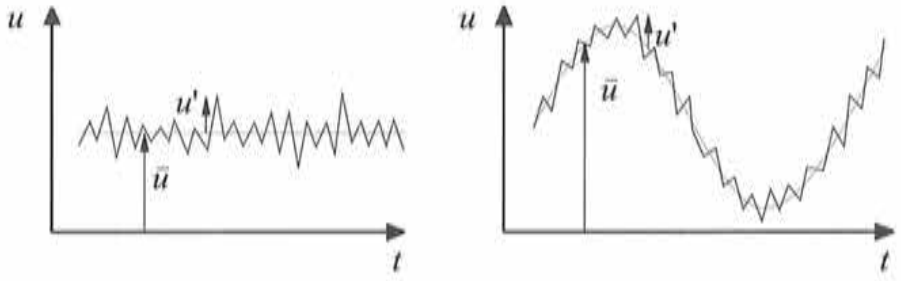


Figure 1.8: A typical turbulent flow. (Left) Mean flow is stationary. (Right) Mean flow is transient.

where $\overline{f}$ is computed performing some averaging (the same definition holds for vectors). We define here the time averaging operation as

$$\overline{f}_{\text{time}}(\boldsymbol{x}) = \lim_{T\to\infty} \frac{1}{T} \int_0^T f(x,t)dt,$$

and the ensemble averaging, defined as

$$\overline{f}_{\text{ensemble}}(\boldsymbol{x},t) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} f_n(x,t),$$

where n denotes the n-th experiment of N identical experiments, all performed with the same set-up and running conditions.

Although the concept of time averaging is more intuitive, the ensemble averaging has the great advantage that any differential operation commutes with the summation sign. For a stationary flow, the two averaging processes are assumed to be equivalent; this is called the ergodic *hypothesis*. The mathematical properties of ensemble averaging are assumed valid while the physical analysis is developed in the framework of time averaging (e.g. for comparisons or calibration of models with experimental or DNS results). The generalization of time averaging to transient flow can be done as long as two time scales can be distinguished: one for the fluctuation and one for the mean flow. This is schematized in Figure 1.8. On the one hand, Figure 1.8 (Left) shows the signal of typical stationary turbulent flow at a given point. On the other hand, Figure 1.8 (Right) shows a transient turbulent flow, where the time scale of the large scale variation is much greater than that of the turbulent fluctuations.

Decomposing the velocity and pressure fields as explained before,

$$\begin{aligned} \boldsymbol{u} &= \overline{\boldsymbol{u}} + \boldsymbol{u}', \\ p &= \overline{p} + p', \end{aligned}$$

an averaged solution of the Navier-Stokes equations can be obtained; the resulting equations are referred to as the Reynolds averaged Navier-Stokes equations (RANS) and were first derived by Reynolds in 1895. They are:

$$\partial_t \overline{\boldsymbol{u}} + (\overline{\boldsymbol{u}} \cdot \nabla)\overline{\boldsymbol{u}} - \nabla \cdot (2\nu\varepsilon(\overline{\boldsymbol{u}}) - \boldsymbol{\tau}) + \nabla\overline{p} = \overline{\boldsymbol{f}}, \tag{1.34}$$

$$\nabla \cdot \overline{\boldsymbol{u}} = 0, \tag{1.35}$$

where τ is called the Reynolds stress tensor. Its components are associated with the correlations between the fluctuation velocities and originate from the non-linear term of the Navier-Stokes equations, the convective term. Its components are:

$$\tau^{ij} = -\overline{u'^i u'^j}.$$

The Reynolds stress has six independent components, for which coupled differential equations can be derived. However, the non-linearity of the Navier-Stokes equations generates higher order correlations terms, including pressure-velocity correlations: this is the so-called *closure problem*. Approximations are therefore needed to solve this closure problems. The models emerging from this stochastic approach and modelling τ are called Reynolds stress closure models (see [35] for a complete review). An excellent introduction to the modelling of turbulence can be found in [36].

1.3.2 The Boussinesq approximation

For practical engineering applications, approximations are needed to avoid solving the six addition equations for the Reynolds stress tensor. Before making any assumption regarding the modelling of the Reynolds stress, it is worth recalling some important physical and mathematical requirements:

- It is known from experience that turbulent effects are more likely in zones of strong velocity gradients; the Reynolds stress would be related to the mean strain velocity tensor $\varepsilon(\overline{u})$. Assuming that its deviatoric part is proportional to $\varepsilon(\overline{u})$ is the Boussinesq approximation [37].
- The Reynolds stress tensor must be symmetrical, i.e. $\tau = \tau^t$.
- It must yield positive energy components, i.e. $\overline{u'^i u'^i} \geq 0 \ \forall \ i = 1, 2, 3$. This is known as *realizability*.
- It must be Galilean invariant.
- It should leave the RANS equations invariant under translation and rotation.
- It must yield similitude under the Reynolds number.
- The fluctuating momentum equations are invariant under an arbitrary translational acceleration. Within the limit of two dimensional turbulence, the Reynolds stress should be completely frame indifferent (see [38] and [39]).

Following the Boussinesq eddy-viscosity approximation, as stated by the first item, the Reynolds stress is modelled as:

$$\tau = 2\nu_t \varepsilon(\overline{u}) - \frac{2}{3} k \boldsymbol{I}, \tag{1.36}$$

where ν_t is the isotropic eddy-viscosity and $\boldsymbol{I}$ is the n_d-dimensional identity. This model is called isotropic because the eddy-viscosity is a scalar. While the validity of the symmetry is evident (from the symmetry of the strain rate tensor), the normal stress components resulting from the Boussinesq eddy-viscosity approximation could violate realizability if ν_t is not chosen properly.

The term k is defined as a positive quantity such that, by contraction of the later equation,

$$k = \frac{1}{2}\overline{\boldsymbol{u}' \cdot \boldsymbol{u}'};$$

with obvious meaning, k is called the *specific turbulence kinetic energy*. The proof of translational and rotational invariance can be found in [40]. The Galilean invariance as well as the similitude under the Reynolds number are easy to check. The two-dimensional invariance is treated in [38] and [39].

For the sake of clarity, we drop the overline sign used to identify averaged variables. Then, the Reynolds-averaged equations together with Boussinesq eddy-viscosity approximation given by Equation (1.36) are

$$\begin{aligned} \partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} - 2\,\nabla\cdot[(\nu+\nu_t)\boldsymbol{\varepsilon}(\boldsymbol{u})] + \nabla p^* &= \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} &= 0, \end{aligned}$$

where ν_t is the kinematic eddy-viscosity and

$$p^* = p + \frac{2}{3}k,$$

k being the turbulence kinetic energy. Note that when using p^* as the independent pressure variable, it is not possible to prescribe boundary conditions involving the physical pressure p, such as in the case in which the traction is prescribed (e.g. prescription of the atmospheric pressure). At most, the prescription of the traction will be an approximation to the real one. When computing forces however, p^* is the pressure variable to consider as the term $2/3k$ acts as an additional normal stress. These equations are the RANS equations, in their divergence form (the viscous term is computed using the strain rate tensor).

1.3.3 Spalart-Allmaras model

The turbulence model chosen to compute the eddy-viscosity is a one-equation turbulence model, namely the Spalart-Allmaras turbulence model [41], referred to as SA from now on. This model was devised "using empiricism and arguments of dimensional analysis, Galilean invariance, and selective dependence on molecular viscosity". It involves an eddy-viscosity variable $\tilde{\nu}$, related to the eddy-viscosity ν_t by:

$$\begin{aligned} \nu_t &= f_{v_1}\tilde{\nu}, \qquad \text{with} \\ f_{v_1} &= \frac{\chi^3}{\chi^3 + c_{v_1}^3}, \qquad \chi := \frac{\tilde{\nu}}{\nu}. \end{aligned} \tag{1.37}$$

The transport equation for $\tilde{\nu}$ is:

$$\frac{\partial\tilde{\nu}}{\partial t} + \boldsymbol{u}\cdot\nabla\tilde{\nu} = c_{b_1}\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\nabla\cdot[(\nu+\tilde{\nu})\nabla\tilde{\nu}] + \frac{c_{b_2}}{\sigma}(\nabla\tilde{\nu})^2 - c_{w_1}f_w\frac{\tilde{\nu}^2}{d^2}, \tag{1.38}$$

where d is the shortest distance to the wall and κ is the Von-Karman constant. The constants of the model are given later on. Equation (1.38) is not the original SA model. For the sake of clarity, some terms have been voluntarily omitted. The laminar region and transition cannot be simulated using the version presented previously; see the original publication of the authors for more information [41].

The function f_w is given by

$$f_w = g\left[\frac{1+c_{w_3}^6}{g^6+c_{w_3}^6}\right]^{\frac{1}{6}}, \qquad \text{with}$$

$$g = r + c_{w_2}(r^6 - r), \qquad r := \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2},$$

g can take relatively high values, so it is preferable to compute f_w as

$$f_w = \left[\frac{1+c_{w_3}^6}{1+c_{w_3}^6/g^6}\right]^{\frac{1}{6}}.$$

The production term, the first term of the right-hand side of (1.38) involves the quantity $\tilde{S}$ which is a function of the magnitude of the vorticity S and is given by

$$\tilde{S} := S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v_2}, \qquad f_{v_2} = 1 - \frac{\chi}{1+\chi f_{v_1}},$$

with

$$S = \sqrt{2\boldsymbol{\Omega}(\boldsymbol{u}) : \boldsymbol{\Omega}(\boldsymbol{u})}, \qquad \text{and}$$

$$\boldsymbol{\Omega}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^t).$$

When the frame of reference is rotating at an angular velocity $\boldsymbol{\omega}$, the velocity gradients $\nabla \boldsymbol{u}$ should be replaced by $\nabla \boldsymbol{u} + \boldsymbol{\Omega}'$, where $\boldsymbol{\Omega}'$ is the anti-symmetric tensor associated with $\boldsymbol{\omega}$:

$$\Omega'_{ij} = e_{kji}\omega_k,$$

where e_{kji} is the permutation "tensor". Therefore, $\boldsymbol{\Omega}(\boldsymbol{u})$ transforms into:

$$\boldsymbol{\Omega}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^t) + \boldsymbol{\Omega}'.$$

The values of the constants of the model are

$$c_{b_1} = 0.1355, c_{b_2} = 0.622, \sigma = 0.667, c_{v_1} = 7.1, \kappa = 0.41,$$

$$c_{w_1} = c_{b_1}/\kappa^2 + (1 + c_{b_2})/\sigma, c_{w_2} = 0.3, c_{w_3} = 2.0.$$

As mentioned earlier, the original model contains laminar and transition corrections. The first correction enables us to predict laminar flows with the solution $\tilde{\nu}$, while the second one enables us to simulate the boundary-layer transition [41]. These corrections are not of interest in our case as they are only effective if the governing equations have to be solved up to the wall.

1.3.4 Boundary conditions

The inflow condition $\tilde{\nu}_\infty$ is computed using Equation (1.37) from an inflow value of $\tilde{\nu}_t$ given as a fraction of the kinematic laminar viscosity. At the outflows, we impose

$$\nabla \tilde{\nu} \cdot \boldsymbol{n} = 0.$$

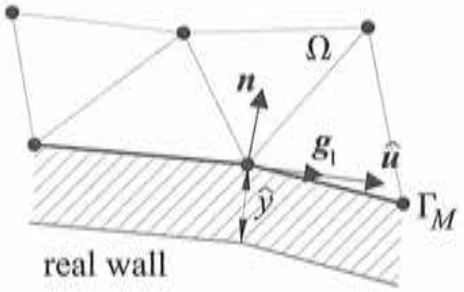


Figure 1.9: Local system on the boundary Γ_M.

Classical approach

The classical approach consists in imposing the eddy viscosity to zero on the walls. However, for practical flows with high Reynolds numbers, the resolution of SA equations would require too many grid points in the normal direction to the wall. In order to avoid solving the whole boundary layer, we introduce a wall function approach.

Wall function approach

The RANS and turbulence equations can be solved using the wall function approach on the boundary of the computational domain. The wall function approach implemented in this work consists in assuming that the computational wall is located sufficiently far from the real wall where the no-slip condition for the velocity holds, in order to avoid solving for the large gradients present in the boundary layer. Then we use a model equation for the traction together with a non-penetrating condition for the velocity (zero normal component) as well as a model equation for the eddy viscosity. Various methods have been proposed to implement the wall functions; see e.g. [42] for their application to two-equation turbulence models, [43] for their application to the present turbulence model and [44] for a general discussion. A possible alternative is now presented.

The wall functions are imposed on the boundary Γ_M, as given by Equation (1.26). In the following, variables on Γ_M are identified with a hat. The distance at which the boundary Γ_M is located from the real wall is assumed to be known and is a user-defined value $\hat{y}$; here y refers to the distance normal to the wall. This approximation is equivalent to considering that the wall is virtually inside the computational domain, at a distance $\hat{y}$ from Γ_M, as schematized by Figure (1.9).

Following [42], another possibility would be to impose the dimensionless $\hat{y}^+$ rather than its dimensional counterpart. Remember that

$$\hat{y}^+ := \frac{\hat{y} U_*}{\nu}, \tag{1.39}$$

where U_* is the friction velocity, defined as

$$U_* = \sqrt{\frac{\tau_{\text{wall}}}{\rho}}, \tag{1.40}$$

where τ_{wall} is the wall shear stress. However, this method would place the real wall at a varying physical distance along Γ_M, with possible infinite corresponding $\hat{y}$ when U_* reaches zero, which occurs near stagnation points. On the other hand, the method proposed in this work gives no control

on $\hat{y}^+$. Although the wall functions presented here can be used up to the wall, the coarse meshes generally accompanying the method would not permit them to capture the near-wall gradients.

Let $(\boldsymbol{g}_1, \boldsymbol{n})$ (or $(\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{n})$ in three dimensions) be the local basis of the current wall point on Γ_M (see Figure (1.9)), $\hat{\boldsymbol{u}}$ its velocity, and define $\hat{U}^+$ as:

$$\hat{U}^+ = \frac{|\hat{\boldsymbol{u}}|}{U_*}. \tag{1.41}$$

Remember that $\hat{\boldsymbol{u}}$ is automatically tangential to Γ_M as zero-normal velocity is imposed on Γ_M (see Equation (1.26)). The law of the wall for smooth walls is mainly divided into three zones, namely the viscous sublayer, the buffer zone and the turbulent zone. The so-called Reichardt's law relates $\hat{U}^+$ and $\hat{y}^+$ within these three zones as:

$$\hat{U}^+ = \frac{1}{\kappa}\ln\left(1 + 0.4\hat{y}^+\right) + 7.8\left[1 - \exp\left(-\frac{\hat{y}^+}{11}\right) - \frac{\hat{y}^+}{11}\exp\left(-0.33\hat{y}^+\right)\right]. \tag{1.42}$$

Using (1.41) and performing a Newton-Raphson scheme for the latter equation enables one to estimate U_*, the value of $\hat{\boldsymbol{u}}$ is known from a previous iteration. Knowing U_*, we model the tangential traction as

$$\boldsymbol{t}_t = -\frac{U_*^2}{|\hat{\boldsymbol{u}}|}\hat{\boldsymbol{u}}, \tag{1.43}$$

and impose it as a natural boundary condition of the Navier-Stokes equations (Equation (1.26)) with

$$\begin{aligned} t_1 &= \boldsymbol{t}_t \cdot \boldsymbol{g}_1, \qquad \text{and} \\ t_2 &= \boldsymbol{t}_t \cdot \boldsymbol{g}_2 \qquad \text{for 3D flows.} \end{aligned}$$

$\boldsymbol{t}_t$ is the shear stress exerted on the fluid by the computational wall which slows down the flow, and this justifies why $\boldsymbol{t}_t$ points in the opposite direction of the local velocity. In addition, recall that the law of the wall was devised for two-dimensional flows. It is known that the velocity component parallel to the wall shear-stress follows the two-dimensional law of the wall. By imposing equation (1.43), the velocity is aligned with the total shear stress; note that only the magnitude of the wall shear-stress can be estimated. This choice raises another controversy. The law of the wall for the velocity is deduced by assuming the following relation

$$U^+ = f(y^+), \tag{1.44}$$

Equation (1.44) states that the velocity scales with the wall shear-stress $t_{\text{wall}} = \rho U_*^2$. However, we have just imposed Equation (1.43) which states that the *local* shear-stress $\boldsymbol{t}_t$ is aligned with the velocity; in fact we had no other choice as we have only information on the magnitude on the characteristic velocity of the wall zone but no physical information on its direction.

Remark 1.2. The wall boundary condition is non-linear as it depends on $|\hat{\boldsymbol{u}}|$; this non-linearity is coupled with the non-linearity due to the convective term of the Navier-Stokes equations.

Remark 1.3. The value of $\hat{y}$ is unknown and can only be *estimated* a priori. It could be calculated, for example, as a fraction of the distance to the first node off the wall. In this work, it is taken to be constant and adjusted in order to be around $\hat{y}^+ = 30$ as an average along the walls. Remember that the high-Reynolds number version of SA model is only valid outside the buffer zone.

Remark 1.4. The distance to the wall must be corrected to $d + \hat{y}$; this is the value used in Equation (1.38).

The wall condition for $\tilde{\nu}$ is computed by first imposing a value for $\hat{\nu}_t$. The classical mixing length hypothesis is used together with Van-Driest damping function, i.e.

$$\begin{aligned} \hat{\nu}_t &= l_{mix}^2 |\boldsymbol{n} \cdot \widehat{\nabla u_g}|, \\ l_{mix} &= \kappa \hat{y}^+ \left[1 - \exp\left(-\frac{\hat{y}^+}{26}\right)\right], \end{aligned} \tag{1.45}$$

where

$$\begin{aligned} u_g &= |\boldsymbol{u} \cdot \boldsymbol{g}_1| && \text{for 2D flows,} \\ u_g &= \sqrt{(\boldsymbol{u} \cdot \boldsymbol{g}_1)^2 + (\boldsymbol{u} \cdot \boldsymbol{g}_2)^2} && \text{for 3D flows,} \end{aligned}$$

is the tangential velocity. Equation (1.45) corresponds to the inner-layer equation of Baldwin-Lomax model, approximating the magnitude of the vorticity by $|\boldsymbol{n} \cdot \widehat{\nabla u_g}|$. The term $\boldsymbol{n} \cdot \widehat{\nabla u_g}$ is the normal derivative of the tangential velocity. The latter equation can be re-expressed in terms of the dimensionless quantities as:

$$\hat{\nu}_t^+ = \kappa^2 (\hat{y}^+)^2 \left[1 - \exp\left(-\frac{\hat{y}^+}{26}\right)\right]^2 \frac{d\hat{U}^+}{d\hat{y}^+}, \qquad \hat{\nu}_t^+ = \frac{\hat{\nu}_t}{\nu}, \tag{1.46}$$

where $d\hat{U}^+/d\hat{y}^+$ is obtained by deriving Reichardt's law (1.42). Finally, the value $\widehat{\tilde{\nu}}$ of $\tilde{\nu}$ on Γ_M is calculated by solving Equation (1.37) using a Newton-Raphson method.

Performing an asymptotic expansion [45] to the first order of the inner-layer of the Navier-Stokes equations, and using the Boussinesq approximation, we find that

$$(1 + \nu_t^+) \frac{\partial u^+}{\partial y^+} = 1, \tag{1.47}$$

This equation states that in the inner region, the total friction is constant and is equal to its value on the wall, even in the presence of pressure gradients. It can be solved by substituting the value ν_t^+ given by the law of the wall (1.46). From there, two different results can be obtained from the integration of (1.47). The first one, referred to as "our wall-law asymptotic-expansion integration" uses $\partial U^+/\partial y^+ = dU^+/dy^+$ obtained by deriving Equation (1.42) with respect to y^+. The second one uses $\partial U^+/\partial y^+$ without any approximation. Note that in this case, the corresponding PDE for (1.47) involves the square of the velocity derivatives. This result will be referred to as "exact asymptotic-expansion integration". Knowing that $U^+ = 0$ at $y^+ = 0$, Equation (1.47) can be integrated numerically for both methods and compared to Reichardt's law. This is shown by Figure (1.10). According to the figure, it can be concluded that the laws of the wall used for the velocity and the eddy-viscosity (Equations (1.43) and (1.46)) are compatible with the first-order asymptotic expansion of the inner region.

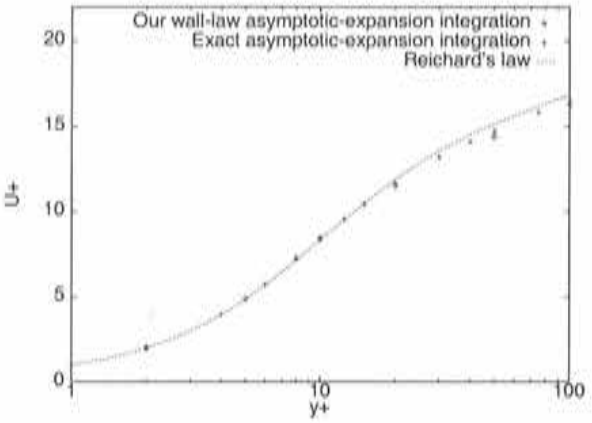


Figure 1.10: Comparison of U^+-profiles in the inner layer.

An estimate for $\hat{y}$

As will be shown in the numerical examples, the estimation of the distance $\hat{y}$ from the real wall to the computational wall is of primary importance when using the wall function approach. This is so that the computational wall lies in the desired fully turbulent zone, which is where we expect the law of the wall to be valid (say in the range $y^+ \in (30, 80)$). We propose two estimates for $\hat{y}$, one for internal flows and another for open flows.

For a fully-developed channel flow at high Reynolds number (10^4 to 10^7), the friction coefficient follows the so-called Halleen and Johnston's correlation [46]:

$$c_f = 0.0706\,\mathrm{Re}^{-1/4},$$

where the Reynolds number is based on the channel height H and the averaged velocity U. By definition,

$$c_f = 2\frac{U_*^2}{U^2}.$$

Hence, from the definition of y^+ (Equation (1.39)), we have that

$$\frac{\hat{y}}{H} = 5.323\hat{y}^+\,\mathrm{Re}^{-7/8}. \tag{1.48}$$

Figure (1.11) (Left) shows the variation of the dimensionless distance where the computational wall should be located, with respect to the Reynolds number, and for various $\hat{y}^+$. This provides a good estimate for the $\hat{y}$ to be chosen.

For an open flow, we could estimate $\hat{y}$ by choosing the skin friction coefficient obtained for a turbulent flow over a flat plate,

$$c_f = 0.074\,\mathrm{Re}^{-1/5},$$

where Re is the Reynolds number based on the length ℓ of the flat plate. Last formula is valid for Re from 10^5 up to 10^7. We obtain in this case

$$\frac{\hat{y}}{\ell} = 5.199\,\hat{y}^+\,\mathrm{Re}^{-9/10}. \tag{1.49}$$

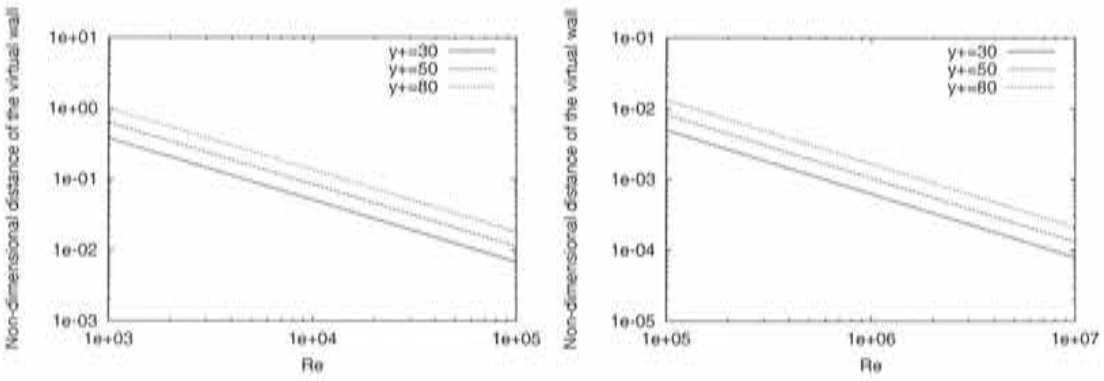


Figure 1.11: A first guess for $\hat{y}$. (Left) Bounded flows. (Right) Open flows.

Figure (1.11) (Right) gives the guess for $\hat{y}$, for different $\hat{y}^+$, as a function of the Reynolds number.

The two estimates we have derived can be viewed as upper bounds as, in general, pressure gradients will tend to decelerate the flow in the wall region. Still, although the value of $\hat{y}$ is problem dependent, Equations (1.48) and (1.49) can provide a good indication for the numerical experimentalist, as the numerical examples will show. Let us just mention that a simple scheme could help to adjust the value $\hat{y}$ iteratively within the program, in order to obtain an average value of $\hat{y}^+$ in the desired range.

1.3.5 Linearization

Before deriving the Galerkin formulation of the problem we have to linearize the SA equation. This stationary transport equation for the eddy-viscosity variable $\tilde{\nu}$ is an advection-diffusion-reaction equation of the form

$$\boldsymbol{u} \cdot \nabla \tilde{\nu} - \nabla \cdot (\varepsilon \nabla \tilde{\nu}) + s\tilde{\nu} = f.$$

We have shown in Section 1.1.3 that, in the simple case $\tilde{\nu} = 0$ on Γ, this equation is 'well behaved' when ε and s are both positive. In this case, the bilinear form associated to the problem is coercive and the finite element approximation to the problem is in principle possible for any positive value of ε and s (see in particular Equation (1.14)). Following this indication, the SA equation is linearized as follows:

$$\begin{aligned} \frac{\partial \tilde{\nu}^{m+1}}{\partial t} + \boldsymbol{u} \cdot \nabla \tilde{\nu}^{m+1} + c_{w_1} f_w \frac{\tilde{\nu}^m}{d^2} \tilde{\nu}^{m+1} - \frac{1}{\sigma} \nabla \cdot [(\nu + \tilde{\nu}^m) \nabla \tilde{\nu}^{m+1}] \\ = \frac{c_{b_2}}{\sigma} (\nabla \tilde{\nu}^m)^2 + c_{b_1} \check{S} \tilde{\nu}^m, \end{aligned} \tag{1.50}$$

where the superscript m stands for the iteration number. In practice, in order to enable convergence classical under-relaxation is introduced and several relaxation iterations are performed. We now present the stabilized finite element formulation.

1.3.6 Time discretization

The time derivative is approximated by a backward-Euler or Crank-Nicolson scheme, as presented in Section 1.1.5.

1.3.7 Finite element formulation

We split the Dirichlet boundary Γ_D which was defined when dealing with the Navier-Stokes equations into an inflow and a wall component, $\Gamma_{D,\infty}$ and $\Gamma_{D,\text{wall}}$, respectively, such that $\Gamma_D = \Gamma_{D,\infty} \cup \Gamma_{D,\text{wall}}$. Let us introduce the following functional spaces:

$$\Psi = \{\phi \in H^1(\Omega) \,|\, \phi_{|\Gamma_D} = 0\},$$
$$\Phi = \{\phi \in H^1(\Omega) \,|\, \phi_{|\Gamma_{D,\infty}} = \tilde{\nu}_\infty,\ \phi_{|\Gamma_{D,\text{wall}}} = 0\},$$

for the classical approach (integration up to the wall), and

$$\Psi = \{\phi \in H^1(\Omega) \,|\, \phi_{|\Gamma_D \cup \Gamma_M} = 0\},$$
$$\Phi = \{\phi \in H^1(\Omega) \,|\, \phi_{|\Gamma_D} = \tilde{\nu}_\infty,\ \phi_{|\Gamma_M} = \widehat{\tilde{\nu}}\},$$

for the wall function approach. Let us construct the functional linear subspaces $\Phi_h \subset \Phi$ and $\Psi_h \subset \Psi$ from the partition of Ω. The finite element formulation for solving equation (1.50) is the variational subgrid scale model described in 1.1.4. The finite element algorithm reads as follows: Given $\tilde{\nu}_h^{n+1,0} \in \Phi_h$, for each time step $n \geq 0$, find for $m = 0, 1, \ldots$ until convergence, $\tilde{\nu}_h^{n+1,m+1} \in \Phi_h$ such that

$$\begin{aligned}
&\int_\Omega \frac{\tilde{\nu}_h^{n+\theta,m+1}}{\theta\delta t}\phi_h \, d\Omega + \int_\Omega (\boldsymbol{u}_h \cdot \nabla \tilde{\nu}_h^{n+\theta,m+1})\phi_h \, d\Omega + \int_\Omega s\, \tilde{\nu}_h^{n+\theta,m+1}\phi_h \, d\Omega \\
&\quad + \int_\Omega \varepsilon \nabla \tilde{\nu}_h^{n+\theta,m+1} \cdot \nabla \phi_h \, d\Omega + \int_{\Omega'} \tau_3 (\boldsymbol{u}_h \cdot \nabla \tilde{\nu}_h^{n+\theta,m+1})(\boldsymbol{u}_h \cdot \nabla \phi_h) \, d\Omega = \int_\Omega f\phi_h \, d\Omega \\
&\quad + \int_\Omega \frac{\tilde{\nu}_h^n}{\theta\delta t}\phi_h \, d\Omega - \int_{\Omega'} \tau_3 (\nabla \varepsilon \cdot \nabla \phi_h - s\phi_h + \varepsilon \Delta \phi_h - \nabla \varepsilon \cdot \nabla \phi_h)\, R\tilde{\nu}_h^{n+\theta,m} \, d\Omega \\
&\quad - \int_{\Omega'} \tau_3 (\boldsymbol{u}_h \cdot \nabla \phi_h)(R\tilde{\nu}_h^{n+\theta,m} - \boldsymbol{u}_h \cdot \nabla \tilde{\nu}_h^{n+\theta,m}) \, d\Omega \qquad \forall\, \phi_h \in \Psi_h.
\end{aligned}$$

We have introduced the following variables computed using the values of the eddy-viscosity variable at the previous linearization step

$$\varepsilon = \frac{1}{\sigma}(\nu + \tilde{\nu}_h^{n+\theta,m}),$$
$$s = c_{w_1} f_w \frac{\tilde{\nu}_h^{n+\theta,m}}{d^2},$$
$$f = \frac{c_{b_2}}{\sigma}(\nabla \tilde{\nu}_h^{n+\theta,m})^2 + c_{b_1} \tilde{S} \tilde{\nu}_h^{n+\theta,m},$$

and where $R\tilde{\nu}_h^{n+\theta,m}$ is the residual of the discrete counterpart of the SA equation defined as

$$R\tilde{\nu}_h^{n+\theta,m} := \frac{\tilde{\nu}_h^{n+\theta,m} - \tilde{\nu}_h^n}{\theta\delta t} + \boldsymbol{u}_h \cdot \nabla \tilde{\nu}_h^{n+\theta,m} - \nabla \cdot (\varepsilon \nabla \tilde{\nu}_h^{n+\theta,m}) + s\tilde{\nu}_h^{n+\theta,m} - f.$$

Note that the ASGS term involving the convection of $\tilde{\nu}$ is integrated into the left-hand side. The value for τ_3 is given by

$$\tau_3 = \left(\frac{c_1}{h_e^2} \varepsilon + \frac{c_2}{h_e} |\boldsymbol{u}_h| + s \right)^{-1},$$

where $c_1 = 4$ and $c_2 = 2$.

Remark 1.5. In order to avoid local oscillations, the discontinuity-capturing crosswind dissipation presented in Section 1.1.4 can be used.

1.3.8 Numerical strategy

The complete numerical strategy used to solve the coupled RANS and SA equations using the integration up to the wall is shown in Algorithm 1.1.

Algorithm 1.1 Solution of the RANS/SA equations: classical approach

```
for all time steps do
  while stopping criterion not reached do
    Solve the RANS equations
    Solve several times the SA equation using under-relaxation
    Update the eddy viscosity
  end while
end for
```

The numerical strategy used to solve the coupled RANS and SA equations using the wall function approach is summarized in Algorithm 1.2.

Algorithm 1.2 Solution of the RANS/SA equations: wall function approach

```
As an initial guess, let t_t = 0 on Γ_M
for all time steps do
  while stopping criterion not reached do
    Solve the RANS equations
    Knowing û on Γ_M, find U_* using Reichardt's law (1.42)
    Compute ν_t^+ on Γ_M using Equation (1.46) and deriving Equation (1.42)
    Compute the corresponding value of ν̃ on the wall from Equation (1.37)
    Solve several times the SA equation using under-relaxation
    Compute t_t from Equation (1.43)
    Update eddy-viscosity
  end while
end for
```

1.4 Examples

We now present three numerical examples. The first one solves the channel flow at a relatively small Reynolds number, for which DNS data are available. The next example is the backward facing step in which we test the behavior of the wall function approach in the presence of separation.

Finally we solve the transient flow past a square cylinder. In all these examples, x and y are the streamwise and crosswise directions respectively, and u and v are their respective averaged velocity components. At the inflows, we prescribe a uniform flow given by

$$\begin{aligned} u &= U, \\ v &= 0, \\ \nu_t &= \nu_{t,\infty}, \end{aligned}$$

where U and $\nu_{t,\infty}$ depend on the problem. At the outflows, we impose

$$\begin{aligned} \boldsymbol{\sigma} \cdot \boldsymbol{n} &= \mathbf{0}, \\ \nabla \tilde{\nu} \cdot \boldsymbol{n} &= 0. \end{aligned}$$

In addition, the symmetry lines are always parallel to the x-axis. They belong to Γ_M for the RANS equations and to Γ_N of the SA equation. On these lines we impose

$$\begin{aligned} \boldsymbol{g}_1 \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n} = t_1 &= 0, \\ \boldsymbol{u} \cdot \boldsymbol{n} = v &= 0, \\ \nabla \tilde{\nu} \cdot \boldsymbol{n} &= 0, \end{aligned}$$

where $\boldsymbol{g}_1$ is a tangent vector to the symmetry line (it is parallel to the x-axis).

1.4.1 Fully developed channel flow

We solve a fully developed turbulent channel flow. Let $2H$ be the height of the channel and U be the uniform inflow velocity (bulk velocity). The geometry is shown in Figure 1.12 together with the boundary conditions. The channel flow is solved at the following Reynolds number

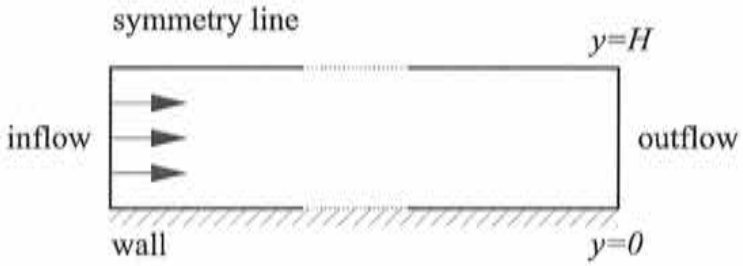


Figure 1.12: Channel flow. Geometry.

$$\mathrm{Re} = \frac{2HU}{\nu} = 13\,750,$$

and the results of the SA model are going to be compared with the theoretical results as well as the DNS results of Mansour *et al.* [47].

The RANS and SA equations are solved using the Q1/Q1 element on three different meshes, adapted near the wall. We denote h as the average element length in the vertical direction. Mesh 1 has 10 elements in the vertical direction, Mesh 2 has 30 elements and Mesh 3, 60 elements. The channel is taken to be sufficiently long to let the flow develop ($100H$). Before presenting the results, let us mention that the shock capturing strategy is not necessary for the iterative scheme to converge.

We now derive the fully-developed channel flow equations. Our starting point are the RANS equations (1.34) and (1.35). We define u, v, w as the averaged components of the velocity field and u', v', w' as the fluctuating components; p is the averaged pressure. Assuming a 2-D steady and fully-developed flow, we set:

$$\begin{aligned} \frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} &= 0, \\ \frac{\partial (\cdot)}{\partial z} &= 0, \\ w &= 0. \end{aligned}$$

Hence, the momentum equation gives

$$u = u(y),$$

while the continuity equation becomes

$$\cancel{\frac{\partial u}{\partial x}} + \frac{\partial v}{\partial y} + \cancel{\frac{\partial w}{\partial z}} = 0,$$

which after applying the no-slip condition at the wall gives

$$v = 0.$$

Simplifying the RANS equations according to the previous results, we have:

$$\begin{aligned} x\text{-momentum:} \quad 0 &= -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y} - \overline{u'v'}\right), \\ y\text{-momentum:} \quad 0 &= \frac{\partial}{\partial y}(\overline{(v')^2} + p), \\ z\text{-momentum:} \quad 0 &= \frac{\partial}{\partial y}(\overline{u'w'}). \end{aligned} \tag{1.51}$$

We drop the third equation as it is unnecessary to compute the two-dimensional field we are interested in. Let us present some results concerning the friction coefficient. The y-momentum equation states that

$$p(x, y) + \overline{(v')^2}(y) = f(x), \tag{1.52}$$

where $f(x)$ is a function of x only. Hence, by denoting p_{wall} as the value of the pressure at the wall, and knowing that the fluctuation velocity is zero on the wall due to the no-slip condition, Equation (1.52) can be rewritten as

$$p(x,y) + \overline{(v')^2}(y) = p_{\text{wall}}(x)$$

and therefore

$$\frac{\partial p(x,y)}{\partial x} = \frac{dp_{\text{wall}}(x)}{dx}.$$

Integrating the x-momentum Equation (1.51) with respect to y and substituting the last equation we obtain

$$\nu \frac{du}{dy} - \overline{u'v'} = \frac{dp_{\text{wall}}(x)}{dx} y + c(x), \tag{1.53}$$

where $c(x)$ is a function of x only. At the symmetry line ($y = H$), the Reynolds stress tensor component $\overline{u'v'}$ and du/dy must vanish and hence, we have that

$$c(x) = -\frac{dp_{\text{wall}}(x)}{dx} H.$$

Since $\overline{u'v'}$ is also zero at $y = 0$ (no-slip condition), Equation (1.53) gives for the wall shear stress

$$\tau_{\text{wall}} = \mu \left| \frac{du}{dy} \right|_{\text{wall}} = \rho H \left| \frac{dp_{\text{wall}}}{dx} \right|$$

which upon substituting into Equation (1.40)

$$U_* = \sqrt{\frac{H}{\rho} \left| \frac{dp_{\text{wall}}}{dx} \right|} \tag{1.54}$$

or, equivalently

$$U_* = \sqrt{\nu \left| \frac{du}{dy} \right|_{\text{wall}}} \tag{1.55}$$

The friction coefficient (or skin friction) is by definition

$$c_f = \tau_{\text{wall}} / (\frac{1}{2} \rho U^2)$$

and can therefore be directly calculated from the evaluation of the friction velocity:

$$c_f = 2 \frac{U_*^2}{U^2}$$

We therefore have two ways of evaluating c_f, using the pressure drop (Equation (1.54)), or using the wall shear stress (Equation (1.55)). Figure 1.13 (Top) (Left) shows the results obtained for

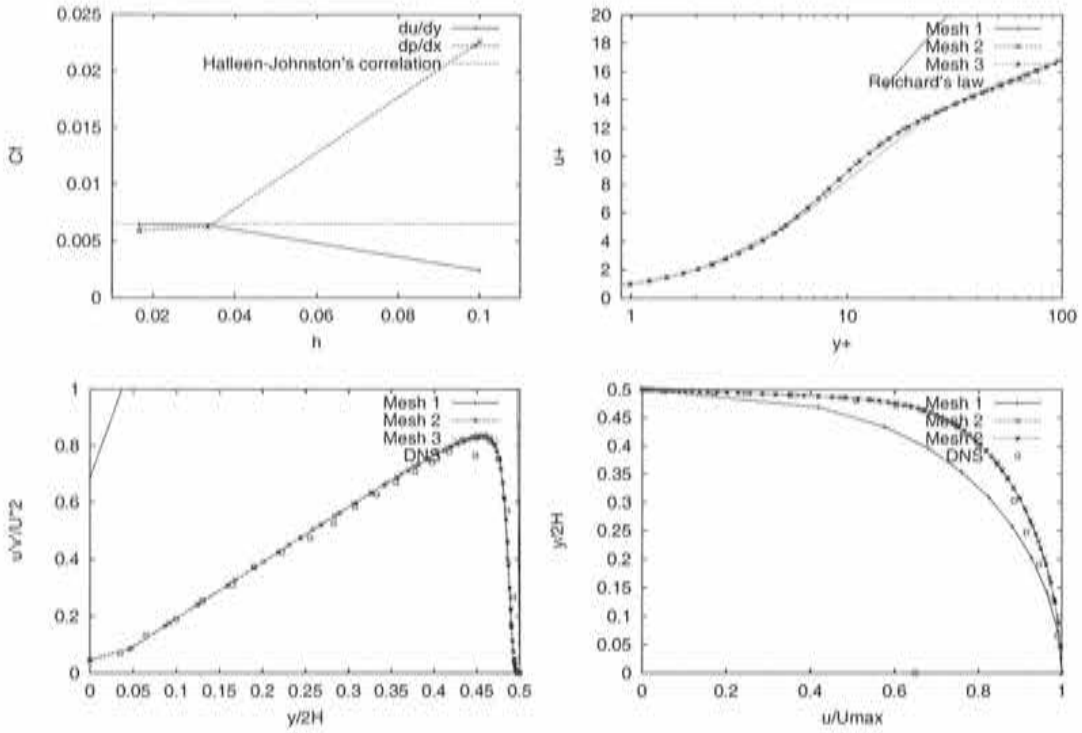


Figure 1.13: Channel flow. Results of the three meshes. (Top) (Left) Friction coefficient. (Top) (Right) Law of the wall. (Bot.) (Left) Reynolds shear stress. (Bot.) (Right) Mean velocity.

c_f, calculated in both ways, as a function of the mesh size h, and compare them to the friction coefficient calculated using Halleen Johnston's correlation,

$$c_f = 0.0706\,\mathrm{Re}^{-1/4} \approx 6.52 \times 10^{-3}.$$

We observe that for a fine mesh, the integration to the wall gives reasonable results while it fails for the *very* coarse mesh.

Table 1.1 gives information on the grid point spacing in the near wall region. Note that the first grid point of the very coarse mesh falls very high in the law of the wall.

Mesh	y/H first node	y^+ first node	Nodes with $y^+ \leq 10$
1	0.06274	16.2	0
2	0.00608	2.5	4
3	0.00032	0.1	23

Table 1.1: Grid spacing in the near wall region.

Figure 1.13 gives some profiles obtained. In Figure 1.13 (Bot.) (Right), we can appreciate that for the *very* coarse mesh, the mean velocity profile is not well captured. When using finer meshes, the results are in good agreements with DNS data. For these meshes, i.e. Mesh 2 and Mesh 3, the Reynolds shear stress as well as the law of the wall are well captured, as confirmed by Figures 1.13 (Top) (Right) and (Bot.) (Left). The first grid point of Mesh 1 is located around $y^+ = 10$, while

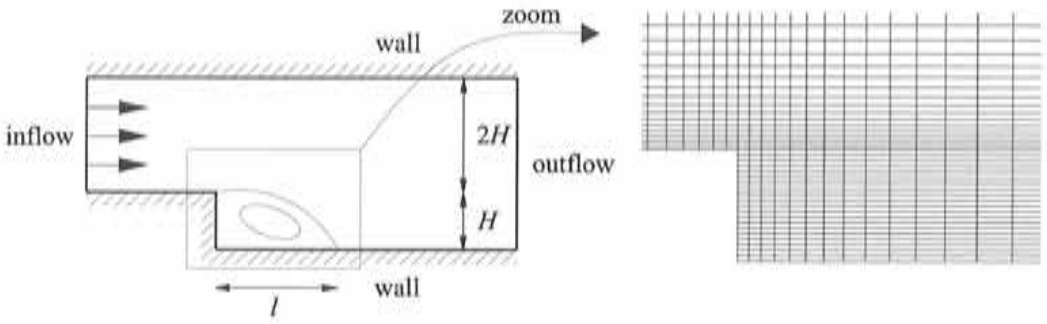


Figure 1.14: Backward facing step. (Left) Geometry. (Right) Detail of Mesh 2.

that of Mesh 2 and 3 are located much closer to the wall, at $y^+ = 1$ and $y^+ = 2$, respectively. Let us mention that as we do not assume any law of the wall, the friction velocity is calculated using a simple difference scheme in the first layer of elements; this explains the quite big departure from the law of the wall, shown in Figure 1.13 (Top) (Right).

1.4.2 Backward facing step

The backward facing step is a great challenge for turbulence models, as well as for the associated numerical strategies. In this work, it will be used to study the influence of the inflow condition $\nu_{t,\infty}$ and the wall-function parameter $\hat{y}$ used in the wall function approach. The step height is H, the channel height $2H$, the channel entrance is $6H$-long and the total length of the computational domain is $50H$. The inlet velocity profile is uniform such that $\boldsymbol{u} = (U, 0)$. Figure 1.14 (Left) shows the geometry of the problem.

Note that the entrance length is too short to let the flow develop before the step entrance: it should be pointed out that we expect that this enhances the effects of the inflow conditions. The Reynolds number is

$$\mathrm{Re} = \frac{UH}{\nu} = 70\,000.$$

As in the previous test case, the backward facing step is solved using three different meshes of Q1/Q1 elements. Mesh 1 has 550 elements, Mesh 2 has 2000 elements, and Mesh 3 has 8000 elements. A detail of Mesh 2 in the step corner is shown in Figure 1.14 (Right). The shock capturing technique is necessary for this test case as well as under-relaxation of the eddy-viscosity. Figure 1.15 shows a typical convergence history obtained using the Picard method as linearization technique for the convective term.

Before starting the computation, we want to estimate $\hat{y}$. Taking formula (1.48) for $y^+ = 50$, we find that $\hat{y}/H \approx 1.5\%$. Taking into account that in the region of interest, i.e. the step corner, the friction velocity is expected to be smaller, we will take this estimate as a minimum value. Numerical experiments show that the greater $\hat{y}/H$, the better the convergence. Also, the integration to the wall seems to be much more robust that the wall function approach.

The first results were obtained on Mesh 2. Table 1.2 shows the variation of the recirculation length l versus the inflow condition for the eddy-viscosity $\nu_{t,\infty}$. The table also shows the effect of the boundary condition imposed on the corner of the step entrance, located at $(6H, H)$. In fact,

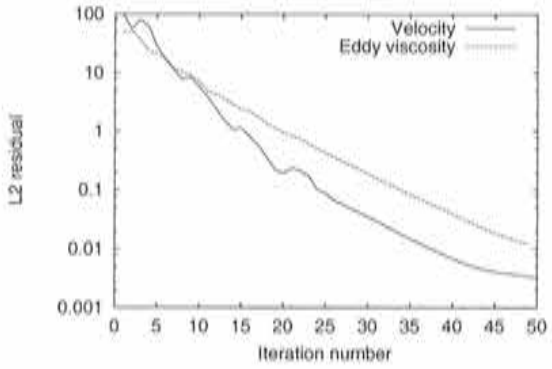


Figure 1.15: Backward facing step. Convergence history.

$\hat{y}/H$ (%)	$\nu_{t,\infty}/\nu$	l/H		
		WF vert.	WF zero	WF horiz.
4.0	1	7.2	6.6	5.6
4.0	10	7.2	6.6	5.6
4.0	100	6.9	6.6	5.6
4.0	1000	6.9	6.3	5.6
2.0	100	7.8	6.9	6.1
4.0	100	7.2	6.6	5.6
8.0	100	6.4	6.0	5.0
[49] standard k-ε with WF				6.6
[49] standard k-ω with WF				4.8
[48] experiments				7.0

Table 1.2: Recirculation length for Mesh 2.

the normal at this point is not well defined. When using the wall function approach, we therefore have three choices. Let the flow go vertical (denoted vert. in the table), impose a zero velocity (zero in the table), or let it go horizontal (horiz. in the table).

We now study the influence of the eddy-viscosity inflow condition. The more diffusive the flow, the more quickly the momentum of the incoming flow will dissipate down to the wall, and the shorter the recirculation length. This is precisely what is measured, although the dependence on the inflow eddy-viscosity is very light. That is good news!

We notice that the boundary condition at the step entrance is really important. If the streamwise velocity is slowed down approaching the corner point, i.e. if $d\bar{u}/dx < 0$, then from the continuity equation, the vertical component must increase. This means that when the horizontal component of the velocity is zero at the corner, the flow will rise up when approaching it, and therefore the recirculation length is expected to be greater. The table confirms this remark.

The results exhibits a high sensitivity of the recirculation length to the wall distance $\hat{y}$. We obtain variations of up to 20% for $\hat{y}/H = 2.0$ to $\hat{y}/H = 8.0$. As an indication, the experimental recirculation length is $7.0 \pm 1h$ and is taken from [48]; we also give the results by Soto [49].

Table (1.3) shows the mesh dependence of the results. When the mesh is refined, the effects of the boundary condition at the step entrance diminish. It also compares the results obtained when integrating the SA and RANS equations up to the wall, to get an insight of how the model

Wall treatment	Mesh 1	Mesh 2	Mesh 3
WF vert.	9.3	7.2	6.3
WF zero	8.2	6.6	6.0
WF horiz.	5.9	5.6	5.7
Up to the wall	5.9	5.5	5.4

Table 1.3: Recirculation length l/H for $\nu_{t,\infty}/\nu = 100$ and $\hat{y}/H = 4\%$.

behaves in the worst situation. With respect to the wall function approach, the model exhibits a much weaker dependence on the mesh size. The recirculation is under-predicted, but remains reasonable. The results of this table should be compared to those given in Table 1.2.

Figure 1.16 shows the results obtained with the SA model for two different values of $\hat{y}$. The top graphs give the variation of y^+ along the bottom wall of the channel beyond the step. The simulation using $\hat{y}/H$ =8% yields values of y^+ much larger than the recommended value of 30. The second simulation, performed with $\hat{y}/H$ =2%, yields acceptable values of y^+ all along the wall.

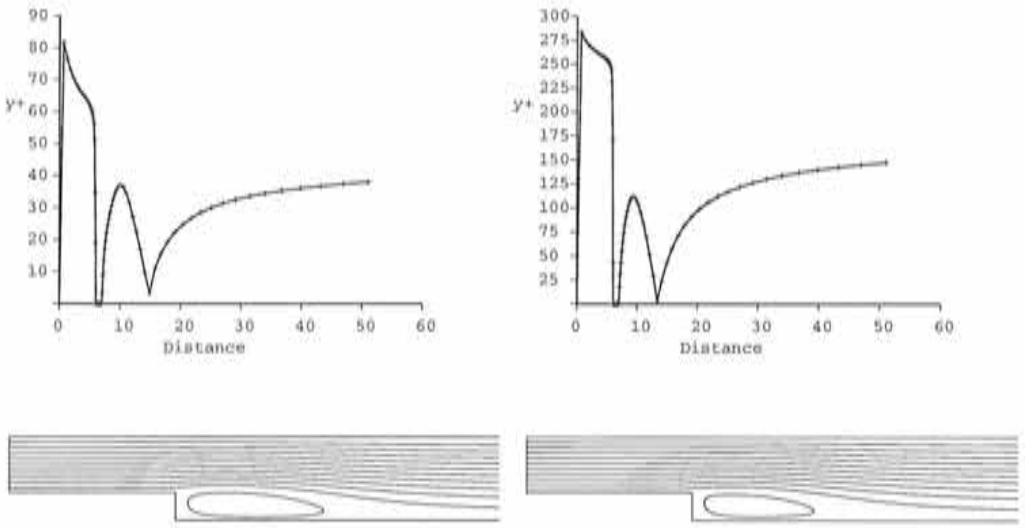
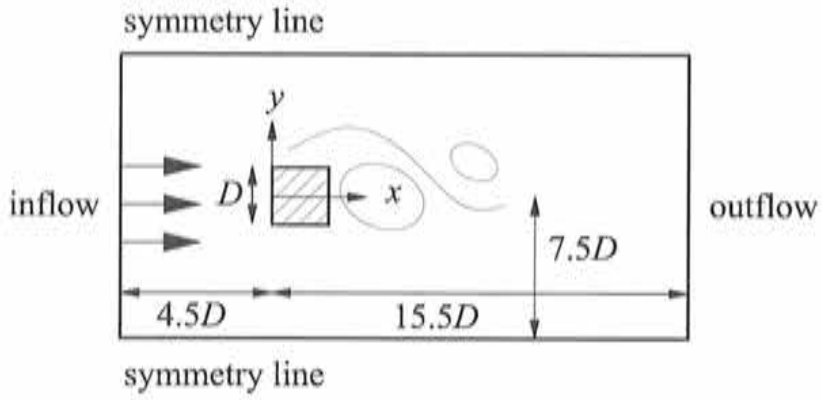


Figure 1.16: Backward facing step. $\nu_{t,\infty}/\nu = 100$. (Top) (Left) $y+$ along bottom wall for $\hat{y}/H = 2\%$. (Bot) (Left) Streamlines for $\hat{y}/H = 2\%$. (Top) (Right) $y+$ along bottom wall for $\hat{y}/H = 8\%$. (Bot.) (Right) Streamlines for $\hat{y}/H = 8\%$.

1.4.3 Flow past a square cylinder

We solve a transient turbulent flow past a square cylinder, shown in Figure 1.17 (Top). When the Reynolds number of the flow is sufficiently high, the turbulent flows undergo transient separation and exhibit periodic vortex shedding beyond the cylinder. The capture of the features of the vortex, i.e. its size and its frequency is of primary importance in engineering as it is responsible for the dynamic loading and torque exerted on the body, for example at the trailing edge of a turbine



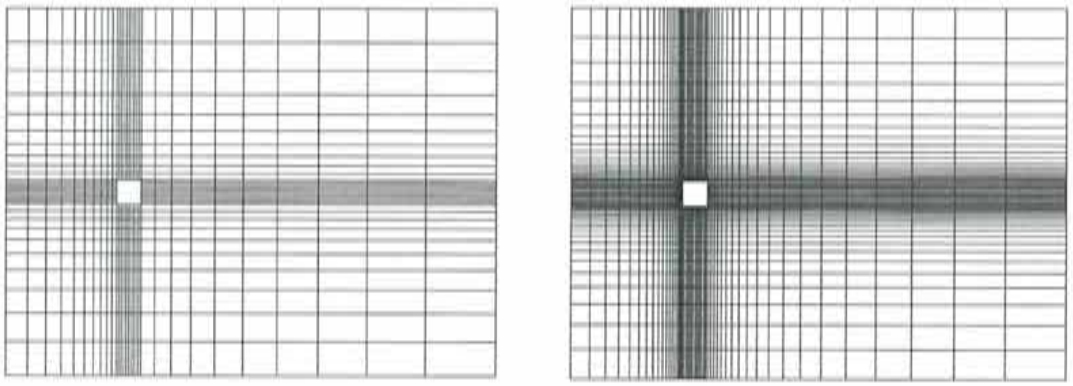


Figure 1.17: Square cylinder. (Top) Geometry. (Bot.) (Left) Mesh 1. (Bot.) (Right) Mesh 2.

blade or an airfoil, behind a building, etc. We base our numerical simulations on the experiment of Lyn *et al.* [50] and on the numerical simulations with k-ε models of Bosh *et al.* [51]. See also the experiments of Durao *et al.* [52].

Through this example, we test the ability of the turbulence model to capture the transient turbulent flow. First of all, we want to check if the numerical scheme is able to distinguish the various scales of motion in play. In the case of a stationary flow, the signal is decomposed into a mean and a fluctuating part with the fluctuating component participating to the extra diffusion through the eddy-viscosity. However, it can occur that another time scale participates. If this is the case, we hope that the turbulence model will be able to manage time-varying mean variables. The large time scale represents a coherent structure, the vortex, while the small time scale represents the random turbulence fluctuations. We will see that both time scales can be accurately captured by the SA turbulence model with wall functions. The situation in schematized in Figure 1.8 and the Reynolds number is

$$\mathrm{Re} = \frac{UD}{\nu} = 22\,000,$$

where U is the inflow velocity.

In view of treating large three-dimensional examples, we want to be able to solve this problem on a rather coarse mesh, using the law of the wall. Mesh 2 is composed of 3200 $Q1/Q1$ elements. The results on Mesh 2 will be compared to the solution obtained on a very coarse mesh, Mesh 1, composed of 800 elements. Both meshes are shown in Figure 1.17 (Bot.) (Left) and (Bot.) (Right).

Using Equation (1.49) to guess the value of $\hat{y}$ for $\hat{y}^+ = 50$ and $L = D$, we find that $\hat{y}/D = 3.2 \times 10^{-2}$. We will take a larger value as the presence of boundary layer separation will tend to diminish the characteristic friction velocities on the wall.

We first solve the stationary problem on Mesh 2 and check that at the wall of the cylinder, $\hat{y}^+$ lies within a reasonable range. Taking $\hat{y}/D = 8\%$ and as inflow condition for $\nu_t = 100\nu$, we obtain the results shown in Figure 1.18 (Top) (Left).

As expected, the stationary simulations give a symmetrical vortex. Figure 1.18 (Top) (Right) shows the profile of the streamwise velocity on the centerline. The size of the vortex is over-predicted in all cases. However, we observe that logically, a higher inflow turbulent eddy-viscosity gives a smaller vortex length. The influence of the value of $\hat{y}$ has no significant influence on the vortex length. For the sake of comparison, the figure also includes the results of the integration up to the wall, which is expected to fail for this very coarse mesh... however, the results are similar to that of the wall function approach.

Table 1.4 gives the value of the drag coefficient computed on the square. The results are very close to those obtained by the references, although the latter are results of transient simulations. Note that the references use a 4410 element mesh.

Wall treatment	$\hat{y}/D$ (%)	$\nu_{t,\infty}/\nu$	$\overline{c}_d$	S
WF	8	100	1.674	-
WF	8	1	1.738	-
WF	12	100	1.646	-
Up to the wall	-	100	1.706	-
[51] Bosh and Rodi, k-ε with WF			1.618	0.126
[53] Kato and Launder, k-ε with WF			1.660	0.127
[50] Experiment			-	0.135
[54] Experiment for $\mathrm{Re} = 5 \times 10^4$			2.19	0.123

Table 1.4: Stationary results, Mesh 2. Drag coefficient and Strouhal number.

Transient flow can be triggered by introducing a perturbation into the stationary solution, for example, by introducing a small vortex near the wall of the cylinder. We have considered two time steps, a large one $\delta t = 0.8\,U/D$ and a small one $\delta t = 0.2\,U/D$. The large time step was chosen in order to resolve the vortex shedding with approximately ten time integration steps. Using the experimental value of Lyn, we find $\delta t = (1/S)/10 \approx 0.74\,U/D$, where S is the Strouhal number, defined as

$$S = f_s \frac{D}{U},$$

f_s being the vortex shedding frequency. Euler time integration is only first order in time, and when using the coarse time step, the vortex is numerically dissipated in few time iterations. The second order scheme is more adequate to study fine transient simulations. Figure 1.18 (Mid.) (Left) presents the vertical velocity profile of a point P located downstream the cylinder as a function of

time. It can be appreciated how that the second order Crank-Nicolson scheme performs well when the Euler scheme fails, i.e. for $\delta t = 0.8\,U/D$.

Figure 1.18 (Mid.) (Right) presents the results of a fast Fourier transform performed on the vertical velocity at P, when the periodic motion is well established (the spectrum were normalized by their maximum values). The spectra shows a clear periodic motion, of higher frequency for the small time step. From this plot, we compute the Strouhal number. Values of S are shown in Table 1.5 for the different time integration schemes and time steps used. The table also gives the mean drag coefficient.

Mesh, time integration			$\overline{c}_d$	S
Mesh 1,	1^{st} order,	$\delta t = 0.8\,U/D$	1.516	-
Mesh 1,	2^{nd} order,	$\delta t = 0.8\,U/D$	1.531	0.092
Mesh 1,	2^{nd} order,	$\delta t = 0.2\,U/D$	1.565	0.107
Mesh 2,	1^{st} order,	$\delta t = 0.8\,U/D$	1.670	-
Mesh 2,	2^{nd} order,	$\delta t = 0.8\,U/D$	1.854	0.099
Mesh 2,	2^{nd} order,	$\delta t = 0.2\,U/D$	1.860	0.126
[51] Bosh and Rodi, k-ε with WF			1.618	0.126
[53] Kato and Launder, k-ε with WF			1.660	0.127
Experiments (see [51])			2.05-2.19	0.135-0.139

Table 1.5: Transient results, Mesh 2. Drag coefficient and Strouhal number.

Finally, the streamwise velocity profile at the centerline is time averaged and compared to experimental and numerical results in Figure 1.18 (Bot.) (Left) for Mesh 1 and 1.18 (Bot.) (Right) for Mesh 2.

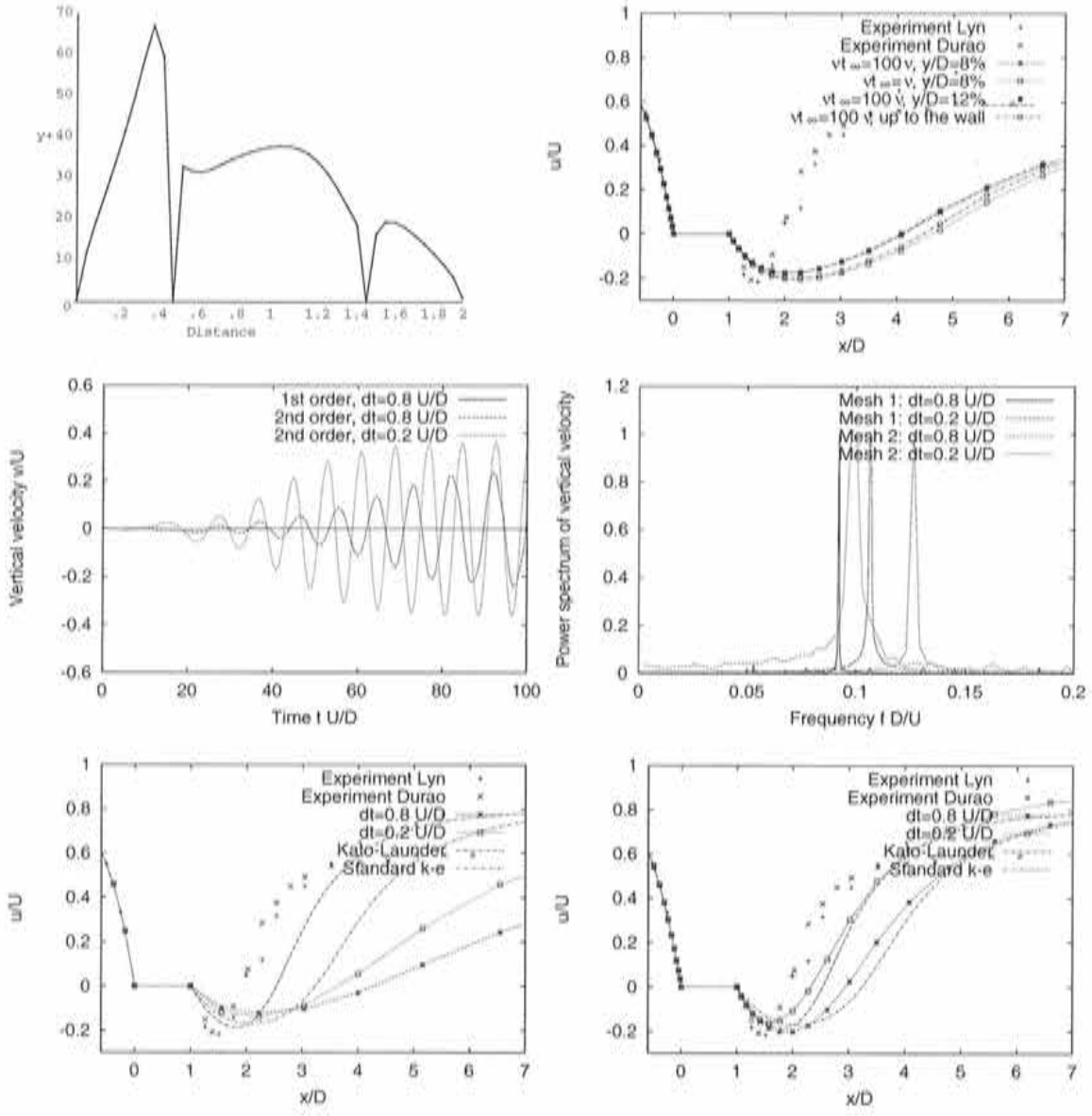


Figure 1.18: Square cylinder. (Top) (Left) Stationary results, Mesh 2. $\hat{y}^+$ on the wall of the cylinder for $\hat{y}/D = 8\%$; by symmetry, only the upper part is shown. (Top) (Right) Stationary results. u/U at centerline on Mesh 2. (Mid.) (Left) Mesh 2. Vertical velocity at point P. (Mid.) (Right) Power spectrum of the signal at point P using Crank-Nicolson scheme. (Bot.) (Left) Transient results. u/U at centerline on Mesh 1. (Bot.) (Right) Transient results. u/U at centerline on Mesh 2.

1.5 Conclusion

In this chapter we have studied a numerical strategy to solve turbulent flows. The numerical examples have shown the possibilities as well as the limitations of the model. The use of the shock capturing technique and under-relaxation for the eddy-viscosity are in general necessary to make the whole numerical strategy reliable and robust. On the physical level, the model has proven to be capable of differentiating the various time scales of unsteady flows; this possibility is auspicious in view of simulating complex unsteady flows, like for example rotodynamic flows which exhibit strong unsteady patterns. For high Reynolds number flows, inasmuch we generally prefer to use the wall function approach as a near-wall treatment but we must be aware of its numerical drawbacks and physical limitations. In particular, we have shown the dependence of the results upon the prescribed distance of the computational boundary to the real wall. However, the wall function approach we have derived can be used for any value of this distance and if unsatisfactory results are obtained, smaller values can be chosen. In addition, small values of the distance are physically preferable as the incidence of the wall functions on the resolution of the boundary layer will be lower.

Chapter 2

Domain Decomposition Methods: A Guide Overview and Proposal

In this chapter, we first introduce domain decomposition methods at the differential level. We start by studying a one-dimensional example in a more intuitive than rigorous manner. Although very simple, this example is sufficient to present all the families of domain decomposition methods we will deal with. In particular, we contemplate the possibility of using mixed DD methods on non-overlapping as well as on overlapping subdomains. Then, we consider the more general advection-diffusion-reaction equation, and describe all the possible improvements that can be achieved to the classical DD methods. In particular, we will mention the adaptive methods. At this point, we will have introduced the necessary terminology to proceed with the variational approach. The weak formulation of the domain decomposed problem will enable us to justify the choice of transmission conditions, involving the essential and natural conditions. Then we discuss some ways of applying these methods to the finite element method and finally we present the proposal of this work.

2.1 Study of a simple problem

We want to solve the following one-dimensional Poisson problem

$$\left\{ \begin{array}{rcll} -\dfrac{d^2u}{dx^2} & = & f & \forall\, x \in \Omega := (-1,1), \\ u & = & 0 & \text{at } x = -1, 1. \end{array} \right. \tag{2.1}$$

We partition Ω into two subdomains Ω_1 and Ω_2, disjoint or overlapping, such that $\Omega_1 = (-1, \delta)$ and $\Omega_2 = (-\delta, 1)$ with $0 \leq \delta < 1$. The solution on each subdomain is uniquely defined by prescribing the unknown u or its first derivative, or, more generally, a linear combination of both. We propose to solve in subdomain Ω_1 a problem of the form

$$\left\{ \begin{array}{rcll} -\dfrac{d^2u_1}{dx^2} & = & f & \forall\, x \in \Omega_1, \\ u_1 & = & 0 & \text{at } x = -1, \\ \alpha_1 u_1 + \beta_1 \dfrac{du_1}{dx} & = & g_1 & \text{at } x = \delta, \end{array} \right. \tag{2.2}$$

where α_1 and β_1 are two constants such that $\alpha_1 \neq 0$ or $\beta_1 \neq 0$, and in subdomain Ω_2 a problem of the form

$$\left\{\begin{array}{rcll} -\dfrac{d^2u_2}{dx^2} & = & f & \forall\, x \in \Omega_2, \\ u_2 & = & 0 & \text{at } x = 1, \\ \alpha_2 u_2 + \beta_2 \dfrac{du_2}{dx} & = & g_2 & \text{at } x = -\delta, \end{array}\right. \tag{2.3}$$

where α_2 and β_2 are two constants such that $\alpha_2 \neq 0$ or $\beta_2 \neq 0$. Boundary conditions $(2.2)_3$ and $(2.3)_3$ are referred to as Robin conditions, i.e. a linear combination of Dirichlet and Neumann conditions. The coefficients α_1 and α_2 will be referred to as the Robin coefficients.

The goal of domain decomposition methods is to construct the solution of the original problem from the solutions u_1 and u_2 on each subdomain. To do so, we need to find expressions for g_1 and g_2 such that

$$\begin{aligned} u_1 &= u_{|\Omega_1}, \\ u_2 &= u_{|\Omega_2}. \end{aligned}$$

For reasons that are going to appear clear along the discussion, we have to treat the disjoint case ($\delta = 0$) and overlapping case ($\delta > 0$) separately. We first consider disjoint subdomains.

Disjoint subdomains

To treat disjoint subdomains, we set $\delta = 0$. Intuitively, in order to have a well defined problem in Ω, the solution must have a certain degree of regularity in Ω. For this second order problem, it is expected that

$$\frac{d^2u}{dx^2} \in L^\infty \Rightarrow \frac{du}{dx} \in \mathcal{C}^0 \Rightarrow u \in \mathcal{C}^1,$$

which means that both the unknown and its first derivative must be continuous anywhere, and in particular on the interface. For this one dimensional problem, the first order derivative at the interface is the flux. To illustrate the need for this double continuity on the interface, let us assume we solve each subproblem $(2.2)_{1-3}$ and $(2.3)_{1-3}$ imposing g_1 and g_2. We first assume that only the continuity of the unknown at $x = 0$ is imposed by setting $\alpha_1 = \alpha_2 = 1$, $\beta_1 = \beta_2 = 0$, and $g_1 = g_2 = g$ where g is arbitrary. Figure 2.1 (Left) shows that the continuity of only the unknown at $x = 0$ is not sufficient to recover the original solution of the original problem, i.e. we have $u_1 \neq u_{|\Omega_1}$ and $u_2 \neq u_{|\Omega_2}$ for an arbitrary g. Now we set $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1$, and $g_1 = g_2 = g$ where g is arbitrary. Figure 2.1 (Right) illustrates that the continuity of the first derivative of the unknown is not sufficient to recover the original solution which is expected to be continuous.

According to this, we set:

$$g_1 = \alpha_1 u_2 + \beta_1 \frac{du_2}{dx}, \tag{2.4}$$

$$g_2 = \alpha_2 u_1 + \beta_2 \frac{du_1}{dx}. \tag{2.5}$$

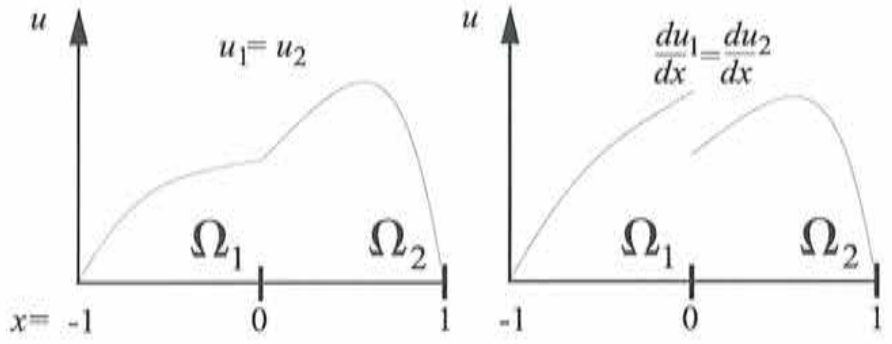


Figure 2.1: Example of solution to the Poisson equation for disjoint subdomains. (a) Continuity of the unknown at the interface. (b) Continuity of the derivative of the unknown at the interface.

The imposition of the continuities of the unknown and its first derivative implies that the coefficients α_1, α_2, β_1 and β_2 must be chosen such that

$$|\alpha_1\beta_2 - \alpha_2\beta_1| \neq 0; \tag{2.6}$$

Then we have at $x = 0$ the following two equalities:

$$\begin{aligned} u_1 &= u_2, \\ \frac{du_1}{dx} &= \frac{du_2}{dx}. \end{aligned}$$

If this is the case, the variable u defined as

$$u = \begin{cases} u_1 \text{ in } \Omega_1, \\ u_2 \text{ in } \Omega_2, \end{cases}$$

is solution of Problem (2.1). In particular, we see that the Dirichlet/Dirichlet coupling ($\beta_1 = \beta_2 = 0$) as well as the Neumann/Neumann ($\alpha_1 = \alpha_2 = 0$) coupling are not possible. Many choices of Robin conditions are possible, for in principle, the coefficients α_1 and β_1 (and/or α_2 and β_2) can take any values different from zero as long as Equation (2.6) holds. However, the Robin condition has no interest in solving such a simple problem; indeed, we remark that only the Neumann condition is a natural condition of the weak formulation. The Robin condition will be presented later on when studying the ADR equation. DD methods employing different type of boundary conditions are called *mixed domain decomposition methods.*

Overlapping subdomains

We now consider the overlapping case. As in the disjoint case, we assume that g_1 and g_2 are given by Equations (2.4) and (2.5). We are first going to show that the solution in the overlapping zone $\Omega_1 \cap \Omega_2$ is the same. Let us define $w = u_1 - u_2$. Subtracting Equation $(2.2)_1$ from $(2.3)_1$, and

Equation $(2.2)_2$ from $(2.3)_2$ using Equations (2.4) and (2.5), we obtain

$$\begin{cases} \dfrac{d^2w}{dx^2} = 0 & \forall\, x \in (-\delta, \delta), \\ \alpha_1 w + \beta_1 \dfrac{dw}{dx} = 0 & \text{at } x = \delta, \\ \alpha_2 w + \beta_2 \dfrac{dw}{dx} = 0 & \text{at } x = -\delta. \end{cases} \tag{2.7}$$

If α_1 or α_2 are not zero at the same time, this problem has a unique solution w, and therefore, $u_1 = u_2$ in $(-\delta, \delta)$. Also, we have necessarily continuity of the derivatives of u_1 at $x = -\delta$ and u_2 at $x = \delta$, as u_1 and u_2 are both solutions of the local differential problems. Hence, the variable u defined as

$$u = \begin{cases} u_1 \text{ in } \Omega_1, \\ u_2 \text{ in } \Omega \backslash \Omega_2, \end{cases}$$

or equivalently defined as

$$u = \begin{cases} u_1 \text{ in } \Omega \backslash \Omega_1, \\ u_2 \text{ in } \Omega_2, \end{cases}$$

is solution of Problem (2.1).

The case $\alpha_1 = \alpha_2 = 0$ corresponds to the overlapping Neumann/Neumann coupling, for which the solution of Problem (2.7) is defined up to a constant.

Relation between disjoint and overlapping methods

Let us consider the additive Schwarz method with a small overlap. The transmission conditions of the Schwarz method are

$$\begin{aligned} u_1 &= u_2 \qquad \text{at } x = \delta, \\ u_2 &= u_1 \qquad \text{at } x = -\delta. \end{aligned}$$

Performing a forward and backward Taylor expansions at $x = 0$, the latter two conditions can be substituted by imposing at $x = 0$ the following two conditions

$$\begin{aligned} u_1 + \delta \frac{du_1}{dx} &= u_2 + \delta \frac{du_2}{dx}, \\ u_2 - \delta \frac{du_2}{dx} &= u_1 - \delta \frac{du_1}{dx}. \end{aligned}$$

This is precisely a Robin/Robin coupling for disjoint subdomains with the choice $\alpha_1 = \alpha_2 = 0$ and $\beta_1 = -\beta_2 = \delta$. This explains why the R/R method is often referred to as the fictitious overlapping method [55].

Iterative solution

We have shown how the solution of the original problem can be constructed from the local solutions on two disjoint or overlapping subdomains, using transmission conditions of Dirichlet, Neumann or more generally of Robin type. These transmission conditions involve the unknown and/or its first derivative and are sufficient to uniquely define the local solutions. In addition, we have shown that they must be different in the case of disjoint subdomains but can be equal in the case of overlapping subdomains, except when both conditions are of Neumann type.

In practice, Problems $(2.2)_{1-3}$ and $(2.3)_{1-3}$ are solved iteratively. Given u_1^0 and u_2^0, for each $k \geq 0$, solve successively the following two problems:

$$\begin{cases} -\dfrac{d^2 u_1^{k+1}}{dx^2} = f & \forall\, x \in \Omega_1, \\ u_1^{k+1} = 0 & \text{at } x = -1, \\ \alpha_1 u_1^{k+1} + \beta_1 \dfrac{du_1^{k+1}}{dx} = \alpha_1 u_2^k + \beta_1 \dfrac{du_2^k}{dx} & \text{at } x = \delta, \end{cases} \tag{2.8}$$

and

$$\begin{cases} -\dfrac{d^2 u_2^{k+1}}{dx^2} = f & \forall\, x \in \Omega_2, \\ u_2^{k+1} = 0 & \text{at } x = 1, \\ \alpha_2 u_2^{k+1} + \beta_2 \dfrac{du_2^{k+1}}{dx} = \alpha_2 u_1^{k'} + \beta_2 \dfrac{du_1^{k'}}{dx} & \text{at } x = -\delta, \end{cases} \tag{2.9}$$

where k' is an iteration index which can be

$$k' = \begin{cases} k & \text{for the parallel version,} \\ k+1 & \text{for the sequential version.} \end{cases}$$

The choice $k' = k$ corresponds to the parallel version also referred to as additive, in reference to the additive Schwarz method; the choice $k' = k+1$ corresponds to the sequential version, in reference to the multiplicative Schwarz method. In addition, one or both transmission conditions can be relaxed in order to gain control on the convergence of the algorithm.

2.2 DD for the advection-diffusion-reaction equation

2.2.1 Disjoint methods

We consider the more general advection-diffusion-reaction (ADR) equation (1.1) in n_d dimensions. For the sake of clarity, we assume that we update the transmission condition of subdomain Ω_i at iteration $k+1$ knowing the solution on subdomain Ω_j at iteration k. Generally, the interface transmission conditions of Neumann and Robin types try to mimic as much as possible the natural conditions of the associated weak forms. We set therefore:

$$\rho_i u_i^{k+1} + \varepsilon \frac{\partial u_i^{k+1}}{\partial n_i} = \rho_i u_j^k + \varepsilon \frac{\partial u_j^k}{\partial n_i}, \tag{2.10}$$

$\boldsymbol{n}_i$ being the outward unit vector normal to the interface of Ω_i and where the coefficients ρ_i have to be determined for all the interfaces. Let us take for example the 1/2-weak formulation presented in Section (1.1.3). If we choose the transmission condition to be the natural condition, then we have to choose

$$\rho_i = -\frac{1}{2}\boldsymbol{a} \cdot \boldsymbol{n}_i.$$

However, this is not a restriction and the transmission condition can be penalized iteratively. Following with our example, we could also impose a transmission condition of the form

$$-\frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_i)u_i^{k+1} + u_i^{k+1} + \varepsilon\frac{\partial u_i^{k+1}}{\partial n_i} = -\frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_i)u_j^k + u_j^k + \varepsilon\frac{\partial u_j^k}{\partial n_i}.$$

We now present the most common DD methods. The family of DD using disjoint subdomains is referred to according to the type of transmission conditions used. All the possible couplings lead to the following methods:

- Dirichlet/Neumann (D/N) method,
- Robin/Robin (R/R) method,
- Robin/Neumann (R/N) and Dirichlet/Robin (D/R) methods,
- Dirichlet/Dirichlet (D/D) method, i.e. the Schwarz method.

The D/N method was first considered in [56] while its first application to the FEM is due to Marini and Quarteroni [57]. They study the D/N method in a variational context for a two-subdomain partition and show the convergence of the iterative method for the discretized interface problem. The method is extensively reviewed in [58]. In [59], the relaxed D/N algorithm is applied to second-order elliptic problems. The authors derive a method to compute iteratively the relaxation parameters to achieve exact convergence in a finite number of iterations.

In [60], Lions introduces the R/R method as a generalization of the Schwarz method to non-overlapping subdomains. He studies a multidomain formulation for the solution of the continuous Poisson problem and the generalization to the ADR equation. In particular, he shows the strong convergence of the algorithm but leaves open the question on the choice of the Robin factors. In [61], the authors reinterpret this method applied to elliptic problems within an augmented Lagrangian framework for two subdomains.

The inclusion of first order derivatives in the ADR equation with respect to the simple Poisson equation adds some difficulties, related to the fact that Dirichlet and Neumann conditions must be imposed in accordance with the direction of the flow when advection is dominant. This requirement is at its turn closely related to the well-posedness of the local variational problems for which essential and natural conditions are needed, as was shown in Section 1.1.3. This was the argument for developing the so-called adaptive methods. Adaptive domain decomposition methods have been derived for disjoint subdomains to take into account the direction of the flow on the interfaces. In [62], a DD method for the solution of a transport equation on two subdomains of the advection-reaction type is presented. A Dirichlet condition is imposed on inflows, and outflows are left free. The authors study the method through an iterative Steklov-Poincaré formulation, and show the convergence of the Richardson procedure.

Carlenzoli and Quarteroni [63] introduce the adaptive D/N for which a Dirichlet transmission condition at inflow and Neumann transmission condition at outflow and also an adaptive R/N method, further considered in [64]. These methods are reviewed in [65].

In [66], Trotta presents some adaptive methods, namely the adaptive D/N, the adaptive R/N and their damped versions. In damped versions, the flux are not calculated when the diffusion is sufficiently small. All these methods are considered and applied in [67].

In its actual form the R/R method can be viewed as an adaptive method as the Robin factors depend on the direction on the flow. Auge, Kapurkin, Lube and Otto [68] consider a R/R method applied to the solution of singularly perturbed elliptic problems. In Equation (2.10), they define ρ_i as

$$\rho_i = -\frac{1}{2}\left(\boldsymbol{a}\cdot\boldsymbol{n}_i - \gamma_i\right),$$

where the presence of the term $\gamma_i \neq 0$ is necessary in order to satisfy Equation (2.6). They show the convergence of the algorithm and suggest that γ_i should be of the form

$$\gamma_i = \sqrt{(\boldsymbol{a}\cdot\boldsymbol{n}_i)^2 + \lambda\varepsilon}, \tag{2.11}$$

λ being a well-chosen positive constant. Once again, we note that the second term of the last equation is required to satisfy Equation (2.6) when the advection is parallel to the interface. By using an approximate factorization of the advection-diffusion operator as transmission conditions, Nataf and Rogier [69] give a similar expression for γ_i:

$$\gamma_i = \sqrt{(\boldsymbol{a}\cdot\boldsymbol{n}_i)^2 + 4s\varepsilon}.$$

We note that a coupling using this transmission condition does not make sense when $s = 0$ and when the advection is parallel to the interface. Finally, we note that the simple choice $\lambda = 0$ gives

$$\gamma_i = |\boldsymbol{a}\cdot\boldsymbol{n}_i|,$$

which is the adaptive R/N method [63]. In fact, we can check that in this case, the transmission condition given by Equation (2.10) involves the following quantities

$$\begin{cases} \varepsilon\dfrac{\partial u_i}{\partial n_i} & \text{at outflows,} \\ \varepsilon\dfrac{\partial u_i}{\partial n_i} - (\boldsymbol{a}\cdot\boldsymbol{n}_i)u_i & \text{at inflows.} \end{cases}$$

This method also fails when the advection is parallel to the interface. The design of the coefficient γ_i can be based, for example, on the study of the singularly perturbed case. On the one hand, when the diffusion ε tends to zero, we want the Neumann part of the Robin condition to dominate in order to avoid artificial exponential layers near the interface so that

$$\lim_{\varepsilon\to 0}\gamma_i = |\boldsymbol{a}\cdot\boldsymbol{n}_i|.$$

On the other hand, in parabolic layers, the Dirichlet and Neumann parts of the Robin condition must balance. This justifies the term proportional to the diffusion in Equation (2.11). See for example [70].

Lube, Müller and Otto [71, 72] review the R/R method presented in [68] and apply it to a transient ADR equation.

Alonso, Trotta and Valli [73] introduce a coercive γ-D/R and a γ-R/R which generalizes the latter R/R method. They consider the case of a two-subdomain partition and for the γ-R/R they set $\gamma_1 = \gamma$ and $\gamma_2 = \gamma$. They propose a variational framework for the study of these methods and show how γ must be chosen in order to achieve convergence for both methods. In the case of the γ-D/R, they obtain linear convergence while no information on the convergence rate is obtained for the other method.

We should also mention the Neumann/Neumann method [74], although it cannot be expressed as an iteration-by-subdomain method as given by Equations (2.8) and (2.9).

2.2.2 Overlapping methods

For overlapping subdomains, we have seen that only the N/N coupling is not possible. The method using a D/D coupling was first studied by Schwarz [75] and recently, Lions [76] reconsidered the method in a new light. See for example [77] for the convergence of the Schwarz method applied to a singularly perturbed advection-diffusion equation. The other choices of couplings on overlapping subdomains can be viewed as a generalization of the methods devised for disjoint subdomains. *The overlapping versions of mixed methods have not received particular attention and are precisely the basis of this work* (see for example [78, 79, 80]). As noted by Lions [1]:

> [...] *the Schwarz algorithm* [...] *presents some properties (like "robustness", or indifference to the type of equations considered...) which do not seem to be enjoyed by other methods.*

So what happens when mixed methods are used on overlapping subdomains? Can we expect some of the robustness of the overlapping Schwarz method to be inherited by overlapping mixed methods? In the next chapter, we apply some overlapping mixed coupling to a one-dimensional ADR equation and try to answer this question. In Chapter 4, we introduce a possible mathematical framework of study of such methods, taking the example of an overlapping D/R method.

2.3 Implementation of DD methods

In this section, we are going to present a general framework for designing mixed domain decomposition methods for finite element applications. The discussion that follows is not absolutely formal; the formal mathematical framework will be presented in Chapter 4.

We want to solve a partial differential equation of the form

$$\begin{aligned} Lu &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned}$$

where L is a differential operator given by Equation (1.1) and f a given function. We assume we look for solutions in $V = H_0^1(\Omega)$. The last equation can be reformulated in a variational way as follows: find $u \in V$ such that

$$a(u, v) = \langle f, v \rangle_\Omega \qquad \forall\, v \in V, \tag{2.12}$$

where the bilinear form can be anyone of the three bilinear forms considered in the last Chapter and given by Equations (1.5), (1.7), or (1.8).

We divide Ω into two non-overlapping subdomains Ω_1 and Ω_2 with interface Γ. Let us introduce the spaces $V_1 = H_0^1(\Omega_1)$ and $V_2 = H_0^1(\Omega_2)$. We also define $\Lambda = H^{1/2}(\Gamma)$. The iterative domain

decomposition problem reads as follows: given initial guesses, for each $k \geq 0$, find $u_1^{k+1} \in V_1$ and $u_2^{k+1} \in V_2$ such that

$$\begin{cases} a_1(u_1^{k+1}, v_1) = \langle f, v_1 \rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1^{k+1} = u_2^k & \text{on } \Gamma, \\ a_2(u_2^{k+1}, v_2) = \langle f, v_2 \rangle_{\Omega_2} & \forall\, v_2 \in V_2^0, \\ a_2(u_2^{k+1}, E_2\mu) = -a_1(u_1^{k+1}, E_1\mu) + \langle f, E_1\mu \rangle_{\Omega_1} + \langle f, E_2\mu \rangle_{\Omega_2} & \forall\, \mu \in \Lambda, \end{cases} \tag{2.13}$$

where E_1 and E_2 can be any extension operators from Λ to V_1 and V_2, respectively. It can be shown that if Algorithm $(2.13)_{1-4}$ converges, then its solution is that of Problem (2.12) (see e.g. [58]). Equation $(2.13)_2$ represents the Dirichlet transmission condition, i.e. the continuity of the primary variable. Let g_i be the quantity involved in the natural boundary conditions of the weak formulation of subdomain i for $i = 1, 2$, i.e. g_i is the boundary term coming from the integration by parts of the bilinear form. Therefore, it can be easily shown that Equation $(2.13)_4$ implies the continuity of g_i across the interface, i.e. we have that

$$g_1^{k+1} = g_2^{k+1} \qquad \text{on } \Gamma.$$

Obviously, the nature of g_i depends on how the terms of the original differential equation are integrated by parts, and it is given by Equations (1.5), (1.7), or (1.8) according to the bilinear form used. If we consider the 0-weak formulation, i.e. if only the diffusive term is integrated by parts, the transmission condition is nothing but the diffusive flux across the interface. By using the 1/2-weak formulation or 1-weak formulation, the transmission condition would involve a Robin condition. This states that Problem $(2.13)_{1-4}$ can be derived from the continuous differential equation using a Dirichlet transmission condition when solving subdomain 1 and a Neumann (or Robin) transmission condition when solving subdomain 2. Therefore, at the differential level, we formulate the DD method as follows: given initial guesses, for each $k \geq 0$, find u_1^{k+1} and u_2^{k+1} such that

$$\begin{cases} Lu_1^{k+1} = f & \text{in } \Omega_1, \\ u_1^{k+1} = 0 & \text{on } \partial\Omega_1 \backslash \Gamma, \\ u_1^{k+1} = u_2^k & \text{on } \Gamma, \\ Lu_2^{k+1} = f & \text{in } \Omega_2, \\ u_2^{k+1} = 0 & \text{on } \partial\Omega_2 \backslash \Gamma, \\ g_2^{k+1} = g_1^{k+1} & \text{on } \Gamma. \end{cases}$$

We recover the mixed domain decomposition method introduced at the beginning of this Chapter.

Now let us go the discrete level and assume that the nodes in the overlapping region and on the interfaces coincide. As in the continuous case, we have that if Algorithm $(2.13)_{1-4}$ converges, then its solution is that of Problem (2.12). However, when dealing with discrete subspaces, Equation $(2.13)_4$ *no* longer implies the continuity of the discrete counter parts of the g_i's. This means that the finite element DD problem cannot be derived directly from the continuous differential problems without special care. This point will be treated in Chapter 5 and is related to the fact that generally $Lu_h \neq f$ where u_h is the finite element solution in Ω_1 or Ω_2. Let us give up the

differential formulation for a moment.

According to the space discretization used, we denote $\mathbf{u}$ the vector of unknowns in Ω. Problem (2.12) leads to an algebraic system of the form:

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \tag{2.14}$$

We denote $\mathbf{u}_1$ and $\mathbf{u}_2$ as the vectors of unknowns of Ω_1 and Ω_2 respectively, excluding the interface vector of unknowns that we denote $\mathbf{u}_a$. By performing a simple node reordering, System (2.14) can be written as:

$$\begin{pmatrix} \mathbf{A}_{11} & 0 & \mathbf{A}_{1a} \\ 0 & \mathbf{A}_{22} & \mathbf{A}_{2a} \\ \mathbf{A}_{a1} & \mathbf{A}_{a2} & \mathbf{A}_{aa} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_a \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_a \end{pmatrix}.$$

The solution of this system yields for the interface unknown the so-called *Schur complement system*

$$\mathbf{Q}_a \mathbf{u}_a = \chi_a,$$

where $\mathbf{S}_a$ and $\mathbf{s}_a$ are given by

$$\begin{aligned} \mathbf{Q}_a &= \mathbf{Q}_a^{(1)} + \mathbf{Q}_a^{(2)}, \\ \mathbf{Q}_a^{(i)} &= \mathbf{A}_{aa}^{(i)} - \mathbf{A}_{ai}\mathbf{A}_{ii}^{-1}\mathbf{A}_{ia} \qquad \text{for } i = 1, 2, \\ \chi_a &= \mathbf{f}_a - \mathbf{A}_{a1}\mathbf{A}_{11}^{-1}\mathbf{f}_1 - \mathbf{A}_{a2}\mathbf{A}_{22}^{-1}\mathbf{f}_2, \end{aligned}$$

and the matrices $\mathbf{A}^{(i)}$ for $i = 1, 2$ are the contributions to the matrix $\mathbf{A}_{aa}$ from each subdomain to the interface unknowns such that

$$\mathbf{A}_{aa} = \mathbf{A}_{aa}^{(1)} + \mathbf{A}_{aa}^{(2)}.$$

We now consider the algebraic equivalent of Problem $(2.13)_{1-4}$. This problem leads to two decoupled algebraic systems to be solved sequentially. By performing some algebraic calculations, these two systems can be reduced to a low dimensional algebraic equation for the unknown on the interface $\mathbf{u}_a$. The resulting algorithm is a preconditioned Richardson procedure for the interface unknown of the form

$$\mathbf{u}_a^{k+1} = \mathbf{P}^{-1}(\chi_a - \mathbf{Q}_a \mathbf{u}_a^k) + \mathbf{u}_a^k,$$

where the preconditioner $\mathbf{P}$ is given by

$$\mathbf{P}^{-1} = \mathbf{Q}_a^{(2)^{-1}},$$

and depends directly on the integration by parts which is performed to obtain the weak formulation. For example using the 0-weak formulation we obtain the so-called Dirichlet/Neumann preconditioner, while the other two weak formulations lead to Dirichlet/Robin-like preconditioners. There are others possibilities, like the Neumann/Neumann method [74] mentioned previously which consists in taking

$$\mathbf{P}^{-1} = \sigma_1 \mathbf{Q}_a^{(1)^{-1}} + \sigma_2 \mathbf{Q}_a^{(2)^{-1}},$$

with σ_1 and σ_2 being two positive constants. This method has also a differential interpretation [58]. In [81], the authors set up a Robin/Robin preconditioner. Berselli and Saleri [82] modify the classical Dirichlet/Neumann and Neumann/Neumann preconditioners to obtain symmetric and positive definite preconditioners in order to solve nonsymmetric elliptic problems.

We have established the link between Dirichlet/Neumann(Robin) iteration-by-subdomain methods and Richardson procedures for solving the Schur complement system. From a more general algebraic point of view, domain decomposition methods [83] aim at devising well preconditioned iterative procedures for solving efficiently the Schur complement system. As a first improvement the Richardson procedure can be *accelerated* by using Conjugate gradient or Krylov subspace methods. In addition, such algorithms are in general not *scalable*, i.e. the convergence deteriorates as the number of subdomains increases. Therefore, a *coarse grid* preconditioner may be necessary to provide a global communication between the subdomains [56, 84, 81]: this is the multilevel approach. For a general discussion, see the survey papers [85, 55] or the books [83, 58].

In the previous section, we presented the Schwarz method in a differential context. Within the algebraic framework, the Schwarz method is used as a preconditioner, generally in its additive form, although the multiplicative form can be used together with a multicoloring technique [86, 83]. The Schwarz method can also be related to an iterative procedure for solving the Schur complement system, as shown in [87].

The domain decomposition methods we have been talking about are called geometric, in contrast with algebraic DD methods [88] which decouple the computation working directly on the global matrix $\mathbf{A}$ without any knowledge of the geometry. The former have the advantage that they can take into account the local characteristic of the flow to design efficient methods; the latter have the practical advantage that they can be used as a "black box" integrated in the algebraic solver, at the risk of being less efficient.

We have briefly explained how geometric DD methods for disjoint subdomains lead to a reduced problem for the interface unknowns, as well as the particular case of the overlapping Schwarz method. In Chapter 4, the variational formulation of the DD problem $(2.13)_{1-4}$ will be extended to the case of overlapping subdomains. We will show that all the previous discussion on non-overlapping subdomains can be extended to the case of mixed DD methods using overlapping subdomains. In particular, we will establish the link and sometimes the equivalence between the differential, variational and algebraic formulations.

2.4 DD for the (Navier-)Stokes equations

In principle, the application of the DD techniques mentioned previously to the solution of the Stokes and Navier-Stokes equations can be done. We refer the reader to Quarteroni and Valli's book [58] (see also [89]) for the Dirichlet/Neumann method applied to the Stokes and [90] for DD methods applied to the Navier-Stokes equations. See also [70] for the R/R applied to the Stokes and Oseen equations and [91] for the R/R method applied to the non-stationary Navier-Stokes equations. For convergence results of the Schwarz method applied to the Stokes equations see [76], and see [92] for the application to the Navier-Stokes equations. When dealing with non-conforming grids, we have several possibilities that will be described in Section 5.6. Here, we just mention the mortar element method [93], the fictitious domain method [94] and the Chimera method [95].

2.5 Proposal

Algebraic as well as geometric methods (in the sense explained above) work on conforming discretizations. Our goal in this work is to design DD methods to be used on non-conforming non-overlapping or overlapping grids, involving moving subdomains. Therefore we have no algebraic equivalent and we must act on the purely geometrical level. More clearly, we would like to use as transmission conditions the essential and natural conditions to build an iteration-by-subdomain method. Such a strategy would enable us to use different finite element approximations on different grids and to solve different physics on each subdomains (heterogeneous DD methods). In addition, the possibility of using overlapping subdomains would provide a very powerful tool to treat complex geometries by simplifying the meshing process, for example by designing Chimera methods with mixed couplings. Finally, if we are able to build a Master/Slave strategy to exchange transmission conditions between the subdomains, then the modifications to the original finite element solver would be minimum. The purpose of this work is to devise such an algorithm.

Chapter 3

One Dimensional Analysis

In this chapter, we analyze three families of overlapping and non-overlapping domain decomposition methods for solving the one-dimensional advection-diffusion-reaction. The first one is the Dirichlet/Dirichlet method, or Schwarz method, which is a purely overlapping method; the second one uses a Dirichlet/Neumann coupling for overlapping or disjoint subdomains, the latter version being the classical Dirichlet/Neumann method; the third one uses a Dirichlet/Robin method for overlapping or disjoint subdomains, the latter version being the classical Dirichlet/Robin method. In the first section, we present the domain decomposition algorithms. Next, we study the convergence of the unrelaxed sequential algorithm, and, in particular, its dependence upon the overlapping length. Apart from the general ADR equation, we study three limiting behaviors of the equation, i.e. the Poisson equation, the advection-diffusion equation, and the hyperbolic limit. Then we study the relaxed sequential versions. In the following two sections, we consider the unrelaxed and relaxed parallel methods. Finally, we sum up the results obtained and draw some conclusions.

3.1 Problem statement

We consider the following advection-diffusion-reaction equation in one dimension

$$\begin{cases} Lu := -\varepsilon \dfrac{d^2u}{dx^2} + a\dfrac{du}{dx} + su = f & \forall\, x \in \Omega = (-\ell_1, \ell_2), \\ u = 0 & \text{at } x = -\ell_1, \ell_2, \end{cases} \tag{3.1}$$

where ℓ_1 and ℓ_2 are positive. We assume ε, and s are constants such that

$$\varepsilon > 0, \qquad s \geq 0,$$

and impose a constant advection a such that

$$a \geq 0.$$

The Péclet number Pe and the reaction number R of the equation are non-dimensional parameters defined as

$$\mathrm{Pe} := \frac{a\ell}{\varepsilon},$$
$$\mathrm{R} := \frac{s\ell}{a},$$

where $\ell = \ell_1 + \ell_2$ is the total length of the domain.

3.2 Domain Decomposition Algorithm

Let $\Omega_1 = (-\ell_1, \delta)$ and $\Omega_2 = (-\delta, \ell_2)$ with $0 \leq \delta < \min(\ell_1, \ell_2)$. We define the following non-dimensional variables $\ell_1^* = \ell_1/\ell$, $\ell_2^* = \ell_2/\ell$ and $\delta^* = \delta/\ell$. Obviously, we have that

$$\delta^* \leq \min(\ell_1^*, \ell_2^*). \tag{3.2}$$

The general iteration-by-subdomain domain decomposition algorithm of overlap 2δ to solve system $(3.1)_{1-2}$ reads: given initial guesses u_1^0 and u_2^0, find u_1^{k+1} and u_2^{k+1} for $k \geq 0$ such that

$$\begin{cases} Lu_1^{k+1} &= f & \forall\, x \in \Omega_1, \\ u_1^{k+1} &= 0 & \text{at } x = -\ell_1, \\ \Phi_1(u_1^{k+1}) &= \theta_1\, \Phi_1(u_2^k) + (1-\theta_1)\, \Phi_1(u_1^k) & \text{at } x = \delta, \end{cases} \tag{3.3}$$

$$\begin{cases} Lu_2^{k+1} &= f & \forall\, x \in \Omega_2, \\ u_2^{k+1} &= 0 & \text{at } x = \ell_2, \\ \Phi_2(u_2^{k+1}) &= \theta_2\, \Phi_2(u_1^{k'}) + (1-\theta_2)\, \Phi_2(u_2^k) & \text{at } x = -\delta, \end{cases} \tag{3.4}$$

where Φ_1 and Φ_2 are the linear functionals representing the transmission conditions at δ and $-\delta$ respectively, θ_1 and θ_2 are positive constants, called the relaxation (or acceleration) parameters, and k' is an iteration index which can be

$$k' = \begin{cases} k+1 & \text{for the sequential version,} \\ k & \text{for the parallel version.} \end{cases}$$

If we assume the solution of problem $(3.1)_{1-2}$ satisfies the transmission conditions, then the error $e_i^{k+1} = u_i^{k+1} - u$ verifies the following homogeneous system of equations for $k \geq 0$

$$\begin{cases} Le_1^{k+1} &= 0 & \forall\, x \in \Omega_1, \\ e_1^{k+1} &= 0 & \text{at } x = -\ell_1, \\ \Phi_1(e_1^{k+1}) &= \theta_1\, \Phi_1(e_2^k) + (1-\theta_1)\, \Phi_1(e_1^k) & \text{at } x = \delta, \end{cases} \tag{3.5}$$

$$\begin{cases} Le_2^{k+1} &= 0 & \forall\, x \in \Omega_2, \\ e_2^{k+1} &= 0 & \text{at } x = \ell_2, \\ \Phi_2(e_2^{k+1}) &= \theta_2\, \Phi_2(e_1^{k'}) + (1-\theta_2)\, \Phi_2(e_2^k) & \text{at } x = -\delta. \end{cases} \tag{3.6}$$

We define three possible transmission conditions of Dirichlet, Neumann and Robin types:

$$\text{Dirichlet}: \quad \Phi_{\mathrm{D}}(u) = u, \tag{3.7}$$

$$\text{Neumann}: \quad \Phi_{\mathrm{N}}(u) = \varepsilon u' n_x,$$

$$\text{Robin}: \quad \Phi_{\mathrm{R}}(u) = \varepsilon u' n_x - \frac{1}{2}(a n_x) u, \tag{3.8}$$

where D holds for Dirichlet, N holds for Neumann, R for Robin, and n_x is the exterior normal which can take the values $n_x = -1$ and $n_x = 1$. The Neumann and Robin conditions considered here correspond to the natural conditions of the 0 and 1/2-weak formulations of the ADR equation, respectively (see Section 1.1.3).

The combinations of the first condition with any of the three at $x = -\delta$ and $x = \delta$ will lead to three families of overlapping domain decomposition methods, Dirichlet/Dirichlet (Schwarz method), Dirichlet/Neumann and Dirichlet/Robin. According to where each of the possible transmission conditions can be prescribed, i.e. at $x = -\delta$ and $x = \delta$, we define the following five couplings:

$$\mathrm{D/D}: \ \Phi_1 = \Phi_2 = \Phi_{\mathrm{D}},$$

$$\mathrm{N/D}: \ \Phi_1 = \Phi_{\mathrm{N}}, \Phi_2 = \Phi_{\mathrm{D}},$$

$$\mathrm{D/N}: \ \Phi_1 = \Phi_{\mathrm{D}}, \Phi_2 = \Phi_{\mathrm{N}},$$

$$\mathrm{R/D}: \ \Phi_1 = \Phi_{\mathrm{R}}, \Phi_2 = \Phi_{\mathrm{D}},$$

$$\mathrm{D/R}: \ \Phi_1 = \Phi_{\mathrm{D}}, \Phi_2 = \Phi_{\mathrm{R}}.$$

For the names of the mixed methods, we have explicitly indicated where the Dirichlet condition is imposed in order to study the effects of the advection, which renders the local equations in each subdomain dependent on the direction of the flow. Note that another coupling involving two Robin conditions is also possible, although it will not considered here.

Let us solve Equations $(3.5)_1$ and $(3.6)_1$. The general solutions are of the form

$$e_i^{k+1} = A_i^{k+1} \exp\left[(\omega + \tau)x\right] + B_i^{k+1} \exp\left[(\omega - \tau)x\right],$$

where A_i^{k+1}'s and B_i^{k+1}'s are constant depending on the boundary conditions, and ω and τ are given by

$$\omega = \frac{a}{2\varepsilon}, \qquad \tau = \frac{1}{2\varepsilon}\sqrt{a^2 + 4\varepsilon s},$$

or, in terms of the non-dimensional quantities Pe and R defined previously,

$$\omega = \frac{\mathrm{Pe}}{2\ell}, \qquad \tau = \frac{1}{\ell}\sqrt{(\mathrm{Pe}/2)^2 + \mathrm{Pe}\,\mathrm{R}}.$$

Applying the boundary conditions given by Equations $(3.5)_2$ and $(3.6)_2$, the errors are given by

$$e_1^{k+1}(x) = C_1^{k+1} \exp(\omega x) \sinh(\tau(\ell_1 + x)),$$
$$e_2^{k+1}(x) = C_2^{k+1} \exp(\omega x) \sinh(\tau(\ell_2 - x)).$$

where C_1^{k+1} and C_2^{k+1} are constants depending on the transmission conditions $(3.5)_3$ and $(3.6)_3$. The following simple estimates hold for the error:

$$\|e_1^{k+1}(x)\|_{\infty,\Omega_1} \leq |C_1^{k+1}| \sup_{x\in\Omega_1} |\exp(\omega x)\sinh(\tau(\ell_1+x))|,$$
$$\|e_2^{k+1}(x)\|_{\infty,\Omega_2} \leq |C_2^{k+1}| \sup_{x\in\Omega_2} |\exp(\omega x)\sinh(\tau(\ell_2-x))|.$$

We now develop the expressions for the error coefficients C_i^k $(i=1,2)$, and also set up a general analysis framework to be able to compare all the DD methods of interest in a systematic way. As a first step, let us introduce the following matrix notation,

$$\mathbf{c}^{k+1} = \begin{bmatrix} C_1^{k+1} \\ C_2^{k+1} \end{bmatrix}.$$

As will be seen in the following sections, all the DD algorithms lead to an algebraic system of the form

$$\mathbf{c}^{k+1} = \mathbf{C}\,\mathbf{c}^k,$$

where $\mathbf{C}$ is called the *iteration matrix*. The errors $e_1^{k+1}(x)$ and $e_2^{k+1}(x)$ will converge to zero as the iteration proceeds if and only if $\lim_{k\to\infty}\mathbf{c}^{k+1}=\mathbf{0}$. It can be shown that a necessary and sufficient condition is that the spectral radius $\varrho(\mathbf{C})$ of $\mathbf{C}$, also called convergence rate, satisfies

$$\varrho(\mathbf{C}) < 1. \tag{3.9}$$

This convergence criterion is an asymptotic statement which does not ensure monotonicity of the convergence. Only in some particular conditions we will be able to obtain a norm estimate, as is for example the L_2 matrix norm. This is precisely the case when the iteration matrix $\mathbf{C}$ is symmetric. For such matrices, we know that $\|\mathbf{C}\|_2 = \varrho(\mathbf{C})$ where $\|\cdot\|_2$ is the matrix norm corresponding to the Euclidean vector norm $\|\cdot\|_2$ (also called root-mean-square vector norm), and defined as

$$\|\mathbf{C}\|_2 = \sup_{\|\mathbf{x}\|\neq 0} \frac{\|\mathbf{C}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

Therefore, if matrix $\mathbf{C}$ is symmetric, we have

$$\begin{aligned} \|\mathbf{c}^{k+1}\|_2 &= \|\mathbf{C}\,\mathbf{c}^k\|_2, \\ &\leq \|\mathbf{C}\|_2\,\|\mathbf{c}^k\|_2, \\ &= \varrho(\mathbf{C})\,\|\mathbf{c}^k\|_2, \end{aligned}$$

and if condition (3.9) is satisfied, the scheme is said to be *monotonically convergent* with respect to the L_2-norm. We will see that this is the case of the unrelaxed and relaxed sequential DD algorithms presented here, whereas the parallel versions are in general not monotonically convergent.

3.3 Unrelaxed sequential version

3.3.1 General case

Five couplings will be studied in detail, namely the D/D, N/D, D/N, R/D and D/R. The differentiation between the N/D and D/N methods, as well as between of the R/D and D/R methods, referred to as *mixed methods*, is necessary because of the asymmetry of domain decomposition methods in general. We will see how the convergence depends on the relative sizes of the subdomains, as well as on the direction and magnitude of the advection field (remember that we have assumed $a \geq 0$). This is obviously the case in the hyperbolic limit, for which the direction of the advection field is an essential characteristic of the problem. As a first approach, the relaxation factors are set to $\theta_1 = \theta_2 = 1$. The effects of under-relaxation will be studied in a separate section in order not to complicate the analysis of the methods.

It can be noted first that the equations for the error coefficients derived from Equations $(3.5)_3$ and $(3.6)_3$ with $k' = k + 1$, can be generally written in the following form when $\theta_1 = \theta_2 = 1$, for any $k \geq 0$:

$$C_1^{k+1} = \varrho_1 C_2^k, \tag{3.10}$$

$$C_2^{k+1} = \varrho_2 C_1^{k+1}, \tag{3.11}$$

where ϱ_1 and ϱ_2 do not depend on the iteration number k. Rearranging Equations (3.10) and (3.11) we obtain simply, this time for $k \geq 1$:

$$C_1^{k+1} = \varrho_1 \varrho_2 C_2^k,$$
$$C_2^{k+1} = \varrho_1 \varrho_2 C_1^k,$$

which gives the following iteration matrix

$$\mathbf{C} = \begin{bmatrix} \varrho_1 \varrho_2 & 0 \\ 0 & \varrho_1 \varrho_2 \end{bmatrix}.$$

Obviously, matrix $\mathbf{C}$ has a single eigenvalue λ of multiplicity two:

$$\lambda = \varrho_1 \varrho_2.$$

Let us define

$$\varrho := \varrho(\mathbf{C}),$$

and we have therefore

$$\varrho = |\varrho_1 \varrho_2|,$$

$\mathbf{C}$ being a diagonal matrix, the convergence of the unrelaxed sequential version of the DD algorithm is monotone with respect to the L_2-norm if $\varrho < 1$.

We now determine the expressions of ϱ for each of the five DD methods, before establishing the conditions for convergence.

D/D method. Dirichlet conditions are imposed at both $x = \delta$ and $x = -\delta$. The transmission conditions give:

$$\begin{aligned}
\Phi_1(e_1^{k+1}) &= C_1^{k+1} \exp(\omega\delta) \sinh(\tau(\ell_1 + \delta)),\\
\Phi_1(e_2^{k+1}) &= C_2^{k+1} \exp(\omega\delta) \sinh(\tau(\ell_2 - \delta)),\\
\Phi_2(e_1^{k+1}) &= C_1^{k+1} \exp(-\omega\delta) \sinh(\tau(\ell_1 - \delta)),\\
\Phi_2(e_2^{k+1}) &= C_2^{k+1} \exp(-\omega\delta) \sinh(\tau(\ell_2 + \delta)).
\end{aligned}$$

Substituting these equations into the equations for the error transmission conditions $(3.5)_3$ and $(3.6)_3$, we find that ϱ_1 and ϱ_2 are given by:

$$\varrho_1 = \frac{\sinh(\tau(\ell_2 - \delta))}{\sinh(\tau(\ell_1 + \delta))}, \tag{3.12}$$

$$\varrho_2 = \frac{\sinh(\tau(\ell_1 - \delta))}{\sinh(\tau(\ell_2 + \delta))}, \tag{3.13}$$

and therefore,

$$\varrho = \frac{\sinh(\tau(\ell_2 - \delta))}{\sinh(\tau(\ell_2 + \delta))} \frac{\sinh(\tau(\ell_1 - \delta))}{\sinh(\tau(\ell_1 + \delta))}. \tag{3.14}$$

N/D method. A Neumann condition is imposed at $x = \delta$ and a Dirichlet condition at $x = -\delta$. Knowing that $n_x = 1$ at $x = \delta$, the transmission conditions give:

$$\begin{aligned}
\Phi_1(e_1^{k+1}) &= C_1^{k+1} \exp(\omega\delta)[\varepsilon\omega \sinh(\tau(\ell_1 + \delta)) + \varepsilon\tau \cosh(\tau(\ell_1 + \delta))],\\
\Phi_1(e_2^{k+1}) &= C_2^{k+1} \exp(\omega\delta)[\varepsilon\omega \sinh(\tau(\ell_2 - \delta)) - \varepsilon\tau \cosh(\tau(\ell_2 - \delta))],\\
\Phi_2(e_1^{k+1}) &= C_1^{k+1} \exp(-\omega\delta) \sinh(\tau(\ell_1 - \delta)),\\
\Phi_2(e_2^{k+1}) &= C_2^{k+1} \exp(-\omega\delta) \sinh(\tau(\ell_2 + \delta)).
\end{aligned}$$

Substituting these equations into the equations for the errors transmission conditions $(3.5)_3$ and $(3.6)_3$, we obtain:

$$\varrho_1 = \frac{\omega \sinh(\tau(\ell_2 - \delta)) - \tau \cosh(\tau(\ell_2 - \delta))}{\omega \sinh(\tau(\ell_1 + \delta)) + \tau \cosh(\tau(\ell_1 + \delta))}, \tag{3.15}$$

$$\varrho_2 = \frac{\sinh(\tau(\ell_1 - \delta))}{\sinh(\tau(\ell_2 + \delta))}. \tag{3.16}$$

Noting that $\tau \geq \omega$, we can show that ϱ_1 is always negative. Therefore, ϱ is given by

$$\varrho = -\frac{\sinh(\tau(\ell_1 - \delta))}{\sinh(\tau(\ell_2 + \delta))} \frac{\omega \sinh(\tau(\ell_2 - \delta)) - \tau \cosh(\tau(\ell_2 - \delta))}{\omega \sinh(\tau(\ell_1 + \delta)) + \tau \cosh(\tau(\ell_1 + \delta))}. \tag{3.17}$$

D/N method. We impose a Dirichlet condition at $x = \delta$ and a Neumann condition at $x = -\delta$. Knowing that $n_x = -1$ at $x = -\delta$, the transmission conditions give:

$$\begin{aligned}
\Phi_1(e_1^{k+1}) &= C_1^{k+1} \exp(\omega\delta) \sinh(\tau(\ell_1 + \delta)),\\
\Phi_1(e_2^{k+1}) &= C_2^{k+1} \exp(\omega\delta) \sinh(\tau(\ell_2 - \delta)),\\
\Phi_2(e_1^{k+1}) &= C_1^{k+1} \exp(-\omega\delta)[-k\omega \sinh(\tau(\ell_1 - \delta)) - k\tau \cosh(\tau(\ell_1 - \delta))],\\
\Phi_2(e_2^{k+1}) &= C_2^{k+1} \exp(-\omega\delta)[-k\omega \sinh(\tau(\ell_2 + \delta)) + k\tau \cosh(\tau(\ell_2 + \delta))],
\end{aligned}$$

and ϱ_1 and ϱ_2 are given by:

$$\varrho_1 = \frac{\sinh(\tau(\ell_2 - \delta))}{\sinh(\tau(\ell_1 + \delta))}, \tag{3.18}$$

$$\varrho_2 = \frac{\omega \sinh(\tau(\ell_1 - \delta)) + \tau \cosh(\tau(\ell_1 - \delta))}{\omega \sinh(\tau(\ell_2 + \delta)) - \tau \cosh(\tau(\ell_2 + \delta))}. \tag{3.19}$$

It can be shown that ϱ_2 is always negative. Therefore, the spectral radius is

$$\varrho = -\frac{\sinh(\tau(\ell_2 - \delta))}{\sinh(\tau(\ell_1 + \delta))} \frac{\omega \sinh(\tau(\ell_1 - \delta)) + \tau \cosh(\tau(\ell_1 - \delta))}{\omega \sinh(\tau(\ell_2 + \delta)) - \tau \cosh(\tau(\ell_2 + \delta))}. \tag{3.20}$$

R/D method. A Robin condition is imposed at $x = \delta$ and a Dirichlet condition at $x = -\delta$. Noting that $n_x = 1$ at $x = \delta$, the transmission conditions give:

$$\begin{aligned}
\Phi_1(e_1^{k+1}) &= C_1^{k+1} \exp(\omega\delta)[k\tau \cosh(\tau(\ell_1 + \delta))],\\
\Phi_1(e_2^{k+1}) &= C_2^{k+1} \exp(\omega\delta)[-k\tau \cosh(\tau(\ell_2 - \delta))],\\
\Phi_2(e_1^{k+1}) &= C_1^{k+1} \exp(-\omega\delta) \sinh(\tau(\ell_1 - \delta)),\\
\Phi_2(e_2^{k+1}) &= C_2^{k+1} \exp(-\omega\delta) \sinh(\tau(\ell_2 + \delta)).
\end{aligned}$$

Substituting these equations into the equations for the errors transmission conditions $(3.5)_3$ and $(3.6)_3$, we have:

$$\varrho_1 = -\frac{\cosh(\tau(\ell_2 - \delta))}{\cosh(\tau(\ell_1 + \delta))}, \tag{3.21}$$

$$\varrho_2 = \frac{\sinh(\tau(\ell_1 - \delta))}{\sinh(\tau(\ell_2 + \delta))}, \tag{3.22}$$

and the spectral radius is

$$\varrho = \frac{\sinh(\tau(\ell_1 - \delta))}{\sinh(\tau(\ell_2 + \delta))} \frac{\cosh(\tau(\ell_2 - \delta))}{\cosh(\tau(\ell_1 + \delta))}.$$

D/R method. We impose a Dirichlet condition at $x = \delta$ and a Robin condition at $x = -\delta$. Noting that $n_x = 1$ at $x = a$, the transmission conditions give:

$$\begin{aligned}
\Phi_1(e_1^{k+1}) &= C_1^{k+1} \exp(\omega\delta) \sinh(\tau(\ell_1 + \delta)),\\
\Phi_1(e_2^{k+1}) &= C_2^{k+1} \exp(\omega\delta) \sinh(\tau(\ell_2 - \delta)),\\
\Phi_2(e_1^{k+1}) &= C_1^{k+1} \exp(-\omega\delta)[-k\tau \cosh(\tau(\ell_1 - \delta))],\\
\Phi_2(e_2^{k+1}) &= C_2^{k+1} \exp(-\omega\delta)[k\tau \cosh(\tau(\ell_2 + \delta))],
\end{aligned}$$

which gives

$$\varrho_1 = \frac{\sinh(\tau(\ell_2 - \delta))}{\sinh(\tau(\ell_1 + \delta))}, \tag{3.23}$$

$$\varrho_2 = -\frac{\cosh(\tau(\ell_1 - \delta))}{\cosh(\tau(\ell_2 + \delta))}, \tag{3.24}$$

and leads to the following spectral radius

$$\varrho = \frac{\sinh(\tau(\ell_2 - \delta))}{\sinh(\tau(\ell_1 + \delta))} \frac{\cosh(\tau(\ell_1 - \delta))}{\cosh(\tau(\ell_2 + \delta))}.$$

Despite the complicated aspects of the expressions developed for the ϱ's, we can draw some general conclusions about the DD methods studied here. First, we note that the D/D method is always convergent for any δ^*; knowing that $\sinh(x)$ is an increasing function, we conclude from Equation (3.14) that $\varrho < 1$ whenever $\delta^* > 0$. We note also that D/D method always behaves indifferently with respect to relative lengths of the subdomains. In fact, interchanging ℓ_1 and ℓ_2 in Equation (3.14), we recover exactly the same rate of convergence. The R/D and D/R methods do not share exactly this property, but nevertheless exhibit an interesting symmetry. In fact, let us call ℓ_{D} the length of the Dirichlet subdomain and ℓ_{R} that of the Robin subdomain. We can easily check that both the R/D and D/R methods give for the spectral radius

$$\varrho = \frac{\sinh(\tau(\ell_{\mathrm{R}} - \delta))}{\sinh(\tau(\ell_{\mathrm{D}} + \delta))} \frac{\cosh(\tau(\ell_{\mathrm{D}} - \delta))}{\cosh(\tau(\ell_{\mathrm{R}} + \delta))}.$$

This property ensures us that the Dirichlet and Robin conditions can be located independently of the direction of the flow. This is not the case of the Dirichlet/Neumann-like coupling.

The convergences of the five DD couplings is now presented for three limiting behaviors of the original equations. The first case is the purely elliptic equation for which we set $a = 0$, and $s = 0$. The second case studied is the original advection-diffusion equation without reaction, i.e. $s = 0$. Finally, the advection-diffusion-reaction equation will be studied in the hyperbolic limit, i.e. when $\mathrm{Pe} \to \infty$. For each of these cases, we give the simplified value of the spectral radius as well as the stability criteria ($\varrho < 1$) in terms the overlap δ^* and finally compare the five methods.

3.3.2 Poisson Equation

To obtain the Poisson Equation, we set $\varepsilon > 0$, $a = 0$, and $s = 0$. The equations for the C_i^{k+1}'s can be found taking the limit $\tau \to 0$ and $\omega \to 0$ in Equations (3.14), (3.17) and (3.20). Obviously, the

Robin type condition is exactly the same than the Neumann condition for there is no advection. We find:

$$\begin{aligned}
\mathrm{D/D}: &\quad \varrho = \frac{\ell_1^* - \delta^*}{\ell_1^* + \delta^*}\,\frac{\ell_2^* - \delta^*}{\ell_2^* + \delta^*},\\
\mathrm{N(R)/D}: &\quad \varrho = \frac{\ell_1^* - \delta^*}{\ell_2^* + \delta^*},\\
\mathrm{D/N(R)}: &\quad \varrho = \frac{\ell_2^* - \delta^*}{\ell_1^* + \delta^*}.
\end{aligned}$$

The stability criteria of the three methods are

$$\begin{aligned}
\mathrm{D/D}: &\quad \delta^* > 0,\\
\mathrm{N(R)/D}: &\quad \delta^* > \frac{\ell_1^* - \ell_2^*}{2},\\
\mathrm{D/N(R)}: &\quad \delta^* > \frac{\ell_2^* - \ell_1^*}{2}.
\end{aligned}$$

We first note that the D/D method is stable for any $\delta^* > 0$ (obviously, this method without overlap does not make sense!) and the rate of convergence increases with the overlap. In order to visualize what happens for the mixed methods, let us introduce some notations.

We define $\Delta\ell^*$ as the difference between the length of the Dirichlet subdomain, noted ℓ_D, with that of the Neumann subdomain, ℓ_N:

$$\Delta\ell^* = \ell_\mathrm{D} - \ell_\mathrm{N},$$

and we have

$$\mathrm{N(R)/D}: \Delta\ell^* = \ell_2^* - \ell_1^*, \tag{3.25}$$

$$\mathrm{D/N(R)}: \Delta\ell^* = \ell_1^* - \ell_2^*. \tag{3.26}$$

For this particular case, the convergence criteria for δ^* of the N/D and D/N methods is exactly the same, i.e.

$$\delta^* > -\frac{\Delta\ell^*}{2}.$$

Remembering also that the value of δ^* is limited from above by inequality (3.2), which can be rewritten as

$$\delta^* < \frac{1 - |\Delta\ell^*|}{2}, \tag{3.27}$$

the permissible range for δ^* is therefore

$$-\frac{\Delta\ell^*}{2} < \delta^* < \frac{1 - |\Delta\ell^*|}{2},$$

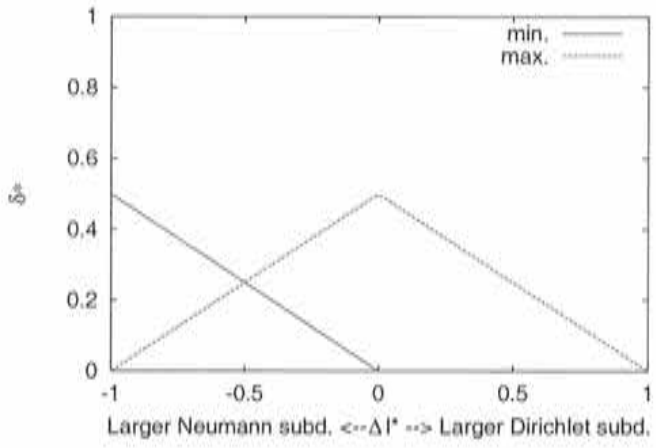


Figure 3.1: Poisson equation. Minimum and maximum overlap δ^* as a function of $\Delta\ell^*$ for the N/D and D/N methods to converge.

as sketched in Figure 3.1. We observe that the mixed methods give a stability criterion which depends on the relative sizes of the subdomains. In particular, if the subdomains are disjoint ($\delta = 0$), the N/D and D/N methods converge only if $\Delta\ell^* > 0$, i.e. if the Dirichlet subdomain is larger than the Neumann subdomain (right part of the axis on Figure 3.1); this is a well known result, see for example [59]. In addition, we note also that we have no way of converging when the Neumann subdomain is such that $\Delta\ell^* \leq -0.5$.

3.3.3 Advection-diffusion equation

We consider the original equation (3.1) without reaction, i.e. $\varepsilon > 0$, $a > 0$ and $s = 0$. This case is interesting because the stability criteria depend only on the difference of lengths of the subdomains $\ell_2^* - \ell_1^*$ and on the Péclet number. Setting $s = 0$ implies that $\omega = \tau$, and the equations for the ϱ's reduce to:

$$\mathrm{D/D} : \varrho = \frac{\sinh[(\ell_1^* - \delta^*)\mathrm{Pe}/2]}{\sinh[(\ell_2^* + \delta^*)\mathrm{Pe}/2]} \frac{\sinh[(\ell_2^* - \delta^*)\mathrm{Pe}/2]}{\sinh[(\ell_1^* + \delta^*)\mathrm{Pe}/2]},$$

$$\mathrm{N/D} : \varrho = \exp(-\mathrm{Pe}/2) \frac{\sinh[(\ell_1^* - \delta^*)\mathrm{Pe}/2]}{\sinh[(\ell_2^* + \delta^*)\mathrm{Pe}/2]},$$

$$\mathrm{D/N} : \varrho = \exp(\mathrm{Pe}/2) \frac{\sinh[(\ell_2^* - \delta^*)\mathrm{Pe}/2]}{\sinh[(\ell_1^* + \delta^*)\mathrm{Pe}/2]},$$

$$\mathrm{R/D} : \varrho = \frac{\sinh[(\ell_1^* - \delta^*)\mathrm{Pe}/2]}{\sinh[(\ell_2^* + \delta^*)\mathrm{Pe}/2]} \frac{\cosh[(\ell_2^* - \delta^*)\mathrm{Pe}/2]}{\cosh[(\ell_1^* + \delta^*)\mathrm{Pe}/2]},$$

$$\mathrm{D/R} : \varrho = \frac{\sinh[(\ell_2^* - \delta^*)\mathrm{Pe}/2]}{\sinh[(\ell_1^* + \delta^*)\mathrm{Pe}/2]} \frac{\cosh[(\ell_1^* - \delta^*)\mathrm{Pe}/2]}{\cosh[(\ell_2^* + \delta^*)\mathrm{Pe}/2]}.$$

The following stability criteria hold:

$$\mathrm{D/D} : \delta^* > 0,$$

$$\mathrm{N/D} : \delta^* > -\frac{1}{\mathrm{Pe}} \ln \left[\frac{\exp\left[(\ell_2^* - \ell_1^* + 1)\mathrm{Pe}/2\right] + \exp\left[(\ell_2^* - \ell_1^* - 1)\mathrm{Pe}/2\right]}{2} \right],$$

$$\mathrm{D/N} : \delta^* > \frac{1}{\mathrm{Pe}} \ln \left[\frac{\exp\left[(\ell_2^* - \ell_1^* + 1)\mathrm{Pe}/2\right] + \exp\left[(\ell_2^* - \ell_1^* - 1)\mathrm{Pe}/2\right]}{2} \right],$$

$$\mathrm{R/D} : \delta^* > \frac{1}{\mathrm{Pe}} \ln \left[-\frac{\sinh[(\ell_2^* - \ell_1^*)\mathrm{Pe}/2]}{\cosh(\mathrm{Pe}/2)} + \sqrt{\frac{\sinh^2[(\ell_2^* - \ell_1^*)\mathrm{Pe}/2]}{\cosh^2(\mathrm{Pe}/2)} + 1} \right],$$

$$\mathrm{D/R} : \delta^* > \frac{1}{\mathrm{Pe}} \ln \left[\frac{\sinh[(\ell_2^* - \ell_1^*)\mathrm{Pe}/2]}{\cosh(\mathrm{Pe}/2)} + \sqrt{\frac{\sinh^2[(\ell_2^* - \ell_1^*)\mathrm{Pe}/2]}{\cosh^2(\mathrm{Pe}/2)} + 1} \right].$$

We first remark that except the D/D method, all convergence criteria on δ^* depend on the Péclet number; in fact, the D/D method converges whenever $\delta^* > 0$. However, an important distinction between the Neumann-type and Robin-type couplings must be pointed out. On the one hand, the D/N and N/D do not behave in the same way with respect to the asymmetry of the subdomains, characterized by the magnitude of $\ell_2^* - \ell_1^*$. On the other hand, the D/R and R/D do, as already pointed out at the end of Section 3.3.1. In the case of the N/D method, for which the Neumann condition is imposed in accordance with the direction of the flow, the convergence criterion requires that the length of the Neumann subdomain (ℓ_1^*) is sufficiently small, this criterion being looser and looser as the Péclet number increases. This characteristic is shared by *both* the D/R and R/D methods. But for the D/N method, the convergence condition on δ^* is more and more restrictive as the Péclet number increase; in order to achieve convergence, the length of the Neumann subdomain must be taken smaller and smaller, or the overlap δ^* greater and greater as Pe increases (when studying the hyperbolic limit of the full ADR equation, we will see that the presence of the reaction term R helps the D/N to converge for high Péclet numbers). Let us illustrate these remarks. As we have done for the Poisson equation, we define $\Delta\ell^*$ the difference between the length of the Dirichlet subdomain with that of the other subdomain, which now can be of Neumann or Robin type. We have

$$\mathrm{N/D \text{ and } R/D} : \Delta\ell^* = \ell_2^* - \ell_1^*,$$
$$\mathrm{D/N \text{ and } D/R} : \Delta\ell^* = \ell_1^* - \ell_2^*.$$

Figure 3.2 shows the permissible range for δ^* to achieve convergence, as a function of the Péclet number, and for four different $\Delta\ell^*$. Of course, the value of δ^* is limited by inequality (3.2), which is represented by the horizontal solid line. Note that the range for δ^* has been extended to negative values only for a graphical reason.

We observe once more that for all the methods the convergence criteria is easier to achieve when the Dirichlet subdomain is larger that the Neumann subdomain, i.e. when $\Delta\ell^* \geq 0$. For the N/D, R/D and D/R methods, we notice that the minimum δ^* decreases as the Péclet number increases whatever $\Delta\ell^*$ is; the minimum δ^* is always smaller that that of the Poisson equation. On the contrary, the D/N method behavior worsens when Pe increases. The hyperbolic limit will be studied in next section.

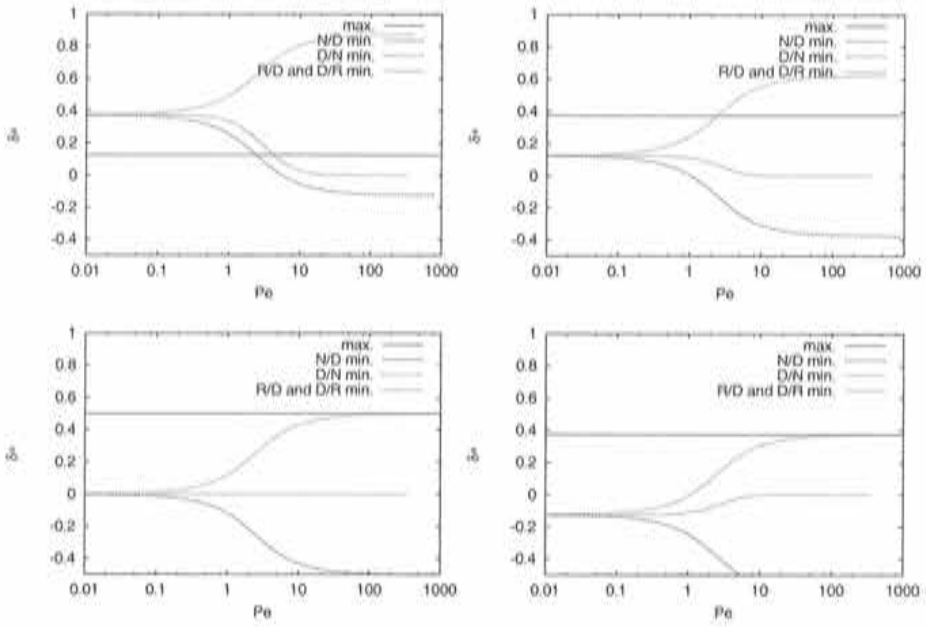


Figure 3.2: Advection-Diffusion equation. Minimum and maximum overlap δ^* as a function of Pe. (Top) (Left) $\Delta\ell^* = -0.75$. (Top) (Right) $\Delta\ell^* = -0.25$. (Bot.) (Left) $\Delta\ell^* = 0$. (Bot.) (Right) $\Delta\ell^* = 0.25$.

Remark. If $\ell_1^* = \ell_2^* = 1/2$, the latter stability criteria become

$$\begin{aligned}
&\mathrm{D/D} : \delta^* > 0, \\
&\mathrm{N/D} : \delta^* \geq 0, \\
&\mathrm{D/N} : \delta^* > \frac{1}{\mathrm{Pe}} \ln[\cosh(\mathrm{Pe}/2)], \\
&\mathrm{R/D} : \delta^* > 0, \\
&\mathrm{D/R} : \delta^* > 0,
\end{aligned} \tag{3.28}$$

The N/D is therefore unconditionally stable for $a > 0$, while the D/N is conditionally stable under condition (3.28). As noted earlier, the higher the Péclet number, the more restrictive is this condition. The D/D, R/D and D/R methods converge if and only if the subdomains overlap.

3.3.4 Hyperbolic limit

We consider the hyperbolic limit of the advection-diffusion-reaction equation, i.e. $\mathrm{Pe} \gg 1$. Obviously, when $\mathrm{Pe} \gg 1$, $\tau \gg 1$. Noting that $\sinh(x) \approx \exp(x)/2$ and $\cosh(x) \approx \exp(x)/2$ when $x \gg 1$, and assuming $s \neq 0$ ($\tau > \omega$), Equations (3.14), (3.17) and (3.20) give the following rates of convergence

$$\mathrm{D/D} : \varrho \approx \exp(-2\mathrm{Pe}\delta^*),$$

$$
\begin{aligned}
&\mathrm{N/D} : \varrho \approx \exp\left(-2\mathrm{Pe}\delta^*\right)\frac{\mathrm{R/Pe} + \exp\left[-\mathrm{Pe}(l_2^* - \delta^*)\right]}{\mathrm{R/Pe} + 1},\\
&\mathrm{D/N} : \varrho \approx \exp\left(-2\mathrm{Pe}\delta^*\right)\frac{\mathrm{R/Pe} + 1}{\mathrm{R/Pe} + \exp\left[-\mathrm{Pe}(l_2^* + \delta^*)\right]},\\
&\mathrm{R/D} : \varrho \approx \exp\left(-2\mathrm{Pe}\delta^*\right),\\
&\mathrm{D/R} : \varrho \approx \exp\left(-2\mathrm{Pe}\delta^*\right).
\end{aligned}
\tag{3.29}
$$

With reaction, $R > 0$. If R has a finite value, the following stability conditions can be established:

$$
\begin{aligned}
&\mathrm{D/D} : \delta^* > 0,\\
&\mathrm{N/D} : \delta^* \geq 0,\\
&\mathrm{D/N} : \delta^* > \frac{1}{2\mathrm{Pe}}\ln(\mathrm{Pe/R}),\\
&\mathrm{R/D} : \delta^* > 0,\\
&\mathrm{D/R} : \delta^* > 0.
\end{aligned}
\tag{3.30}
$$

We first notice that unlike the preceding two equations studied, i.e. the Poisson and advection-diffusion equations, the condition on the minimum value for δ^* does not depend on the lengths of the subdomain.

On the one hand, the N/D method, for which transmission conditions are consistent with the subdomains boundary conditions in the hyperbolic limit, i.e. Neumann condition at outflow and Dirichlet condition at inflow, is unconditionally stable. On the other hand, the D/N coupling is conditionally stable. Imposing a Dirichlet transmission condition at the outflow of Ω_1 creates an artificial boundary layer at $x = \delta$. The derivative of the unknown in the vicinity of the boundary layer tends to infinity when the Péclet number tends to infinity, and if the Neumann transmission condition is to be imposed inside this artificial boundary layer, the domain decomposition might not converge. Let us estimate the width d of the artificial boundary layer. When the diffusion term is balanced with the advection term, we have $\varepsilon u/d^2 \sim au/d$ so that the boundary layer width scales like

$$\frac{d}{\ell} \sim \mathrm{Pe}^{-1}.$$

Inequality (3.30) states therefore that the overlapping must be larger than the artificial boundary layer created at the interface at $x = \delta$. This results holds only when a reaction term is present. On the contrary, we have seen that the convergence criteria is $\delta > 1/2$ when $\mathrm{Pe} \to \infty$.

As expected, the R/D and the D/R behave exactly as the D/D method, for the Robin transmission condition (3.8) tends to the Dirichlet transmission condition (3.7) when $\mathrm{Pe} \gg 1$. Of course, this remark only applies to this one-dimensional example; this result can only be extrapolated to two and three-dimension problems when the normal is "sufficiently" aligned with the advection vector.

Without reaction, $R = 0$. The N/D method is unconditionally stable, while the D/D, R/D and D/R methods converge whenever $\delta^* > 0$. The rate of convergence of the N/D method is in this case $\exp[\mathrm{Pe}(l_2^* - \delta^*)]$ smaller than that of the D/D, R/D and D/R methods. The stability condition for the D/N method requires that:

$$\mathrm{D/N} : \delta^* \to \ell_2^*,$$

Remembering that $\delta^* < \min(\ell_1^*, \ell_2^*)$, the latter equation can not be satisfied if the Neumann subdomain is larger than the Dirichlet subdomain ($\ell_2 > \ell_1$). The algorithm is neither viable if the Dirichlet subdomain is larger, as the latter condition states that the overlap must be of the size of the Neumann subdomain, which is not of practical interest!

3.3.5 Summary

All the stability conditions computed in the previous sections hold for $\theta_1 = \theta_2 = 1$. The conclusions are the following:

The D/D method is always stable whenever $\delta^* > 0$, and its convergence rate depends on the relative lengths of the subdomains in general, and exponentially on the overlap in the hyperbolic limit. When the Péclet number is zero or small, all the four mixed methods studied require that the Neumann subdomain be "sufficiently" smaller than the Dirichlet subdomains. If this is not the case, convergence can be achieved using "sufficient" overlap. The behaviors of the mixed methods differ drastically when the Péclet number is high, although their convergence criteria no longer depend on the relative lengths of the subdomains. In the hyperbolic limit, the R/D and D/R methods tend to the D/D method and converge whenever $\delta^* > 0$, at the same rates. In this limit, the N/D method is the most adequate method for this simple one-dimensional example, as it exhibits the best convergence rate. As for the Dirichlet/Neumann method, we have seen that in the hyperbolic limit, when no reaction term is present in the equation, we have no way of making the method converge. However, the presence of the reaction enables convergence if the overlap is larger than the artificial boundary layer created at the Dirichlet interface.

3.4 Relaxed sequential version

We now study the effects of under-relaxation. In order to compare the results of the relaxed versions with that of the unrelaxed version, we define ϱ_{seq} the rate of convergence of the unrelaxed versions, i.e.

$$\varrho_{\mathrm{seq}} = |\varrho_1 \varrho_2|.$$

It will be shown that under-relaxation can, in some cases, help the DD algorithms converge. In particular, in the presence of a geometrical asymmetry between the subdomains, as already observed, under relaxation is necessary for the N/D and D/N to converge. We have also observed that the D/N method for solving the advection-diffusion equation without reaction only converges under sever restrictions if $\varepsilon \to 0$. This section will present the effects of under-relaxation on the five DD algorithms studied. In particular, a convergence condition for the under-relaxation factor will be determined, as well as the optimum θ_{opt} for which the rate of convergence ϱ_{opt} is minimum. Finally, the two cases mentioned previously will be studied as illustrations of the under-relaxation.

For all the methods studied, the under-relaxed version of the sequential DD methods can be obtained from Equations $(3.5)_3$ and $(3.6)_3$ for $k \geq 0$:

$$C_1^{k+1} = \theta_1 \varrho_1 C_2^k + (1-\theta_1) C_1^k, \tag{3.31}$$

$$C_2^{k+1} = \theta_2 \varrho_2 C_1^{k+1} + (1-\theta_2) C_2^k, \tag{3.32}$$

where ϱ_1 and ϱ_2 are the convergence rates of the error coefficients already calculated for the unrelaxed version. They are given by Equations (3.12)-(3.13), (3.15)-(3.16), (3.18)-(3.19), (3.21)-(3.22) and (3.23)-(3.24) for the D/D, N/D, D/N, R/D and D/R methods, respectively. The iteration matrix $\mathbf{C}$ is for $k \geq 0$:

$$\mathbf{C} = \begin{bmatrix} 1-\theta_1 & \theta_1 \varrho_1 \\ \theta_2(1-\theta_1)\varrho_2 & \theta_1\theta_2\varrho_1\varrho_2 + (1-\theta_2) \end{bmatrix}.$$

Ideally, one wants to manipulate the least external parameters as possible to control the convergence of the algorithm. We enable the following possible choices, $(\theta_1, \theta_2) = (1, \theta)$ or $(\theta_1, \theta_2) = (\theta, 1)$. By rearranging the error coefficients given by Equations (3.31) and (3.32), we notice that the iteration matrix $\mathbf{C}$ for both relaxed versions can be re-written for $k \geq 1$ as:

$$\mathbf{C} = \begin{bmatrix} 1+\theta(\varrho_1\varrho_2 - 1) & 0 \\ 0 & 1+\theta(\varrho_1\varrho_2 - 1) \end{bmatrix}.$$

The diagonal form of the iteration matrix is very convenient as we know that the $\|\cdot\|_2$ norm of a diagonal matrix is precisely its spectral radius, i.e.

$$\|\mathbf{c}^{k+1}\|_2 \leq \varrho \, \|\mathbf{c}^k\|_2. \tag{3.33}$$

Therefore, the convergence condition $\varrho < 1$ implies also monotonicity of the convergence with respect to the $\|\cdot\|_2$ norm. The condition for monotonicity cannot in general be established if both θ_1 and θ_2 are different from unity.

As obtained for the case with no relaxation, we have an eigenvalue λ of multiplicity two given by:

$$\lambda = 1 + \theta(\varrho_1\varrho_2 - 1),$$

which gives the following rate of convergence

$$\varrho = |1 + \theta(\varrho_1\varrho_2 - 1)|.$$

The convergence condition $\varrho < 1$ gives

$$|1 + \theta(\varrho_1\varrho_2 - 1)| < 1,$$

which is equivalent to

$$-2 < \theta(\varrho_1\varrho_2 - 1) < 0.$$

The convergence condition is therefore subject to the value of the product $\varrho_1\varrho_2$. Let us now obtain the value of $\varrho_1\varrho_2$ for the five DD methods considered. It can be easily shown from Equations (3.12)

and (3.13) that the D/D method always gives $0 < \varrho_1\varrho_2 < 1$ provided $\delta^* > 0$. When $\delta^* = 0$, we are in the case $\varrho_1\varrho_2 = 1$. For the N/D and D/N methods, we have already pointed in Section 3.3.1 that the product $\varrho_1\varrho_2$ is negative; as for the R/D and D/R methods, from Equations (3.21)-(3.22) and (3.23)-(3.24), we easily check that $\varrho_1\varrho_2 < 0$. Therefore, the following two situations are possible:

- If $\varrho_1\varrho_2 < 1$. The condition for convergence becomes

$$0 < \theta < \frac{2}{1 - \varrho_1\varrho_2}, \tag{3.34}$$

  and, in addition, $\varrho = \varrho_{\text{opt}} = 0$ if

$$\theta_{\text{opt}} = \frac{1}{1 - \varrho_1\varrho_2}.$$

- If $\varrho_1\varrho_2 = 1$. $\varrho = 1$ for all θ, and the D/D method does not converge.

Let us sum up the results we have obtained. We have shown that all the DD methods studied admit a maximum relaxation parameter $\theta_{\max}$ such that they converge for any $\theta < \theta_{\max}$, as given by Equation (3.34). We have also shown that the quantity $\varrho_1\varrho_2$ is such that

$$\text{D/D :} \qquad 0 \leq \varrho_1\varrho_2 \leq 1, \tag{3.35}$$

$$\text{Mixed methods :} \quad \varrho_1\varrho_2 \leq 0, \tag{3.36}$$

which implies the following rates of convergence

$$\text{D/D :} \qquad \varrho = |1 + \theta(\varrho_{\text{seq}} - 1)|$$

$$\text{Mixed methods :} \quad \varrho = |1 - \theta(\varrho_{\text{seq}} + 1)|,$$

and the following equations for $\theta_{\max}$,

$$\text{D/D :} \qquad \theta_{\max} = \frac{2}{1 - \varrho_{\text{seq}}},$$

$$\text{Mixed methods :} \quad \theta_{\max} = \frac{2}{1 + \varrho_{\text{seq}}}.$$

In addition, all the DD methods admit an optimum relaxation parameter θ_{opt} for which $\varrho = 0$. Let us define an iteration as a complete cycle for which we solve each subdomain; remembering that Equation (3.33) is valid only for $k \geq 1$ (and not $k \geq 0$), therefore the DD methods converge in at most two iterations when $\theta = \theta_{\text{opt}}$ with

$$\text{D/D :} \qquad \theta_{\text{opt}} = \frac{1}{1 - \varrho_{\text{seq}}} \geq 1,$$

$$\text{Mixed methods :} \quad \theta_{\text{opt}} = \frac{1}{1 + \varrho_{\text{seq}}} \leq 1.$$

Figure 3.3 shows how the relaxation parameters acts on the rates of convergence of the relaxed sequential algorithms, as a function of the sequential rate of convergence ϱ_{seq}. It enables also to appreciate how an optimum relaxation parameter can be chosen so that the mixed methods converge, whatever the corresponding sequential rate of convergence ϱ_{seq} is.

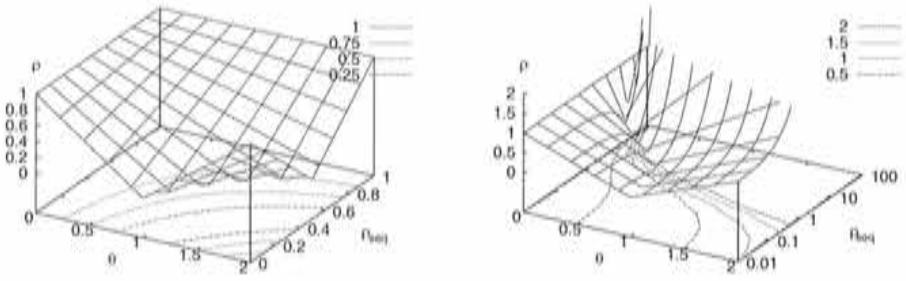


Figure 3.3: Relaxed sequential version. Rate of convergence as a function of θ and ϱ_{seq}. (Left) D/D method. (Right) Mixed methods.

Remark 1. In the hyperbolic limit, we have shown that the D/D, R/D and D/R methods coincide. Their rates of convergence ρ tend to zero, and therefore, their optimum relaxation parameter is $\theta_{\text{opt}} = 1$.

Example 1. As a first illustration of the under-relaxation, we consider the D/N method for solving the advection-diffusion (without reaction) equation in the hyperbolic limit. This case is precisely the most difficult as we have shown that it requires an overlap of the size of the Neumann subdomain to achieve convergence! From Equation (3.29) with R = 0, we find that

$$\varrho_1 \varrho_2 \approx -\exp\left[\mathrm{Pe}(\ell_2^* - \delta^*)\right].$$

The maximum permissible θ is therefore

$$\theta_{\max} = \frac{2}{1 + \exp\left[\mathrm{Pe}(\ell_2^* - \delta^*)\right]}, \tag{3.37}$$

which in the hyperbolic limit gives

$$\lim_{\mathrm{Pe}\to\infty} \theta_{\max} = 0^+.$$

This asymptotic value is represented by the convergence of the contour line of value 1 in Figure 3.3 (Right). Theoretically, condition (3.37) can be fulfilled. However, small values of the relaxation parameters can be prohibitive when performing numerical experimentations, for which the effects of round off errors become important.

Example 2. The second example studies the convergence criterion for θ, using the N/D and D/N methods for solving the Poisson equation. The maximum permissible θ is

$$\mathrm{N/D}: \quad \theta_{\max} = \frac{2}{1 + (\ell_1^* - \delta^*)/(\ell_2^* + \delta^*)},$$

$$\mathrm{D/N}: \quad \theta_{\max} = \frac{2}{1 + (\ell_2^* - \delta^*)/(\ell_1^* + \delta^*)}.$$

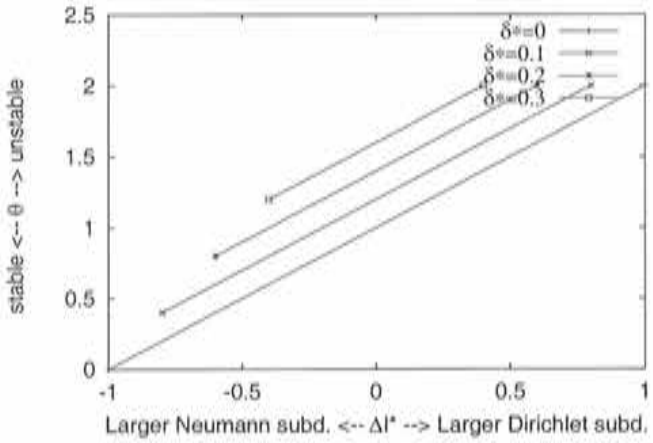


Figure 3.4: Poisson equation. Stability curve of the N/D and D/N methods for various δ^*'s.

Using the definition of $\Delta\ell^*$ introduced in the study of the unrelaxed DD algorithm for the Poisson equation (see Equations (3.25) and (3.26)), the stability criterion of the N/D and D/N methods in terms of θ is exactly the same and can be written as

$$\theta_{\max} = 1 + \Delta\ell^* + 2\delta^*.$$

Figure (3.4) shows the stability curve obtained for different overlaps δ^*. Remember that, owing to Equation (3.27), $\Delta\ell^*$ is bounded from below and above with respect to δ^* according to

$$|\Delta\ell^*| < 1 - 2\delta^*.$$

Choosing θ in the range below the curve leads to a stable scheme, whereas the zone above the curve leads to an unstable scheme. We see that in any case, the instability due to the asymmetry of the problem can be circumvented by choosing a sufficiently small relaxation parameter. We note also that the larger the Dirichlet subdomain, the larger the maximum permissible θ. Finally, we remark that the maximum possible value of θ over all $\Delta\ell^*$ and δ^* is $\theta = 2$.

3.5 Unrelaxed parallel version

The parallel version of the domain decomposition algorithm to solve system $(3.1)_{1-2}$ is given by Equations $(3.3)_{1-3}$ and $(3.4)_{1-3}$ choosing $k' = k$ in Equation $(3.4)_3$.

The parallel version of the DD algorithms is first presented without under-relaxation, and compared to its sequential counterpart. When no under-relaxation is used, all the parallel versions of the DD algorithms lead to the following systems of equations for $k \geq 0$:

$$\begin{aligned} C_1^{k+1} &= \varrho_1 C_2^k, \\ C_2^{k+1} &= \varrho_2 C_1^k, \end{aligned}$$

which gives the following iteration matrix:

$$\mathbf{C} = \begin{bmatrix} 0 & \varrho_1 \\ \varrho_2 & 0 \end{bmatrix}.$$

The equation for the eigenvalues is

$$\lambda^2 = \varrho_1 \varrho_2,$$

which gives the following rate of convergence

$$\begin{aligned} \varrho &= \sqrt{|\varrho_1 \varrho_2|} \\ &= \sqrt{\varrho_{\text{seq}}}. \end{aligned} \tag{3.38}$$

Therefore, the convergence condition $\varrho < 1$ for the parallel version is the same as that of the sequential version, although its rate of convergence is the square of that of the sequential method. In addition, we have no guarantee of monotonicity. Nevertheless, the simple form of the iteration matrix allows us to derive easily the expression for the $\|\cdot\|_2$ norm of $\mathbf{C}$:

$$\|\mathbf{C}\|_2 = \max\left(|\varrho_1|, |\varrho_2|\right)$$

and therefore we can establish the following norm estimate

$$\|\mathbf{e}^k\|_2 \leq \max\left(|\varrho_1|, |\varrho_2|\right) \|\mathbf{e}^{k-1}\|_2.$$

3.6 Relaxed parallel version

3.6.1 General relaxation

The iteration matrix of the relaxed parallel version is

$$\mathbf{C} = \begin{bmatrix} 1-\theta_1 & \theta_1 \varrho_1 \\ \theta_2 \varrho_2 & 1-\theta_2 \end{bmatrix},$$

which gives the following characteristic equation

$$\lambda^2 + (\theta_1 + \theta_2 - 2)\lambda + 1 - \theta_1 - \theta_2 + \theta_1 \theta_2 (1 - \varrho_1 \varrho_2) = 0, \tag{3.39}$$

which corresponding determinant Δ is given by

$$\Delta = (\theta_1 - \theta_2)^2 + 4\theta_1 \theta_2 \varrho_1 \varrho_2.$$

We first note that the action of both relaxation parameters is symmetric, i.e. we can interchange their values without modifying the resulting algorithm. In Section 3.4, we have shown that, for the D/D method, the product $\varrho_1 \varrho_2$ is always positive, while the mixed methods give a negative product, as stated by Equations (3.35) and (3.36). We must therefore treat the D/D apart from the mixed methods.

D/D method. Owing to Equation (3.35), the product $\varrho_1\varrho_2 = \varrho_{\text{seq}}$ is positive and therefore the eigenvalues solution of Equation (3.39) are real; they are

$$\lambda_1 = \frac{1}{2}\left((2-\theta_1-\theta_2) - \sqrt{(\theta_1-\theta_2)^2 + 4\theta_1\theta_2\varrho_{\text{seq}}}\right),$$

$$\lambda_2 = \frac{1}{2}\left((2-\theta_1-\theta_2) + \sqrt{(\theta_1-\theta_2)^2 + 4\theta_1\theta_2\varrho_{\text{seq}}}\right),$$

which give the following rate of convergence:

$$\varrho = \frac{1}{2}\left(|2-\theta_1-\theta_2| + \sqrt{(\theta_1-\theta_2)^2 + 4\theta_1\theta_2\varrho_{\text{seq}}}\right).$$

Owing to the expression of ϱ, we conclude that there does not exist a pair of relaxation parameter for which $\varrho = 0$. In addition, we see that the optimum rate of convergence is achieved for $\theta_1 = \theta_2 = 1$; therefore, we have no way to improve the rate of convergence achieved by the unrelaxed parallel version, given by Equation (3.38). We are now going to see that the situation is different for the mixed methods.

Mixed methods. Owing to Equation (3.36), product $\varrho_1\varrho_2 = -\varrho_{\text{seq}}$. Thus, the determinant of the characteristic equation is

$$\Delta = (\theta_1-\theta_2)^2 - 4\theta_1\theta_2\varrho_{\text{seq}}, \tag{3.40}$$

and the equation for the eigenvalues (3.39) admits real or complex solutions, according to the sign of Δ. We can easily see that the rate of convergence can be written as

$$\varrho = \frac{1}{2}\begin{cases} \sqrt{(2-\theta_1-\theta_2)^2 - \Delta} & \text{if } \Delta < 0 \\ |2-\theta_1-\theta_2| + \sqrt{\Delta} & \text{if } \Delta \geq 0. \end{cases} \tag{3.41}$$

Let us try to simplify this expression. Noting that

$$\frac{1}{2}(|\Delta| + \Delta) = \begin{cases} 0 & \text{if } \Delta < 0 \\ \Delta & \text{if } \Delta \geq 0, \end{cases} \quad \text{and}$$

$$\sqrt{(2-\theta_1-\theta_2)^2 + \frac{|\Delta|-\Delta}{2}} = \begin{cases} \sqrt{(2-\theta_1-\theta_2)^2 - \Delta} & \text{if } \Delta < 0 \\ |2-\theta_1-\theta_2| & \text{if } \Delta \geq 0, \end{cases}$$

we can rewrite Equation (3.41) for ϱ in the following form

$$\varrho = \frac{1}{2}\left(\sqrt{\frac{|\Delta|+\Delta}{2}} + \sqrt{(2-\theta_1-\theta_2)^2 + \frac{|\Delta|-\Delta}{2}}\right).$$

At a first glance, we see that we can achieve $\varrho = 0$ by setting both Δ and the term $2-\theta_1-\theta_2$ to zero. The latter term vanishes if we introduce a unique relaxation parameter θ such that $0 < \theta < 2$, and let

$$\theta_1 = \theta, \tag{3.42}$$

$$\theta_2 = 2-\theta. \tag{3.43}$$

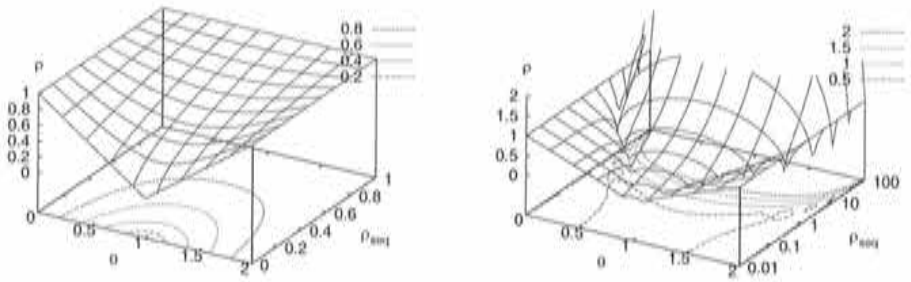


Figure 3.5: Optimum relaxed parallel version. Rate of convergence as a function of θ and ϱ_{seq}. (Left) D/D method. (Right) Mixed methods.

The following section studies the convergence of this "optimum relaxation".

3.6.2 Optimum relaxation

This optimum relaxation uses a unique relaxation parameter to define θ_1 and θ_2, as given by Equations (3.42) and (3.43).

D/D method. We have mentioned that the optimum rate of convergence of the relaxed parallel D/D method is obtain setting $\theta_1 = \theta_2 = 1$, i.e. no relaxing at all. Anyway, it is interesting to study the behavior of the D/D method using the optimum relaxation of the mixed method. For the rate of convergence, we find:

$$\varrho = \sqrt{1 + \theta(\theta - 2)(1 - \varrho_{\text{seq}})}.$$

Imposing $\varrho < 1$, we can derive the following inequality for the maximum possible relaxation parameter

$$\theta_{\max} = 2,$$

which is independent of ϱ_{seq}. Figure (3.5) (Left) shows the variation of the rate of convergence as a function of θ and ϱ_{seq}.

Mixed methods. The rate of convergence is

$$\varrho = \frac{1}{2\sqrt{2}} \left(\sqrt{|\Delta| + \Delta} + \sqrt{|\Delta| - \Delta} \right), \tag{3.44}$$

with Δ given by

$$\Delta = 4(\theta - 1)^2 - 4\theta(2 - \theta)\varrho_{\text{seq}}.$$

Equation (3.44) is represented by Figure (3.5) (Right). In order to achieve $\varrho = 0$, we just have to set the determinant to zero. Solving Equation (3.40) for $\Delta = 0$, we find two optimum relaxation

factors θ^+_{opt} and θ^-_{opt}

$$\begin{aligned} \theta^{\pm}_{\text{opt}} &= 1 \pm \sqrt{\frac{\varrho_{\text{seq}}}{\varrho_{\text{seq}}+1}}, \qquad \text{and} \\ \varrho_{\text{opt}} &= 0. \end{aligned} \tag{3.45}$$

The range of admissible relaxation factor can be determined by solving the problem $\varrho < 1$. We find that the mixed methods are convergent when

$$\begin{aligned} &0 < \theta < \theta_{\max} \text{ or } (2-\theta_{\max}) < \theta < 2 \qquad \text{if } \varrho_{\text{seq}} \geq 1, \\ &0 < \theta < 2 \qquad \text{if } \varrho_{\text{seq}} < 1, \end{aligned}$$

with $\theta_{\max}$ given by

$$\theta_{\max} = 1 - \sqrt{\frac{\varrho_{\text{seq}}-1}{\varrho_{\text{seq}}+1}}.$$

We note that when ϱ_{seq} tends to infinity, $\theta_{\max}$ tends to zero.

3.6.3 Other relaxations

In this section we compare some relaxation strategies. In addition to the optimum relaxation we have already presented, we introduce two other strategies. We call the first one "Unique relaxation". As explicitly indicated by its name, this strategy uses only one relaxation, the other being set to unity. We also introduce an "Equal relaxation", for which both relaxation parameters take the same value. We do not derive the equations for the rates of convergence and we limit ourselves to presenting some plots to have an idea of how the different relaxation algorithms act. Let us sum up the choice of relaxation parameters for the three versions studied:

$$\begin{aligned} &\text{Optimum relaxation :} && \theta_1 = \theta, \theta_2 = 2-\theta, \\ &\text{Equal relaxation :} && \theta_1 = \theta_2 = \theta. \\ &\text{Unique relaxation :} && \theta_1 = \theta, \theta_2 = 1. \end{aligned}$$

Figures (3.6) shows the dependence of the rate of convergence on the relaxation parameter θ, for various sequential rates of convergence ϱ_{seq}. We can appreciate the fact that for the D/D method, the optimum relaxation factors are $\theta_1 = \theta_2 = 1$ for the three relaxation strategies studied. Nevertheless, it should be pointed out that for given θ and ϱ_{seq}, the rate of convergence ϱ of the optimum relaxation is always smaller than that of the other ones and that, taking $0 < \theta < 2$, we are always ensured that $\varrho < 1$. In addition, the derivative of ϱ near the optimum point at $\theta = 1$ is continuous and the function smooth. This relaxation should be preferred to the other ones.

As for the mixed method, the conclusions are much more evident. We have shown that the optimum relaxation admits an optimum θ for which the rate of convergence is zero. This property is not shared by the other two strategies. Nevertheless, let us mention that the derivative of the rate of convergence near the optimum point is higher for this strategy, which makes it more sensitive to changes in θ near the optimum point with respect to the other ones, around their respective optimum point.

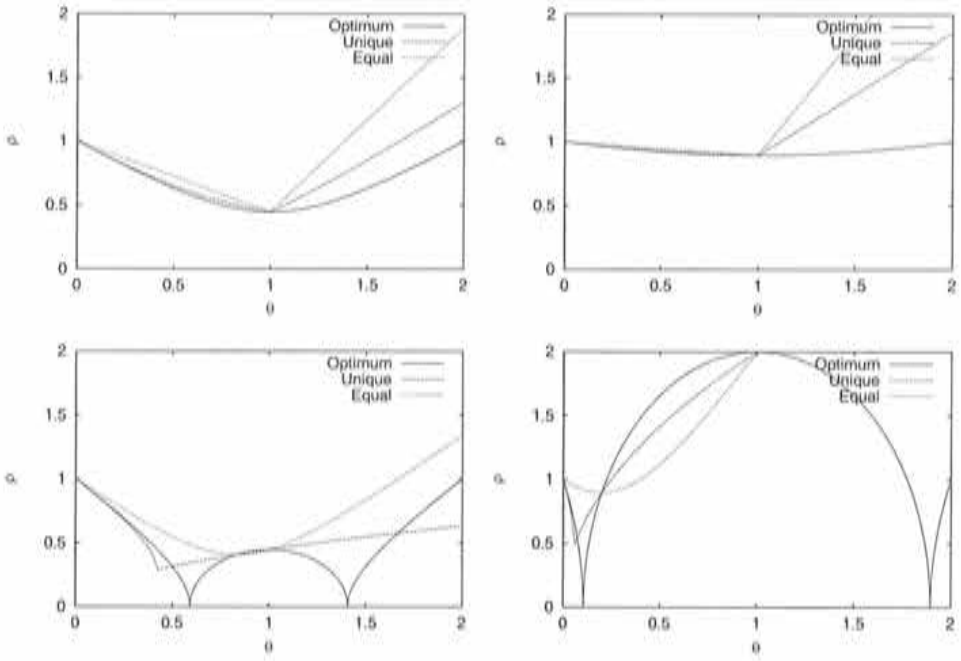


Figure 3.6: Comparison of different relaxed parallel algorithms. (Top) (Left) D/D with $\varrho_{\text{seq}} = 0.2$. (Top) (Right) D/D with $\varrho_{\text{seq}} = 0.8$. (Bot.) (Left) Mixed methods with $\varrho_{\text{seq}} = 0.2$. (Bot.) (Right) Mixed methods with $\varrho_{\text{seq}} = 4$.

Let us conclude this section about the relaxed parallel version. We have shown that there always exist a θ for which parallel version of the DD methods converge. In addition, the parallel version is obviously computationally more attractive than its sequential counterpart. However, monotonicity of the convergence is not guaranteed and there does not exist an optimal θ for which $\varrho = 0$. Numerical parallelism can therefore be only achieved at the expense of lower stability and slower convergence speed.

3.7 Conclusions

During the analysis of the one-dimensional problem, we have been studying the behaviors of five domain decomposition methods. We have first considered the unrelaxed sequential versions and established the conditions on the overlapping length so that the DD methods converge. The following remarks have been made:

1. The D/D converges whenever the overlapping length is not zero. The higher the Péclet number, the smaller the rate of convergence.

2. The N/D is the most appropriated method for advection dominated flows, as the Neumann and Dirichlet conditions are imposed according to the hyperbolic character of the equation.

3. The D/N, which contradicts the latter remark, cannot converge for high Péclet numbers, when the differential equation has no reaction term. However, when the reaction term is present, we have shown that a minimum overlapping length enables to achieve convergence in any case. Note that in the transient case, the reaction term comes from the time discretization

with $s \sim 1/\delta t$, δt being the time step. We expect therefore transient problems to be even more favorable to the overlapping D/N method.

4. The R/D and D/R methods acts symmetrically with respect to the transmission conditions. Therefore, the Dirichlet and Robin conditions can be imposed without regards to the direction of the flow. In addition, the overlapping length can be chosen so that the methods convergence for all Péclet numbers. When $\mathrm{Pe} \gg 1$, the rates of convergence of the R/D and D/R methods tend to that of the D/D method.

5. In all cases, convergence is monotone.

In order to enable convergence when no control was possible on the overlapping length, we have introduced relaxation of the transmission conditions. We have found that setting one of the relaxation parameters to unity, we could achieve convergence for all the DD methods, and for any Péclet number; as in the case of the unrelaxed sequential version, this convergence is monotone. In addition, we have shown that there always exists an optimum relaxation parameter for which the DD methods converge in at most two iterations.

Then we have presented the parallel versions of the D/D method. The conditions for convergence are exactly the same as those of the sequential method. All methods exhibit a rate of convergence which is the square root of that of the sequential versions.

Finally, we have relaxed the transmission conditions of the parallel version. On the one hand, we have shown that we have no way of finding relaxation parameters so that the D/D method converges in at most two iterations. On the other hand, this optimal convergence can be achieved for the mixed method choosing the so-called optimum relaxation, using $\theta_1 = \theta$ and $\theta_2 = 2 - \theta$ for θ given by Equation (3.45). Finally, let us comment that we have not tried to calculate a norm estimate; in general, the methods are not monotone.

Table 3.1 shows the rates of convergence obtained for the relaxed sequential and parallel methods, as well as for the respective relaxed versions. Table 3.2 sums up the values of the optimum

Method	Seq.	Par.	Rel. Seq. $\theta_1 = \theta, \theta_2 = 1$	Rel. Par. $\theta_1 = \theta, \theta_2 = 2 - \theta$
D/D	ϱ_{seq}	$\sqrt{\varrho_{\mathrm{seq}}}$	$\lvert 1 + \theta(\varrho_{\mathrm{seq}} - 1)\rvert$	$\sqrt{1 + \theta(\theta - 2)(1 - \varrho_{\mathrm{seq}})}$
Mixed	ϱ_{seq}	$\sqrt{\varrho_{\mathrm{seq}}}$	$\lvert 1 - \theta(\varrho_{\mathrm{seq}} + 1)\rvert$	$\frac{1}{2\sqrt{2}}\left(\sqrt{\lvert\Delta\rvert + \Delta} + \sqrt{\lvert\Delta\rvert - \Delta}\right)^{(*)}$

Table 3.1: Rates of convergence, with $\varrho_{\mathrm{seq}} = |\varrho_1 \varrho_2|$. $(*)$ $\Delta = 4(\theta - 1)^2 - 4\theta(2 - \theta)\varrho_{\mathrm{seq}}$.

relaxation factors and their corresponding optimum rates of convergence.

This one-dimensional study has enabled to point out the importance of three main factors in the convergence of overlapping methods, namely the relative geometries of the subdomains, the relaxation factor, and the overlapping length. Next chapter treats the multidimensional problem at the variational level. In particular, we will recognize easily the importance of the relaxation factor. However, the effects of the sizes of the subdomains and that of the overlapping zone will

Method	Parameter	Sequential $\theta_1 = \theta, \theta_2 = 1$	Parallel $\theta_1 = \theta, \theta_2 = 2 - \theta$
D/D	θ_{opt}	$\dfrac{1}{1 - \varrho_{\text{seq}}}$	1
	ϱ_{opt}	0	$\sqrt{\varrho_{\text{seq}}}$
Mixed	θ_{opt}	$\dfrac{1}{1 + \varrho_{\text{seq}}}$	$\pm\sqrt{\dfrac{\varrho_{\text{seq}}}{\varrho_{\text{seq}} + 1}}$
	ϱ_{opt}	0	0

Table 3.2: Optimum relaxation parameter and optimum rate of convergence.

not be so clear, as they will be embedded inside the norm estimates of the operators in play. In any case, the one-dimensional frame is a preview of what happens in multidimensional problems and is a precious hint before performing a numerical experimentation.

Chapter 4

An Overlapping Domain Decomposition Method

In Chapter 2 we contemplated the possibility of using overlapping mixed methods for solving iteration-by-subdomain problems. We justified their mathematical foundation through a simple example and intuited their advantage over their disjoint counterparts. In the next chapter, the study of a one-dimensional ADR equation confirmed what we had advanced concerning the effects of the overlapping length, and in particular, the possibility of achieving convergence in the hyperbolic limit even when the Neumann and Dirichlet interfaces were not placed in accordance to the direction of the flow. In this chapter, we propose to make one step beyond and to study the convergence of overlapping mixed methods within a variational framework applied to the solution of a scalar advection-diffusion-reaction equation. This chapter constitutes therefore a theoretical basis for the study of overlapping mixed methods [96, 97].

As a beginning, we introduce the continuous problem, derive the corresponding variational formulation, and look for a weak solution. Then we present a new overlapping domain decomposition method. The starting point is a two-domain variational formulation of the problem, originating from a geometrical decomposition of the original domain of study; we follow the strategy presented in [98] for the classical Dirichlet/Neumann method and extensively studied in [58]. We show how the formulation can be reformulated into an overlapping domain decomposition method based on a Dirichlet/Robin coupling and how this formulation can be simply derived from a differential problem. Next, the domain decomposition method for the subdomains is re-written in terms of a problem for the interface unknowns. An iterative and relaxed sequential scheme is then introduced in order to solve the DD problem. The convergence is studied through the interface equations. We present the generalization of the overlapping DD method introduced to other types of overlapping mixed couplings, in particular to an overlapping Dirichlet/Neumann method. Afterwards, we consider the discrete counterpart of the formulation. We then build an iterative strategy to solve the two-domain problem. This strategy is studied algebraically, using a finite element method for the spatial discretization and solving for the Schur complement system, and the overlapping D/N and D/R methods are illustrated by four numerical examples. We mention briefly the possibility of parallelizing the algorithm using two or many subdomains. Finally, we introduce the extension of the overlapping Dirichlet/Neumann method to the Stokes and Navier-Stokes equations.

4.1 Problem statement

We propose to study the following ADR problem:

$$\begin{cases} Lu := -\varepsilon\Delta u + \nabla\cdot(\boldsymbol{a}u) + su &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{cases}$$

already introduced in the first chapter. We choose to study the 1/2-weak formulation presented in Section 1.1.3, and keep using the same notation. The weak formulation of the original problem reads: find $u \in V$ such that

$$a(u,v) = \langle f, v\rangle_\Omega \qquad \forall\, v \in V, \tag{4.1}$$

where the bilinear form is

$$a(w,v) := \varepsilon(\nabla w, \nabla v) + \frac{1}{2}(\boldsymbol{a}\cdot\nabla w, v) - \frac{1}{2}(w, \boldsymbol{a}\cdot\nabla v) + (s_0 w, v),$$

with

$$s_0 = s + \frac{1}{2}\nabla\cdot\boldsymbol{a}.$$

From the Lax-Milgram lemma, if $s_0 \geq 0$ almost everywhere, Problem (4.1) has a unique solution. From now on, we assume that $s_0 \geq 0$ almost everywhere.

4.2 Overlapping Dirichlet/Robin method

4.2.1 Domain partitioning and definitions

We perform a geometrical decomposition of the original domain Ω into three disjoint and connected subdomains Ω_3, Ω_4 and Ω_5 such that

$$\Omega = \text{int}\left(\overline{\Omega_3 \cup \Omega_4 \cup \Omega_5}\right).$$

From this partition, we define Ω_1 and Ω_2 as two overlapping subdomains:

$$\Omega_1 := \text{int}\left(\overline{\Omega_3 \cup \Omega_4}\right), \quad \Omega_2 := \text{int}\left(\overline{\Omega_5 \cup \Omega_4}\right).$$

Finally, we define Γ_a as the part of $\partial\Omega_2$ lying in Ω_1, and Γ_b as the part of $\partial\Omega_1$ lying in Ω_2, formally given by

$$\Gamma_a := \overline{\partial\Omega_2 \cap \Omega_1}, \quad \Gamma_b := \overline{\partial\Omega_1 \cap \Omega_2}.$$

The geometrical nomenclature is shown in Figure 4.1. Γ_b and Γ_a are the *interfaces* of the domain decomposition method we now present. Ω_4 is the overlap zone. In the following, index i or j refer to a subdomain or an interface.

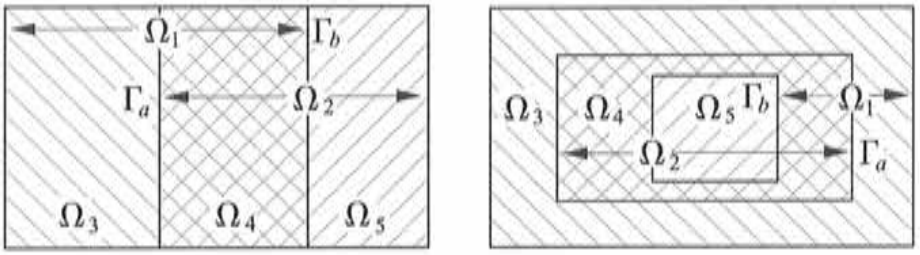


Figure 4.1: Examples of decomposition of domain Ω into two overlapping subdomains Ω_1 and Ω_2.

We set the following definitions to be used for the variational formulation of the two-domain problem:

$$\begin{aligned}
(w,v)_{\Omega_i} &:= \int_{\Omega_i} wv\, d\Omega, \qquad \text{for } w \text{ and } v \text{ in } L_2(\Omega_i),\\
a_i(w,v) &:= \varepsilon(\nabla w, \nabla v)_{\Omega_i} + (s_0 w, v)_{\Omega_i} + \frac{1}{2}(\boldsymbol{a}\cdot\nabla w, v)_{\Omega_i} - \frac{1}{2}(w, \boldsymbol{a}\cdot\nabla v)_{\Omega_i}, \qquad (4.2)\\
V_i &:= \{v \in H^1(\Omega_i) \mid v_{|\partial\Omega\cap\partial\Omega_i} = 0\},\\
V_i^0 &:= H_0^1(\Omega_i),
\end{aligned}$$

where i can be any of the five subdomains introduced previously, i.e. $i = 3, 4, 5, 1$ or 2.

We now introduce the trace operator. From the trace theorem (see e.g. [2, 3]), we know that there exists a unique linear continuous map $\gamma_{0,i}$, called the trace operator, defined as

$$\gamma_{0,i} : V_i \longrightarrow H^{1/2}(\partial\Omega_i), \text{ such that } \gamma_{0,i} v_i = v_{i|\partial\Omega_i}\ \forall\, v_i \in V_i,\ i = 3,4,5,1,2,$$

and that this result applies equivalently to any Lipschitz continuous subset of $\partial\Omega_i$. Let us denote T_a and T_b as the trace operators restricted to Γ_a and Γ_b, respectively. They are defined by:

$$\begin{aligned}
T_a &: V \longrightarrow H_{00}^{1/2}(\Gamma_a), \qquad T_a v = v_{|\Gamma_a}\ \forall\, v \in V,\\
T_b &: V \longrightarrow H_{00}^{1/2}(\Gamma_b), \qquad T_b v = v_{|\Gamma_b}\ \forall\, v \in V.
\end{aligned}$$

In addition, we explicitly define the trace spaces on Γ_a and Γ_b as

$$\Lambda_a := \{\mu_a \in H_{00}^{1/2}(\Gamma_a)\}, \quad \Lambda_b := \{\mu_b \in H_{00}^{1/2}(\Gamma_b)\};$$

obviously, for any $v \in V$, we have $v_{|\Gamma_a} \in \Lambda_a$, and $v_{|\Gamma_b} \in \Lambda_b$.

We also introduce some basic properties of the spaces we are working with; as many constants are going to be introduced, we adopt a general nomenclature. We enunciate three inequalities (Poincaré-Friedrichs, trace inequalities and an a-priori estimate) that characterize the functions belonging to our work spaces, i.e. $H^1(\Omega)$ and $H_0^1(\Omega)$. The domains of study are the original domain Ω and its five partitions Ω_i with $i = 3, 4, 5, 1, 2$. The Poincaré-Friedrichs inequality reads

$$\|v\|_{0,\Omega_i}^2 \le C_{\Omega_i} \|\nabla v\|_{0,\Omega_i}^2 \qquad \forall\, v \in H_0^1(\Omega_i). \qquad (4.3)$$

where C_{Ω_i} is a positive constant depending on the size of the domain Ω_i. The space of application $H_0^1(\Omega_i)$ can be actually extended to any subspace of $H^1(\Omega_i)$ for which the trace is specified "somewhere" on $\partial\Omega_i$.

The trace inequality is a direct consequence of the trace theorem; it states that there exists a positive constant C_i^* such that

$$\|v_{|\partial\Omega_i}\|_{1/2,\partial\Omega_i} \leq C_i^* \|v\|_{1,\Omega_i} \qquad \forall\, v \in H^1(\Omega_i). \tag{4.4}$$

Finally, the following a-priori estimate for the solution v of homogeneous elliptic problems with Dirichlet data holds (see e.g. [2, 3]):

$$\|v\|_{1,\Omega_i} \leq C_i \|v_{|\partial\Omega_i}\|_{1/2,\partial\Omega_i}. \tag{4.5}$$

This establishes the continuous dependence of the solution on the boundary data and closes the list of properties.

4.2.2 Variational formulation

We propose to solve the following problem: find $u_1 \in V_1$ and $u_2 \in V_2$ such that

$$\begin{cases} a_1(u_1, v_1) = \langle f, v_1 \rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1 = u_2 & \text{on } \Gamma_b, \\ a_2(u_2, v_2) = \langle f, v_2 \rangle_{\Omega_2} & \forall\, v_2 \in V_2^0, \\ a_3(u_1, E_3\mu_a) + a_2(u_2, E_2\mu_a) = \langle f, E_3\mu_a \rangle_{\Omega_3} + \langle f, E_2\mu_a \rangle_{\Omega_2} & \forall\, \mu_a \in \Lambda_a, \end{cases} \tag{4.6}$$

where E_i denotes any possible extension operator, such that

$$\begin{aligned} &E_i : \Lambda_a \longrightarrow H^1(\Omega_i), \\ &T_a E_i \mu_a = \mu_a \qquad \forall\, \mu_a \in \Lambda_a. \end{aligned}$$

Equations $(4.6)_1$ and $(4.6)_3$ are the equations for the unknown in subdomains Ω_1 and Ω_2 respectively; in fact, observe that both test functions v_1 and v_2 vanish on the whole boundaries of Ω_1 and Ω_2 respectively, including on their interfaces. Equation $(4.6)_2$ is the condition which ensures continuity of the primary variable across Γ_b, and levels the solution in both subdomains. Finally, equation $(4.6)_4$ is the equation for the primary variable on the interface Γ_a.

Theorem 4.1. *Problems* (4.6) *and* (4.1) *are equivalent.*

Proof. We first show that the solution is the same in both subdomains inside the overlap zone Ω_4, i.e. that the two transmission conditions on the interfaces are sufficient to uniquely define the solution. For any $v_4 \in V_4^0$, construct

$$v_1 = \begin{cases} 0 & \text{in } \Omega_3, \\ v_4 & \text{in } \Omega_4, \end{cases}$$

$$v_2 = \begin{cases} v_4 & \text{in } \Omega_4, \\ 0 & \text{in } \Omega_5. \end{cases}$$

Clearly, $v_1 \in V_1^0$ and $v_2 \in V_2^0$ and therefore subtracting $(4.6)_1$ and $(4.6)_3$, we obtain

$$a_4(u_1 - u_2, v_4) = 0 \qquad \forall\, v_4 \in v_4^0$$

together with the condition $u_1 - u_2 = 0$ on Γ_b, derived from $(4.6)_2$. Now, we need to derive a boundary condition on Γ_a in order to close the problem for the unknown $u_1 - u_2$. For any $\mu_a \in \Lambda_a$ define

$$v_1 = \begin{cases} E_3\mu_a & \text{in } \Omega_3, \\ E_4\mu_a & \text{in } \Omega_4. \end{cases}$$

Since $v_1 \in V_1^0$, $(4.6)_1$ gives

$$a_3(u_1, E_3\mu_a) + a_4(u_1, E_4\mu_a) = \langle f, E_3\mu_a\rangle_{\Omega_3} + \langle f, E_4\mu_a\rangle_{\Omega_4} \qquad \forall\, \mu_a \in \Lambda_a.$$

Substituting last expression into equation $(4.6)_4$, we find

$$a_2(u_2, E_2\mu_a) - a_4(u_1, E_4\mu_a) = \langle f, E_2\mu_a\rangle_{\Omega_2} - \langle f, E_4\mu_a\rangle_{\Omega_4} \qquad \forall\, \mu_a \in \Lambda_a. \tag{4.7}$$

Now we define for all $\mu_a \in \Lambda_a$

$$v_2' = \begin{cases} E_4\mu_a & \text{in } \Omega_4, \\ 0 & \text{in } \Omega_5. \end{cases}$$

Equation (4.7) can be rewritten as

$$\begin{aligned} &a_2(u_2, E_2\mu_a - v_2') + a_2(u_2, v_2') - a_4(u_1, E_4\mu_a) \\ &\quad = \langle f, E_2\mu - v_2'\rangle_{\Omega_2} + \langle f, v_2'\rangle_{\Omega_2} - \langle f, E_4\mu_a\rangle_{\Omega_4} \qquad \forall\, \mu_a \in \Lambda_a. \end{aligned} \tag{4.8}$$

According to the definition of v_2', $(E_2\mu_a - v_2') \in V_2^0$ and consequently, applying $(4.6)_3$, we obtain

$$a_2(u_2, E_2\mu_a - v_2') = \langle f, E_2\mu - v_2'\rangle_{\Omega_2}.$$

Equation (4.8) gives therefore

$$a_4(u_2, E_4\mu_a) - a_4(u_1, E_4\mu_a) = \langle f, E_4\mu_a\rangle_{\Omega_4} - \langle f, E_4\mu_a\rangle_{\Omega_4} \qquad \forall\, \mu_a \in \Lambda_a,$$

which is equivalent to

$$a_4(u_1 - u_2, E_4\mu_a) = 0 \qquad \forall\, \mu_a \in \Lambda_a.$$

As a result, the complete system of equations for $w = u_1 - u_2$ is

$$\begin{cases} a_4(w, v_4) = 0 & \forall\, v_4 \in V_4^0, \\ w = 0 & \text{on } \Gamma_b, \\ a_4(w, E_4\mu_a) = 0 & \forall\, \mu_a \in \Lambda_a. \end{cases}$$

From Lax-Milgram lemma, this problem has a unique solution $w = 0$; this implies that $u_1 = u_2$ in Ω_4.

We now show that the solution of the original problem is also solution of the domain decomposition problem. Let u be solution of equation (4.1), and define $u_i = u_{|\Omega_i}$ for $i = 1, 2$. Clearly, $u_i \in V_i$ and therefore equations $(4.6)_1$, $(4.6)_2$ and $(4.6)_3$ are trivially satisfied. Now for all $\mu_a \in \Lambda_a$ define γ as

$$\gamma = \begin{cases} E_3\mu_a & \text{in } \Omega_3, \\ E_2\mu_a & \text{in } \Omega_2. \end{cases}$$

We have $\gamma \in V$, which implies that

$$a(u, \gamma) = \langle f, \gamma \rangle_\Omega,$$

and substituting the definition of γ into this equation we recover equation $(4.6)_4$. We now prove the reciprocal. Let

$$u = \begin{cases} u_{1|\Omega_3} & \text{in } \Omega_3, \\ u_2 & \text{in } \Omega_2. \end{cases}$$

We proved that $u_1 = u_2$ in Ω_4 and in particular that $u_1 = u_2$ on Γ_a. This implies that $u \in V$; as a result, we have

$$a(u, v) = a_3(u_1, v) + a_2(u_2, v) \qquad \forall\, v \in V. \tag{4.9}$$

For each $v \in V$, set $\mu_a = T_a v \in \Lambda_a$. Let us define

$$\begin{aligned} \gamma_3 &= v_{|\Omega_3} - E_3\mu_a, \\ \gamma_2 &= v_{|\Omega_2} - E_2\mu_a. \end{aligned}$$

and rewrite equation (4.9) as

$$a(u, v) = a_3(u_1, \gamma_3) + a_3(u_1, E_3\mu_a) + a_2(u_2, \gamma_2) + a_2(u_2, E_2\mu_a) \qquad \forall\, \mu_a \in \Lambda_a. \tag{4.10}$$

From its definition, $\gamma_3 \in V_3^0$. Let us now define γ_1 as

$$\gamma_1 = \begin{cases} \gamma_3 & \text{in } \Omega_3, \\ 0 & \text{in } \Omega_4. \end{cases}$$

$\gamma_1 \in V_1^0$ and therefore, applying $(4.6)_1$,

$$a_1(u_1, \gamma_1) = \langle f, \gamma_1 \rangle_{\Omega_1},$$

which gives

$$a_3(u_1, \gamma_3) = \langle f, \gamma_3 \rangle_{\Omega_3}.$$

Knowing also that $\gamma_2 \in V_2^0$, we have

$$a_2(u_2, \gamma_2) = \langle f, \gamma_2 \rangle_{\Omega_2},$$

and from the latter two equations, equation (4.10) becomes

$$a(u,v) = \langle f, \gamma_3 \rangle_{\Omega_3} + a_3(u_1, E_3\mu_a) + \langle f, \gamma_2 \rangle_{\Omega_2} + a_2(u_2, E_2\mu_a) \qquad \forall\, \mu_a \in \Lambda_a.$$

From equation $(4.6)_4$, the last equation reads

$$a(u,v) = \langle f, \gamma_3 \rangle_{\Omega_3} + \langle f, E_3\mu_a \rangle_{\Omega_3} + \langle f, \gamma_2 \rangle_{\Omega_2} + \langle f, E_2\mu_a \rangle_{\Omega_2} \qquad \forall\, \mu_a \in \Lambda_a,$$

which gives from the definitions of γ_3 and γ_2,

$$\begin{aligned} a(u,v) &= \langle f, v_{|\Omega_3} \rangle_{\Omega_3} + \langle f, v_{|\Omega_2} \rangle_{\Omega_2}, \\ &= \langle f, v \rangle_{\Omega} \qquad \forall\, v \in V, \end{aligned}$$

and hence the theorem follows. □

Remark 4.1. The variational formulation given by Equations $(4.6)_{1-4}$ provides a general setting for an overlapping domain decomposition method. On the one hand, we have a Dirichlet condition on Γ_b; on the other hand, the transmission condition $(4.6)_4$ on Γ_a depends on the bilinear form chosen to represent the original differential operator in the weak formulation. For the particular case of the ADR problem, this condition can be written in the more familiar form presented next.

4.2.3 Alternative formulation

We develop an alternative formulation for the domain decomposition method given by Equations $(4.6)_{1-4}$.

Lemma 4.1. *The solution of the domain decomposition problem satisfies*

$$\varepsilon \frac{\partial u_1}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_1 = \varepsilon \frac{\partial u_2}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_2 \qquad \text{on } \Gamma_a, \text{ in the sense of } \Lambda_a,$$

where $\partial(\cdot)/\partial n_2 = \boldsymbol{n}_2 \cdot \nabla(\cdot)$, $\boldsymbol{n}_2$ *being the outward unit vector normal to* Ω_2 *on* Γ_a.

Proof. Note first that according to Green's formula, we have for all $\mu_a \in \Lambda_a$

$$a_3(u_1, E_3\mu_a) = -\langle \varepsilon \frac{\partial u_1}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_1, \mu_a \rangle_{\Gamma_a} + \langle L u_1, E_3\mu_a \rangle_{\Omega_3}, \tag{4.11}$$

$$a_2(u_2, E_2\mu_a) = \langle \varepsilon \frac{\partial u_2}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_2, \mu_a \rangle_{\Gamma_a} + \langle L u_2, E_2\mu_a \rangle_{\Omega_2}. \tag{4.12}$$

In addition, from Equations $(4.6)_1$ and $(4.6)_3$, we have

$$L u_1 = f \qquad \text{in } \Omega_1, \qquad \text{and} \tag{4.13}$$

$$L u_2 = f \qquad \text{in } \Omega_2, \tag{4.14}$$

in the sense of distributions. As a result, Equations (4.11) and (4.12) become

$$\begin{aligned} a_3(u_1, E_3\mu_a) &= -\langle \varepsilon \frac{\partial u_1}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1, \mu_a\rangle_{\Gamma_a} + \langle f, E_3\mu_a\rangle_{\Omega_3}, \\ a_2(u_2, E_2\mu_a) &= \langle \varepsilon \frac{\partial u_2}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_2, \mu_a\rangle_{\Gamma_a} + \langle f, E_2\mu_a\rangle_{\Omega_2}. \end{aligned} \tag{4.15}$$

Adding up these two equations, and substituting the result into Equation $(4.6)_4$, we find

$$\langle -\varepsilon \frac{\partial u_1}{\partial n_2} + \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1 + \varepsilon \frac{\partial u_2}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_2, \mu_a\rangle_{\Gamma_a} = 0, \qquad \forall\, \mu_a \in \Lambda_a,$$

and thus the lemma holds. □

Theorem 4.2. *System of equations* $(4.6)_{1-4}$ *can be reformulated as follows: find* $u_1 \in V_1$ *and* $u_2 \in V_2$ *such that*

$$\begin{cases} a_1(u_1, v_1) = \langle f, v_1\rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1 = u_2 & \text{on } \Gamma_b, \\ a_2(u_2, v_2') = \langle f, v_2'\rangle_{\Omega_2} + \langle \varepsilon \dfrac{\partial u_1}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1, v_2'\rangle_{\Gamma_a} & \forall\, v_2' \in V_2. \end{cases} \tag{4.16}$$

Proof. We first substitute Equation (4.15) into Equation $(4.6)_4$, and add the result to Equation $(4.6)_3$:

$$a_2(u_2, v_2 + E_2\mu_a) = \langle \varepsilon \frac{\partial u_1}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1, \mu_a\rangle_{\Gamma_a} + \langle f, v_2 + E_2\mu_a\rangle_{\Omega_2} \qquad \forall\, v_2 \in V_2^0, \mu_a \in \Lambda_a.$$

Let us define $v_2' = v_2 + E_2\mu_a$. Clearly, $v_2' \in V_2$ and $\mu_a = T_a v_2'$; consequently, the last equation is equivalent to

$$a_2(u_2, v_2') = \langle \varepsilon \frac{\partial u_1}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1, v_2'\rangle_{\Gamma_a} + \langle f, v_2'\rangle_{\Omega_2} \qquad \forall\, v_2' \in V_2,$$

The proof is completed by substituting Equation $(4.6)_3$ and $(4.6)_4$ of the system of equations $(4.6)_{1-4}$ by the last equation. □

The interpretation of the domain decomposition method now appears clearly. On the one hand, a Dirichlet problem is solved in Ω_1 using as Dirichlet data on the interface Γ_b the solution in Ω_2. On the other hand, a mixed Dirichlet/Robin problem is solved in Ω_2 using as Robin data on Γ_a the solution in Ω_1. This formulation justifies the choice of the name overlapping Dirichlet/Robin (O-D/R) method to designate the DD method.

Remark 4.2. The system of equations $(4.16)_{1-3}$ could have been derived directly from the following DD algorithm applied at the differential level:

$$\begin{cases} Lu_1 = f & \text{in } \Omega_1, \\ u_1 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ u_1 = u_2 & \text{on } \Gamma_b, \\ Lu_2 = f & \text{in } \Omega_2, \\ u_2 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ \varepsilon\dfrac{\partial u_2}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_2 = \varepsilon\dfrac{\partial u_1}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1 & \text{on } \Gamma_a. \end{cases} \tag{4.17}$$

The interface conditions on Γ_a and Γ_b are usually referred to as *matching conditions* or *transmission conditions*. The first one is of Dirichlet type while the second one is of Robin type. At the variational level, we have just shown they correspond to essential and natural boundary conditions when choosing the bilinear form $a = a^{1/2}$.

Remark 4.3. Although formulation $(4.16)_{1-3}$ was derived from formulation $(4.6)_{1-4}$, it should be pointed out that the original formulation does not explicitly involve the normal derivatives on the interfaces Γ_a, whose definition can be ambiguous, e.g. at corners. In addition, we will see that the variational formulation can be useful to derive a corresponding algebraic approach.

4.2.4 Interface equations

A convenient way to study domain decomposition methods is to derive equations for the interface unknown(s). To do so, the domain decomposition problem is first rewritten into two purely Dirichlet problems for which the Dirichlet data are the unknowns on the interfaces. The development of the interface equations is first achieved at the differential level, starting form Equations $(4.17)_{1-6}$. Then, the variational equivalent is derived directly from Equations $(4.6)_{1-4}$. Finally, we enunciate the properties of the operators involved in the interface equations and state the existence and unicity of the solutions.

Differential interface equations

We propose to solve the following two problems:

$$\begin{cases} Lw_1 = f & \text{in } \Omega_1, \\ w_1 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ w_1 = \lambda_b & \text{on } \Gamma_b, \end{cases} \tag{4.18}$$

$$\begin{cases} Lw_2 = f & \text{in } \Omega_2, \\ w_2 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ w_2 = \lambda_a & \text{on } \Gamma_a. \end{cases} \tag{4.19}$$

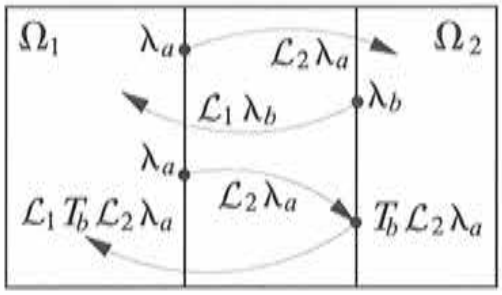


Figure 4.2: Extension operators.

Now let us decompose w_1 and w_2 into L-homogeneous and Dirichlet-homogeneous parts:

$$w_1 = u_1^0 + u_1^*, \tag{4.20}$$

$$w_2 = u_2^0 + u_2^*, \tag{4.21}$$

where the L-homogeneous parts u_1^0 and u_2^0 are the solutions of the following systems

$$\begin{cases} Lu_1^0 = 0 & \text{in } \Omega_1, \\ u_1^0 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ u_1^0 = \lambda_b & \text{on } \Gamma_b, \end{cases} \tag{4.22}$$

$$\begin{cases} Lu_2^0 = 0 & \text{in } \Omega_2, \\ u_2^0 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ u_2^0 = \lambda_a & \text{on } \Gamma_a, \end{cases} \tag{4.23}$$

and the Dirichlet-homogeneous parts u_1^* and u_2^* are the solutions of the following systems

$$\begin{cases} Lu_1^* = f & \text{in } \Omega_1, \\ u_1^* = 0 & \text{on } \partial\Omega_1, \end{cases} \tag{4.24}$$

$$\begin{cases} Lu_2^* = f & \text{in } \Omega_2, \\ u_2^* = 0 & \text{on } \partial\Omega_2. \end{cases} \tag{4.25}$$

u_1^0 is referred to as the L-homogeneous extension of λ_b into Ω_1, and is denoted $\mathcal{L}_1\lambda_b$. Similarly, u_2^0 is called the L-homogeneous extension of λ_a into Ω_2, and is denoted $\mathcal{L}_2\lambda_a$; see Figure (4.2). In the case when $L = -\Delta$, $\mathcal{L}$ is called the harmonic extension and is usually denoted H. The Dirichlet-homogeneous parts u_1^* and u_2^* are rewritten as $\mathcal{G}_1 f$ and $\mathcal{G}_2 f$, respectively. Comparing systems of equations (4.18) and (4.19) with system (4.17), we have $w_i = u_i$ for $i = 1, 2$ if and only if the following two conditions are satisfied:

$$\begin{cases} \varepsilon\dfrac{\partial w_2}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)w_2 = \varepsilon\dfrac{\partial w_1}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)w_1 \qquad \text{on } \Gamma_a, \\ \\ w_1 = w_2 \qquad \text{on } \Gamma_b. \end{cases} \tag{4.26}$$

Using the previous definitions of the L-homogeneous extension operators and those of the L-homogeneous components of the w_i's, and noting that $T_b\mathcal{G}_1 f = 0$, the system of equations (4.26)

can be rewritten as

$$\begin{cases} \varepsilon\dfrac{\partial\mathcal{L}_2\lambda_a}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_2\lambda_a = \varepsilon\dfrac{\partial\mathcal{L}_1\lambda_b}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_1\lambda_b \\ \qquad\qquad +\varepsilon\dfrac{\partial\mathcal{G}_1 f}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_1 f - \varepsilon\dfrac{\partial\mathcal{G}_2 f}{\partial n_2} + \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_2 f \qquad \text{on } \Gamma_a, \\ \\ \lambda_b = T_b\mathcal{L}_2\lambda_a + T_b\mathcal{G}_2 f \qquad \text{on } \Gamma_b. \end{cases}$$

Let us clean up the last system by introducing some definitions. In the first equation, we recognize the Steklov-Poincaré operator S_2 associated to subdomain Ω_2, and defined as

$$\begin{aligned} S_2 &: \Lambda_a \longrightarrow H^{-1/2}(\Gamma_a), \\ S_2\lambda_a &:= \varepsilon\frac{\partial\mathcal{L}_2\lambda_a}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_2\lambda_a. \end{aligned}$$

We define $\tilde{S}_b$, a Steklov-Poincaré-like operator acting on Γ_b as

$$\begin{aligned} \tilde{S}_b &: \Lambda_b \longrightarrow H^{-1/2}(\Gamma_a), \\ \tilde{S}_b\lambda_b &:= -\varepsilon\frac{\partial\mathcal{L}_1\lambda_b}{\partial n_2} + \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_1\lambda_b. \end{aligned}$$

We also define $\tilde{T}_b$, the trace on Γ_b of the L-extension of λ_a into Ω_2 by

$$\begin{aligned} \tilde{T}_b &: \Lambda_a \longrightarrow \Lambda_b, \\ \tilde{T}_b\lambda_a &:= T_b\mathcal{L}_2\lambda_a. \end{aligned}$$

χ and χ' are defined as follows

$$\begin{aligned} \chi &= \varepsilon\frac{\partial\mathcal{G}_1 f}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_1 f - \varepsilon\frac{\partial\mathcal{G}_2 f}{\partial n_2} + \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_2 f, \\ \chi' &= T_b\mathcal{G}_2 f, \end{aligned}$$

where we have $\chi \in H^{-1/2}(\Gamma_a)$ and $\chi' \in \Lambda_b$. Owing to the previous definitions, the system of two equations for the interface unknowns reads

$$\begin{cases} S_2\lambda_a &= -\tilde{S}_b\lambda_b + \chi \qquad \text{in } H^{-1/2}(\Gamma_a), \\ \lambda_b &= \tilde{T}_b\lambda_a + \chi' \qquad \text{in } \Lambda_b. \end{cases} \tag{4.27}$$

Let us introduce the following operator

$$\begin{aligned} \tilde{S}_1 &: \Lambda_a \longrightarrow H^{-1/2}(\Gamma_a), \\ \tilde{S}_1\lambda_a &:= \tilde{S}_b\tilde{T}_b\lambda_a, \end{aligned}$$

and define S as

$$S = \tilde{S}_1 + S_2,$$

After substituting λ_b given by Equation $(4.27)_2$ into Equation $(4.27)_1$, we finally obtain the following system of equations for the interface unknowns

$$\left\{ \begin{array}{rcll} S\lambda_a & = & \chi - \tilde{S}_b\chi' & \text{in } H^{-1/2}(\Gamma_a), \\ \lambda_b & = & \tilde{T}_b\lambda_a + \chi' & \text{in } \Lambda_b. \end{array} \right. \tag{4.28}$$

Once λ_a and λ_b are obtained, we can solve the two Dirichlet problems (4.22) and (4.23) to obtain the L-homogeneous parts u_1^0 and u_2^0. The Dirichlet-homogeneous parts u_1^* and u_2^* are obtained by solving Equations (4.24) and (4.25). Hence, the solutions u_1 and u_2 are calculated by adding up their respective L and Dirichlet homogeneous contributions.

Remark 4.4. Let us consider the limit of disjoint subdomains, and define λ the unique interface unknown, i.e. $\lambda_a = \lambda_b = \lambda$. The operator $\tilde{S}_1$ is precisely the Steklov-Poincaré operator given by

$$\tilde{S}_1\lambda = S_1\lambda = \varepsilon\frac{\partial}{\partial n_1}\mathcal{L}_1\lambda - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_1)\lambda, \tag{4.29}$$

where $\boldsymbol{n}_1 = -\boldsymbol{n}_2$ is the outward unit vector normal to Ω_1. We can easily check that Equations $(4.28)_1$ simply reduce to the following equation for λ:

$$(S_1 + S_2)\lambda = \chi.$$

As expected, this equation coincides with the Steklov-Poincaré equation for the interface unknown of disjoint subdomains using the classical Dirichlet/Robin method, or the classical Dirichlet/Neumann method if the equation is purely diffusive (see [58]).

Although Equation $(4.28)_1$ has been derived at the differential level, it should formally be understood in a weak sense. Let us find a variational interpretation to the operators in play. From Equation $(4.28)_1$, we have

$$\langle (S_2 + \tilde{S}_1)\lambda_a, \mu_a\rangle_{\Gamma_a} = \langle \chi - \tilde{S}_b\chi', \mu_a\rangle_{\Gamma_a} \qquad \forall\, \mu_a \in \Lambda_a. \tag{4.30}$$

Lemma 4.2. *The variational counterpart of the Steklov-Poincaré operators are*

$$\langle \tilde{S}_1\lambda_a, \mu_a\rangle_{\Gamma_a} = a_3(\mathcal{L}_1\tilde{T}_b\lambda_a, E_3\mu_a) \qquad \forall\, \mu_a \in \Lambda_a, \tag{4.31}$$

$$\langle S_2\lambda_a, \mu_a\rangle_{\Gamma_a} = a_2(\mathcal{L}_2\lambda_a, E_2\mu_a) \qquad \forall\, \mu_a \in \Lambda_a, \tag{4.32}$$

for any extension operators E_2 and E_3.

Proof. According to the definition of $\check{S}_1$, and using Green's formula, we have

$$\begin{aligned}
\langle \check{S}_1\lambda_a, \mu_a\rangle_{\Gamma_a} &= \int_{\Gamma_a}\left(-\varepsilon\frac{\partial \mathcal{L}_1\check{T}_b\lambda_a}{\partial n_2}+\frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_1\check{T}_b\lambda_a\right)\mu_a\,d\Gamma\\
&= \int_{\Gamma_a}\left(-\varepsilon\nabla(\mathcal{L}_1\check{T}_b\lambda_a)+\frac{1}{2}(\mathcal{L}_1\check{T}_b\lambda_a)\boldsymbol{a}\right)\mu_a\cdot\boldsymbol{n}_2\,d\Gamma\\
&= \int_{\Omega_3}\nabla\cdot\left[\left(-\varepsilon\nabla(\mathcal{L}_1\check{T}_b\lambda_a)+\frac{1}{2}(\mathcal{L}_1\check{T}_b\lambda_a)\boldsymbol{a}\right)\mu_a\right]d\Omega\\
&= \int_{\Omega_3}\varepsilon\nabla(\mathcal{L}_1\check{T}_b\lambda_a)\cdot\nabla(E_3\mu_a)\,d\Omega+\int_{\Omega_3}\varepsilon\Delta(\mathcal{L}_1\check{T}_b\lambda_a)E_3\mu_a\,d\Omega\\
&\quad -\frac{1}{2}\int_{\Omega_3}(E_3\mu_a)\,\boldsymbol{a}\cdot\nabla(\mathcal{L}_1\check{T}_b\lambda_a)\,d\Omega-\frac{1}{2}\int_{\Omega_3}(\mathcal{L}_1\check{T}_b\lambda_a)\,\boldsymbol{a}\cdot\nabla(E_3\mu_a)\,d\Omega\\
&\quad -\frac{1}{2}\int_{\Omega_3}(\mathcal{L}_1\check{T}_b\lambda_a)(E_3\mu_a)\nabla\cdot\boldsymbol{a}\,d\Omega \qquad \forall\,\mu_a\in\Lambda_a,
\end{aligned}\tag{4.33}$$

for any extension operators E_3. Knowing also that according to Equation (4.22) we have

$$\varepsilon\Delta(\mathcal{L}_1\check{T}_b\lambda_a)=\boldsymbol{a}\cdot\nabla(\mathcal{L}_1\check{T}_b\lambda_a)+(s+\nabla\cdot\boldsymbol{a})\mathcal{L}_1\check{T}_b\lambda_a,$$

in the sense of distributions, Equation (4.33) simplifies to

$$\langle \check{S}_1\lambda_a,\mu_a\rangle_{\Gamma_a}=a_3(\mathcal{L}_1\check{T}_b\lambda_a,E_3\mu_a)\qquad\forall\,\mu_a\in\Lambda_a.$$

The first argument of the last bilinear form is represented in Figure (4.2).

Let us now consider S_2. According to its definition and using Green's formula, we have

$$\begin{aligned}
\langle S_2\lambda_a,\mu_a\rangle_{\Gamma_a} &= \int_{\Gamma_a}\left(\varepsilon\frac{\partial\mathcal{L}_2\lambda_a}{\partial n_2}-\frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_2\lambda_a\right)\mu_a\,d\Gamma\\
&= \int_{\Omega_2}\varepsilon\nabla(\mathcal{L}_2\lambda_a)\cdot\nabla(E_2\mu_a)\,d\Omega+\int_{\Omega_2}\varepsilon\Delta(\mathcal{L}_2\lambda_a)E_2\mu_a\,d\Omega\\
&\quad -\frac{1}{2}\int_{\Omega_2}(E_2\mu_a)\,\boldsymbol{a}\cdot\nabla(\mathcal{L}_2\lambda_a)\,d\Omega-\frac{1}{2}\int_{\Omega_2}(\mathcal{L}_2\lambda_a)\,\boldsymbol{a}\cdot\nabla(E_2\mu_a)\,d\Omega\\
&\quad -\frac{1}{2}\int_{\Omega_2}(\mathcal{L}_2\lambda_a)(E_2\mu_a)\nabla\cdot\boldsymbol{a}\,d\Omega \qquad \forall\,\mu_a\in\Lambda_a,
\end{aligned}\tag{4.34}$$

for any extension operators E_2. Knowing also that according to Equation (4.23) we have

$$\varepsilon\Delta(\mathcal{L}_2\lambda_a)=\boldsymbol{a}\cdot\nabla(\mathcal{L}_2\lambda_a)+(s+\nabla\cdot\boldsymbol{a})\mathcal{L}_2\lambda_a,$$

in the sense of distributions, Equation (4.34) simplifies to

$$\langle S_2\lambda_a,\mu_a\rangle_{\Gamma_a}=a_2(\mathcal{L}_2\lambda_a,E_2\mu_a)\qquad\forall\,\mu_a\in\Lambda_a,$$

which completes the proof of the lemma. □

As for the right hand-side of Equation $(4.28)_1$, we can show that

$$\begin{aligned}\langle \chi, \mu_a\rangle_{\Gamma_a} &= \langle f, E_2\mu_a\rangle_{\Omega_2} - a_2(\mathcal{G}_2 f, E_2\mu_a) + \langle f, E_3\mu_a\rangle_{\Omega_3} - a_3(\mathcal{G}_1 f, E_3\mu_a), \qquad \forall\, \mu_a \in \Lambda_a\\ \langle \tilde{S}_b\chi', \mu_a\rangle_{\Gamma_a} &= a_3(\mathcal{L}_1 T_b \mathcal{G}_2 f, E_3\mu_a) \qquad \forall\, \mu_a \in \Lambda_a,\end{aligned}$$

for any extension operators E_2 and E_3. This completes the definition of the variational form of Equation $(4.28)_1$:

$$\begin{aligned}a_2(\mathcal{L}_2\lambda_a, E_2\mu_a) + a_3(\mathcal{L}_1\tilde{T}_b\lambda_a, E_3\mu_a) &= \langle f, E_2\mu_a\rangle_{\Omega_2} - a_2(\mathcal{G}_2 f, E_2\mu_a)\\ &\quad + \langle f, E_3\mu_a\rangle_{\Omega_3} - a_3(\mathcal{G}_1 f, E_3\mu_a) - a_3(\mathcal{L}_1 T_b \mathcal{G}_2 f, E_3\mu_a) \qquad \forall\, \mu_a \in \Lambda_a.\end{aligned}$$

Variational interface equations

Equation (4.30) can also be obtained by formulating Problems (4.18) and (4.19) in a variational form as follows: find $w_1 \in V_1$ and $w_2 \in V_2$ such that

$$\begin{cases} a_1(w_1, v_1) = f & \forall\, v_1 \in V_1^0\\ w_1 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega,\\ w_1 = \lambda_b & \text{on } \Gamma_b, \end{cases} \tag{4.35}$$

$$\begin{cases} a_2(w_2, v_2) = f & \forall\, v_2 \in V_2^0\\ w_2 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega,\\ w_2 = \lambda_a & \text{on } \Gamma_a, \end{cases} \tag{4.36}$$

As given by Equations (4.20) and (4.21), we decompose w_1 and w_2 into L-homogeneous and Dirichlet-homogeneous parts. The L-homogeneous parts are solutions of the following equations

$$\begin{cases} a_1(u_1^0, v_1) = 0 & \forall\, v_1 \in V_1^0\\ u_1^0 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega,\\ u_1^0 = \lambda_b & \text{on } \Gamma_b, \end{cases}$$

$$\begin{cases} a_2(u_2^0, v_2) = 0 & \forall\, v_2 \in V_2^0\\ u_2^0 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega,\\ u_2^0 = \lambda_a & \text{on } \Gamma_a, \end{cases}$$

while the Dirichlet-homogeneous parts satisfy the following equations with homogeneous data

$$\begin{cases} a_1(u_1^*, v_1) = 0 & \forall\, v_1 \in V_1^0\\ u_1^* = 0 & \text{on } \partial\Omega_1, \end{cases}$$

$$\begin{cases} a_2(u_2^*, v_2) = 0 & \forall\, v_2 \in V_2^0\\ u_2^* = 0 & \text{on } \partial\Omega_2, \end{cases}$$

We now proceed similarly to what has been done at the differential level. Comparing Equations $(4.35)_{1-3}$ and $(4.36)_{1-3}$ with Equations $(4.6)_{1-4}$, we have $w_i = u_i$ for $i = 1, 2$ if and only if the following two conditions are satisfied:

$$\begin{cases} a_2(\mathcal{L}_2\lambda_a, E_2\mu_a) + a_3(\mathcal{L}_1\lambda_b, E_3\mu_a) = & \\ \qquad \langle f, E_2\mu_a\rangle_{\Omega_2} - a_2(\mathcal{G}_2 f, E_2\mu_a) & \\ \qquad + \langle f, E_3\mu_a\rangle_{\Omega_3} - a_3(\mathcal{G}_1 f, E_3\mu_a) & \forall\, \mu_a\ \Lambda_a,\\ \lambda_b = \tilde{T}_b\lambda_a + \chi' & \text{in } \Lambda_b. \end{cases} \tag{4.37}$$

By substituting Equation $(4.37)_2$ into $(4.37)_1$, we recover the weak form of Equation $(4.28)_1$ derived at the differential level.

Solution of the interface equations

Let us go back to system (4.28). We first state some useful properties of operators S_2 and $\tilde{S}_1$.

Lemma 4.3. *S_2 is both continuous and coercive and $\tilde{S}_1$ is continuous and non-negative.*

Proof. We have shown that Equations (4.32) and (4.31) hold for any extension operators E_2 and E_1. This leaves us the choice to find appropriate expressions for S_2 and $\tilde{S}_1$ to facilitate their analysis. A straightforward choice consists in taking $E_2 = \mathcal{L}_2$, and $E_1 = \mathcal{L}_1\tilde{T}_b$. Thus, we have

$$\begin{aligned}\langle S_2\lambda_a, \mu_a\rangle_{\Gamma_a} &= a_2(\mathcal{L}_2\lambda_a, \mathcal{L}_2\mu_a),\\ \langle \tilde{S}_1\lambda_a, \mu_a\rangle_{\Gamma_a} &= a_3(\mathcal{L}_1\tilde{T}_b\lambda_a, \mathcal{L}_1\tilde{T}_b\mu_a) \qquad \forall\, \mu_a \in \Lambda_a,\end{aligned}$$

We first show that S_2 is both continuous and coercive. Using the definition of a_2 given by Equation (4.2) and applying the Cauchy-Schwartz inequality, we obtain

$$\langle S_2\eta_a, \mu_a\rangle_{\Gamma_a} \leq \kappa_{\Omega_2}\|\mathcal{L}_2\eta_a\|_{1,\Omega_2}\|\mathcal{L}_2\mu_a\|_{1,\Omega_2} \qquad \forall\, \eta_a, \mu_a \in \Lambda_a, \tag{4.38}$$

where

$$\kappa_{\Omega_2} = \varepsilon + \|\boldsymbol{a}\|_{\infty,\Omega_2} + \|s_0\|_{\infty,\Omega_2}.$$

According to the a-priori estimate given by Equation (4.5), we have that

$$\|\mathcal{L}_2\mu_a\|_{1,\Omega_2} \leq C_2\|\mu_a\|_{1/2,\Gamma_a} \qquad \forall\, \mu_a \in \Lambda_a.$$

As a result, Equation (4.38) gives

$$\langle S_2\eta_a, \mu_a\rangle_{\Gamma_a} \leq M_{S_2}\|\eta_a\|_{1/2,\Gamma_a}\|\mu_a\|_{1/2,\Gamma_a} \qquad \forall\, \eta_a, \mu_a \in \Lambda_a, \tag{4.39}$$

which states that S_2 is continuous, with

$$M_{S_2} = \kappa_{\Omega_2}C_2^2$$

the continuity constant.

We now show the coercivity of S_2. Owing to the skew-symmetry of the convective term of a_2, for any $\mu_a \in \Lambda_a$ we have

$$\begin{aligned}\langle S_2\mu_a, \mu_a\rangle_{\Gamma_a} &= a_2(\mathcal{L}_2\mu_a, \mathcal{L}_2\mu_a)\\ &= \varepsilon\,\|\nabla\mathcal{L}_2\mu_a\|_{0,2}^2 + \int_{\Omega_2} s_0(\mathcal{L}_2\mu_a)^2\, d\Omega\\ &\geq \varepsilon\|\nabla\mathcal{L}_2\mu_a\|_{0,\Omega_2}^2 \quad (s_0 \geq 0 \text{ almost everywhere}).\end{aligned} \tag{4.40}$$

From the trace inequality (see Equation (4.4)), we know that there exists a constant $C_2^* > 0$ such that

$$\|\mathcal{L}_2\mu_{a|\partial\Omega_2}\|_{1/2,\partial\Omega_2} \leq C_2^*\|\mathcal{L}_2\mu_a\|_{1,\Omega_2} \quad \forall\, \mu_a \in \Lambda_a,$$

Using the Poincaré-Friedrichs inequality (4.3), Equation (4.40) yields

$$\langle S_2\mu_a, \mu_a\rangle \geq N_{S_2}\|\mu_a\|^2_{1/2,\Gamma_a} \quad \forall\, \mu_a \in \Lambda_a, \tag{4.41}$$

where

$$N_{S_2} := \frac{\varepsilon}{(C_{\Omega_2}+1)(C_2^*)^2}$$

is the coercivity constant.

Let us finally prove the continuity and non-negativeness of $\tilde{S}_1$. Applying the Cauchy-Schwarz inequality to Equation (4.31), we obtain

$$\begin{aligned}\langle \tilde{S}_1\eta_a, \mu_a\rangle_{\Gamma_a} &\leq \kappa_{\Omega_3}\|\mathcal{L}_1T_b\mathcal{L}_2\eta_a\|_{1,\Omega_3}\|\mathcal{L}_1T_b\mathcal{L}_2\mu_a\|_{1,\Omega_3}\\ &\leq \kappa_{\Omega_3}\|\mathcal{L}_1T_b\mathcal{L}_2\eta_a\|_{1,\Omega_1}\|\mathcal{L}_1T_b\mathcal{L}_2\mu_a\|_{1,\Omega_1} \quad (\Omega_3 \subset \Omega_1)\end{aligned}$$

for any $\eta_a, \mu_a \in \Lambda_a$ and where $\kappa_{\Omega_3} = \varepsilon + \|\boldsymbol{a}\|_{\infty,\Omega_3} + \|s_0\|_{\infty,\Omega_3}$. From the a-priori estimate given by Equation (4.5), we have that

$$\begin{aligned}\langle \tilde{S}_1\eta_a, \mu_a\rangle_{\Gamma_a} &\leq \kappa_{\Omega_3}C_1^2\|T_b\mathcal{L}_2\eta_a\|_{1/2,\partial\Omega_1}\|T_b\mathcal{L}_2\mu_a\|_{1/2,\partial\Omega_1}\\ &= \kappa_{\Omega_3}C_1^2\|T_b\mathcal{L}_2\eta_a\|_{1/2,\Gamma_b}\|T_b\mathcal{L}_2\mu_a\|_{1/2,\Gamma_b} \qquad (4.42)\\ &= \kappa_{\Omega_3}C_1^2\|\gamma_{0,5}\mathcal{L}_2\eta_a\|_{1/2,\partial\Omega_5}\|\gamma_{0,5}\mathcal{L}_2\mu_a\|_{1/2,\partial\Omega_5}\\ &\leq \kappa_{\Omega_3}C_1^2C_5^{*2}\|\mathcal{L}_2\eta_a\|_{1,\Omega_5}\|\mathcal{L}_2\mu_a\|_{1,\Omega_5} \quad \text{(trace inequality (4.4))}\\ &\leq \kappa_{\Omega_3}C_1^2C_5^{*2}\|\mathcal{L}_2\eta_a\|_{1,\Omega_2}\|\mathcal{L}_2\mu_a\|_{1,\Omega_2} \quad (\Omega_5 \subset \Omega_2)\\ &\leq \kappa_{\Omega_3}C_1^2C_5^{*2}C_2^2\|\eta_a\|_{1/2,\partial\Omega_2}\|\mu_a\|_{1/2,\partial\Omega_2} \quad \text{(a-priori estimate (4.5))}\\ &= M_{\tilde{S}_1}\|\eta_a\|_{1/2,\Gamma_a}\|\mu_a\|_{1/2,\Gamma_a}, \qquad (4.43)\end{aligned}$$

which proves the continuity of $\tilde{S}_1$. Finally, owing to the skew-symmetry of a_1, for any $\mu_a \in \Lambda_a$ we have

$$\begin{aligned}\langle \tilde{S}_1\mu_a, \mu_a\rangle_{\Gamma_a} &= a_1(\mathcal{L}_1\tilde{T}_b\mu_a, \mathcal{L}_1\tilde{T}_b\mu_a)\\ &= \varepsilon\|\nabla\mathcal{L}_1\tilde{T}_b\mu_a\|^2_{0,\Omega_3} + \int_{\Omega_3} s_0(\mathcal{L}_1\tilde{T}_b\mu_a)^2\, d\Omega\\ &\geq 0 \quad (s_0 \geq 0 \text{ almost everywhere}),\end{aligned}$$

and the lemma holds. □

The following result is a direct consequence of the previous properties:

Theorem 4.3. *System* (4.28) *has a unique solution* $\{\lambda_a, \lambda_b\}$.

Proof. We first prove that S is invertible, showing that it is both continuous and coercive. We have

$$\langle S\eta_a, \mu_a\rangle_{\Gamma_a} = \langle \tilde{S}_1\eta_a, \mu_a\rangle_{\Gamma_a} + \langle S_2\eta_a, \mu_a\rangle_{\Gamma_a} \qquad \forall\, \eta_a, \mu_a \in \Lambda_a.$$

Therefore, the continuity of S follows from that of S_2 and $\tilde{S}_1$, i.e.

$$\langle S\eta_a, \mu_a\rangle \leq M_S \|\eta_a\|_{1/2,\Gamma_a} \|\mu_a\|_{1/2,\Gamma_a} \qquad \forall\, \eta_a, \mu_a \in \Lambda_a,$$

with continuity constant M_S given by

$$M_S = M_{\tilde{S}_1} + M_{S_2},$$

and where M_{S_2} and $M_{\tilde{S}_1}$. We now show the coercivity of S without trying to obtain sharp estimates. We have already shown the coercivity of S_2 and the non-negativeness of $\tilde{S}_1$. Therefore,

$$\begin{aligned} \langle S\mu_a, \mu_a\rangle_{\Gamma_a} &= \langle S_2\mu_a, \mu_a\rangle_{\Gamma_a} + \langle \tilde{S}_1\mu_a, \mu_a\rangle_{\Gamma_a} \\ &\geq \langle S_2\mu_a, \mu_a\rangle_{\Gamma_a} \\ &\geq N_S \|\mu_a\|^2_{1/2,\Gamma_a} \qquad \forall\, \mu_a \in \Lambda_a, \end{aligned}$$

where N_S is given by

$$N_S = N_{S_2} = \frac{\varepsilon}{C_2^{*2}(C_{\Omega_2} + 1)}. \tag{4.44}$$

Thus S is a continuous and coercive operator. According to Lax-Milgram Lemma, it is therefore invertible and Equation $(4.28)_1$ has a unique solution λ_a. The existence and uniqueness of λ_b follows from that of λ_a, by applying Equation $(4.28)_2$. Remember that we have

$$\lambda_b = \tilde{T}_b \mathcal{L}_2 \lambda_a + \chi'.$$

$\mathcal{L}_2\lambda_a$ is the unique solution of Problem (4.23). Since the trace operator T_b is well defined, from $H^1(\Omega_2)$ onto Λ_b, we know that λ_b exists and is unique. Inverting S in Equation $(4.28)_1$, we find that

$$\begin{cases} \lambda_a &= S^{-1}(\chi - \tilde{S}_b\chi') & \text{in } \Lambda_a, \\ \lambda_b &= \tilde{T}_b S^{-1}(\chi - \tilde{S}_b\chi') + \chi' & \text{in } \Lambda_b, \end{cases}$$

are the solutions of our interface problem. □

4.3 Iterative scheme

4.3.1 Relaxed sequential algorithm

In this section, we derive an iterative procedure to solve the domain decomposition problem (4.6). The sequential version of the iterative overlapping D/R algorithm is defined as follows. Given an

initial guess u_2^0 on Γ_b, for each $k \geq 0$, find $u_1^{k+1} \in V_1$ and $u_2^{k+1} \in V_2$ such that

$$\begin{cases} a_1(u_1^{k+1}, v_1) = \langle f, v_1 \rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1^{k+1} = u_2^k & \text{on } \Gamma_b, \\ a_2(u_2^{k+1}, v_2) = \langle f, v_2 \rangle_{\Omega_2} & \forall\, v_2 \in V_2^0, \\ a_2(u_2^{k+1}, E_2\mu_a) = -a_3(u_1^{k+1}, E_3\mu_a) \\ \qquad + \langle f, E_3\mu_a \rangle_{\Omega_3} + \langle f, E_2\mu_a \rangle_{\Omega_2} & \forall\, \mu_a \in \Lambda_a. \end{cases} \tag{4.45}$$

for any extension operators E_3 and E_2. If this algorithm converges, the solutions on both subdomains satisfy Equations $(4.6)_{1-4}$. The corresponding algorithm for the differential problem reads: given an initial guess u_2^0 on Γ_b, for each $k \geq 0$, find u_1^{k+1} and u_2^{k+1} such that

$$\begin{cases} Lu_1^{k+1} = f & \text{in } \Omega_1, \\ u_1^{k+1} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_b, \\ u_1^{k+1} = u_2^k & \text{on } \Gamma_b, \\ Lu_2^{k+1} = f & \text{in } \Omega_2, \\ u_2^{k+1} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_a, \\ \varepsilon \dfrac{\partial u_2^{k+1}}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_2^{k+1} = \varepsilon \dfrac{\partial u_1^{k+1}}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_1^{k+1} & \text{on } \Gamma_a. \end{cases} \tag{4.46}$$

If this algorithm converges, the solutions on both subdomains satisfy Equations $(4.17)_{1-6}$. For the sake of clarity, we have omitted the relaxation of the transmission conditions; for example, the Dirichlet condition $(4.46)_3$ could be replaced by

$$u_1^{k+1} = \theta u_2^k + (1 - \theta) u_1^k,$$

where $\theta > 0$ is the relaxation parameter.

We now investigate the interface iterates produced by this relaxed iterative procedure. The set of equations for the w_i's is the following:

$$\begin{cases} Lw_1^{k+1} = f & \text{in } \Omega_1, \\ w_1^{k+1} = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ w_1^{k+1} = \lambda_b^k & \text{on } \Gamma_b, \end{cases}$$

$$\begin{cases} Lw_2^{k+1} = f & \text{in } \Omega_2, \\ w_2^{k+1} = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ w_2^{k+1} = \lambda_a^{k+1} & \text{on } \Gamma_a. \end{cases}$$

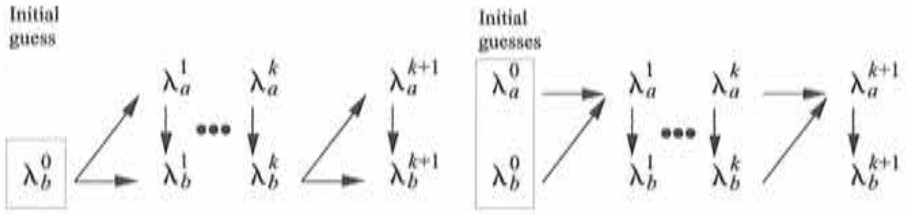


Figure 4.3: Relaxed sequential algorithms for the interface unknowns. (Left) $\mathrm{D}_\theta/\mathrm{R}$. (Right) $\mathrm{D}/\mathrm{R}_\theta$.

The choice for taking as Dirichlet conditions λ_b at iteration k for w_1^{k+1}, and λ_a at iteration $k+1$ for w_2^{k+1} is arbitrary. According to this choice, we have

$$\begin{aligned} w_1^{k+1} &= \mathcal{L}_1 \lambda_b^k + \mathcal{G}_1 f, \\ w_2^{k+1} &= \mathcal{L}_2 \lambda_a^{k+1} + \mathcal{G}_2 f. \end{aligned}$$

We have $w_i^k = u_i^k$ for $i = 1, 2$ if and only if the w_i^k's satisfy the transmission conditions $(4.46)_3$ and $(4.46)_6$. By noting that the Dirichlet-homogeneous solutions $\mathcal{G}_1 f$ and $\mathcal{G}_1 f$ do not change along the iterative process, the Dirichlet-relaxed iterative scheme, denoted $\mathrm{D}_\theta/\mathrm{R}$, gives for any $k \geq 0$

$$\left\{ \begin{array}{l} S_2 \lambda_a^{k+1} = -\bar{S}_b \lambda_b^k + \chi, \\ \lambda_b^{k+1} = \theta(\bar{T}_b \lambda_a^{k+1} + \chi') + (1-\theta)\lambda_b^k. \end{array} \right. \tag{4.47}$$

The Robin transmission condition can be relaxed as well, by replacing Equation $(4.46)_6$ by

$$\begin{aligned} \varepsilon \frac{\partial u_2^{k+1}}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_2^{k+1} = \theta \left(\varepsilon \frac{\partial u_1^{k+1}}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_1^{k+1} \right) \\ + (1-\theta) \left(\varepsilon \frac{\partial u_2^k}{\partial n_2} - \frac{1}{2}(\boldsymbol{a} \cdot \boldsymbol{n}_2) u_2^k \right). \end{aligned}$$

In terms of the interface unknowns, the Robin-relaxed iterative scheme, denoted $\mathrm{D}/\mathrm{R}_\theta$, produces the following iterates for any $k \geq 0$

$$\left\{ \begin{array}{rcl} S_2 \lambda_a^{k+1} &=& \theta(-\bar{S}_b \lambda_b^k + \chi) + (1-\theta) S_2 \lambda_a^k, \\ \lambda_b^{k+1} &=& \bar{T}_b \lambda_a^{k+1} + \chi'. \end{array} \right. \tag{4.48}$$

The dependence of λ_a^{k+1} and λ_b^{k+1} on the values at previous iterations is sketched in Figure 4.3, given two initial values λ_a^0 and λ_b^0; note that the value of λ_a^0 is only needed when using the $\mathrm{D}/\mathrm{R}_\theta$ method.

The continuity and coercivity of S_2 has been proven in last section. According to Lax-Milgram Lemma, S_2 is therefore invertible. We can therefore reformulate the system for the interface unknowns (4.27) as follows:

$$\left\{ \begin{array}{l} Q_a \lambda_a = \chi_a, \\ Q_b \lambda_b = \chi_b, \end{array} \right. \tag{4.49}$$

where we have defined Q_a, Q_b, χ_a and χ_b as

$$\begin{aligned} Q_a &= I_a + S_2^{-1}\tilde{S}_b\tilde{T}_b \ (= I_a + S_2^{-1}\tilde{S}_1), \\ Q_b &= I_b + \tilde{T}_b S_2^{-1}\tilde{S}_b, \\ \chi_a &= S_2^{-1}\chi - S_2^{-1}\tilde{S}_b\chi', \\ \chi_b &= \tilde{T}_b S_2^{-1}\chi + \chi'. \end{aligned}$$

and where I_a is the identity on Λ_a and I_b is the identity on Λ_b.

By solving the Dirichlet-relaxed and Robin-relaxed systems for λ_a^{k+1} and λ_b^{k+1}, we can show that both schemes lead to the same following iterates for any $k \geq 1$:

$$\left\{ \begin{array}{rcl} \lambda_a^{k+1} &=& \theta(\chi_a - Q_a\lambda_a^k) + \lambda_a^k, \\ \lambda_b^{k+1} &=& \theta(\chi_b - Q_b\lambda_b^k) + \lambda_b^k. \end{array} \right. \tag{4.50}$$

We recognize here two *stationary Richardson procedures* for solving Equations $(4.49)_1$ and $(4.49)_2$. The Richardson procedure for solving λ_a is similar to that produced by the classical Dirichlet/Neumann method; in fact, by multiplying Equation $(4.50)_1$ by S_2 we obtain the following equivalent iterate

$$\lambda_a^{k+1} = \theta S_2^{-1}[(\chi - \tilde{S}_b\chi') - S\lambda_a^k] + \lambda_a^k,$$

which is a preconditioned Richardson method for solving Equation $(4.28)_1$, using S_2 as preconditioner for S.

Remark 4.5. As pointed out above, the Richardson procedures (4.50) are valid only for $k \geq 1$. The D$_\theta$/R and D/R$_\theta$ are therefore not completely equivalent, as the first iterative values λ_a^1 and λ_b^1 may differ, although λ_a^0 and λ_b^0 are chosen to be equal.

4.3.2 Convergence

This section studies the convergence of the D$_\theta$/R and D/R$_\theta$ iterative schemes given by Equations $(4.46)_{1-6}$ at the differential level, or $(4.45)_{1-4}$ at the variational level. Rather than directly studying the whole system of equations for u_1 and u_2, we base our analysis on the interface equation systems, i.e. Equations $(4.47)_{1-2}$ for the D$_\theta$/R method and Equations $(4.48)_{1-2}$ for the D/R$_\theta$ method. The result we can prove is

Theorem 4.4. *Assume that ε is large enough so that*

$$\kappa^* := 2N_{S_2} - 2\|\boldsymbol{a}\|_{\infty,\Gamma_a} C_2^2 \frac{M_{\tilde{S}_1} + M_{S_2}}{N_{S_2}} > 0, \tag{4.51}$$

where the constants N_{S_2}, $M_{\tilde{S}_1}$ and M_{S_2} have been introduced in Equations (4.41), (4.43) and (4.39), respectively. Then, there exists $\theta_{\max}$ such that for any given $\lambda_a^0 \in \Lambda_a$ and $\lambda_b^0 \in \Lambda_b$ and for all $\theta \in (0, \theta_{\max})$, the sequences $\{\lambda_a^k\}$ and $\{\lambda_b^k\}$ given by (4.50) converge in Λ_a and Λ_b, respectively. The upper bound of the relaxation parameter $\theta_{\max}$ can be estimated by

$$\theta_{\max} = \frac{\kappa^* N_{S_2}^2}{M_{S_2}(M_{\tilde{S}_1} + M_{S_2})^2}. \tag{4.52}$$

Proof. The proof is split into two steps. We first show the Richardson procedure for the sequence $\{\lambda_a^k\}$ given by Equation $(4.50)_1$ converges. The proof is based on the abstract Theorem 3.1. of [73], or Theorem 4.2.2. of [58]. Secondly, we show that if the sequence $\{\lambda_a^k\}$ converges, then $\{\lambda_b^k\}$ does as well.

Let us start with the first step and define R_a the Richardson iteration operator as

$$\begin{aligned} &R_a : \Lambda_a \longrightarrow H^{-1/2}(\Gamma_a), \\ &R_a\mu_a := (I_a - \theta Q_a)\mu_a = (I_a - \theta S_2^{-1}S)\mu_a. \end{aligned}$$

If we define $e_a^k = \lambda_a^k - \lambda_a$ as the error with respect to λ_a at iteration k, λ_a being solution of problem $(4.49)_1$, the error equation reads

$$e_a^{k+1} = R_a e_a^k.$$

The Richardson procedure $(4.50)_1$ is therefore convergent if the operator R_a is a contraction with respect to some norm. Let us introduce the following application:

$$\begin{aligned} &(\cdot,\cdot)_{S_2} : \Lambda_a \times \Lambda_a \longrightarrow \mathbb{R}, \\ &(\eta_a, \mu_a)_{S_2} := \frac{1}{2}(\langle S_2\eta_a, \mu_a\rangle_{\Gamma_a} + \langle S_2\mu_a, \eta_a\rangle_{\Gamma_a}). \end{aligned}$$

It is easy to check that this application is a scalar product, and that it induces the following S_2-norm

$$\|\mu_a\|_{S_2} := \langle S_2\mu_a, \mu_a\rangle_{\Gamma_a}^{1/2},$$

which, owing to both the coercivity and continuity of S_2, is equivalent to the natural norm on Λ_a, i.e.

$$N_{S_2}^{1/2}\|\mu_a\|_{1/2,\Gamma_a} \leq \|\mu_a\|_{S_2} \leq M_{S_2}^{1/2}\|\mu_a\|_{1/2,\Gamma_a} \quad \forall\, \mu_a \in \Lambda_a. \tag{4.53}$$

By definition we have

$$\begin{aligned} \|R_a\mu_a\|_{S_2}^2 = \|\mu_a\|_{S_2}^2 \; &+\theta^2\langle S\mu_a, S_2^{-1}S\mu_a\rangle_{\Gamma_a} \\ &-\theta\left(\langle S_2\mu_a, S_2^{-1}S\mu_a\rangle_{\Gamma_a} + \langle S\mu_a, \mu_a\rangle_{\Gamma_a}\right). \end{aligned} \tag{4.54}$$

Using the same strategy as in [58], it can be checked that

$$\langle S_2\mu_a, S_2^{-1}S\mu_a\rangle_{\Gamma_a} + \langle S\mu_a, \mu_a\rangle_{\Gamma_a} \geq \kappa^*\|\mu_a\|_{1/2,\Gamma_a}^2 \quad \forall\, \mu_a \in \Lambda_a, \tag{4.55}$$

with κ^* defined in Equation (4.51); this point will be studied just after this proof. Since the norm of S_2^{-1} is $1/N_{S_2}$, and owing to the continuity of S_2 and $\tilde{S}_1$ and to the assumption of the theorem, Equation (4.54) yields

$$\|R_a\mu_a\|_{S_2}^2 \leq K_\theta\|\mu_a\|_{S_2}^2,$$

with K_θ given by

$$K_\theta = 1 + \theta^2 \frac{(M_{S_2} + M_{S_1})^2}{N_{S_2}^2} - \theta \frac{\kappa^*}{M_{S_2}},$$

The Richardson procedure is a contraction in the S_2-norm if $K_\theta < 1$, i.e. if $0 < \theta < \theta_{\max}$, with $\theta_{\max}$ given by Equation (4.52).

Let us now go on to the second step of the proof, i.e. the convergence of the sequence $\{\lambda_a^k\}$ implies that of the sequence $\{\lambda_b^k\}$. Although the Dirichlet and Robin-relaxed methods lead to the same Richardson procedure for λ_b (Equation $(4.50)_2$) for $k \geq 1$, we have to treat their convergence separately. We define $e_b^k = \lambda_b^k - \lambda_b$. Since the converged solution satisfies $\lambda_b = \tilde{T}_b \lambda_a + \chi'$, Equation $(4.48)_2$ for the Robin-relaxed scheme gives for any $k \geq 1$,

$$e_b^k = \tilde{T}_b e_a^k.$$

Therefore, we have

$$\begin{aligned}
\|e_b^k\|_{1/2,\Gamma_b} &= \|T_b \mathcal{L}_2 e_a^k\|_{1/2,\Gamma_b} \\
&\leq C_2^* \, \|\mathcal{L}_2 e_a^k\|_{1,\Omega_2} && \text{(trace inequality (4.4))} \\
&\leq C_2^* \, C_2 \, \|e_a^k\|_{1/2,\Gamma_a} && \text{(a-priori estimate (4.5))} \\
&\leq \frac{C_2^* \, C_2}{N_{S_2}^{1/2}} \, \|e_a^k\|_{S_2} && \text{(norm equivalence (4.53))} \\
&\leq K_\theta^k \frac{C_2^* \, C_2}{N_{S_2}^{1/2}} \, \|e_a^0\|_{1/2,\Gamma_a},
\end{aligned}$$

which shows that the sequence $\{\lambda_b^k\}$ converges whenever $K_\theta < 1$.

Now we study the convergence of the Dirichlet-relaxed algorithm (for $\theta \neq 1$). From Equation $(4.47)_2$, we have that, for any $k \geq 1$,

$$e_b^k = \theta \tilde{T}_b e_a^k + (1-\theta) e_b^{k-1}.$$

According to this equation, we can generate the following sequence

$$\begin{aligned}
e_b^k &= \theta \, \tilde{T}_b e_a^k + (1-\theta) \, e_b^{k-1}, \\
(1-\theta) \, e_b^{k-1} &= \theta(1-\theta) \, \tilde{T}_b e_a^{k-1} + (1-\theta)^2 \, e_b^{k-2}, \\
(1-\theta)^2 \, e_b^{k-2} &= \theta(1-\theta)^2 \, \tilde{T}_b e_a^{k-2} + (1-\theta)^3 \, e_b^{k-3}, \\
&\vdots \qquad\qquad\qquad\qquad\qquad \vdots \\
(1-\theta)^{k-2} \, e_b^2 &= \theta(1-\theta)^{k-2} \, \tilde{T}_b e_a^2 + (1-\theta)^{k-1} \, e_b^1, \\
(1-\theta)^{k-1} \, e_b^1 &= \theta(1-\theta)^{k-1} \, \tilde{T}_b e_a^1 + (1-\theta)^k \, e_b^0.
\end{aligned}$$

Adding up all the terms, we find the following equality

$$e_b^k = (1-\theta)^k e_b^0 + \theta(1-\theta)^k \sum_{n=1}^{k} (1-\theta)^{-n} \tilde{T}_b e_a^k,$$

which gives

$$\begin{aligned}\|e_b^k\|_{1/2,\Gamma_b} &\leq |1-\theta|^k \|e_b^0\|_{1/2,\Gamma_b} + \theta|1-\theta|^k \sum_{n=1}^{k} |1-\theta|^{-n} \|T_b \mathcal{L}_2 e_a^n\|_{1/2,\Gamma_b} \\ &\leq |1-\theta|^k \|e_b^0\|_{1/2,\Gamma_b} + \frac{\theta}{K_\theta}|1-\theta|^{k-1} \frac{C_2^* C_2}{N_{S_2}^{1/2}} \|e_a^1\|_{S_2} \sum_{n=1}^{k} \left(\frac{K_\theta}{|1-\theta|}\right)^n.\end{aligned}$$

The geometric progression is

$$\sum_{n=1}^{k} \left(\frac{K_\theta}{|1-\theta|}\right)^n = \begin{cases} \dfrac{1}{2}k(k+1) & \text{if } K_\theta = |1-\theta|, \\ \dfrac{K_\theta}{|1-\theta|^k} \dfrac{|1-\theta|^k - K_\theta^k}{|1-\theta| - K_\theta} & \text{otherwise,} \end{cases}$$

and thus we find the following two expressions for the norm of the error

$$\|e_b^k\|_{1/2,\Gamma_b} \leq \begin{cases} |1-\theta|^k \left(\|e_b^0\|_{1/2,\Gamma_b} + \dfrac{1}{2}k(k+1)\theta \dfrac{C_2^* C_2}{N_{S_2}^{1/2}} \|e_a^0\|_{S_2}\right) & \\ \qquad\qquad \text{if } K_\theta = |1-\theta|, & \\ |1-\theta|^k \|e_b^0\|_{1/2,\Gamma_b} + \theta \dfrac{|1-\theta|^k - K_\theta^k}{|1-\theta| - K_\theta} \dfrac{C_2^* C_2}{N_{S_2}^{1/2}} \|e_a^0\|_{S_2} \quad \text{otherwise.} \end{cases}$$

Owing to these inequalities and since $\theta < 2$ (see Equations (4.51)-(4.52)), we conclude that if $K_\theta < 1$ the sequence $\{\lambda_b^k\}$ converges. □

Note that once $\lambda_a = \lim_{k\to\infty} \lambda_a^k$ and $\lambda_b = \lim_{k\to\infty} \lambda_b^k$ are found, the solutions in Ω_1 and Ω_2 are obtained by solving the two Dirichlet problems given by Equations (4.19). As a consequence, the convergences of sequences $\{\lambda_a^k\}$ and $\{\lambda_b^k\}$ imply the convergence of the whole algorithm.

We show how Equation (4.55) and Equation (4.51) for κ^* are equivalent. Once again, we follow the authors of [58]. Let us split operator a_2 into a symmetric part and a skew-symmetric part as follows:

$$a_2(w_2, v_2) = a_2^{\mathrm{s}}(w_2, v_2) + a_2^{\mathrm{ss}}(w_2, v_2), \qquad \forall\, u_2, w_2 \in V_2,$$

with

$$\begin{aligned} a_2^{\mathrm{s}}(w_2, v_2) &:= \varepsilon(\nabla w_2, \nabla v_2)_{\Omega_2} + (s_0 w_2, v_2)_{\Omega_2} \\ a_2^{\mathrm{ss}}(w_2, v_2), &:= \frac{1}{2}(\boldsymbol{a} \cdot \nabla w_2, v_2)_{\Omega_2} - \frac{1}{2}(w_2, \boldsymbol{a} \cdot \nabla v_2)_{\Omega_2}. \end{aligned}$$

Let $\eta_a = S_2^{-1} S\mu_a$. We have

$$\begin{aligned}
\langle S_2\mu_a, S_2^{-1}S\mu_a\rangle_{\Gamma_a} + \langle S\mu_a, \mu_a\rangle_{\Gamma_a} &= \langle S_2\mu_a, \eta_a\rangle_{\Gamma_a} - \langle S\mu_a, \mu_a\rangle_{\Gamma_a} + 2\langle S\mu_a, \mu_a\rangle_{\Gamma_a} \\
&= \langle S_2\mu_a, \eta_a\rangle_{\Gamma_a} - \langle S_2\eta_a, \mu_a\rangle_{\Gamma_a} + 2\langle S\mu_a, \mu_a\rangle_{\Gamma_a} \\
&= a_2^s(\mathcal{L}_2\mu_a, \mathcal{L}_2\eta_a) + a_2^{ss}(\mathcal{L}_2\mu_a, \mathcal{L}_2\eta_a) \\
&\quad -a_2^s(\mathcal{L}_2\eta_a, \mathcal{L}_2\mu_a) - a_2^{ss}(\mathcal{L}_2\eta_a, \mathcal{L}_2\mu_a) + 2\langle S\mu_a, \mu_a\rangle_{\Gamma_a} \\
&= 2a_2^{ss}(\mathcal{L}_2\mu_a, \mathcal{L}_2 S_2^{-1} S\mu_a) + 2\langle S\mu_a, \mu_a\rangle_{\Gamma_a}.
\end{aligned}$$

We have shown in Section 4.2.4 that S is coercive with coercivity constant N_S given by Equation (4.44). Hence, last equations yields

$$\langle S_2\mu_a, S_2^{-1}S\mu_a\rangle_{\Gamma_a} + \langle S\mu_a, \mu_a\rangle_{\Gamma_a} \geq 2N_S\|\mu_a\|^2_{1/2,\Gamma_a} - 2|a_2^{ss}(\mathcal{L}_2\mu_a, \mathcal{L}_2 S_2^{-1} S\mu_a)| \tag{4.56}$$

Let us try to bound the last term. From Cauchy-Schwartz inequality and the a-priori estimate (4.5), and using the continuity of S and S_2^{-1}, we obtain the following (coarse) estimate

$$\begin{aligned}
|a_2^{ss}(\mathcal{L}_2\mu_a, \mathcal{L}_2 S_2^{-1} S\mu_a)| &\leq \|\boldsymbol{a}\|_{\infty,\Gamma_a} \|\mathcal{L}_2 S_2^{-1} S\mu_a\|_{1,\Omega_2} \|\mathcal{L}_2\mu_a\|_{1,\Omega_2} \\
&\leq \|\boldsymbol{a}\|_{\infty,\Gamma_a} C_2^2 \|S_2^{-1} S\mu_a\|_{1/2,\partial\Omega_2} \|\mu_a\|_{1/2,\partial\Omega_2} \\
&\leq \|\boldsymbol{a}\|_{\infty,\Gamma_a} C_2^2 \|S_2^{-1}\| \, \|S\mu_a\|_{-1/2,\Gamma_a} \|\mu_a\|_{1/2,\Gamma_a} \\
&\leq \|\boldsymbol{a}\|_{\infty,\Gamma_a} C_2^2 \frac{M_S}{N_{S_2}} \|\mu_a\|^2_{1/2,\Gamma_a}.
\end{aligned}$$

Therefore, Equation (4.56) becomes:

$$\langle S_2\mu_a, S_2^{-1}S\mu_a\rangle_{\Gamma_a} + \langle S\mu_a, \mu_a\rangle_{\Gamma_a} \geq \kappa^* \|\mu_a\|^2_{1/2,\Gamma_a},$$

with

$$\kappa^* = 2N_S - 2\|\boldsymbol{a}\|_{\infty,\Gamma_a} C_2^2 \frac{M_S}{N_{S_2}}, \tag{4.57}$$

or equivalently κ^* given by Equation (4.51).

Remark 4.6. Let us examine closer the continuity of $\tilde{S}_1$. If the subdomains are disjoint, we have already shown that $\tilde{S}_1$ is precisely the Steklov-Poincaré operator acting on Γ_a associated to subdomain Ω_1, i.e. $\tilde{S}_1 = S_1$ and $\tilde{T}_b = T_b\mathcal{L}_2 = I_a$ (see Equation (4.29)). In the proof of the continuity assumption, Equation (4.42) is thus the bifurcation point between the overlapping and disjoint Dirichlet/Robin methods. So what does the overlapping method tell us more? $T_b\mathcal{L}_2\mu_a$ is the value on Γ_b of the extension of μ_a in Ω_2. If the norm appearing in Equation (4.42) were the infinite norm, i.e. $\|T_b\mathcal{L}_2\mu_a\|_{\infty,\Gamma_b}$, we could apply the maximum principle and state that

$$\|T_b\mathcal{L}_2\mu_a\|_{\infty,\Gamma_b} \leq k_1\|\mu_a\|_{\infty,\Gamma_a}$$

where $k_1 \in (0,1)$ (see [1]) (see the graphical representation of operator $T_b\mathcal{L}_2$ in Figure 4.2). Therefore, in addition to constant C_1 already present in the continuity constant M_1 of $\tilde{S}_1$ for the classical

D/R method, the present method would include the factor $k_1 < 1$. According to the well-known estimates for k_1 (see e.g. [1, 77]), we would expect k_1 to decrease with increasing overlapping length. In particular, the smaller the continuity constant M_1, the greater the maximum authorized relaxation parameter $\theta_{\max}$, as confirmed by Equation (4.52). Unfortunately, the infinite norm cannot be bounded by the $H^{1/2}$ norm in which we are measuring the continuity of $\tilde{S}_1$. The extrapolation of the previous comment to our situation is here only intuitive.

Remark 4.7. We have shown that the convergence of the D_θ/R and D/R_θ schemes was conditioned by the smallness assumption on the skew-symmetric part of S_2, given by Equation (4.57). When the subdomains are disjoint, there is a way to circumvent this unpleasant condition; the resulting method is called the γ-D/R method devised in [73].

4.4 Generalization to other mixed DD methods

When we showed the equivalence between the one-domain and two-domain variational formulations, i.e. the equivalence between Problems (4.6) and (4.1), we did not restrict ourselves to any particular bilinear form. We could have chosen identically any of the three bilinear forms derived in Chapter 1, namely a^0, $a^{1/2}$ and a^1. The important point in the choice of a bilinear form only appears when we write down the interface equation. As an example, let us consider the bilinear form $a = a^0$, which natural condition is a Neumann condition.

Proposition 4.1. *System of Equations* $(4.6)_{1-4}$ *can be reformulated as follows: find* $u_1 \in V_1$ *and* $u_2 \in V_2$ *such that*

$$\begin{cases} a_1(u_1, v_1) = \langle f, v_1 \rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1 = u_2 & \text{on } \Gamma_b, \\ a_2(u_2, v_2') = \langle f, v_2' \rangle_{\Omega_2} + \langle \varepsilon \dfrac{\partial u_1}{\partial n_2}, v_2' \rangle_{\Gamma_a} & \forall\, v_2' \in V_2. \end{cases}$$

Proof. The proposition is easily shown following the steps of the proof of the corresponding proposition for the Dirichlet/Robin method in Section 4.2.3. □

The DD method is therefore an overlapping Dirichlet/Neumann method.

Let us go back to the derivation of the interface equations, as done in Section 4.2.4 for the Dirichlet/Robin method. We redefine some of the operators in play in the interface equation. We first define S_2 as

$$S_2 : \Lambda_a \longrightarrow H^{-1/2}(\Gamma_a),$$
$$S_2 \lambda_a := \varepsilon \frac{\partial \mathcal{L}_2 \lambda_a}{\partial n_2},$$

and the operator $\tilde{S}_b$ as

$$\tilde{S}_b : \Lambda_b \longrightarrow H^{-1/2}(\Gamma_a),$$
$$\tilde{S}_b \lambda_b := -\varepsilon \frac{\partial \mathcal{L}_1 \lambda_b}{\partial n_2},$$

The right-hand side χ is

$$\chi = \varepsilon \frac{\partial \mathcal{G}_1 f}{\partial n_2} - \varepsilon \frac{\partial \mathcal{G}_2 f}{\partial n_2}.$$

Then, we can show that the system of two equations for the interface unknowns reads

$$\begin{cases} S_2 \lambda_a &= -\tilde{S}_b \lambda_b + \chi & \text{in } H^{-1/2}(\Gamma_a), \\ \lambda_b &= \tilde{T}_b \lambda_a + \chi' & \text{in } \Lambda_b. \end{cases} \tag{4.58}$$

We also introduce the operator $\tilde{S}_1$ defined as

$$\begin{aligned} &\tilde{S}_1 : \Lambda_a \longrightarrow H^{-1/2}(\Gamma_a), \\ &\tilde{S}_1 \lambda_a := \tilde{S}_b \tilde{T}_b \lambda_a, \end{aligned}$$

and define S as

$$S = \tilde{S}_1 + S_2.$$

After substituting λ_b given by Equation $(4.58)_2$ into Equation $(4.58)_1$, we finally obtain the following system of equations for the interface unknowns

$$\begin{cases} S \lambda_a &= \chi - \tilde{S}_b \chi' & \text{in } H^{-1/2}(\Gamma_a), \\ \lambda_b &= \tilde{T}_b \lambda_a + \chi' & \text{in } \Lambda_b. \end{cases}$$

We now look for a solution to the last system, i.e. we want to be able to invert S. Let us consider the properties of the operator S. As for the Dirichlet/Robin method, we can show that Lemma 4.2 holds. Moreover, we can easily show that both $\tilde{S}_1$ and S_2 are continuous, and so is S. We are now left with the coercivity of S. Choosing $E_2 = \mathcal{L}_2$, and $E_3 = \mathcal{L}_1 \tilde{T}_b$ in Equations (4.31) and (4.32), we have

$$\begin{aligned} \langle S\mu_a, \mu_a \rangle_{\Gamma_a} &= \langle \tilde{S}_1 \mu_a, \mu_a \rangle_{\Gamma_a} + \langle S_2 \mu_a, \mu_a \rangle_{\Gamma_a} \\ &= a_3(\mathcal{L}_1 \tilde{T}_b \mu_a, \mathcal{L}_1 \tilde{T}_b \mu_a) + a_2(\mathcal{L}_2 \mu_a, \mathcal{L}_2 \mu_a). \end{aligned}$$

Applying equation (1.14) for $\flat = 0$, the last equation gives

$$\begin{aligned} \langle S\mu_a, \mu_a \rangle_{\Gamma_a} \geq\ & \frac{\varepsilon}{1 + C_{\Omega_3}} \|\mathcal{L}_1 \tilde{T}_b \mu_a\|^2_{1,\Omega_3} - \frac{1}{2} \int_{\Gamma_a} (\boldsymbol{a} \cdot \boldsymbol{n}_2)(\mathcal{L}_1 \tilde{T}_b \mu_a)^2 \, d\Gamma \\ & + \frac{\varepsilon}{1 + C_{\Omega_2}} \|\mathcal{L}_2 \mu_a\|^2_{1,\Omega_2} + \frac{1}{2} \int_{\Gamma_a} (\boldsymbol{a} \cdot \boldsymbol{n}_2)(\mathcal{L}_2 \mu_a)^2 \, d\Gamma. \end{aligned} \tag{4.59}$$

Before going any further, let us recast last equation in the case of non-overlapping subdomains. The two contour integrals cancel each other as we have $\mathcal{L}_1 \tilde{T}_b \mu_a = \mathcal{L}_2 \mu_a = \mu_a$ on Γ_a. In addition, we can apply the Poincaré-Friedrichs inequality (4.3) to bound from below the first and third terms in terms of the trace norm. Therefore, S is coercive. In the overlapping case the contour integrals do not cancel each other, and some estimates are to be found to be able to proceed; for example, by assuming ε large enough, it is possible to prove the coercivity of S in the same way Equation

(1.15) was derived. If the coercivity is assumed, then the interface problem has a unique solution.

Now let us study the iterative procedures for solving the interface problem presented in Section 4.3. We consider the Dirichlet and Neumann-relaxed versions given by Equations $(4.47)_{1-2}$ and $(4.48)_{1-2}$, respectively, which both lead to the same stationary Richardson procedure $(4.50)_{1-2}$. In order to prove the convergence of the relaxed scheme, we need the continuities of $\tilde{S}_1$ and S_2, and the coercivity of the preconditioner S_2. Applying equation (1.14) for $\flat = 0$ to our case, we have

$$\langle S_2\mu_a, \mu_a\rangle_{\Gamma_a} \geq \frac{\varepsilon}{1 + C_{\Omega_2}} \|\mathcal{L}_2\mu_a\|^2_{1,\Omega_2} + \frac{1}{2}\int_{\Gamma_a} (\boldsymbol{a}\cdot\boldsymbol{n}_2)(\mathcal{L}_2\mu_a)^2\, d\Gamma.$$

In addition, we showed in Section 1.1.3 that if Γ_a is an outflow or if

$$\varepsilon > C' \|\boldsymbol{a}\cdot\boldsymbol{n}_2\|_{\infty,\Gamma_a}$$

where C' is a constant independent of $\boldsymbol{a}$ and ε, then S_2 is coercive. If this is the case, then we can apply Theorem 4.4 to prove the convergence of the Richardson procedure.

In this section, we have shown that the extension of the overlapping DD method derived for the Dirichlet/Robin algorithm is in principle possible, although additional assumptions on the data have to be made. In particular, we showed that the solution to the interface equation, which depends on the coercivity of S, is not as straightforward as in the non-overlapping case (see Equation (4.59)). The iterative procedure was established and its convergence is submitted to the same condition as in the non-overlapping case. However, we hope that in general the overlapping helps convergence, as has been already intuited in Remark 4.6. This will be confirmed by the numerical experiments at the end of this chapter.

Note finally that if we consider the bilinear form a^1, we are faced we the same problems than those encountered in the study of the overlapping Dirichlet/Neumann method. The 1/2-weak formulation is therefore the most appropriate for solving the scalar ADR equation using the overlapping mixed method.

4.5 Finite element approximation

In order to get further insight on the overlapping DD method introduced in this chapter, we will derive an algebraic system for the interface unknowns and see how the overlapping method solves the Schur complement system of the interface unknowns. The derivation of the Schur system does not depend on any particular choice of the original bilinear form, and therefore it can be applied to the overlapping Dirichlet/Robin as well as to the overlapping Dirichlet/Neumann methods. Before developing this algebraic system, we need to introduce the finite element partition and set up the finite element overlapping DD method. The numerical setting is valid for any arbitrary triangulation as long as the grids match in the overlapping zone and on the interfaces.

4.5.1 Discrete problem

Let us discretize the complete geometrical domain Ω into finite elements, such that the interfaces Γ_b and Γ_a lye on some elements segments of the triangulation $\mathcal{T}_h$, and define V_h the associated

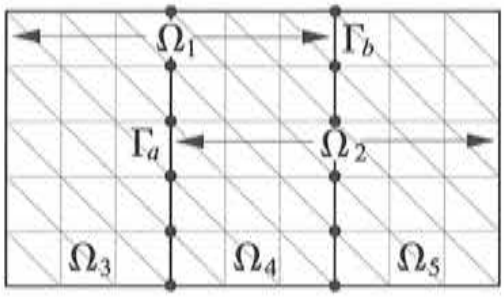


Figure 4.4: Finite element triangulation of the subdomains Ω_1 and Ω_2.

finite dimensional subspace of V. From the definition of V, we construct the finite dimensional subspaces $V_{1,h}$ and $V_{2,h}$, $V_{1,h}^0$ and $V_{2,h}^0$ of V_h such that

$$\begin{aligned} V_{1,h} &= \{v_h \in V_h \mid v_{h|\Omega_b} = 0\}, \\ V_{2,h} &= \{v_h \in V_h \mid v_{h|\Omega_3} = 0\}, \\ V_{1,h}^0 &= \{v_h \in V_h \mid v_{h|\overline{\Omega}_b} = 0\}, \\ V_{2,h}^0 &= \{v_h \in V_h \mid v_{h|\overline{\Omega}_3} = 0\}. \end{aligned}$$

Let $T_{a,h}$ be the discrete counterpart of T_a. We define $\Lambda_{a,h}$ the finite element subspace of $T_{a,h}v_h$,

$$\Lambda_{a,h} = \{T_{a,h}v_h \in V_h\}.$$

Clearly, $V_{1,h}^0$ and $V_{2,h}^0$ are finite dimensional subspaces of (the extensions to Ω of) V_1^0 and V_2^0, respectively, and $\Lambda_{a,h}$ is a finite dimensional subspace of Γ_a. Figure 4.4 shows an example of finite element triangulations of Ω_1 and Ω_2. The requirement that both $V_{1,h}$ and $V_{2,h}$ are constructed from V_h is necessary in order to make sure that nodes inside the overlapping zone coincide; this simplifies further analysis of the model problem.

The finite element formulation of the overlapping D/R method reads: find $u_{1,h} \in V_{1,h}$ and $u_{2,h} \in V_{2,h}$ such that

$$\begin{cases} a_{1,h}(u_{1,h}, v_{1,h}) = \langle f, v_{1,h}\rangle_{\Omega_1} & \forall\, v_{1,h} \in V_{1,h}^0, \\ u_{1,h} = u_{2,h} & \text{on } \Gamma_b, \\ a_2(u_{2,h}, v_{2,h}) = \langle f, v_{2,h}\rangle_{\Omega_2} & \forall\, v_{2,h} \in V_{2,h}^0, \\ a_3(u_{1,h}, E_{3,h}\mu_{a,h}) + a_2(u_{2,h}, E_{2,h}\mu_{a,h}) \\ \quad = \langle f, E_{3,h}\mu_{a,h}\rangle_{\Omega_3} + \langle f, E_{2,h}\mu_{a,h}\rangle_{\Omega_2} & \forall\, \mu_{a,h} \in \Lambda_{a,h}, \end{cases} \tag{4.60}$$

where $E_{i,h}$ denotes any possible extension operator, from the finite dimensional trace space it applies to, to $V_{i,h}$.

Remark 4.8. Contrary to the continuous case, we cannot derive an alternative formulation for the finite dimensional case. Remember that in the continuous case, the alternative formulation

was derived by using Equations (4.13) and (4.14). However, in the discrete case we have in general

$$\begin{aligned} Lu_{1,h} &\neq f \quad \text{in } \Omega_1, \quad \text{and} \\ Lu_{2,h} &\neq f \quad \text{in } \Omega_2. \end{aligned}$$

This is an important point as in principle we are not allowed to use the alternative formulation at the discrete level. In the next Chapter we will present a numerical scheme that enables us to deal with this formulation in order to preserve the order of convergence in h of the finite element method.

4.5.2 Iterative procedure

We now derive an iterative procedure to solve the domain decomposition problem (4.60). The original formulation was preferred to the alternative formulation for the corresponding algebraic scheme is much easier to set up. The sequential version of the iterative overlapping D/R algorithm is defined as follows. Given an initial guess $u_{2,h}^0$ on Γ_b, for each $k \geq 0$, find $u_{1,h}^{k+1} \in V_{1,h}$ and $u_{2,h}^{k+1} \in V_{2,h}$ such that

$$\left\{ \begin{array}{ll} a_1(u_{1,h}^{k+1}, v_{1,h}) = \langle f, v_{1,h} \rangle_{\Omega_1} & \forall\, v_{1,h} \in V_{1,h}^0, \\ u_{1,h}^{k+1} = u_{2,h}^k & \text{on } \Gamma_b, \\ a_2(u_{2,h}^{k+1}, v_{2,h}) = \langle f, v_{2,h} \rangle_{\Omega_2} & \forall\, v_{2,h} \in V_{2,h}^0, \\ a_2(u_{2,h}^{k+1}, E_{2,h}\mu_{a,h}) = -a_3(u_{1,h}^{k+1}, E_{3,h}\mu_{a,h}) & \\ \quad + \langle f, E_{3,h}\mu_{a,h} \rangle_{\Omega_3} + \langle f, E_{2,h}\mu_{a,h} \rangle_{\Omega_2} & \forall\, \mu_{a,h} \in \Lambda_{a,h}. \end{array} \right. \tag{4.61}$$

Note that all the results concerning the convergence of the algorithm derived in the continuous case are valid here and if this algorithm converges, the solutions on both subdomains satisfy Equations $(4.60)_{1-4}$. For the sake of clarity, we have omitted the relaxation of the transmission conditions.

We now go on with the study of the algebraic systems resulting from the three formulations of the ADR problem studied up to now, i.e. the one-domain formulation (4.1), the two-domain formulation $(4.60)_{1-4}$, and the iterative two-domain formulation $(4.61)_{1-4}$. In a first step, we construct the algebraic equation of the original one-domain problem. We then reorder the unknowns according to the geometrical decomposition used in the earlier section, and derive an exact equation for the unknown on the interfaces of the domain decomposition. We will recognize the ordering explicitly introduced by the domain decomposition algorithm (4.60). Afterwards, we construct the algebraic system of equations deriving from the iterative overlapping D/R method, and this system is solved for the unknowns on the interfaces. We show how the resulting iterative scheme can be related to the exact equations for the unknowns on the interfaces.

4.5.3 Schur complement equations

At this point, we need to define precisely the finite dimensional space V_h. Let us choose the space of linear piecewise polynomials. We denote x_r the coordinate of the node $\mathbf{x}_r$ of the triangulation of domain Ω, and n_p the total number of nodes. We define $N^{(r)}$ as the shape function associated

to node r and such that that $N^{(r)}(x_s) = \delta_{rs}$ for each $r, s = 1, \cdots, n_{\mathrm{p}}$. Then we construct the vector $\mathbf{N}$ of components $N^{(r)}$ for $r = 1, \ldots, n_{\mathrm{p}}$, i.e

$$\mathbf{N} = [N^{(1)}, \cdots, N^{(n_{\mathrm{p}})}]^t.$$

At any point of coordinates x inside the triangulation, the approximated solution u_h is

$$u_h(x) = \mathbf{N}^t \mathbf{u},$$

where the shape functions are evaluated at x and $\mathbf{u}$ is the vector of nodal unknowns. According to these definitions, we define the vector of forces as

$$\mathbf{f} = (f, \mathbf{N})_\Omega,$$

and the matrix $\mathbf{A}$ such that

$$\mathbf{A} = a(\mathbf{N}^t, \mathbf{N}),$$

or equivalently in terms of there respective coefficients

$$\begin{aligned} f_r &= (f, N^{(r)})_\Omega \qquad && \forall\, r = 1, \ldots, n_{\mathrm{p}}, \\ A_{r,s} &= a(N^{(s)}, N^{(r)}) \qquad && \forall\, r, s = 1, \ldots, n_{\mathrm{p}}. \end{aligned}$$

Hence, the finite dimensional variational form of the ADR equation leads to the following system:

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \tag{4.62}$$

We now decompose the matrix geometrically, according to the decomposition introduced in Section 4.2.1 and sketched in Figure 4.1. Let us define the matrices $\mathbf{A}_{ij}$ for $i, j = 3, 4, 5, a, b$ whose coefficients are

$$(A_{ij})_{r,s} = a(N^{(s)}, N^{(r)}) \qquad \forall\, r = 1, \ldots, n_{\mathrm{p}_i}, \text{ and } s = 1, \ldots, n_{\mathrm{p}_j},$$

where n_{p_i} is the number of nodes of the interior of the partition of Ω_i when $i = 3, 4, 5$ and the number of nodes of the partition of Γ_i when $i = a, b$. By performing a simple reordering of the unknowns, the system of equations (4.62) can be equivalently written as

$$\begin{pmatrix} \mathbf{A}_{33} & 0 & 0 & \mathbf{A}_{3a} & 0 \\ 0 & \mathbf{A}_{44} & 0 & \mathbf{A}_{4a} & \mathbf{A}_{4b} \\ 0 & 0 & \mathbf{A}_{55} & 0 & \mathbf{A}_{5b} \\ \mathbf{A}_{a3} & \mathbf{A}_{a4} & 0 & \mathbf{A}_{aa} & 0 \\ 0 & \mathbf{A}_{b4} & \mathbf{A}_{b5} & 0 & \mathbf{A}_{bb} \end{pmatrix} \begin{pmatrix} \mathbf{u}_3 \\ \mathbf{u}_4 \\ \mathbf{u}_5 \\ \mathbf{u}_a \\ \mathbf{u}_b \end{pmatrix} = \begin{pmatrix} \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{f}_5 \\ \mathbf{f}_a \\ \mathbf{f}_b \end{pmatrix}. \tag{4.63}$$

System (4.63) can be obtained as well by deriving the algebraic form of system $(4.60)_{1-4}$ using special extension operators. Let us set the operators $E_{i,h}$ $(i = 3, 2)$ for all $\mu_{a,h} \in \Lambda_{a,h}$ as follows:

$$\begin{cases} E_{i,h}\mu_{a,h}(x_s) &= \mu_{a,h}(x_s) & \text{when } \mathbf{x}_s \text{ is on } \Gamma_a, \\ &= 0 & \text{otherwise.} \end{cases}$$

$E_{i,h}\mu_{a,h}$ equals to zero on internal nodes and takes the value of its argument on the interface Γ_a. Let us define $\hat{u}_{i,h}$: it takes the value of $u_{i,h}$ on internal nodes and is zero on Γ_a; we define $\hat{v}_h$ in the same way. Consequently, we have the following equalities

$$u_{i,h} = \hat{u}_{i,h} + E_{i,h}T_a u_{i,h},$$
$$v_{i,h} = \hat{v}_{i,h} + E_{i,h}T_a v_{i,h},$$

which enable us to decompose the unknowns and test functions into internal and interface components. We are now able to construct the algebraic system given by Equation (4.60). Let $u_{i,j}$ be the unknown belonging to subdomain $i = 3, 4, 5, a, b$ and computed solving the finite element problem in Ω_j with $j = 1, 2$. Quantities without superscript refer to the non-iterative algorithm. The following algebraic equations show the contribution of each one of the equations of system $(4.60)_{1-4}$:

Contribution of Equation $(4.60)_1$:

$$\begin{pmatrix} \mathbf{A}_{33} & 0 & \mathbf{A}_{3a} & 0 \\ 0 & \mathbf{A}_{44} & \mathbf{A}_{4a} & \mathbf{A}_{4b} \\ \mathbf{A}_{a3} & \mathbf{A}_{a4} & \mathbf{A}_{aa} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{3,1} \\ \mathbf{u}_{4,1} \\ \mathbf{u}_{a,1} \\ \mathbf{u}_{b,2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{f}_a \end{pmatrix}. \tag{4.64}$$

Contribution of Equation $(4.60)_3$:

$$\begin{pmatrix} \mathbf{A}_{44} & 0 & \mathbf{A}_{4a} & \mathbf{A}_{4b} \\ 0 & \mathbf{A}_{55} & 0 & \mathbf{A}_{5b} \\ \mathbf{A}_{b4} & \mathbf{A}_{b5} & 0 & \mathbf{A}_{bb} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{4,2} \\ \mathbf{u}_{5,2} \\ \mathbf{u}_{a,2} \\ \mathbf{u}_{b,2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_4 \\ \mathbf{f}_5 \\ \mathbf{f}_b \end{pmatrix}. \tag{4.65}$$

Contribution of Equation $(4.60)_4$:

$$\begin{pmatrix} \mathbf{A}_{a3} & \mathbf{A}_{a4} & \mathbf{A}_{aa}^{(3)} & \mathbf{A}_{aa}^{(4)} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{3,1} \\ \mathbf{u}_{4,2} \\ \mathbf{u}_{a,1} \\ \mathbf{u}_{a,2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_a \end{pmatrix}. \tag{4.66}$$

We observe that the global system induced by the three previous systems of equations leads to the algebraic system (4.63) derived from the original problem, whenever the unknowns in Ω_1 and Ω_2 are the same in the overlapping zone.

Remark 4.9. Matrix $\mathbf{A}_{aa}$ was explicitly split into two components $\mathbf{A}_{aa}^{(3)}$ and $\mathbf{A}_{aa}^{(4)}$. The first component corresponds to the contribution of the first term of the left-hand side of Equation $(4.60)_4$, while the second contribution corresponds to the second term of the left-hand side of Equation $(4.60)_4$.

Remark 4.10. In practical implementations for which the solution on each subdomains is performed by separate finite element solvers, the Robin type condition is calculated using the normal derivatives. This means that Equation (4.66) is not assembled, but instead, Equation (4.65) is modified.

We now look for the solution $\mathbf{u}_a$ and $\mathbf{u}_b$ on the interfaces by eliminating the interior unknowns of Equation (4.63). The resulting system is the *Schur complement system*:

$$\begin{pmatrix} \mathbf{M}_a & \mathbf{P}_b \\ \mathbf{P}_a & \mathbf{M}_b \end{pmatrix} \begin{pmatrix} \mathbf{u}_a \\ \mathbf{u}_b \end{pmatrix} = \begin{pmatrix} \mathbf{g}_a \\ \mathbf{g}_b \end{pmatrix} \tag{4.67}$$

where

$$\begin{aligned} \mathbf{M}_a &= \mathbf{A}_{aa} - \mathbf{A}_{a3}\mathbf{A}_{33}^{-1}\mathbf{A}_{3a} - \mathbf{A}_{a4}\mathbf{A}_{44}^{-1}\mathbf{A}_{4a}, \\ \mathbf{M}_b &= \mathbf{A}_{bb} - \mathbf{A}_{b4}\mathbf{A}_{44}^{-1}\mathbf{A}_{4b} - \mathbf{A}_{b5}\mathbf{A}_{55}^{-1}\mathbf{A}_{5b}, \\ \mathbf{P}_a &= -\mathbf{A}_{b4}\mathbf{A}_{44}^{-1}\mathbf{A}_{4a}, \\ \mathbf{P}_b &= -\mathbf{A}_{a4}\mathbf{A}_{44}^{-1}\mathbf{A}_{4b}, \\ \mathbf{g}_a &= \mathbf{f}_a - \mathbf{A}_{a3}\mathbf{A}_{33}^{-1}\mathbf{f}_3 - \mathbf{A}_{a4}\mathbf{A}_{44}^{-1}\mathbf{f}_4, \\ \mathbf{g}_b &= \mathbf{f}_b - \mathbf{A}_{b4}\mathbf{A}_{44}^{-1}\mathbf{f}_4 - \mathbf{A}_{b5}\mathbf{A}_{55}^{-1}\mathbf{f}_5. \end{aligned} \tag{4.68}$$

This gives the following exact equations for $\mathbf{u}_a$ and $\mathbf{u}_b$:

$$\begin{aligned} \mathbf{Q}_a\mathbf{u}_a &= \chi_a, \\ \mathbf{Q}_b\mathbf{u}_b &= \chi_b, \end{aligned}$$

where

$$\begin{aligned} \mathbf{Q}_a &= \mathbf{M}_a - \mathbf{P}_b\mathbf{M}_b^{-1}\mathbf{P}_a, \\ \mathbf{Q}_b &= \mathbf{M}_b - \mathbf{P}_a\mathbf{M}_a^{-1}\mathbf{P}_b, \\ \chi_a &= \mathbf{g}_a - \mathbf{P}_b\mathbf{M}_b^{-1}\mathbf{g}_b, \\ \chi_b &= \mathbf{g}_b - \mathbf{P}_a\mathbf{M}_a^{-1}\mathbf{g}_a. \end{aligned}$$

Let us now identify the algebraic system resulting from the iterative algorithm $(4.61)_{1-4}$. We define $u_{i,j}^k$ as the unknown belonging to subdomain $i = 3, 4, 5, a, b$ and computed solving the finite element problem in Ω_j with $j = 1, 2$, at iteration k. For the sake of clarity, we consider the first cycle, i.e. we set $k = 0$. By remembering the contributions of each equations, explicitly given by Equations (4.64), (4.65) and (4.66), we construct the following iterative algebraic algorithm:

$$\begin{pmatrix} \mathbf{A}_{33} & 0 & \mathbf{A}_{3a} \\ 0 & \mathbf{A}_{44} & \mathbf{A}_{4a} \\ \mathbf{A}_{a3} & \mathbf{A}_{a4} & \mathbf{A}_{aa} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{3,1}^1 \\ \mathbf{u}_{4,1}^1 \\ \mathbf{u}_{a,1}^1 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_3 \\ \mathbf{f}_4 - \mathbf{A}_{4b}\mathbf{u}_{b,2}^0 \\ \mathbf{f}_a \end{pmatrix}, \tag{4.69}$$

$$\begin{pmatrix} \mathbf{A}_{44} & 0 & \mathbf{A}_{4a} & \mathbf{A}_{4b} \\ 0 & \mathbf{A}_{55} & 0 & \mathbf{A}_{5b} \\ \mathbf{A}_{a4} & 0 & \mathbf{A}_{aa}^{(4)} & 0 \\ \mathbf{A}_{b4} & \mathbf{A}_{b5} & 0 & \mathbf{A}_{bb} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{4,2}^1 \\ \mathbf{u}_{5,2}^1 \\ \mathbf{u}_{a,2}^1 \\ \mathbf{u}_{b,2}^1 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_4 \\ \mathbf{f}_5 \\ \mathbf{f}_a - \mathbf{A}_{a3}\mathbf{u}_{3,1}^1 - \mathbf{A}_{aa}^{(3)}\mathbf{u}_{a,1}^1 \\ \mathbf{f}_b \end{pmatrix}. \tag{4.70}$$

The first equation is the Dirichlet step on subdomain Ω_1, using $\mathbf{u}_{b,2}^0$ as Dirichlet condition on Γ_b (as initial guess). The second equation is the Neumann step, the Neumann condition on Γ_a being

given by the right-hand side of the third row of the system. During one complete iteration, we obtain therefore two estimates of the unknowns on the overlap zone $\mathbf{u}_{4,1}^1$ and $\mathbf{u}_{4,2}^1$, but also on the Neumann interface, $\mathbf{u}_{a,1}^1$ and $\mathbf{u}_{a,2}^1$. Note that the unknown on Γ_b never appears as belonging to subdomain 1, but always like $\mathbf{u}_{b,2}^k$, since this unknown is fixed as a Dirichlet condition when solving the algebraic problem on subdomain 1.

We now derive the equations for the three interface iterates involved in one iterative cycle, i.e. $\mathbf{u}_{a,1}^1$, $\mathbf{u}_{a,2}^1$ and $\mathbf{u}_{b,2}^1$. Before going further, we need to introduce some additional definitions. We split matrix $\mathbf{M}_a$, given by Equation (4.68), into two components, each one coming from the contribution of the two adjacent subdomains of Γ_a:

$$\begin{aligned}
\mathbf{M}_a &= \mathbf{M}_a^{(3)} + \mathbf{M}_a^{(4)}, \qquad \text{with} \\
\mathbf{M}_a^{(3)} &= \mathbf{A}_{aa}^{(3)} - \mathbf{A}_{a3}\mathbf{A}_{33}^{-1}\mathbf{A}_{3a}, \\
\mathbf{M}_a^{(4)} &= \mathbf{A}_{aa}^{(4)} - \mathbf{A}_{a4}\mathbf{A}_{44}^{-1}\mathbf{A}_{4a}.
\end{aligned}$$

By eliminating the interior unknowns, we can show that the algebraic iterative system represented by Equations (4.69) and (4.70) leads to the following system for the unknowns on the interfaces:

$$\begin{pmatrix} \mathbf{M}_a & 0 & 0 \\ \mathbf{M}_a^{(3)} & \mathbf{M}_a^{(4)} & \mathbf{P}_b \\ 0 & \mathbf{P}_a & \mathbf{M}_b \end{pmatrix} \begin{pmatrix} \mathbf{u}_{a,1}^1 \\ \mathbf{u}_{a,2}^1 \\ \mathbf{u}_{b,2}^1 \end{pmatrix} = \begin{pmatrix} \mathbf{g}_a - \mathbf{P}_b\mathbf{u}_b^0 \\ \mathbf{g}_a \\ \mathbf{g}_b \end{pmatrix}. \tag{4.71}$$

Let us take a breath and look closer at the last system. In reference [87], the authors showed that the multiplicative version of the Schwarz algorithm can be viewed as a preconditioned Richardson iteration for the Schur complement of the interface unknowns, as stated by Equation (4.67). According to the geometrical decomposition and nomenclature used in this section, a Schwarz iteration produces the following sequence for $\mathbf{u}_{a,1}^1$ and $\mathbf{u}_{b,2}^1$:

$$\begin{aligned}
\mathbf{M}_a\mathbf{u}_{a,1}^1 &= \mathbf{g}_a - \mathbf{P}_b\mathbf{u}_b^0, \\
\mathbf{M}_b\mathbf{u}_{b,2}^1 &= \mathbf{g}_b - \mathbf{P}_a\mathbf{u}_{a,1}^1.
\end{aligned}$$

We recognize here the iterates that would be produced by our algorithm by ignoring the intermediate iteration step for $\mathbf{u}_{a,2}^1$ and letting $\mathbf{u}_{a,2}^1 = \mathbf{u}_{a,1}^1$ to compute the update of $\mathbf{u}_{b,2}^1$ in Equation (4.71). As a result, we intuit that the overlapping D/R algorithm gives more information than the multiplicative Schwarz method during one cycle. Let us now go back to Equation (4.71). We find that the algorithm produces the following iterates:

$$\begin{aligned}
(\mathbf{Q}_a - \mathbf{M}_a^{(3)})\mathbf{u}_{a,2}^1 &= \chi_a - \mathbf{M}_a^{(3)}\mathbf{u}_{a,1}^1, \\
\mathbf{u}_{b,2}^1 &= \mathbf{M}_b^{-1}(\mathbf{g}_b - \mathbf{P}_a\mathbf{u}_{a,2}^1).
\end{aligned}$$

We note that the first iterate is equivalent to a Richardson iteration using as preconditioner matrix $(\mathbf{Q}_a - \mathbf{M}_a^{(3)})$, i.e.

$$\mathbf{u}_{a,2}^1 = (\mathbf{Q}_a - \mathbf{M}_a^{(3)})^{-1}(\chi_a - \mathbf{Q}_a\mathbf{u}_{a,1}^1) + \mathbf{u}_{a,1}^1.$$

This equation is similar to the Richardson iteration produced by the classical (disjoint) D/R preconditioner, as devised in [98]. In fact, in the limit of non-overlapping subdomains, we recover

exactly the same Richardson iteration. This result is obtained by collapsing the interface unknowns, i.e. by choosing $\mathbf{u}^1_{b,2} = \mathbf{u}^1_{a,2}$, $\mathbf{M}_b = \mathbf{I}$, $\mathbf{P}_a = \mathbf{0}$, $\mathbf{P}_b = \mathbf{0}$ and $\mathbf{g}_b = \mathbf{u}^1_{a,1}$. As a result,

$$\mathbf{u}^1_{a,2} = \mathbf{M}_a^{(4)^{-1}}(\mathbf{g}_a - \mathbf{M}_a\mathbf{u}^1_{a,1}) + \mathbf{u}^1_{a,1}.$$

Nevertheless, the present algorithm gives additional information, provided by the second iterate. In fact, we observe that the iteration for $\mathbf{u}^1_{b,2}$ is a block Gauss-Seidel iteration of the Schur complement system (4.67). This Gauss-Seidel iteration is the same iteration than that produced by the multiplicative Schwarz algorithm, as shown in [87]. In the view of the iterates produced for the interface unknowns, we can therefore expect the overlapping D/R method inherits some properties of the classical D/R method (independence on the mesh size) together with that of the Schwarz method (convergence can be improved with the overlapping length for a given mesh size). In the limit of zero overlapping length, the classical D/R method is recovered.

We can also develop a Richardson procedure for solving $\mathbf{u}_b$ as well. Simple algebraic calculations give

$$\mathbf{u}^1_{b,2} = (\mathbf{M}_b - \mathbf{P}_a\mathbf{M}_a^{(4)^{-1}}\mathbf{P}_b)^{-1}(\chi_b - \mathbf{Q}_b\mathbf{u}^0_b) + \mathbf{u}^0_b.$$

4.6 Numerical examples

We present four numerical examples to test the overlapping D/R method in the diffusion as well as in the advection limits. In addition, several flow configurations are considered: skew advection, normal and tangential advections, curved advection and, finally, a rotating advection field.

4.6.1 Skew advection

Through this example, which was used as a first test case of the classical γ-D/R method in [73], we want to compare the disjoint and overlapping versions of the D/R method, for a skew advection field. As an additional indication when using overlapping grids, we will systematically give the results of the Schwarz method (D/D) for overlapping subdomains, and that of the adaptive D/N method (A-D/N) for both disjoint and overlapping subdomains. The overlapping version of the A-D/N method uses a Neumann interface at outflow and a Dirichlet interface at inflow, as in the classical disjoint case. We propose to solve the equation

$$-\varepsilon\Delta u + \boldsymbol{a}\cdot\nabla u = f \qquad \text{in } \Omega = (0,1)\times(0,1),$$

with a skew advection field $\boldsymbol{a} = [1,1]^t$, and look for the exact solution $u = u(x,y) = x + 5y$, which belongs to the finite element space of work. According to this choice, we impose $f = 6$, and exact Dirichlet conditions on the boundary; see Figure 4.5.

We define three different meshes, with $h = 1/10$, $h = 1/20$ and $h = 1/40$. In addition, we define three different partitionings. The splitting of the two subdomains is always performed vertically and symmetrically with respect to the line $x = 0.5$. The first partition splits Ω into two disjoint subdomains, the second into two overlapping subdomains with horizontal overlapping length $\delta = 0.2$, and the third one with $\delta = 0.4$. The numerical strategy is classical. We use the Q1 element together with the variational subgrid scale model (indispensable for small ε). In order to introduce as few extrinsic errors to the DD methods themselves as possible, all the matrices

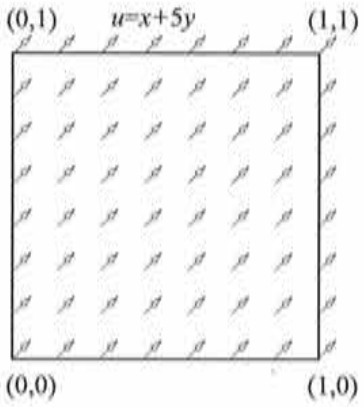


Figure 4.5: Computational domain and boundary conditions.

involved in the Schur complement system are inverted using a direct solver. When considering disjoint subdomains, the convergence criterion is the *interface L_2 residual*

$$100\,\frac{\|\mathbf{u}_a^{k+1}-\mathbf{u}_a^k\|_2}{\|\mathbf{u}_a^k\|_2}\leq 10^{-10},$$

while for overlapping subdomains it is given by

$$100\,\frac{\|\mathbf{u}_a^{k+1}-\mathbf{u}_a^k+\mathbf{u}_b^{k+1}-\mathbf{u}_b^k\|_2}{\|\mathbf{u}_a^k+\mathbf{u}_b^k\|_2}\leq 10^{-10}.$$

Tables 4.1 and 4.2 present the already known results of the disjoint D/R and adaptive D/N methods. The former confirms the mesh independence of both methods, while the latter gives the optimum relaxation parameter θ_{opt} and the corresponding numbers of iterations needed to achieve convergence. Possible values of θ have been limited to two decimal figures. As expected, we note that θ_{opt} for the D/R method is always 0.5, while that of the A-D/N method is somewhere between 0.5 and 1, and depends on ε.

	D/R			A-D/N		
$\varepsilon \backslash\, h$	1/10	1/20	1/40	1/10	1/20	1/40
10^{1}	2	2	2	8	8	8
10^{0}	5	4	2	15	15	15
10^{-1}	7	6	5	31	31	31
10^{-2}	8	8	7	39	39	39
10^{-3}	8	8	8	40	40	40
10^{-4}	9	8	8	41	41	41
10^{-5}	9	8	8	41	41	41

Table 4.1: Number of iterations ($\theta = 0.5$, $\delta = 0$)

Tables 4.3, 4.4, 4.5 and 4.6 present the same results as the former ones, but this time for the overlapping methods. The tables show that the overlapping D/R method behaves like the classical

	D/R		A-D/N	
ε	θ_{opt}	#	θ_{opt}	#
10^1	0.50	2	0.50	8
10^0	0.50	4	0.54	11
10^{-1}	0.50	6	0.65	19
10^{-2}	0.50	8	0.81	17
10^{-3}	0.50	8	0.90	17
10^{-4}	0.50	9	0.93	18
10^{-5}	0.50	9	0.93	18

Table 4.2: θ_{opt} and number of iterations ($\delta = 0$)

D/N method for ε high, and like the D/D method for ε small. We observe that when $\varepsilon \ll 1$, the convergence of the D/R will improve with decreasing h, and that the number of iterations is bounded as ε goes to zero. We also note that for all the DD methods tested, the optimum θ is close to unity in the diffusion-dominated range, while it is exactly one in the advection-dominated range. This contrasts completely with the disjoint counterparts of the DD methods.

	D/R			A-D/N			D/D		
$\varepsilon \backslash\ h$	1/10	1/20	1/40	1/10	1/20	1/40	1/10	1/20	1/40
10^1	23	23	23	23	23	23	21	21	21
10^0	23	23	23	19	19	19	21	21	21
10^{-1}	10	11	11	7	8	8	10	11	11
10^{-2}	10	6	3	7	4	3	10	6	3
10^{-3}	12	7	5	7	5	4	11	7	5
10^{-4}	12	7	5	7	5	4	12	7	5
10^{-5}	12	7	5	7	5	4	12	7	5

Table 4.3: Number of iterations ($\theta = 1.0$, $\delta = 0.2$)

	D/R		A-D/N		D/D	
ε	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#
10^1	0.87	14	0.87	14	1.14	16
10^0	0.87	14	0.90	13	1.14	15
10^{-1}	0.98	9	1.00	8	1.02	9
10^{-2}	1.00	6	1.00	4	1.00	6
10^{-3}	1.00	7	1.00	5	1.00	7
10^{-4}	1.00	7	1.00	5	1.00	7
10^{-5}	1.00	7	1.00	5	1.00	7

Table 4.4: θ_{opt} and number of iterations ($\delta = 0.2$)

Table 4.7 gives the number of iterations needed to achieve convergence for the different methods, as a function of the overlapping length, and for the second finest mesh $h = 1/20$. We observe that for $\varepsilon = 10^1$ and $\varepsilon = 10^0$, the overlapping does not improve the convergence. This is rather a coincidence than a rule. For example, locating the interface at $x = 0.75$, the disjoint D/R method converges in 14 iterations at least in both cases!

	D/R			A-D/N			D/D		
$\varepsilon \setminus h$	1/10	1/20	1/40	1/10	1/20	1/40	1/10	1/20	1/40
10^{1}	12	12	12	12	12	12	11	11	11
10^{0}	12	12	12	11	11	11	11	11	11
10^{-1}	6	6	6	5	5	5	6	6	6
10^{-2}	6	4	2	5	3	2	6	4	2
10^{-3}	7	4	2	5	4	3	7	4	3
10^{-4}	7	4	3	5	4	3	7	4	3
10^{-5}	7	4	3	5	4	3	7	4	3

Table 4.5: Number of iterations ($\theta = 1.0$, $\delta = 0.4$)

	D/R		A-D/N		D/D	
ε	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#
10^{1}	0.96	10	0.96	10	1.03	9
10^{0}	0.97	10	0.97	9	1.01	10
10^{-1}	1.00	6	1.00	5	1.00	6
10^{-2}	1.00	4	1.00	3	1.00	4
10^{-3}	1.00	4	1.00	4	1.00	4
10^{-4}	1.00	4	1.00	4	1.00	4
10^{-5}	1.00	4	1.00	4	1.00	4

Table 4.6: Number of iterations and θ_{opt} ($\delta = 0.4$)

	D/R			A-D/N			D/D		
$\varepsilon \setminus \delta$	0	0.2	0.4	0	0.2	0.4	0	0.2	0.4
10^{1}	2	14	10	8	14	10	-	16	10
10^{0}	4	14	10	11	13	9	-	15	10
10^{-1}	6	9	6	19	8	5	-	9	6
10^{-2}	8	6	4	17	4	3	-	6	4
10^{-3}	8	7	4	17	5	4	-	7	4
10^{-4}	8	7	4	18	5	4	-	7	4
10^{-5}	8	7	4	18	5	4	-	7	4

Table 4.7: Number of iterations ($\theta = \theta_{\text{opt}}$)

Finally, Figure 4.6 compares the behaviors of the DD methods around their respective optimum relaxation parameters.

Before closing the analysis of this example, let us examine how the error is reduced by the disjoint and overlapping D/R methods ($\delta = 0.2$), for high advection ($\varepsilon = 10^{-4}$). We choose θ such that the rate of convergence of each method is more or less the same, to be able to compare the error reduction using the same scale; this choice corresponds to $\theta = 0.44$ in the case of the disjoint D/R method, and $\theta = 0.9$ in the case of the overlapping D/R method. The initial solution is the exact solution, on which we superimpose an error with respect to the analytical solution somewhere on the interface. In the case of the disjoint D/R method, we introduce the error at point $(0.5, 0.5)$, while for the overlapping version, we introduce the error at point $(0.4, 0.5)$. The magnitude of the

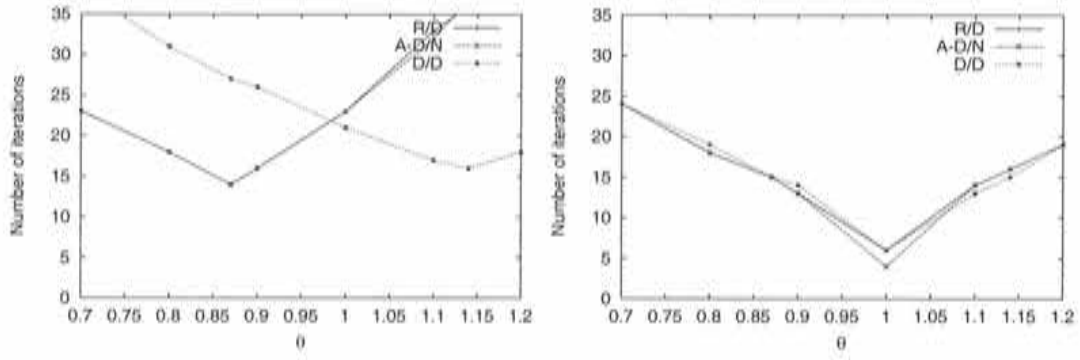


Figure 4.6: Number of iterations. (Left) $\varepsilon = 10^{1}$. (Right) $\varepsilon = 10^{-2}$.

error in both cases is 0.5, normalized by the maximum exact value over the domain, i.e. 6. On the one hand, Figures 4.7 (Top Left) and (Top Right) show how the error is advected along the streamlines of the flow, at iterations 2 and 4, respectively. On the other hand, Figures 4.7 (Bot. Left) and (Bot. Right) show how the error is mostly confined between the interfaces, located at $x = 0.4$ and $x = 0.6$.

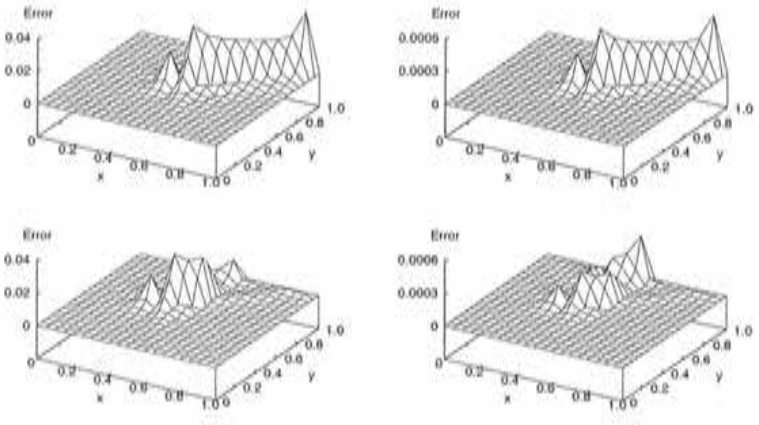


Figure 4.7: Error. (Top) (Left) Disjoint D/R, iteration 2. (Top) (Right) Disjoint D/R, iteration 4. (Bot.) (Left) Overlapping D/R, iteration 2. (Bot.) (Right) Overlapping D/R, iteration 4.

4.6.2 Normal and tangential advections

This example studies the solution of a thermal boundary layer, also presented in [73],

$$-\varepsilon\Delta u + \boldsymbol{a}\cdot\nabla u = 0 \qquad \text{in } \Omega = (0,1)\times(0,0.5),$$

with an horizontal advection field $\boldsymbol{a} = [2y, 0]^t$ and the following boundary conditions:

$$\begin{aligned} u &= 1 && \text{at } x = 0, \text{ and y=0.5,}\\ u &= 2y && \text{at } x = 1,\\ u &= 0 && \text{elsewhere.} \end{aligned}$$

The geometry as well as the boundary conditions are shown in Figure 4.8 (Left). Let us mention that the solution to this problem exhibits a parabolic layer near $y = 0$ and an exponential layer at $x = 1$ for small diffusion coefficients, as shown in Figure 4.8 (Right) for $\varepsilon = 10^{-2}$.

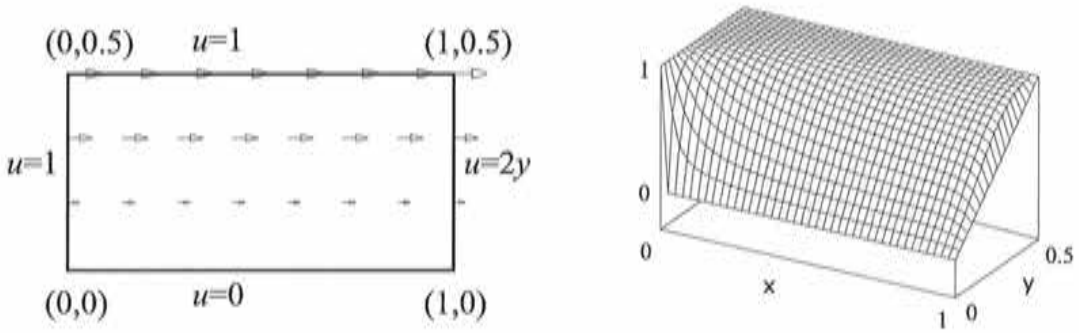


Figure 4.8: (Left) Computational domain and boundary conditions. (Right) Solution for $\varepsilon = 10^{-2}$.

This example is solved using the same numerical strategy as that of the previous example. The mesh convergence shares sensibly the same characteristics as that of the first example so only the results run with a mesh size of $h = 1/20$ are reported here. Two different partitionings are performed. First we consider a symmetric vertical partitioning of the domain, i.e. the interface is placed normal to the advection field. Tables 4.8 and 4.9 compare the optimum relaxation parameters and the associated number of iterations of the disjoint and overlapping versions of the different DD methods. As was already observed in last example, we note that the θ_{opt} of the disjoint D/R method is 0.5, while that of the overlapping D/R is 1. The results of the A-D/N method are more mitigated. On the one hand, the θ_{opt} of the disjoint version tends to unity very slowly for decreasing ε. On the other hand, the θ_{opt} of the overlapping version is, as in the case of the overlapping D/R, unity for $\varepsilon \leq 10^{-2}$. As in the previsous example, the number of iterations is bounded as ε goes to zero.

We now partition Ω horizontally. In this case, the Neumann and Robin conditions coincide as $\boldsymbol{a}\cdot\boldsymbol{n} = 0$. Table 4.10 gives the results obtain for the classical D/N method. As in the case of the Normal advection, we observe that the optimum relaxation parameter of all methods tends to unity rapidly when $\varepsilon \leq 10^{-2}$, while that of the disjoint D/N method remains around 0.5.

	D/R		A-D/N	
ε	θ_{opt}	#	θ_{opt}	#
10^{1}	0.50	5	0.50	7
10^{0}	0.50	6	0.51	9
10^{-1}	0.50	9	0.61	15
10^{-2}	0.50	10	0.74	21
10^{-3}	0.50	7	0.92	12
10^{-4}	0.50	4	0.99	8
10^{-5}	0.50	3	1.00	6

Table 4.8: Normal advection. θ_{opt} and number of iterations ($\delta = 0$)

	D/R		A-D/N		D/D	
ε	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#
10^{1}	0.96	10	0.96	10	1.04	10
10^{0}	0.96	10	0.96	10	1.04	10
10^{-1}	0.97	10	0.98	8	1.03	10
10^{-2}	1.00	5	1.00	5	1.00	6
10^{-3}	1.00	4	1.00	3	1.00	4
10^{-4}	1.00	5	1.00	3	1.00	5
10^{-5}	1.00	5	1.00	3	1.00	5

Table 4.9: Normal advection. θ_{opt} and number of iterations ($\delta = 0.2$)

	D/N $\delta = 0$		D/N $\delta = 0.1$		D/N $\delta = 0.2$		D/D $\delta = 0.1$		D/D $\delta = 0.2$	
ε	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#
10^{1}	0.50	5	0.79	18	0.89	14	1.24	20	1.08	12
10^{0}	0.50	6	0.79	18	0.89	14	1.24	20	1.07	12
10^{-1}	0.49	10	0.80	18	0.91	13	1.21	19	1.06	11
10^{-2}	0.47	10	0.99	10	1.00	6	1.01	9	1.00	6
10^{-3}	0.48	8	1.00	6	1.00	4	1.00	6	1.00	4
10^{-4}	0.47	8	1.00	7	1.00	4	1.00	6	1.00	4
10^{-5}	0.47	7	1.00	7	1.00	4	1.00	7	1.00	4

Table 4.10: Tangential advection. θ_{opt} and number of iterations

4.6.3 Curved advection

We increase a bit the difficulty. We consider a curved advection field and impose a discontinuity in the Dirichlet condition. This example was proposed by Toselli in [99] and consists in solving

$$-\varepsilon\Delta u + \boldsymbol{a}\cdot\nabla u + su = 0 \qquad \text{in } \Omega = (-1,1)\times(-1,1),$$

where the advection field and the source term are given by

$$\begin{aligned} \boldsymbol{a} &= \frac{1}{2}[(1-x^2)(1+y), -x(4-(1+y)^2)]^t, \\ s &= 10^{-4}, \end{aligned}$$

and the Dirichlet boundary conditions for u are

$$\begin{aligned} u &= 1 \qquad \text{at } y = -1,\ 0 < x < 0.5, \\ u &= 0 \qquad \text{elsewhere.} \end{aligned}$$

See Figure 4.9 (Left) for a sketch of the problem. We present here the results obtained on three meshes composed of constant element length h such that $h = 1/10$ for the coarse mesh, $h = 1/20$ for the medium mesh and $h = 1/40$ for the fine mesh. Figure 4.9 (Right) shows the solution obtained on the medium mesh for $\varepsilon = 10^{-2}$.

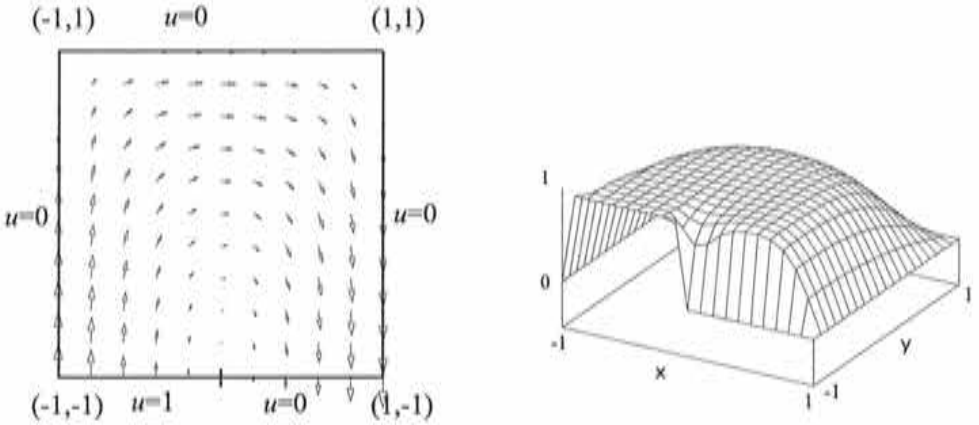


Figure 4.9: (Left) Computational domain and boundary conditions. (Right) Solution for $\varepsilon = 10^{-2}$.

In this example, we want to compare the results of the overlapping and disjoint D/R method without trying to adjust the relaxation parameter. For the disjoint versions, we take $\theta = 0.5$ and for the overlapping versions we take $\theta = 1.0$. We consider symmetrical horizontal and vertical partitionings, with an overlap of $\delta = 0.4$ for the overlapping partitions. As different results have been found (in the disjoint version) depending on where the Dirichlet and Robin interfaces are imposed, the Dirichlet/Robin method is referred to as D/R method when the Dirichlet condition is imposed on the top and left subdomain interfaces in the case of horizontal and vertical partitionings, respectively. On the contrary, the Dirichet/Robin methid is referred to as R/D method.

Tables 4.11, 4.12 and 4.13 gives the numbers of iterations needed to achieve convergence for all the methods. We notice that in the diffusion range, the disjoint versions converge better than the overlap versions. The tendency is inverted as soon as the advection compensates and overcomes the diffusion, i.e. when $\varepsilon \leq 10^{-1}$. In addition, the overlapping version shows much less sensitivity to the positioning of the interface when the mesh is coarse. In all cases, the number of iterations is bounded as the diffusion decreases.

	Disjoint				Overlapping			
	D/R		R/D		D/R		R/D	
ε	horiz.	verti.	horiz.	verti.	horiz.	verti.	horiz.	verti.
10^{1}	6	7	6	7	23	23	23	23
10^{0}	10	10	10	10	23	22	23	23
10^{-1}	16	18	16	18	13	15	12	15
10^{-2}	23	16	17	16	8	6	8	6
10^{-3}	40	16	21	16	9	9	9	9
10^{-4}	46	18	22	17	9	10	9	10
10^{-5}	47	18	22	18	9	10	9	10

Table 4.11: Number of iterations. Coarse mesh: $h = 1/10$

	Disjoint				Overlapping			
	D/R		R/D		D/R		R/D	
ε	horiz.	verti.	horiz.	verti.	horiz.	verti.	horiz.	verti.
10^{1}	6	7	6	7	24	23	24	23
10^{0}	10	10	10	10	23	23	23	23
10^{-1}	17	18	17	18	11	14	12	14
10^{-2}	15	17	14	17	5	5	5	5
10^{-3}	23	16	17	16	6	5	6	5
10^{-4}	26	16	18	16	6	7	6	7
10^{-5}	27	17	18	17	6	7	6	7

Table 4.12: Number of iterations. Medium mesh: $h = 1/20$

	Disjoint				Overlapping			
	D/R		R/D		D/R		R/D	
ε	horiz.	verti.	horiz.	verti.	horiz.	verti.	horiz.	verti.
10^{1}	6	7	7	7	24	23	24	23
10^{0}	10	10	10	10	23	23	23	23
10^{-1}	17	19	17	19	11	14	12	14
10^{-2}	16	18	12	18	4	6	4	6
10^{-3}	16	16	14	16	4	3	4	3
10^{-4}	18	16	15	16	4	5	4	5
10^{-5}	18	17	15	17	4	5	4	5

Table 4.13: Number of iterations. Fine mesh: $h = 1/40$

4.6.4 Rotating advection

Now let us try to solve a more challenging problem. We consider once more the exact linear solution $u = u(x, y) = x + 5y$ of the first test case, but this time using a rotating advection field centered in $(0.6, 0.6)$ such that $\boldsymbol{a} = [-y + 0.6, x - 0.6]$.

This choice leads us to choose the following force term: $f = 5x - y$. We have chosen this case because of its complicated local behavior. Around the center of the rotating advection filed,

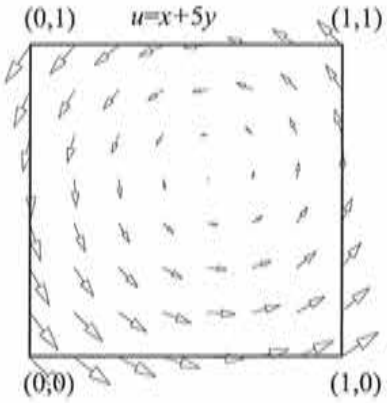


Figure 4.10: Computational domain and boundary conditions.

diffusion dominates. In addition, the interfaces considered are both inflow and outflow. The results presented here have been obtained on a 20×20 element mesh, and the interfaces are the same as those of the first test case.

Table 4.14 shows the number of iterations needed to achieve convergence for the optimum relaxation parameter.

	D/R $\delta = 0$		R/D $\delta = 0$		D/R $\delta = 0.2$		R/D $\delta = 0.2$	
ε	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#	θ_{opt}	#
10^{1}	0.50	5	0.50	5	0.87	14	0.87	14
10^{0}	0.50	8	0.50	8	0.87	14	0.87	14
10^{-1}	0.50	14	0.50	13	0.88	14	0.88	14
10^{-2}	0.49	40	0.50	34	0.97	11	1.07	13
10^{-3}	0.46	243	0.48	200	1.43	37	1.54	49
10^{-4}	0.47	1864	0.49	1493	1.85	221	1.87	242
10^{-5}	0.47	8460	0.51	6257	1.94	753	1.95	816

Table 4.14: θ_{opt} and number of iterations ($\delta = 0$, $\delta = 0.2$)

In this example, we have observed notable differences in the results depending on which interfaces the Robin and Dirichlet conditions are imposed; we denote them D/R when the first subdomain is assigned a Dirichlet condition and R/D when it is assigned a Robin condition. We observe that for the disjoint and overlapping versions with $\delta = 0.2$ the number of iterations blows up when ε decreases. However, the overlapping decreases this figure by approximately one order of magnitude. In addition, we have considered the case of $\delta = 0.4$. The compared results are shown in Figure Figures 4.11 (Left) and (Right). They confirm the improvement in convergence when using overlapping.

As in the first test case, we now introduce a perturbation (an error peak) on the interface, of magnitude 0.5. The difficulty of solving this case lies in the fact that, for small diffusion coefficients, the error is advected around and around, flowing along the streamlines. If the error is introduced near the center of the vortex, it can remain for a long time within the domain before being diffused and absorbed by the boundary conditions. On the contrary, if the perturbation is put sufficiently

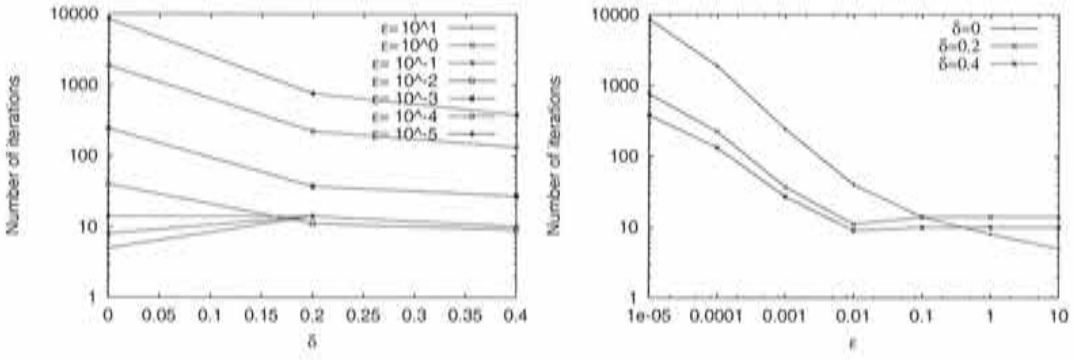


Figure 4.11: Number of iterations as a function of δ and ε.

far from the center of rotation, the error will be advected rapidly away from the domain and absorbed by the boundary conditions. We consider here the case $\varepsilon = 10^{-4}$. As an illustration, we have also solved the unpreconditioned Richardson procedure for the interface unknowns, using disjoint subdomains; in this case, the error is introduced at point $(0.5, 0.5)$. The error magnitude is 0.5 (normalized by the maximum value, i.e. 6). Figure 4.12 shows the error obtained after 1000 and 4000 iterations, using $\theta = 0.50$. After 1000 iterations, we still recognize the error peak

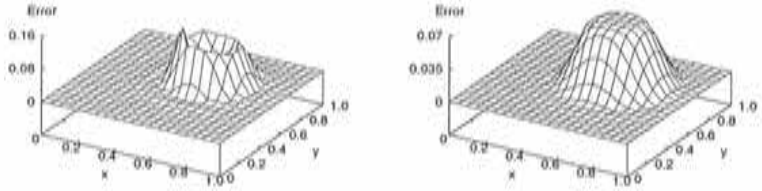


Figure 4.12: Error. Unpreconditioned Richardson procedure. (Left) Iteration 1000. (Right) Iteration 4000.

introduced at point $(0.5, 0.5)$; we also note that the error has been totally advected around. After 4000 iterations, the error has been diffused inside and outside the advection circle. Let us now go back to the analysis of the disjoint and overlapping ($\delta = 0.2$) D/R methods. In the case of the disjoint D/R method, we introduce the error at point $(0.5, 0.5)$, while for the overlapping version, we introduce the error at point $(0.6, 0.5)$. Figure 4.13 compares the convergence histories of both versions, using $\theta = 0.5$ and $\theta = 1.0$ respectively. We observe that the convergence of the disjoint D/R method is far from monotone.

Figure 4.14 represents the error with respect to the exact solution and normalized by the maximum exact solution at iteration 1,6,11,16,21,26,31 and 36. These iterations are labeled in Figure 4.13. We notice that after few iterations the error of the disjoint D/R exhibits more or less the same error profile as the unpreconditioned Richardson procedure, although the error is diffused much more rapidly (in terms of iterations). However, after having decreased one order of magnitude, the error bounces up, before decreasing once again, and so on, until convergence. This phenomenon can be clearly identified in the convergence history of the method. The error profiles of the overlapping versions at iterations 1,6,11,16,21,26,31 and 36 are shown in Figure 4.15

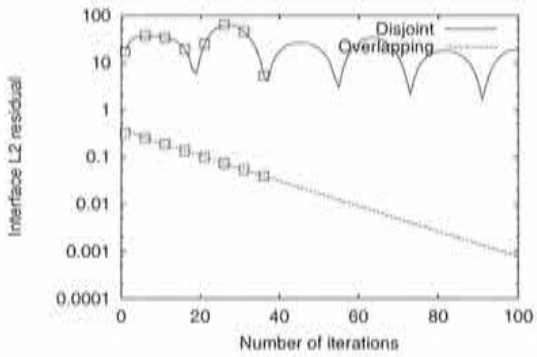


Figure 4.13: Convergence histories of the disjoint and overlapping D/R methods.

(Bot.). They confirm the improvements achieved by the overlapping method. We conclude that the overlapping can be useful when a vortex passes near the interface.

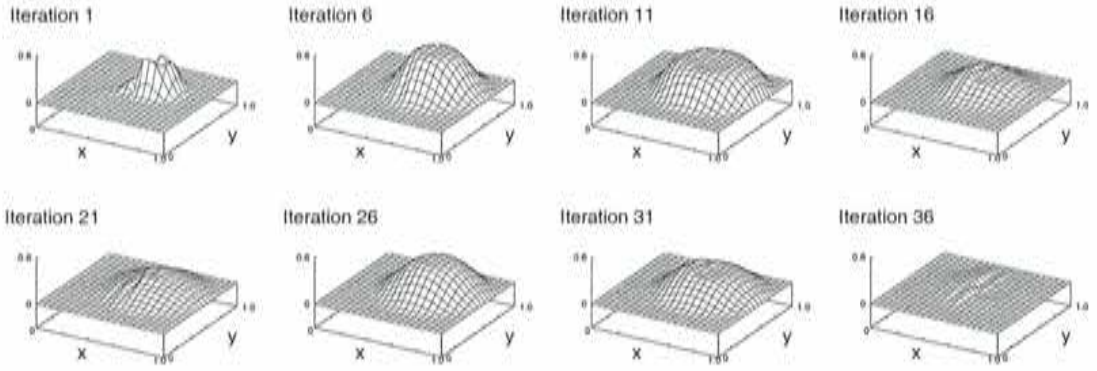


Figure 4.14: Error. Disjoint D/R.

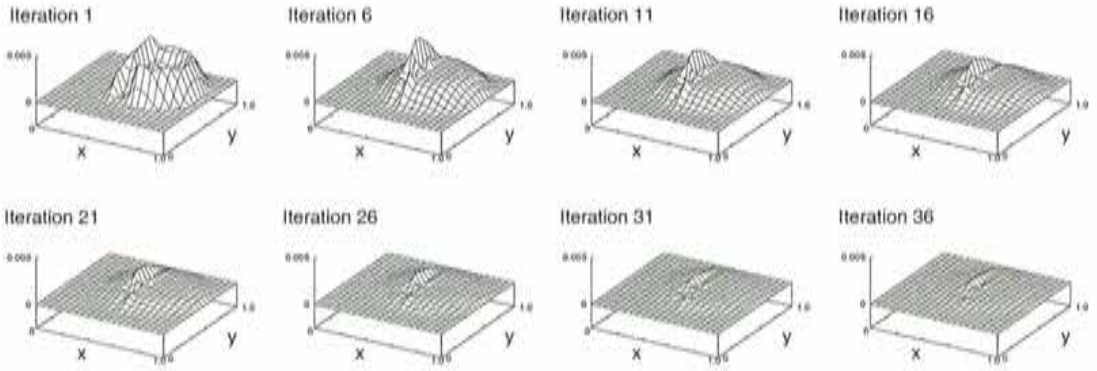


Figure 4.15: Error. Overlapping D/R.

4.7 Summary

From a geometrical decomposition of the domain Ω into two overlapping subdomains Ω_1 and Ω_2, we split the variational formulation of the ADR equation. The resulting variational formulation consists in solving three local equations together with a matching condition on interface Γ_b. These three equations are two equations for the interior of Ω_1 and Ω_2, and one equation for the interface Γ_a. We showed that this variational formulation was equivalent to the original one. Then, we observed that the formulation implied the continuity of the fluxes on Γ_a; this observation justified the name overlapping Dirichlet/Robin domain decomposition method. We also reexpressed the domain decomposition method as a method for solving the interface unknowns. Then we presented a relaxed iterative scheme for solving the decoupled problem. We first noticed that relaxation of the Dirichlet condition or that of the Robin condition had the same effects on the interface iterates. We identified these iterates as Richardson methods for solving the interface unknowns. From the finite element formulation of the DD method, we set up a sequential iterative scheme to decouple the solution on Ω_1 and Ω_2 into two subproblems; a Dirichlet problem on Ω_1 and a Robin problem on Ω_2. The derivation of the Schur complement system brought to light the link between the iterative strategy and two common approaches, namely the classical D/R method and the multiplicative Schwarz method. Finally we presented four numerical examples. They outlined the ambivalent nature of the overlapping D/R method which behaves sometimes like the classical Dirichlet/Neumann method and sometimes like the Schwarz method. In any case, the overlapping can be very usefull to accelerate the convergence, in particular in the presence of a vortex in the vicinity of the interface.

4.7.1 Parallel version and algorithm for many subdomains

The extension of the sequential overlapping mixed methods to a parallel version, as well as the case of many subdomains (for example in the view of parallel implementation via a multicoloring technique) can be treated exactly as for non-overlapping methods. However, this does not fall within the scope of this work and we refer the reader to the survey papers [85, 55] or the books [83, 58].

4.8 Extension to the Stokes and Navier-Stokes equations

We want to apply the overlapping mixed domain decomposition method to the following stationary Stokes problem in n_d dimensions

$$\begin{aligned} -2\nu\nabla\cdot\varepsilon(\boldsymbol{u}) + \nabla p &= \boldsymbol{f} && \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{0} && \text{on } \partial\Omega. \end{aligned}$$

Let us introduce the following functional spaces

$$\begin{aligned} V^0 &= H_0^1(\Omega)^{n_d}, \\ Q &= L^2(\Omega), \\ P &= \{p \in L^2(\Omega) \mid \int_\Omega p \, d\Omega = 0\}. \end{aligned}$$

Keeping in mind the notation used in Section 1.2.5, this would correspond to the case $\Gamma_D = \partial\Omega$ and $\Gamma_N = \Gamma_M = \emptyset$. Let $\boldsymbol{U} = [\boldsymbol{u}, p]^t$ be the unknowns and $\boldsymbol{V} = [\boldsymbol{v}, q]^t$ the associated test functions. The weak formulation of the stationary Stokes equations consists in finding $\boldsymbol{U} \in V^0 \times P$ such that

$$a(\boldsymbol{U}, \boldsymbol{V}) = l(\boldsymbol{V}) \qquad \forall\, \boldsymbol{V} \in V^0 \times Q,$$

where

$$\begin{aligned} a(\boldsymbol{U}, \boldsymbol{V}) &= 2\nu \int_\Omega \varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{v})\, d\Omega + \int_\Omega q\, \nabla \cdot \boldsymbol{u}\, d\Omega - \int_\Omega p\, \nabla \cdot \boldsymbol{v}\, d\Omega, \\ l(\boldsymbol{V}) &= \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega. \end{aligned}$$

Let us introduce for each subdomain $i = 1, 2$ the following bilinear and linear forms

$$\begin{aligned} a_i(\boldsymbol{U}, \boldsymbol{V}) &= 2\nu \int_{\Omega_i} \varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{v})\, d\Omega + \int_{\Omega_i} q\, \nabla \cdot \boldsymbol{u}\, d\Omega - \int_{\Omega_i} p\, \nabla \cdot \boldsymbol{v}\, d\Omega, \\ l_i(\boldsymbol{V}) &= \int_{\Omega_i} \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega. \end{aligned}$$

as well as the following functional spaces

$$\begin{aligned} V_1^0 &= H_0^1(\Omega_1)^{n_d}, \\ V_i &= \{\boldsymbol{v} \in H^1(\Omega_i)^{n_d} \,|\, \boldsymbol{v}_{|\partial\Omega \cap \partial\Omega_i} = \boldsymbol{0}\}, \\ Q_i &= L^2(\Omega_i), \qquad \text{for } i = 1, 2, \\ P_1 &= \{p \in L^2(\Omega_1) \,|\, \int_{\Omega_1} p\, d\Omega = 0\}, \\ P_2 &= L^2(\Omega_2). \end{aligned}$$

The domain decomposition method presented here is a generalization of the classical Dirichlet/Neumann method [90, 89, 58] applied to overlapping subdomains. The algorithm reads: given $\boldsymbol{u}_2^0 \in V_2$, for each $k \geq 0$, find $\boldsymbol{U}_1^{k+1} \in V_1 \times P_1$ and $\boldsymbol{U}_2^{k+1} \in V_2 \times P_2$ such that

$$\begin{cases} a_1(\boldsymbol{U}_1^{k+1}, \boldsymbol{V}_1) = l_1(\boldsymbol{V}_1) & \forall\, \boldsymbol{V}_1 \in V_1^0 \times P_1, \\ \boldsymbol{u}_1^{k+1} = \boldsymbol{u}_2^k & \text{on } \Gamma_b, \\ a_2(\boldsymbol{U}_2^{k+1}, \boldsymbol{V}_2) = l_2(\boldsymbol{V}_2) + \langle \boldsymbol{\sigma}_1^{k+1} \cdot \boldsymbol{n}_2, \boldsymbol{v}_2 \rangle_{\Gamma_a} & \forall\, \boldsymbol{V}_2 \in V_2 \times P_2, \end{cases} \tag{4.72}$$

where the stress tensor is for $i = 1, 2$

$$\boldsymbol{\sigma}_i^{k+1} = -p_i^{k+1} \boldsymbol{I} + 2\nu\varepsilon(\boldsymbol{u}_i^{k+1}).$$

The reason for choosing $\boldsymbol{u}_2^0 \in V_2^0$ is due to the need of satisfying the incompressibility constraint in subdomain 1, at least at the first iteration. In the case of the approximate problem (for example

using finite elements) and if during the iterative process it happens that

$$\int_{\Gamma_b} \boldsymbol{u}_2^k \cdot \boldsymbol{n}\, d\Gamma \neq 0,$$

then the incompressibility constraint is violated and a conservative algorithm must be used. This point will be discussed in Section 5.4.

Latter iterative scheme may need relaxation of one (or both) of the transmission conditions. By introducing two relaxation parameters $\theta_1 > 0$ and $\theta_2 > 0$, we substitute Equations $(4.72)_2$ and $(4.72)_3$ by the following updates

$$\begin{aligned} \boldsymbol{u}_1^{k+1} &= \theta_1 \boldsymbol{u}_2^k + (1-\theta_1)\boldsymbol{u}_1^k, \\ a_2(\boldsymbol{U}_2^{k+1}, \boldsymbol{V}_2) &= l_2(\boldsymbol{V}_2) + \langle [\theta_2 \boldsymbol{\sigma}_1^{k+1} + (1-\theta_2)\boldsymbol{\sigma}_2^k] \cdot \boldsymbol{n}_2, v_2 \rangle_{\Gamma_a}. \end{aligned}$$

with $\boldsymbol{u}_2^0 = \boldsymbol{0}$ on Γ_b so that $\boldsymbol{u}_1^1 = \boldsymbol{0}$ on Γ_b. This iteration-by-subdomain method is clearly of Dirichlet/Neumann type.

Finally we consider the following stationary Navier-Stokes problem

$$\begin{aligned} -2\nu \nabla \cdot \varepsilon(\boldsymbol{u}) + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p &= \boldsymbol{f} \qquad \text{in } \Omega, \\ \nabla \cdot \boldsymbol{u} &= 0 \qquad \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{0} \qquad \text{on } \partial\Omega. \end{aligned}$$

Applying the Picard linearization scheme to linearize the convective term, we obtain the following weak formulation. Given an initial guess $\boldsymbol{u}^0 \in V^0$, find $\boldsymbol{U} \in V^0 \times P$ for $m \geq 0$, such that

$$a^m(\boldsymbol{U}^{m+1}, \boldsymbol{V}) = l(\boldsymbol{V}) \qquad \forall\, \boldsymbol{V} \in V^0 \times Q$$

until convergence, where

$$\begin{aligned} a^m(\boldsymbol{U}^{m+1}, \boldsymbol{V}) &= 2\nu \int_\Omega \varepsilon(\boldsymbol{u}^{m+1}) : \varepsilon(\boldsymbol{v})\, d\Omega + \int_\Omega [(\boldsymbol{u}^m \cdot \nabla)\boldsymbol{u}^{m+1}] \cdot \boldsymbol{v}\, d\Omega \\ &\quad + \int_\Omega q\, \nabla \cdot \boldsymbol{u}^{m+1}\, d\Omega - \int_\Omega p^{m+1}\, \nabla \cdot \boldsymbol{v}\, d\Omega, \\ l(\boldsymbol{V}) &= \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, d\Omega. \end{aligned}$$

We now apply the iteration-by-subdomain method to this problem. The algorithm reads as follows: given $\boldsymbol{u}_1^{m+1,0} = \boldsymbol{u}_1^m \in V_1$ and $\boldsymbol{u}_2^{m+1,0} = \boldsymbol{u}_2^m \in V_2$, for each $m \geq 0$, find until convergence $\boldsymbol{U}_1^{m+1,k+1} \in V_1 \times P_1$ and $\boldsymbol{U}_2^{m+1,k+1} \in V_2 \times P_2$ for $k = 0, 1, 2, \ldots$ such that

$$\begin{cases} a_1^m(\boldsymbol{U}_1^{m+1,k+1}, \boldsymbol{V}_1) = l_1(\boldsymbol{V}_1) & \forall\, \boldsymbol{V}_1 \in V_1^0 \times P_1, \\ \boldsymbol{u}_1^{m+1,k+1} = \boldsymbol{u}_2^{m+1,k} & \text{on } \Gamma_b, \\ a_2^m(\boldsymbol{U}_2^{m+1,k+1}, \boldsymbol{V}_2) = l_2(\boldsymbol{V}_2) + \langle \boldsymbol{\sigma}_1^{m+1,k+1} \cdot \boldsymbol{n}_2, v_2 \rangle_{\Gamma_a} & \forall\, \boldsymbol{V}_2 \in V_2 \times P_2. \end{cases} \tag{4.73}$$

where the bilinear form is for $i = 1, 2$

$$a_i^m(\boldsymbol{U}^{m+1,k+1}, \boldsymbol{V}) = 2\nu \int_{\Omega_i} \varepsilon(\boldsymbol{u}^{m+1,k+1}) : \varepsilon(\boldsymbol{v})\, d\Omega + \int_{\Omega_i} [(\boldsymbol{u}^m \cdot \nabla)\boldsymbol{u}^{m+1,k+1}] \cdot \boldsymbol{v}\, d\Omega$$
$$+ \int_{\Omega_i} q\, \nabla \cdot \boldsymbol{u}^{m+1,k+1}\, d\Omega - \int_{\Omega_i} p^{m+1,k+1}\, \nabla \cdot \boldsymbol{v}\, d\Omega,$$

and the stress tensor is for $i = 1, 2$

$$\boldsymbol{\sigma}_i^{m+1,k+1} = -p_i^{m+1,k+1}\boldsymbol{I} + 2\nu\varepsilon(\boldsymbol{u}_i^{m+1,k+1}).$$

This iterative procedure is an overlapping Dirichlet/Neumann DD, which can be relaxed as in the case of the Stokes problem. Note that integrating by parts the convective term, we would obtain an overlapping Dirichlet/Robin type coupling. The Dirichlet/Robin coupling will be considered briefly in Chapter 6, by integrating one-half of the convective term, as was done for the ADR equations in Section 1.1.3. In this iterative procedure, we have chosen to nest the linearization and DD iterative loops, the inner loop being the DD one. Letting $\boldsymbol{u}^m = \boldsymbol{u}^{m+1,k}$ in the convective term, we couple both loops. In the discrete finite element problem, it is computationally preferable to choose the DD loop as the inner loop; if it were the outer loop, the matrix of the discrete system would have to be computed at each iteration. All the details concerning the implementation of this algorithm and the results are presented in the next chapter.

Chapter 5

Implementation aspects

In this chapter, we derive a possible finite element implementation of two overlapping domain decomposition methods, namely the classical Schwarz method and an overlapping Dirichlet/Neumann method, with particular attention on the latter. The chapter is organized like a recipe, with each section constituting a new ingredient. The final dishes are two conservative Chimera methods, based on Dirichlet/Dirichlet and Dirichlet/Neumann couplings and able to deal with moving subdomains.

We first identify the transmission conditions from the alternative formulations of the DD derived in the last chapter. Then we set an iteration-by-subdomain method applied to the solution of the ADR and Navier-Stokes based on a Master/Slave strategy. We briefly describe an element search algorithm, which consists in looking for the host elements (in the underlying mesh) of the nodes involved in the iterative process. This operation is one of the first operations to be performed by the Master. At this stage we are ready to interpolate the transmission conditions. We present the interpolation of the Dirichlet data and two interpolation schemes for the Neumann (or Robin) data involved in the natural transmission conditions. We then explain the need for using a conservative interpolation and present two algorithms: an interface constraining and a conservative interpolator. In order to be able to deal with complex geometry, we introduce a Chimera method, using all the ingredients presented previously. Finally, the domain decomposition method is applied to moving subdomains by the way of tensorial transformations.

5.1 Iteration-by-subdomain algorithm

5.1.1 Introduction

The domain decomposition methods presented in the previous chapter have been applied at the algebraic level. According to a geometrical partitioning of the computational subdomain, we set up a Richardson procedure to solve the Schur complement system for the interface unknowns. In this chapter, we want to take advantage of the variational equivalent of the DD method to define a geometrical DD algorithm based on the iterative updates of the boundary conditions on the subdomains. More precisely, we want to use the alternative formulation given by Equation $(4.16)_{1-3}$ for the scalar ADR equation and its extension to the solution of the Navier-Stokes equations $(4.73)_{1-3}$. The solution of a transient problem is straightforward as it consists in applying the DD algorithm at each time step.

In the previous chapter, we studied three different types of transmission conditions that we can

deal with, at the continuous and variational level, namely:

- the Dirichlet condition, corresponding to the essential boundary condition of the weak formulations;
- the Neumann and Robin conditions, both corresponding to the natural boundary condition of the weak formulations.

We now go to the discrete level and assume that we have a precise way to compute all the transmission conditions introduced previously, although we have already commented in Remark 4.8 that the use of the alternative formulation is not possible at the discrete level. This issue will be treated in Section 5.3.2.

5.1.2 A Master/Slave-coupling algorithm

In this section, we will treat the domain decomposition algorithm as generally as possible, so that we allow for any type of coupling between subdomains, namely i and j, with transmission conditions of Dirichlet, Neumann and Robin types, involving overlapping and non-overlapping meshes. For the sake of clarity, we assume that the subdomains are steady; the case of moving subdomains will be treated in Section 5.6.2. The iterative domain decomposition algorithm consists of three steps: the pre-process, the process and the post-process.

- Pre-process. The pre-process consists in dividing the computational domain into overlapping and/or non-overlapping subdomains. The interface of a subdomain with another is defined as the part of the boundary of the former lying inside the latter. Along with the geometrical coupling of the subdmomains, some coupling variables must be carefully chosen in order to obtain a global solution from the local solutions on each subdomain. This global solution will be obtained iteratively, by exchanging variables between subdomains, according to some transmission conditions. The nodes involved in the transmission process are called the *interface nodes* or *transmission nodes*.
- Process. The control of the iterative process is performed by a master code. Communication between the master code and the slave codes (ADR or Navier-Stokes solver on each subdomain) can be achieved by any of the communication libraries like PVM or MPI. Each subproblem runs on different processes of the ADR or Navier-Stokes solver. The master code controls the iterative process by imposing successive updates of the interface boundary conditions (transmission conditions), using the information of the others. The required operations of the master code are:
  - find the host elements of the interface nodes in the adjacent subdomain;
  - interpolate the variables from one subdomain to another;
  - update of the boundary condition of each one;
  - pass the data (the new boundary condition) back and forth to the slaves (the processes of the finite element code).
- Post-process. Eventually, the post-process defines the global solution. For example, in the case of overlapping grids, one has to define the solution in the regions in common.

We now describe the specific tasks to be carried out by the Master and the Slaves. Within a standard implicit Navier-Stokes solver, the DD algorithm loop fits within a multi-loop algorithm as shown by Algorithm 5.3. Each loop is controlled by a tolerance and a maximum number of

Algorithm 5.3 Slave's point of view

```
for time steps do
  for linearization steps do
    for DD steps do
      Import transmission condition update from Master
      for solver steps do
        Solve Algebraic system
      end for
    end for
    Export new solution to Master
  end for
end for
```

steps. For the case of time iterations, the tolerance is needed only when a steady state is required. With respect to a classical solver, as shown by Algorithm 5.3, the DD algorithm is just an additive loop that can be coupled with the others. For example, it may be convenient to couple it with the linearization loop (Newton or Picard linearization); in all the examples presented in this work, this is the technique employed. When dealing with explicit codes, the DD loop can also be coupled with the time loop.

As mentioned earlier, the master code is in charge of controlling the iterative process and performing all the necessary operations to leave the slaves unworried. Let us assume we want to couple n_s subdomains. We denote by Γ_{ij} the interface of subdomain i with subdomain j. The n_s slave processes are distributed via a multicoloring technique: each subdomain is assigned a color color(i) so that subdomains of the same color have no common interface. The colors are ordered from 1 to n_c, where n_c is the total number of colors used. Subdomains of smaller color are run first.

The algorithm as seen from the master code is shown in Algorithm 5.4. The first task is to find the host elements of all the interface nodes to enable further interpolation of the transmission conditions. When the subdomains are steady, this operation must be performed only once, as a pre-process work. The search technique used in this work will be described in detail in Section 5.2. The stopping criterion is based on some norm of the interface unknown changes between two successive iterates, k and $k+1$. We define the *interface* L_2 *residual* of variable f as the following quantity

$$\left[\frac{\sum_{i=1}^{n_s} \sum_j \left(\frac{1}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} (f_i^{k+1} - f_i^k)^2 d\Gamma \right)}{\sum_{i=1}^{n_s} \sum_j \left(\frac{1}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} (f_i^{k+1})^2 d\Gamma \right)} \right]^{1/2},$$

where $|\Gamma_{ij}|$ is the measure of Γ_{ij} and f_i^k is the approximate solution on Γ_{ij} obtained in solving subdomain i at iteration k. The sum in j is extended to all subdomains connected to i.

5.2 Search algorithm

We have seen that once the domain decomposition has been performed, one of the first tasks of the Master is to find host elements for all the interface nodes. In addition, if the subdomains

Algorithm 5.4 Master's point of view

```
Impose initial conditions
for time steps do
  Set iteration number k = 0
  Find the host elements of all the Γ_ij interface nodes of subdomain i in subdomain j
  while stopping criterion not reached do
    for color=1 to n_c do
      Export transmission conditions to subdomains i such that color(i)=color
      Run subdomains i in parallel
      Import solutions from subdomains i
      for subdomains j connected to subdomain i do
        Interpolate and compute transmission conditions for subdomains j
      end for
    end for
    k = k + 1
    Check convergence of the DD scheme
  end while
end for
```

are moving, the element search strategy (ESS) must be performed at each time step. The ESS used in this work is based on a quad-tree strategy (in 2D, and oct-tree in 3D). The strategy can be decomposed in two steps, the pre-process (which constructs the tree-like structure) and the process (*range searching*). In the pre-process, the computational domain is first embedded in a box, taking the minimum and maximum of the node coordinates to define its corners. The algorithm recursively partitions the box(es) into smaller boxes, until each box contains less than a prescribed number of nodes. A box is divided into 4 boxes in two dimensions (quad-tree) and into 8 boxes in three dimensions (oct-tree). If a box is not divided further because it contains too few nodes, we start filling it with elements. To get the list of elements located inside a box, the node/element connectivity is used; all the elements connected to the node of a box belong to this box. Note that this criterion does not require any calculation of the intersections of the faces of elements and boxes. Figure 5.1 shows a two-dimensional example of a quad-tree subdivision of a mesh of a NACA0012.

The process step is known as range searching. Knowing the coordinates $\boldsymbol{x}$ of the test point `ipoin`, we proceed down through the tree until we find a box containing elements where the test point must be. Now we perform a loop over the elements `jelem` belonging to the box. Let n_n be the number of nodes of each element and $\boldsymbol{x}^{(r)}$ the node coordinates for $r = 1, \ldots, n_\mathrm{n}$. We define $N^{(r)}_\mathrm{local}$ as the local shape function of node r and $\boldsymbol{\xi}$ the local coordinate of $\boldsymbol{x}$. We solve the following equation for $\boldsymbol{\xi}$ using a Newton-Raphson scheme

$$\boldsymbol{x} = \sum_{r=1}^{n_\mathrm{n}} N^{(r)}_\mathrm{local}(\boldsymbol{\xi}) \boldsymbol{x}^{(r)}$$

If $\boldsymbol{\xi}$ is inside the local domain (master element), then the test point belongs to `jelem`.

The condition for an element to belong to a box (based on the connectivity) can seem restrictive; in fact, an element can intersect a box without having any node in it. Nevertheless, it has proved to be sufficient for most cases and the search almost never fails. In case a point has no host element, a new search is performed using a less restrictive method.

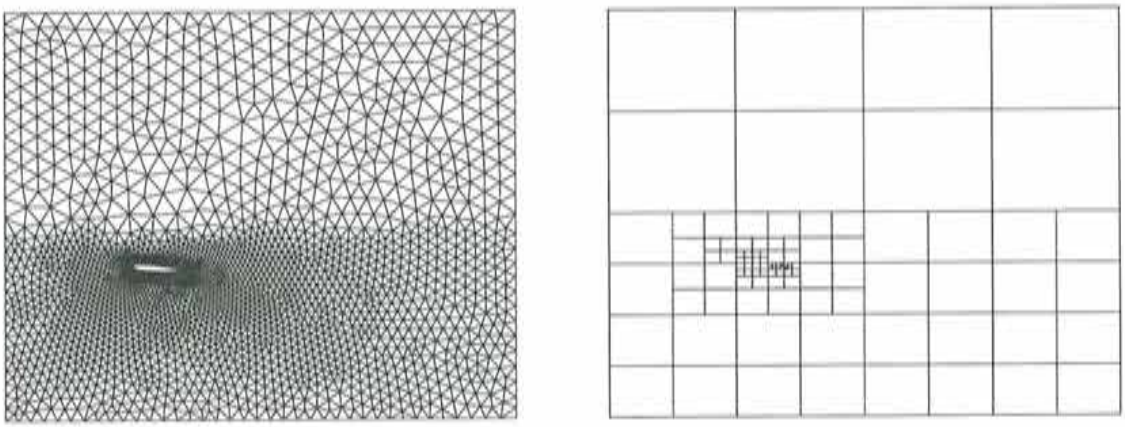


Figure 5.1: A quad-tree division of a mesh (NACA0012). (Left) Mesh. (Right) Quad-tree structure.

5.3 Interpolation of the transmission conditions

Up to now we have studied how the transmission conditions are passed from one subdomain to another to set up the iterative algorithm. We now study the interpolation technique performed at the finite element level. In the following, subscripts i and j refer to values computed in subdomains i and j, respectively, and we assume we want to update the solution of subdomain i knowing the solution of subdomain j. Remember that Γ_{ij} is the interface of subdomain i in the adjacent subdomain j. The notation we use in the following was introduced at the beginning of Section 4.5.3.

5.3.1 Interpolation of Dirichlet data

The primary variables considered here can be the unknowns of the ADR equations or all the velocity components in the case of the Navier-Stokes equations. For the sake of clarity, we drop the h subscript to identify finite element solutions. Although we are still going to deal with continuous solutions, the distinction will always be obvious.

Let n_t be the number of transmission nodes on Γ_{ij}. We define $\mathbf{u}_{i|\Gamma_{ij}}$ as the vector containing the n_t nodal unknowns, of coordinates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_t}$, that have to be updated. The interpolation of primary variables is carried out using the classical Lagrange interpolation. We define I^{ij} as the continuous interpolation operator from mesh j to mesh i, and $\mathbf{I}^{ij}$ its discrete counterpart, made of global Lagrange interpolation functions. The operator $\mathbf{I}^{ij}$ is given by

$$\mathbf{I}^{ij} = \begin{bmatrix} \mathbf{N}^t(\boldsymbol{x}_1) \\ \vdots \\ \mathbf{N}^t(\boldsymbol{x}_{n_t}) \end{bmatrix},$$

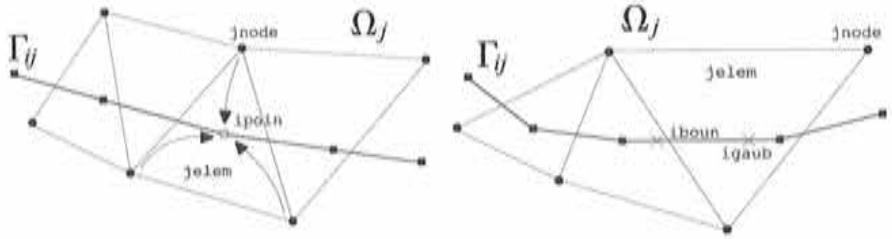


Figure 5.2: Interpolation. (Left) Dirichlet data. (Right) Neumann/Robin data.

and we have

$$\mathbf{u}_{i|\Gamma_{ij}} = \mathbf{I}^{ij}\mathbf{u}_j. \tag{5.1}$$

The last equation is only symbolic as in practice we work at the element level. In fact, from the element search strategy, we have identified the host elements `jelem` of each interface node `ipoin`, as sketched in Figure 5.2 (Left). Therefore, only the natural coordinates of the node `ipoin` inside `jelem` are needed to perform the interpolation of the primary variables.

Let us finally mention that this interpolation is "diffusive" as some information can be missed during the interpolation. For example, this may be the case when the mesh of subdomain i is coarser than the mesh of subdomain j; this point is known as *conservation* and will be treated in Section 5.4.

5.3.2 Interpolation of Neumann/Robin data

The Neumann or Robin data are involved in the natural transmission conditions. They are the flux in the case of the ADR equation and the traction in the case of the Navier-Stokes equations. In the last chapter, we mentioned in Remark 4.8 that the alternative formulation of the DD method that we use here cannot be directly extended to the discrete case. So what can we do? Through the following one-dimensional example we are going to explain where the key is for deriving a "correct" discrete alternative formulation.

Discrete normal derivatives

We propose to solve the following one-dimensional problem

$$\begin{cases} -\Delta u &= f & \forall\, x \in \Omega = (-1,1), \\ u &= 0 & \text{at } x = -1, 1. \end{cases} \tag{5.2}$$

with f a given function of x, and $\Delta u = d^2u/dx^2$. We divide the domain into two three disjoint subdomains $\Omega_3 = [-1,-a]$, $\Omega_4 = [-a,a]$ and $\Omega_5 = [a,1]$, with $a > 0$. These subdomains are partitioned into linear elements of constant length h. We define $\Omega_1 = \Omega_3 \cup \Omega_4 = (-1,a)$ and $\Omega_2 = \Omega_4 \cup \Omega_5 = (-a,1)$. As shown in Figure 5.3, we only number five nodes on both subdomains, around the position $x = -a$: three nodes on Ω_1, namely $(1,1)$, $(a,1)$ and $(2,1)$; two nodes on Ω_2, namely $(a,2)$ and $(2,2)$. The three elements we are going to refer to are element 1, with nodes $(1,1)$ and $(a,1)$, element 2, with nodes $(a,1)$ and $(2,1)$, and element 3, with nodes $(a,2)$ and $(2,2)$.

Problem (5.2) is solved using the overlapping Dirichlet/Neumann method, by imposing a Dirichlet transmission condition on the last node of Ω_1 at $x = a$, and a Neumann transmission condition

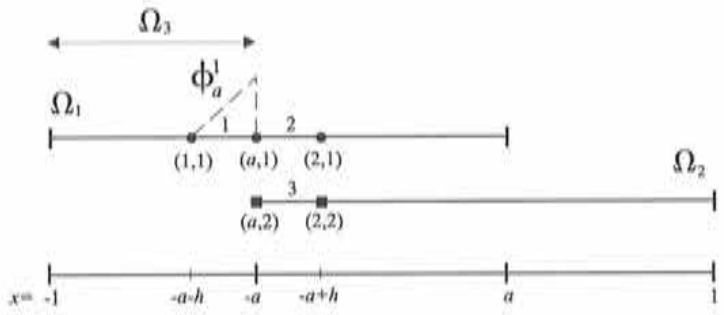


Figure 5.3: Overlapping subdmomains with coinciding nodes.

on Ω_2 at node $(a, 2)$, at $x = -a$.

In the following, a superscript without parenthesis refers to an elemental value while a superscript with parenthesis refers to a nodal value; subscript i for $i = 1, 2$ refers to values considered in subdomain i. In addition, let ϕ_a^1 be the shape function of node $(a, 1)$ evaluated in element 1, and set to zero elsewhere, and ϕ_a^3 be the shape function of node $(a, 2)$ evaluated in element 3, and set to zero elsewhere. According to the notation used for this example, Equation $(4.60)_4$ reads:

$$a_2(u_{2,h}, \phi_a^3) = \langle f, \phi_a^3 \rangle_{\Omega_2} + \langle f, \phi_a^1 \rangle_{\Omega_3} - a_3(u_{1,h}, \phi_a^1). \tag{5.3}$$

From this equation, we are going to derive the discrete counterpart of the Neumann condition $(4.16)_3$ of the 3-equation formulation. By integrating the last term of the last equation by parts, we obtain:

$$\langle f, \phi_a^1 \rangle_{\Omega_3} - a_3(u_{1,h}, \phi_a^1) = \int_{\Omega_3} f \phi_a^1 d\Omega + \int_{\Omega_3} \Delta u_{1,h}^1 \phi_a^1 d\Omega - \left. \frac{du_{1,h}}{dx} \right|_{x=-a}, \tag{5.4}$$

where the derivative has to be calculated in element 3. How do we go on and get rid of the first two terms of the right-hand side? Remember that at the continuous level, by using Equation (4.13) to substitute the Laplacian by minus the force term, we would be left only with the derivative on the right hand side. At the discrete level, however, this substitution is not straightforward, as the Laplacian must be understood in the sense of distributions. Within the elements we have, in general

$$\Delta u_{1,h}^1 \neq -f.$$

Indeed, when using linear interpolation the Laplacian is zero inside each element but it is undefined at $x = a$. Let us find the formal expression of the Laplacian; we introduce the Heaviside function $H(x)$:

$$H(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0. \end{cases}$$

By restricting ourselves to the region consisting of elements 1 and 2, the derivative of $u_{1,h}$ with respect to x at $x = -a$ can be expressed as a function of $H(x + a)$ as:

$$\frac{du_{1,h}}{dx} = \frac{du_{1,h}^1}{dx} + \left(\frac{du_{1,h}^2}{dx} - \frac{du_{1,h}^1}{dx} \right) H(x + a), \tag{5.5}$$

where $du^1_{1,h}/dx$ and $du^2_{1,h}/dx$ are the derivatives calculated in elements 1 and 2, respectively. They are:

$$\frac{du^1_{1,h}}{dx} = \frac{u^{(a,1)} - u^{(1,1)}}{h}, \tag{5.6}$$

$$\frac{du^2_{1,h}}{dx} = \frac{u^{(2,1)} - u^{(a,1)}}{h}. \tag{5.7}$$

We introduce the jump in velocity derivatives across node $(a, 1)$:

$$[\frac{du_h^{(a)}}{dx}] = \left(\frac{du^2_{1,h}}{dx} - \frac{du^1_{1,h}}{dx}\right).$$

By deriving Equation (5.5) with respect to x, we find the following expression for the Laplacian operator

$$\begin{aligned}\Delta u_{1,h} &= \frac{d}{dx}\left(\frac{du_{1,h}}{dx}\right) \\ &= \Delta u^1_{1,h} + (\Delta u^2_{1,h} - \Delta u^1_{1,h})H(x+a) + [\frac{du_h^{(a)}}{dx}]\delta(x+a).\end{aligned} \tag{5.8}$$

where, with obvious meaning, we have defined

$$\Delta u^1_{1,h} = \frac{d^2 u^1_{1,h}}{dx^2},$$

$$\Delta u^2_{1,h} = \frac{d^2 u^2_{1,h}}{dx^2}.$$

Note that in the case of linear interpolation, the first two terms of Equation (5.8) are identically zero; we have left them to be able to follow their trace along the following calculations. Now, we multiply Equation (5.8) by ϕ^1_a and integrate the result in Ω_3; we find

$$\begin{aligned}\int_{\Omega_3} \Delta u_{1,h}\phi^1_a d\Omega &= \int_{\Omega_3} \Delta u^1_{1,h}\phi^1_a d\Omega + \int_{\Omega_3} (\Delta u^2_{1,h} - \Delta u^1_{1,h})H(x+a)\phi^1_a d\Omega \\ &\quad + \int_{\Omega_3} [\frac{du_h^{(a)}}{dx}]\delta(x+a)\phi^1_a d\Omega \\ &= \int_{\Omega_3} \Delta u^1_{1,h}\phi^1_a d\Omega + \frac{1}{2}[\frac{du_h^{(a)}}{dx}].\end{aligned}$$

The left-hand side term of the last equation is precisely the Laplacian in the sense of distributions. We can therefore apply Equation (4.13) to substitute $\Delta u_{1,h}$ by $-f$ in the last equation and we obtain

$$\int_{\Omega_3} f\phi^1_a d\Omega + \int_{\Omega_3} \Delta u^1_{1,h}\phi^1_a d\Omega = -\frac{1}{2}[\frac{du_h^{(a)}}{dx}],$$

which, together with Equation (5.4) gives:

$$\langle f, \phi_a^1 \rangle_{\Omega_1} - a_1(u_{1,h}, \phi_a^1) = -\frac{du_{1,h}^1}{dx}\Big|_{x=-a} - \frac{1}{2}\left[\frac{du_h^{(a)}}{dx}\right].$$

Finally, the Neumann condition we have been looking for is simply obtained by substituting the last equation into Equation (5.3), and by using the values of the derivatives given by Equations (5.6) and (5.7):

$$a_2(u_{2,h}, \phi_a^3) = \langle f, \phi_a^3 \rangle_{\Omega_2} - \frac{u^{(2,1)} - u^{(1,1)}}{2h},$$

For this simple example, we conclude that the correct way to use the discrete counterpart of the 3-equation formulation $(4.16)_{1-3}$ is to compute the normal derivative at $x = -a$ using a *second order* centered scheme.

A one-dimensional example

As an illustration, let us consider a special case of the one-dimensional problem introduced previously. We want to solve the following problem: $(5.2)_{1-2}$ with $f = x^2 - 3x + 1$. We decompose Ω into two overlapping subdomains $\Omega_1 = (-1, a)$ and $\Omega_2 = (-a, 1)$ with $a = 0.2$ that we mesh with a constant mesh size h such that the nodes in the overlapping zone coincide. The subdomains are coupled using the overlapping Dirichlet/Neumann method. We impose a Dirichlet transmission condition on Ω_1 at $x = 0.2$ and a Neumann condition on Ω_2 at $x = -0.2$. We use the nomenclature shown in Figure 5.3.

We take advantage of the simple one-dimensional geometry to compute the Neumann condition using a finite difference scheme around the point $x = -0.2$. We approximate the derivative at $x = -0.2$ according to the following three finite difference schemes:

$$\frac{du_{1,h}}{dx}\Big|_{x=-0.2} \approx \begin{cases} \text{Left :} & \dfrac{u^{(a,1)} - u^{(1,1)}}{h}, \\ \text{Right :} & \dfrac{u^{(2,1)} - u^{(a,1)}}{h}, \\ \text{Centered :} & \dfrac{u^{(2,1)} - u^{(1,1)}}{2h}, \end{cases} \tag{5.9}$$

where the superscript denotes the node at which the solution is considered.

Figure 5.4 shows the results obtained on both subdomains using the three approximations as well as the solution obtained on one subdomain of the same mesh size $h = 0.01$. The disjoint D/N method is the non-overlapping Dirichlet/Neumann method computing the derivative using the Left scheme, as only information on the left of node $(a, 2)$ is available on subdomain 1. We observe that both the left and right finite difference schemes exhibit a strong jump in the the solution at the transmission node, whereas the centered scheme gives results much closer to those of the one-domain solution. We guess that this could be explained by the fact that the centered scheme is of second order in space while the other two are only of first order. This fact is actually confirmed by Figure 5.5 which shows the mesh convergence for three mesh sizes, where the error

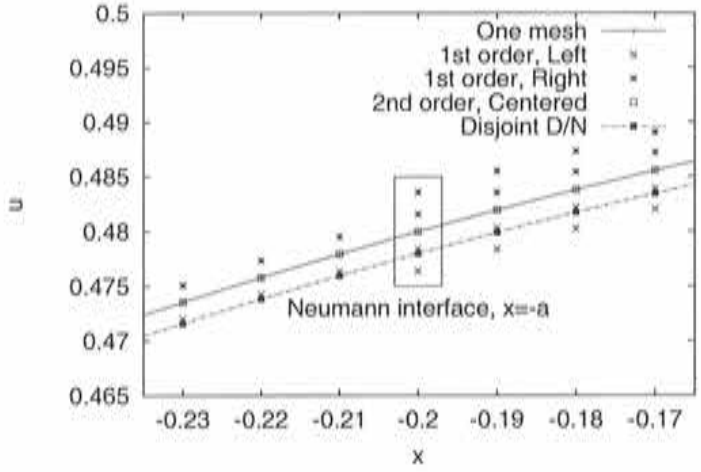


Figure 5.4: Poisson Equation. Solution using different approximation schemes of the derivative.

is defined as

$$\frac{1}{u_{\text{one}}} \max\left(|u^{(1,2)} - u_{\text{one}}|, |u^{(a,2)} - u_{\text{one}}|\right),$$

and u_{one} is the solution of the one-domain solution obtained at $x = -0.2$.

Although the discrete counterpart of the alternative formulation of the DD methods cannot be justified, we have in hand a numerical tool for solving the problem and conserving the space order of convergence of the finite element method. We now present two interpolation schemes of first and second order to compute general transmission conditions.

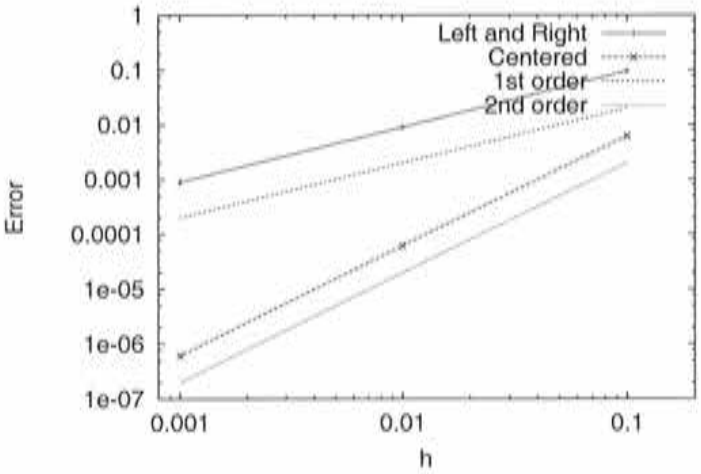


Figure 5.5: Poisson Equation. Mesh convergence of the approximation schemes of the derivative.

Classical interpolation

For the sake of clarity, we consider the Neumann transmission condition involved in the DD method for the ADR problem, by considering the simple 0-weak formulation. The discrete counterpart of the Neumann condition of the alternative formulation of the DD method (Equation $(4.16)_3$) consists in calculating the following contour integral on the interface Γ_{ij}:

$$\int_{\Gamma_{ij}} \varepsilon \frac{\partial u_j}{\partial n} v_i \, d\Gamma = \int_{\Gamma_{ij}} \varepsilon \nabla u_j \cdot \boldsymbol{n}, \, v_i \, d\Gamma \tag{5.10}$$

where u_j is known from the previous solution on Ω_j. We now present the first order interpolation scheme for the first order derivatives of u_j involved in the contour integral. The extension to the calculation of the stress tensor involved in the DD method for the Navier-Stokes equations is straightforward.

We note that the flux is needed at the integration points of the boundaries in order to perform the numerical integration of the Neumann condition. Let us consider the element boundary `iboun` and define `igaub` as an integration point on this boundary. Once the host element `jelem` of `igaub` in subdomain j is found, we obtain the first order interpolation by direct interpolation of the derivatives from the node to the boundary integration points. The strategy to compute the force term (5.10) is shown in Algorithm 5.5.

Algorithm 5.5 First order interpolation

for all boundary elements `iboun` **do**
  for all integration points `igaub` **do**
    Find host element `jelem` of `igaub`
    Interpolate derivatives ∇u_j from nodes `inode` to integration point `igaub`
    Calculate outward unit normal $\boldsymbol{n}$ at `igaub`
    Calculate test function v_i at `igaub`
    Calculate product $(\varepsilon \nabla u_j \cdot \boldsymbol{n}) v_i$ at `igaub` and multiply the result by the weight of the numerical integration
    Assemble result
  end for
end for

Note that if the subdomains are steady, the host elements `jelem` of the boundary integration points `igaub` must be calculated only once.

Least-square interpolation

In the one dimensional example presented at the beginning of this section, we showed that using a centered finite difference scheme we could obtain a second order convergence in space for the Dirichlet/Neumann algorithm. From the notation used in Equation (5.9), we observe that the centered scheme uses the unknowns of the background mesh on both sides of the node (2), in the sense that

$$\frac{u^{(2,1)} - u^{(1,1)}}{2h} = \frac{1}{2} \left(\frac{u^{(a,1)} - u^{(1,1)}}{h} + \frac{u^{(2,1)} - u^{(a,1)}}{h} \right)$$

One way of applying this result to a general finite element problem is to perform a least-squares smoothing to compute the derivatives of the unknown at the nodes of subdomain j. By doing so,

the values of the derivatives at a node `jnode` will depend on the derivatives calculated on all its neighboring element, i.e. on the values of the function at all the nodes of the elements connected to `jnode`. Remember that the first order interpolation only considers the derivative inside the host element of the boundary integration point. The previous example showed that we could obtain a second order convergence in one dimension: we expect the method to be presented here for two and three dimensions will be of second order as well.

The least-squares smoothing used here is standard. Let ϕ_d be a discontinuous function across elements, in our case known at the element integration points. Then, the continuous function ϕ_c is computed by minimizing the following functional:

$$||\phi_\mathrm{d} - \phi_\mathrm{c}||^2 = \int_\Omega (\phi_\mathrm{d} - \phi_\mathrm{c})^2 d\Omega.$$

The function ϕ_d represents any component of ∇u_j calculated at the integration points. Let $\mathbf{N}_j$ be the vector of Lagrange interpolation functions of subdomain j and $\mathbf{\Phi}$ the nodal vector which components are the nodal values $\phi_c^{(r)}$ of ϕ_c at this point such that

$$\mathbf{\Phi} = [\phi_c^{(1)}, \cdots, \phi_c^{(n_{\mathrm{p}_j})}]^t,$$

Re-expressed in a matrix form, our problem consists in minimizing

$$L(\mathbf{\Phi}) = \int_\Omega (\phi_\mathrm{d} - \mathbf{N}_j^t \mathbf{\Phi})^2 d\Omega.$$

The solution of this minimization problem consists in finding a stationary point of L, which satisfies

$$\frac{dL(\mathbf{\Phi})}{d\mathbf{\Phi}} = \int_\Omega 2\mathbf{N}_j (\phi_\mathrm{d} - \mathbf{N}_j^t \mathbf{\Phi})\, d\Omega = \mathbf{0},$$

The latter equation leads to the following system:

$$\mathbf{M}\,\mathbf{\Phi} = \mathbf{r}, \tag{5.11}$$

where the mass matrix $\mathbf{M}$ and the right-hand side $\mathbf{r}$ are given by

$$\mathbf{M} = \int_{\Omega_j} \mathbf{N}_j\, \mathbf{N}_j^t\, d\Omega,$$

$$\mathbf{r} = \int_{\Omega_j} \phi_\mathrm{d}\, \mathbf{N}_j\, d\Omega.$$

Equation (5.11) can be solved efficiently using a closed quadrature rule to compute the coefficients of the mass matrix. A closed quadrature rule is an integration rule for which the integration points are located on the nodes. By noting that $N^{(r)}(\boldsymbol{x}_s) = \delta_{rs}$, the resulting mass matrix is diagonal and $\mathbf{M}$ is trivially inverted.

Once the derivatives are obtained at the nodes of the background mesh, we proceed as in the case of the first order interpolation, as shown in Algorithm 5.6.

Algorithm 5.6 Least-square interpolation

```
Perform least-squares smoothing for the derivatives
for all boundary elements iboun do
  for all integration points igaub do
    Interpolate derivatives ∇u_j from nodes jnode to integration point igaub
    Calculate outward unit normal n at igaub
    Calculate test function v_i at igaub
    Calculate product (ε∇u_j · n)v_i at igaub and multiply the result by the weight of the
    numerical integration
    Assemble result
  end for
end for
```

Remark 5.1. In the case of non-overlapping subdomains, for which we only have information on one side of the underlying mesh, the classical and least-square interpolations are equivalent and are both of first order. The overlapping seems therefore necessary to obtain a second order convergence.

Remark 5.2. The first order strategy leads to discontinuous derivatives across the elements, since they are directly obtained from those inside the elements. On the other hand, the least-squares smoothing used in the second order scheme leads to continuous derivatives.

Navier-Stokes equations: note on the pressure

Up to now, we have derived algorithms for calculating the derivatives of a function at the integration points of the interface. We are therefore able to compute the velocity strain rates present in the Neumann transmission condition, as given by the discrete counterpart of Equation $(4.72)_3$. In order to complete the approximation of the transmission condition, we need the pressure. On the one hand, when using continuous pressure spaces, the pressure is interpolated at the boundary integration points in a classical way, i.e. like the velocity using Equation (5.1). On the other hand, when using discontinuous pressure spaces the pressure is first smoothed using the least-squares smoothing described previously, before being interpolated from the nodes to the boundary integration points.

Example

We present a simple example of application of the classical and least-square interpolations to the solution of the Stokes equations. We solve the following system

$$\begin{aligned} -\Delta \boldsymbol{u} + 2\boldsymbol{\omega} \times \boldsymbol{u} + \nabla p &= \boldsymbol{f}, \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned} \tag{5.12}$$

in a two-dimensional domain Ω made of two concentric circles, as shown in Figure (5.6), and where the force $\boldsymbol{f}$ is chosen so that the exact solution of the problem is

$$\begin{aligned} u_e &= 2y[r - 1/2], \\ v_e &= -2x[r - 1/2], \\ p_e &= r. \end{aligned}$$

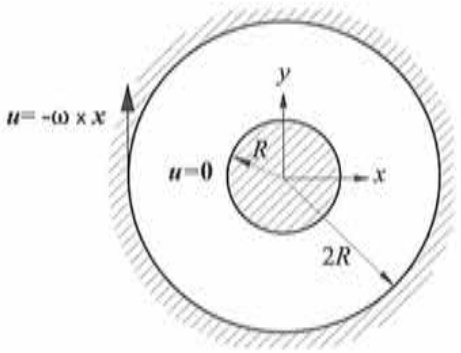


Figure 5.6: Concentric circles. Geometry and boundary conditions.

with $\boldsymbol{u}_e = [u_e, v_e]^t$ and $r = (x^2 + y^2)^{1/2}$. We construct subdomain 1 of inner diameter 0.5 and outer radius 1, and subdomain 2 of inner radius 0.5 and outer radius 2. In order to test the interpolation technique of the secondary variables, we first solve the problem in subdomain 2 using exact Dirichlet boundary conditions on its boundary, and then update the Neumann condition on the outer circle of subdmomain 1. The solution is a radial Poiseuille-like flow and does not depend on the rotation, although we are going to show that the error of the finite element solution does. The rotation is first chosen to be sufficiently small as we want to avoid any possible instability due to the Coriolis term, so we take $\boldsymbol{\omega} = |\omega|[0, 0, 1]^t$ with $|\omega| = 0.1$. Figure (5.7) (Top) (Left) shows the rate of convergence of the error in subdomain 1 computed for the classical and least-square interpolations, and confirms the results obtained with the one-dimensional example presented at the beginning of the Section. Now what happens if we increase the rotation? Let us denote $\boldsymbol{u}$ and p as an approximate solution, for example a finite element solution or the solution at a certain iteration of a DD method. From the BB condition, and using the notation introduced in Section 1.2.5, we know that

$$||q||_{P,\Omega} \leq \frac{1}{\beta} \sup_{\boldsymbol{v} \in V} \frac{(q, \nabla \cdot \boldsymbol{v})}{||\boldsymbol{v}||_{V,\Omega}} \qquad \forall\, q \in Q.$$

Taking $q = p - p_e$, and from Equation (5.12) knowing that

$$(p - p_e, \nabla \cdot \boldsymbol{v}) = (\nabla(\boldsymbol{u} - \boldsymbol{u}_e), \nabla \boldsymbol{v}) - (2\boldsymbol{\omega} \times (\boldsymbol{u} - \boldsymbol{u}_e), \boldsymbol{v}),$$

we obtain

$$\begin{aligned} ||p - p_e||_{P,\Omega} &\leq \frac{1}{\beta} \sup_{\boldsymbol{v} \in V} \frac{(\nabla(\boldsymbol{u} - \boldsymbol{u}_e), \nabla \boldsymbol{v}) + (2\boldsymbol{\omega} \times (\boldsymbol{u} - \boldsymbol{u}_e), \boldsymbol{v})}{||\boldsymbol{v}||_{V,\Omega}} \\ &\leq \frac{1}{\beta} \left(\sup_{\boldsymbol{v} \in V} \frac{(\nabla(\boldsymbol{u} - \boldsymbol{u}_e), \nabla \boldsymbol{v})}{||\boldsymbol{v}||_{V,\Omega}} + 2|\boldsymbol{\omega}|\, ||\boldsymbol{u} - \boldsymbol{u}_e||_{-1,\Omega} \right). \end{aligned}$$

When $\boldsymbol{\omega}$ is high, the second term dominates, i.e.

$$||p - p_e||_{P,\Omega} \sim \frac{2|\boldsymbol{\omega}|}{\beta} ||\boldsymbol{u} - \boldsymbol{u}_e||_{-1,\Omega}, \tag{5.13}$$

so we expect that the pressure becomes out of control when we have an error in the velocity. Hopefully, when passing Neumann transmission conditions, the error in pressure remains and does

not affect the velocity. The mesh convergence for $|\omega| = 10^4$ is shown in Figure 5.7 (Bot.) (Left). We see that the pressure convergence is entirely dominated by the rotation term. In addition, Figure 5.7 (Top) (Right) gives the dependence of the errors with respect to $|\omega|$. The velocity is not negatively affected by the rotation while the error in pressure goes linearly with $|\omega|$, as predicted by Equation (5.13). Finally, Figure 5.7 (Bot.) (Right) gives the mesh convergence of the least-square interpolation for the pressure and velocity. We observe that the velocity error for $|\omega| = 10^4$ is always below the velocity error for $|\omega| = 10^{-1}$ for the range of mesh sizes studied, while that of the pressure is four orders of magnitude greater.

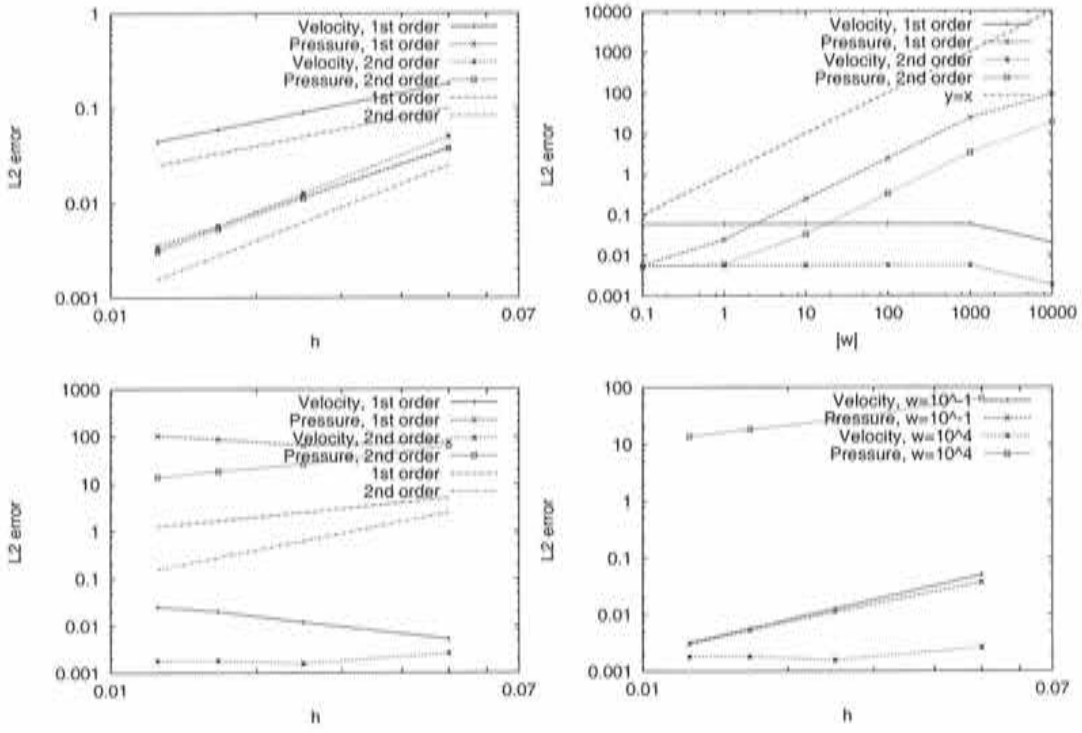


Figure 5.7: Concentric circles. L_2 errors. (Top) (Left) $|\omega| = 10^{-1}$. (Top) (Right) $h = 1/60$. (Bot.) (Left) $|\omega| = 10^4$. (Bot.) (Right) Least-square interpolation.

5.4 Conservation

This section addresses an important aspect of the implementation of DD method: *conservation*. We first present the shortcomings of the classical interpolation. Then we propose an interface constraining method to enable us to have control on some well-chosen conservation properties. The method is illustrated by three examples. Then, we present an interpolation operator. This operator is tested for the Dirichlet/Neumann method as applied to a simple cavity flow.

5.4.1 Classical Interpolation

The interpolation strategy of a variable, as given by Equation (5.1), is simple and easy to implement but it is non-conservative. Figure 5.8 illustrates the importance of using a conservative algorithm when interpolating a variable from a fine grid to a coarse grid; note that the same occurs when interpolating between grids whose nodes do not coincide. In this simple illustration, although the

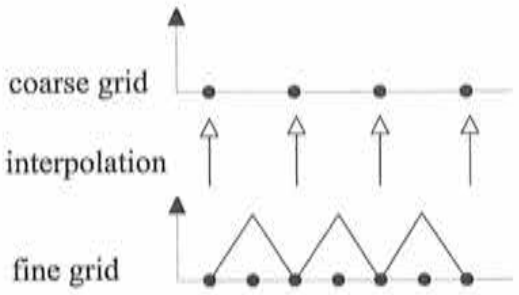


Figure 5.8: Interpolation from a fine grid to a coarse grid.

continuity of the interpolated variable is guaranteed on each node of the coarse grid, the global information is not necessarily well captured. High frequency modes may be filtered out if the variable exhibits strong variations along the interpolation domain. Several techniques are available to overcome the lack of conservation of classical interpolations; e.g., Cebral and Löhner [100] apply a weighted residual method to conserve the force when solving coupled fluid-structure problems. Different grid sizes are not the only reason for applying a conservative scheme; it can also be needed if the interpolated data are not compatible with the numerical formulation, as will be illustrated with the second and third numerical examples of this section. The interface constraining presented here enables one to make a compromise between the continuity of the variable (if the nodes coincide) and the global information it carries.

At the end of this section, we also devise a "conservative" operator to interpolate the variables from a fine mesh to a coarse mesh.

5.4.2 Constrained transmission conditions

The idea of the interface constraining technique [101, 102] is to impose the continuity of a variable in a weak sense via a classical interpolation and to relax it by the conservation of a global quantity. Let us assume we want to update the variable u_i of subdomain i knowing the variable u_j of the adjacent subdomain j. Remember that we denoted I^{ij} the continuous interpolation operator from subdomain j to subdomain i. We propose to find u_i from u_j by solving the following system:

$$\begin{aligned} &\text{minimize} && \int_{\Gamma_{ij}} [u_i - I^{ij}(u_j)]^2 \, \mathrm{d}\Gamma, \\ &\text{under the constraint} && f(u_i) = 0, \end{aligned}$$

where $f(u_i)$ is a linear function of the unknown u_i that determines the quantity to be conserved. For example, one can conserve the flux of u_i across the interface by choosing

$$f(u_i) = \int_{\Gamma_{ij}} \nabla u_i \cdot \boldsymbol{n}_i \, d\Gamma - \int_{\Gamma_{ij}} \nabla u_j \cdot \boldsymbol{n}_i \, d\Gamma,$$

where n_i is the outward unit vector normal to subdomain i. Let $\mathbf{N}_i$ be the vector of classical Lagrange interpolation functions for the boundary elements. We can re-express the latter system in a matrix form as:

$$\begin{aligned} &\text{minimize} && \int_{\Gamma_{ij}} [\mathbf{N}_i^t \mathbf{u}_{i|\Gamma_{ij}} - \mathbf{N}_i^t \mathbf{I}^{ij} \mathbf{u}_j]^2 \, d\Gamma, \\ &\text{under the constraint} && \mathbf{r}^t \mathbf{u}_{i|\Gamma_{ij}} = r, \end{aligned}$$

where $\mathbf{r}$ and r are the vector and scalar representing the function f, respectively. This system can be solved by introducing the Lagrange multiplier λ of the constraint. The Lagrangian is:

$$L(\mathbf{u}_{i|\Gamma_{ij}}, \lambda) = \int_{\Gamma_{ij}} [\mathbf{N}_i^t \mathbf{u}_{i|\Gamma_{ij}} - \mathbf{N}_i^t \mathbf{I}^{ij} \mathbf{u}_j]^2 \, d\Gamma - \lambda(\mathbf{r}^t \mathbf{u}_{i|\Gamma_{ij}} - r).$$

Searching for the optimal point of the Lagrangian, i.e. the point which satisfies

$$\frac{\partial L(\mathbf{u}_{i|\Gamma_{ij}}, \lambda)}{\partial \mathbf{u}_{i|\Gamma_{ij}}} = \mathbf{0}, \qquad \frac{\partial L(\mathbf{u}_{i|\Gamma_{ij}}, \lambda)}{\partial \lambda} = 0,$$

and defining $\mu_0 = \lambda/2$, leads to solving the following system:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{r} \\ \mathbf{r}^t & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{i|\Gamma_{ij}} \\ \mu_0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{I}^{ij}\mathbf{u}_j \\ r \end{bmatrix}, \tag{5.14}$$

where $\mathbf{M}$ is the mass matrix

$$\mathbf{M} = \int_{\Gamma_{ij}} \mathbf{N}_i \, \mathbf{N}_i^t \, d\Gamma.$$

Solving (5.14) for $\mathbf{u}_{i|\Gamma_{ij}}$, we finally find:

$$\mathbf{u}_{i|\Gamma_{ij}} = \mathbf{I}^{ij}\mathbf{u}_j + (\mathbf{M}^{-1}\mathbf{r})(\mathbf{r}^t\mathbf{M}^{-1}\mathbf{r})^{-1}(r - \mathbf{r}^t\mathbf{I}^{ij}\mathbf{u}_j). \tag{5.15}$$

The first term of the last equation represents the classical interpolation, while the second term is due to the constraint. Using a closed quadrature rule to compute $\mathbf{M}$, this equation is trivial since the resulting approximation to $\mathbf{M}$ is diagonal. Obviously, if instead of only having one scalar constraint there are n_c of them, exactly the same procedure can be applied. Matrix $\mathbf{r}^t\mathbf{M}^{-1}\mathbf{r}$ will then have $n_c \times n_c$ components.

We now present three examples to illustrate the interface constraining technique. The second numerical example shows show that this conservation scheme not only enables to treat conservation problems due to different grids sizes but it can be necessary to conserve the mass when one of the subdomains is confined.

Example 1: analytical solution

We present a simple two-dimensional domain decomposition problem involving two disjoint subdomains. Figure 5.9 shows the nomenclature of both meshes on the interface, a one-dimensional line. The letters identify the coarse mesh nodes while the figures identify the fine mesh nodes, and the capital letters refer to the coarse mesh solution while the small letters refer to the fine

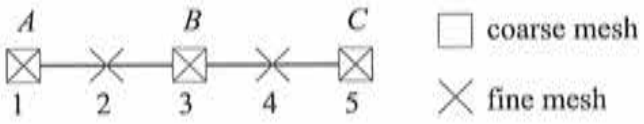


Figure 5.9: Analytical solution. Coarse and fine meshes.

mesh solution. We propose to update the solution U of the coarse mesh, knowing the solution u of the fine mesh and imposing as a constraint the conservation of the integral of the solution. The corresponding problem is to

$$\begin{aligned} &\text{minimize} && \int_0^{2h} [U - I(u)]^2 \, dx, \\ &\text{under the constraint} && \int_0^{2h} U \, dx = \int_0^{2h} u \, dx. \end{aligned} \tag{5.16}$$

We introduce $\mathbf{U}$ and $\mathbf{u}$ as the discrete vectors of unknowns of U and u, respectively. Using linear interpolation, the solution of the system, is

$$\mathbf{U} = \mathbf{Iu} + (\mathbf{M}^{-1}\mathbf{r})(\mathbf{r}^t\mathbf{M}^{-1}\mathbf{r})^{-1}(r - \mathbf{r}^t\mathbf{Iu}),$$

with

$$\begin{aligned} \mathbf{Iu} &= [u^{(1)}, u^{(3)}, u^{(5)}]^t, \\ \mathbf{M} &= \mathrm{diag}(h/2, h, h/2), \\ \mathbf{r} &= [h/2, h, h/2]^t, \\ r &= h/2[u^{(1)}/2 + u^{(2)} + u^{(3)} + u^{(4)} + u^{(5)}/2]. \end{aligned}$$

All the integrals have been computed using a closed quadrature rule. Further calculations give

$$\begin{bmatrix} U^{(A)} \\ U^{(B)} \\ U^{(C)} \end{bmatrix} = \begin{bmatrix} u^{(1)} \\ u^{(3)} \\ u^{(5)} \end{bmatrix} + \frac{1}{4}\left(-\frac{u^{(1)}}{2} + u^{(2)} - u^{(3)} + u^{(4)} - \frac{u^{(5)}}{2}\right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Now let us compare these results with those obtained with the classical interpolation for the three triangle solutions shown in Figure 5.10.

The solutions using the classical and constrained interpolations are drawn together with the fine mesh triangle solutions. Table 5.1 gives the results for the integration of the function along the interface. Obviously, only the constrained interpolation gives the right integral of the solution, the quantity conserved by solving system (5.16). Observe that in this case the nodal quadrature rule is exact, and therefore the integral of the unknown is exactly conserved.

Example 2: mass conservation, cavity

The second example presented involves mass conservation for an incompressible flow problem when one of the transmission conditions is of Dirichlet type and the Dirichlet subdomain is confined. Let us first tackle the origin of the need for conserving mass. We want to solve the stationary

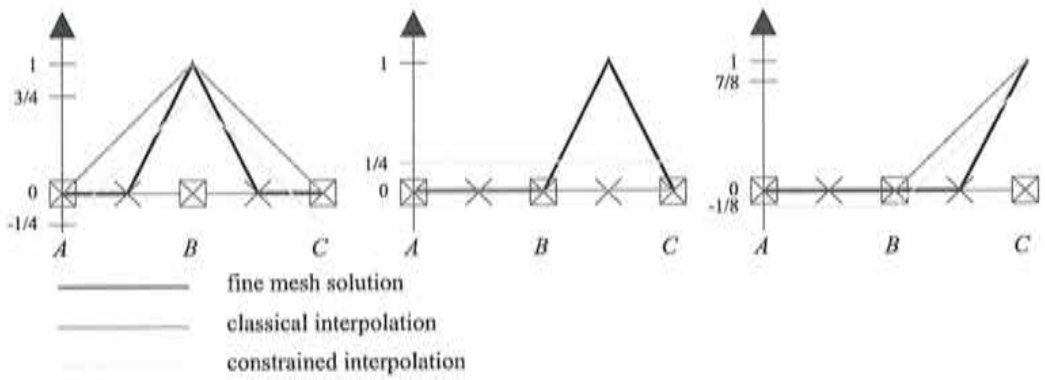


Figure 5.10: Analytical solution. Interpolation of three triangle solutions.

Solution	Fine mesh	Classical	Constrained
$u^{(3)} = 1$	$h/2$	h	$h/2$
$u^{(4)} = 1$	$h/2$	0	$h/2$
$u^{(5)} = 1$	$h/4$	$h/2$	$h/4$

Table 5.1: Analytical solution. Integration of triangle solutions.

Stokes equation in a domain Ω. The weak form of the continuity equation using the stabilized finite element method is derived taking $\boldsymbol{v}_h = \mathbf{0}$ in Equation (1.31):

$$\sum_{e=1}^{n_e} \int_{\Omega_e} \tau_1 \nabla q_h \cdot [-\mu \Delta \boldsymbol{u}_h + \nabla p_h] \, d\Omega + \int_{\Omega} q_h \nabla \cdot \boldsymbol{u}_h \, d\Omega = 0,$$

which must hold for all $q_h \in Q_h$. Taking $q_h = 1$ in Ω, which is an admissible pressure test function, and integrating by parts, we obtain the following compatibility equation

$$\oint_{\partial\Omega} \boldsymbol{u}_h \cdot \boldsymbol{n} \, d\Gamma = 0, \tag{5.17}$$

which is the same as for the continuous problem. Consider two overlapping subdomains i and j. We want to update the interface Γ_{ij} boundary condition of i using a Dirichlet transmission condition. We assume equation (5.17) is satisfied for subdomain j across the whole domain; however, zero mass flow rate across any interior section, and therefore across the interface Γ_{ij}, is not guaranteed. Therefore, we have that

$$\int_{\Gamma_{ij}} I^{ij}(\boldsymbol{u}_j) \cdot \boldsymbol{n}_i \, d\Gamma \neq 0, \qquad \text{generally.} \tag{5.18}$$

Furthermore, if subdomain i is confined, the non-zero mass flow rate passing through the interface Γ_{ij} remains inside the subdomain and, therefore, the boundary condition of i does not satisfy the compatibility equation. This is illustrated by Figure 5.11. Note that for discontinuous pressure spaces, q_h can be taken piecewise constant and if Γ_{ij} coincides with some element boundaries of subdomain j, the net flux across Γ_{ij} is zero.

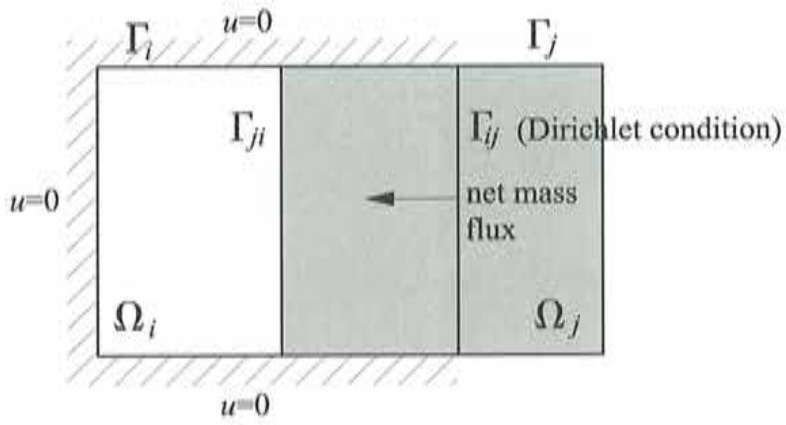


Figure 5.11: Mass conservation, cavity. A confined subdomain with Dirichlet condition on its interface.

A solution to circumvent the incompatibility of the transmission boundary condition and the weak formulation is to decrease the mass flux in subdomain i by means of the constraining technique. We propose to obtain the boundary data $\boldsymbol{u}_i$ by solving the problem:

$$\begin{aligned} &\text{minimize} && \int_{\Gamma_{ij}} |\boldsymbol{u}_i - I^{ij}(\boldsymbol{u}_j)|^2 \, d\Gamma, \\ &\text{under the constraint} && \oint_{\partial\Omega_i} \boldsymbol{u}_i \cdot \boldsymbol{n}_i \, d\Gamma = 0. \end{aligned}$$

The system is solved using the strategy defined previously, with the constraint re-expressed as:

$$\int_{\Gamma_{ij}} \boldsymbol{u}_i \cdot \boldsymbol{n}_i \, d\Gamma = -\int_{\partial\Omega_i \backslash \Gamma_{ij}} \boldsymbol{u}_i \cdot \boldsymbol{n}_i \, d\Gamma \quad (= 0 \quad \text{in the case of Figure 5.11}).$$

Note that the integral on the right-hand side of the constraint only involves the solution of i on $\partial\Omega_i \backslash \Gamma_{ij}$, which is known; the integral can therefore be calculated accurately using the same closed quadrature than that used to compute the boundary mass matrix $\mathbf{M}$. In the next example, a special integration rule will have to be designed as the right-hand side of the constraint will depend on the solution of the fine mesh.

We solve the Stokes cavity flow using the Q1/Q1 element (piecewise bilinear velocities and pressures) on two subdomains. The DD method used to couple the subdomains is the Schwarz method. Figure 5.12 shows the pressure contours obtained. The contours are only shown for the right-hand side subdomain. Figure 5.12 (Top) (Left) shows the inconsistency in the pressure field induced by the non-conservation of mass in the two subdomains. The wriggles appear precisely where the pressure is prescribed, i.e. at the top left corner. Figure 5.12 (Top) (Right) shows the pressure contours obtained using the zero mass flow rate constraint on the interface. The last solution corrects the zone of pressure instabilities in the upper left corner, where the value of the pressure is imposed (remember that the flow is confined and therefore, the pressure must be prescribed at one point in the subdomain). This is directly related to the fact that the algorithm enables us to reduce the mass flow rate. This is confirmed by table 5.4.2 which gives the net mass flow rates in the two parts of the cavity. They are normalized by the velocity at the top wall times

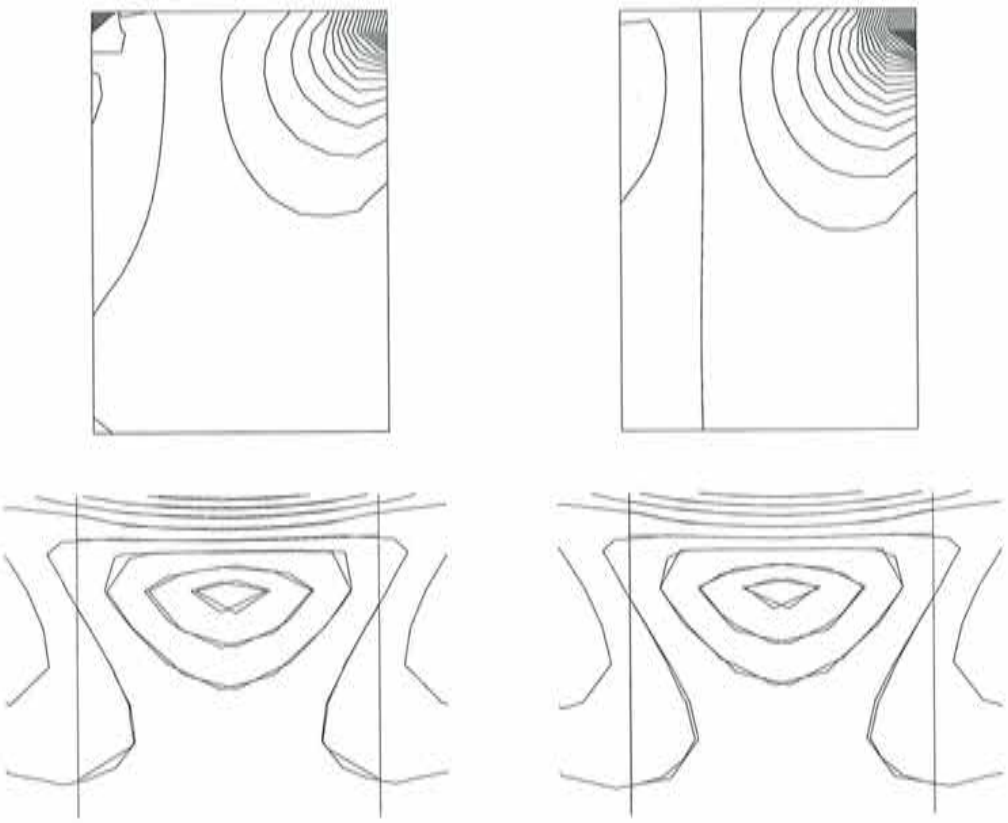


Figure 5.12: Mass conservation, cavity. Pressure contours of the right-hand side subdomain. (Top) Pressure contours. (Bot.) Velocity module contours. (Left) Classical interpolation. (Right) Constrained interpolation.

the length of the interface. The residual mass flow rate obtained for the constrained method is the mass flow rate passing through the top and bottom first elements, over which we have no control.

Subdomain	Classical	Constrained
Left-hand side	1.5	-0.1
Right-hand side	-3.0	0.1

Table 5.2: Mass conservation, cavity. Mass flow rates ($\times 10^{-3}$).

Finally, Figure 5.12 (Bot.) (Left)-(Right) shows details of the velocity module in the center of the cavity. In the overlapping region, the mass flow rate constraining method gives the best results. It should also be pointed out that the convergence of the problem is not affected by the constraint.

The mass conservation was illustrated for a simple flow because it enables us to clearly estimate the effects of the constraining. However, the problem of incompatibility of the data is very likely to occur when using a Chimera method because it uses a Dirichlet transmission condition for the

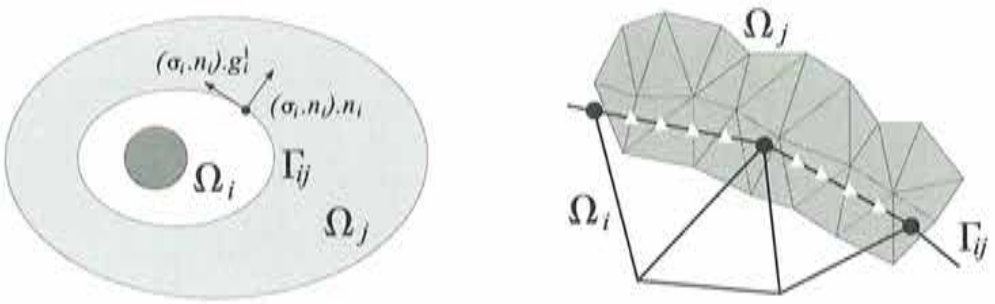


Figure 5.13: Force conservation. (Left) Components of the force on the interface. (Right) Integration strategy. •: nodes of i. △: integration points.

interface of patch meshes with the background mesh.

Example 3: force conservation

We will now present a strategy to conserve the components of the force (or traction) acting on the interface; see Figure 5.13 (Left). This algorithm can be applied only to the least-square interpolation, once the stress is obtained at the nodes of the interface. We propose to update the stress $\boldsymbol{\sigma}_i$ from the known stress $\boldsymbol{\sigma}_j$ by solving the following problem in two dimensions:

$$\text{minimize} \qquad \int_{\Gamma_{ij}} [\boldsymbol{\sigma}_i \cdot \boldsymbol{n}_i - I^{ij}(\boldsymbol{\sigma}_j \cdot \boldsymbol{n}_i)]^2 \, d\Gamma,$$

under the constraints

$$\begin{cases} \displaystyle\int_{\Gamma_{ij}} (\boldsymbol{\sigma}_i \cdot \boldsymbol{n}_i) \cdot \boldsymbol{n}_i \, d\Gamma &= \displaystyle\int_{\Gamma_{ij}} (\boldsymbol{\sigma}_j \cdot \boldsymbol{n}_i) \cdot \boldsymbol{n}_i \, d\Gamma, \\ \displaystyle\int_{\Gamma_{ij}} (\boldsymbol{\sigma}_i \cdot \boldsymbol{n}_i) \cdot \mathbf{g}_i^1 \, d\Gamma &= \displaystyle\int_{\Gamma_{ij}} (\boldsymbol{\sigma}_j \cdot \boldsymbol{n}_i) \cdot \mathbf{g}_i^1 \, d\Gamma. \end{cases}$$

By introducing two Lagrange multipliers for the constraints, this problem leads to the solution of a matrix system of the form

$$\begin{bmatrix} \mathbf{M} & -\mathbf{r} & -\mathbf{s} \\ \mathbf{r}^t & 0 & 0 \\ \mathbf{s}^t & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t}_i \\ \mu_r \\ \mu_s \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{I}^{ij}\mathbf{t}_j \\ r \\ s \end{bmatrix},$$

where $\mathbf{t}$ is the nodal vector of the traction components, $\mathbf{r}$ and r are the vector and scalar representing the normal force constraint, and $\mathbf{s}$ and s are the vector and scalar representing the tangential force constraint. The calculations of $\mathbf{r}$ and $\mathbf{s}$ are straightforward. However, the success of the conservation stems from accurately calculating the total force contribution of the adjacent subdomain j. In the example discussed previously, the constraint depended only on the solution in subdomain i. In the present problem, the force is known from j and if the mesh of i is too coarse with respect to the mesh in j, a special integration strategy has to be found to integrate r and s. In order to take into account the possible loss of conservation, the calculation of r and s will be performed by injecting a sufficient number of integration points on Γ_{ij}. The strategy is illustrated in Figure 5.13 (Right). Note that an efficient element search strategy is therefore necessary in order to find a host element for each of these integration points. For example, the number of integration points

to be chosen on each element boundary could be related to the ratio of the local density of nodes of i to that of j.

Conclusion

We have developed a method for constraining the interface boundary conditions in the framework of domain decomposition methods. As a first application, we have constrained Dirichlet interfaces with the zero mass flow rate equation and good results have been obtained. The method is general and can be applied to the conservation of any quantity involving the variable of the transmission condition.

5.4.3 An interpolation operator

We now present a "conservative" interpolation operator specially designed for interpolating variable from a fine mesh to a coarse mesh. As a first and simple approach, the transmission variables at the interpolation and interface nodes are obtained using the Lagrange interpolation functions, as given by Equation (5.1). The interface constraining technique presented previously enables one to conserve some properties across the interface. We propose here to work directly on the interpolation operator. When Lagrange interpolation functions are used, the method will be referred to as *classical interpolation* (Class.). Let us denote n_{p_i} as the number of nodes of subdomain i and n_{p_j} the number of nodes of subdomain j from which we want to interpolate the solution. The classical method gives:

$$\mathbf{u}_i = \mathbf{I}^{ij}\mathbf{u}_j,$$

where this time we denote $\mathbf{I}^{ij}$ as the operator $\mathbf{I}^{ij}$ is the interpolation operator to the whole mesh i from mesh j, and it is a $n_{\mathrm{p}_i} \times n_{\mathrm{p}_j}$ matrix. Obviously, only the interpolation nodes are updated with this formula, and the matrix coefficients corresponding to the other nodes are meaningless.

One way to avoid losing information when interpolating from one mesh to another, and particularly from a fine to a coarse mesh, is to take into account the values of the interpolated function not just locally, but using a cloud of nodes. The idea is illustrated in Figure 5.14. On the upper part, a sharp stencil uses only the value of the interpolated function on one node of the fine grid, while on the lower part, the dense stencil uses the values of all its neighbors. Dense stencils are the key for developing conservative interpolation. In some sense, the interface constraining method can be viewed as a general method for devising conservative dense stencils.

Inspired by transfer operators of multigrid methods (see for instance [103] or [104]), we suggest an alternative method to the classical Lagrange interpolation to obtain $\mathbf{u}_i$. We seek for a kind of transfer operator which is *conservative* in the sense that the mean of a scalar field computed from its integral in both domains is equal. As a guideline, we want our operator to satisfy the following requirements:

1. as in the classical interpolation, a constant field must be transferred as constant field of the same value;

2. for identical meshes, identical fields should be obtained;

3. additionally, the information contained in high frequency oscillations in the fine mesh should be partially present in the coarse one.

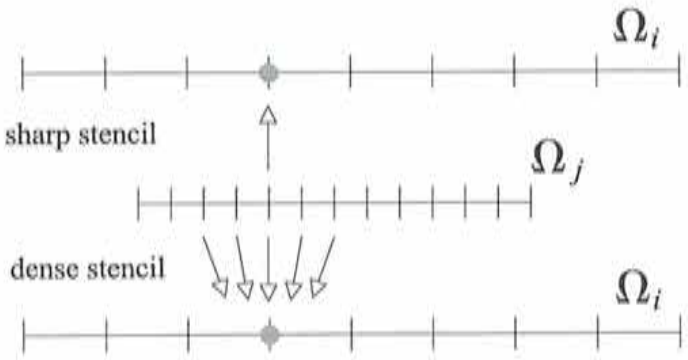


Figure 5.14: Interpolation stencils.

Let us define the interpolation matrix $\mathbf{I}^{ji}$ of the interpolation coefficients from mesh i to mesh j. Then,

$$\mathbf{u}_j = \mathbf{I}^{ji}\mathbf{u}_i.$$

The idea is to use the information contained in $\mathbf{I}^{ji}$, which transfers variables from background to patch to improve transferring from patch to background taking into account conservation properties. This is something normally considered in multigrid methods when right hand sides are passed from finer to coarser meshes. In that case, the transpose of matrix $\mathbf{I}^{ji}$ can be plainly used, even though that the local different characteristic element sizes introduces a scale factor. It can be shown that although this scale factor is helpful in RHS' multigrid transferring, it leads to a violation of conservation when passing variables [104], as in the case of DD methods. For that reason, we propose the following

$$\mathbf{u}_i = (\bar{\mathbf{I}}^{ji})^t\mathbf{u}_j, \tag{5.19}$$

where $\bar{\mathbf{I}}^{ji}$ is the column-wise normalized interpolation matrix defined as

$$\bar{\mathbf{I}}^{ji} = \mathbf{I}^{ji}\mathrm{diag}\left(1/\textstyle\sum_{i=1}^{n_{\mathrm{p}_j}} \mathbf{I}^{ji}_{i,1}, 1/\sum_{i=1}^{n_{\mathrm{p}_j}} \mathbf{I}^{ji}_{i,2}, \ldots, 1/\sum_{i=1}^{n_{\mathrm{p}_j}} \mathbf{I}^{ji}_{i,n_{\mathrm{p}_i}}\right).$$

This method will be referred to as the *normalized transpose interpolation*, denoted NTI. To show the positive effect of the normalization, we will also momentarily consider the plain transpose interpolation which uses directly $(\mathbf{I}^{ji})^t$, denoted PTI.

Example: analytical solution

As an illustration, the three interpolations defined previously are analyzed for a very simple one-dimensional example. We use the meshes sketched in Figure 5.9. Let us denote $u^{(r)}$ for $r = 1, 2, 3, 4, 5$ the solution on the fine (patch) mesh and $U^{(r)}$ for $r = A, B, C$ the solution on the coarse (background) mesh. The node spacing on the coarse mesh is h while that on the fine mesh mesh

is twice as small. We obtain the following interpolation matrices:

$$\text{Class.:} \begin{bmatrix} U^{(A)} \\ U^{(B)} \\ U^{(C)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ u^{(3)} \\ u^{(4)} \\ u^{(5)} \end{bmatrix},$$

$$\text{PTI:} \begin{bmatrix} U^{(A)} \\ U^{(B)} \\ U^{(C)} \end{bmatrix} = \begin{bmatrix} 1 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 1 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ u^{(3)} \\ u^{(4)} \\ u^{(5)} \end{bmatrix},$$

$$\text{NTI:} \begin{bmatrix} U^{(A)} \\ U^{(B)} \\ U^{(C)} \end{bmatrix} = \begin{bmatrix} 2/3 & 1/3 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 0 & 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ u^{(3)} \\ u^{(4)} \\ u^{(5)} \end{bmatrix}.$$

The problem of the classical interpolation is obvious: the solution on the coarse mesh does not explicitly depend on $u^{(2)}$ and $u^{(4)}$. In physical terms, this means that the operator filters out the high frequencies. We will now consider some triangle solutions on the fine mesh and examine how such solutions are interpolated on the coarse mesh for the three interpolation methods described previously. Figure 5.15 sketches the solution obtained using the NTI and PTI operators. It is interesting to check how well those three methods integrate the function in the interval. Consider three different triangle functions defined on the figure; table 5.3 shows the result of the integration of the function as calculated for each method. As expected, the Class. method gives the wrong

Solution	Fine mesh	Class.	PTI	NTI
$u^{(3)} = 1$	$h/2$	h	h	$h/2$
$u^{(4)} = 1$	$h/2$	0	$3h/4$	$5h/12$
$u^{(5)} = 1$	$h/4$	$h/2$	$h/2$	$h/3$

Table 5.3: Integration of a triangle solution for different interpolations.

results for the integrals of the three triangles solutions. The integral is either underpredicted or overpredicted. The PTI always overpredicts these integrals. Finally, only the NTI approximately captures the area of the three triangles solutions.

Apart from its conservation property, the normalized transposed interpolation has one more advantage. The interpolation matrix involves only the interpolation coefficients of the patch mesh nodes of the corresponding host elements of the background mesh. If the background mesh is structured (Q1 elements), the search for host elements is therefore trivial. This could be an important property if the patch is moving with time; in this case, the interpolation operator would have to be calculated at each time step.

The NTI operator has been successfully applied to multigrid techniques by Vázquez *et al.* in [105] and to the present DD methods applied to an explicit flow solver [106].

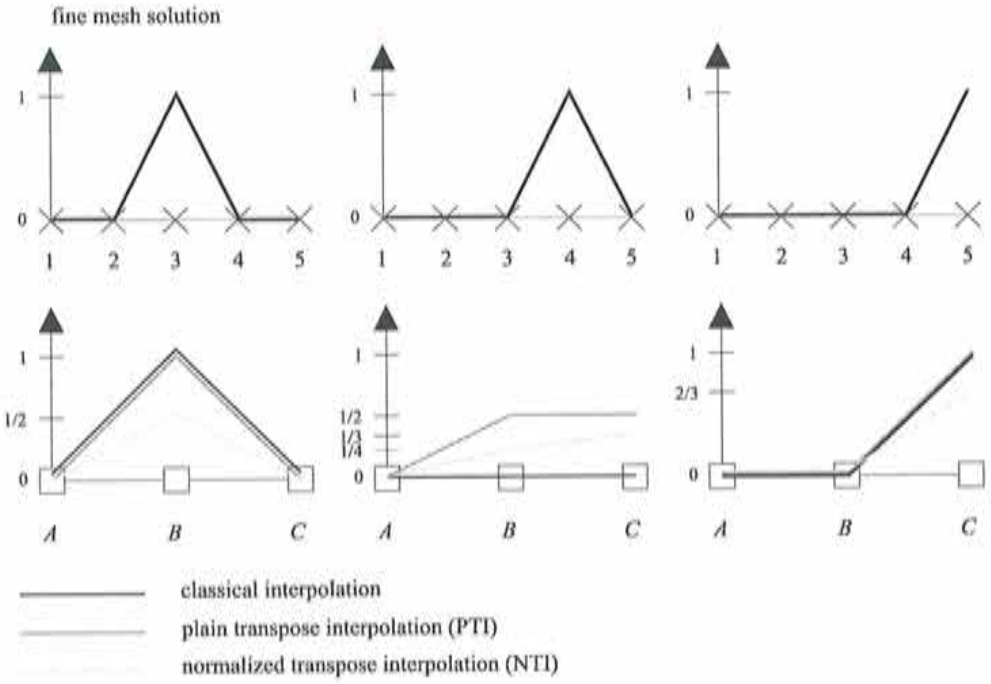


Figure 5.15: Interpolations of triangle solutions on the fine mesh.

Example 2: a conservative mesh refinement technique, cavity flow

Through this example we propose to test the conservative operator for the D/N method used as a local refinement technique. We solve the cavity flow on a domain $(0,1) \times (0,1)$ at a relatively high Reynolds number $\mathrm{Re} = 5\,000$, based on the cavity length and the velocity at the cavity top. The geometry is shown in Figure 1.4 (Left). The results are going to be compared to the results of the standard reference for this flow by Ghia *et al.* [107].

We first solve the flow on an adapted mesh of 900 Q1/Q1 elements shown in Figure 5.16 (Top) (Mid.) and use its solution for the sake of comparisons. Figure 5.16 (Bot.) (Mid.) gives the velocity field while Figures 5.17 (Left) or (Right) show the horizontal velocity along a vertical cut at the middle of the cavity. We also generate a very coarse mesh of 400 uniformly distributed Q1/Q1 elements, shown in Figure 5.16 (Top) (Left). Figures 5.16 (Bot.) (Left) 5.17 (Left) or (Right) show that the momentum imposed at the cavity top is not transmitted at all at lower parts of the cavity, where the velocity is almost zero. Can we obtain a satisfactory solution by way of local refinement? Let us overset a fine mesh on the top of the very coarse mesh. The fine mesh occupies the first top quarter of the cavity and has 450 $Q1/Q1$ elements: this is the Neumann subdomain. The Dirichlet subdomain is the coarse mesh: the first three element layers of this coarse mesh are cut so that the resulting Dirichlet subdomain is composed of 340 elements. The fine mesh is refined near the top of the cavity so that it looks more or less like the adapted mesh of the one-domain solution; the resulting composite mesh is shown in Figure 5.16 (Top) (Right). We hope the fine mesh will help the flow enter the cavity and will be sufficient to capture the leading flow scales on the top of the cavity.

We tested the D/N method using the classical interpolation as well as the normalized transpose interpolation (NTI), using an overlap of two layers of elements. Figure 5.17 compares the horizontal

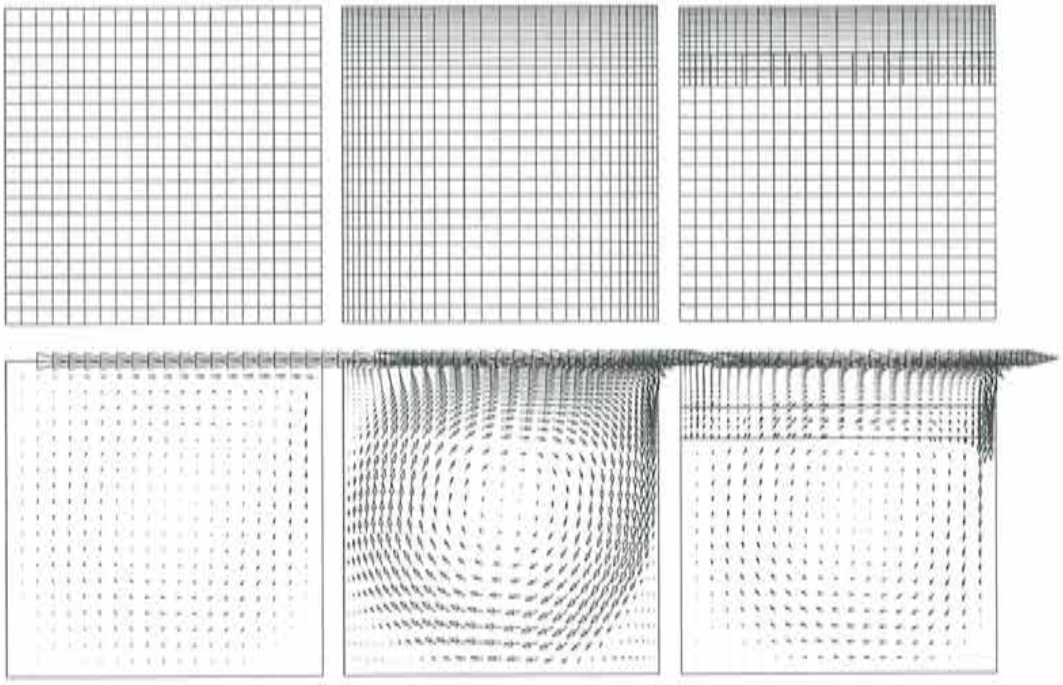


Figure 5.16: Cavity flow. (Top) Meshes. (Bot.) Velocity vectors. (Left) Coarse mesh. (Mid.) One domain. (Right) D/N+NTI.

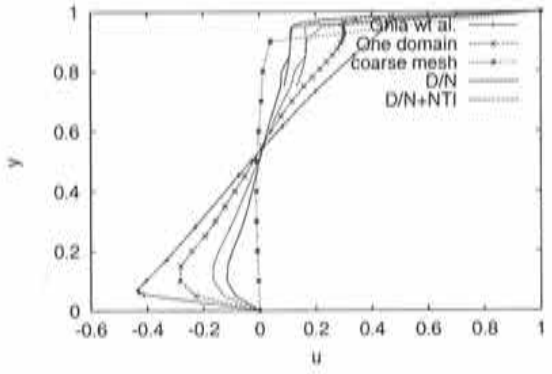


Figure 5.17: Cavity flow. Horizontal velocity at $x = 0.5$. D/N method.

velocity profile at $x = 0.5$ obtained by the D/N couplings together with the solution on the refined mesh, the very coarse mesh, and with Ghia's results. We observe that the NTI improves the results obtained with the D/N and the classical interpolation. Figure 5.16 (Bot.) (Right) shows the velocity vectors in the cavity. We have also tested the D/D method with the same overlap: the D/D method does not converge to a stationary state.

5.5 Chimera Method

This section studies a Chimera method intended to solve incompressible flows on complex geometries. We first describe the purpose of the Chimera approach by giving an insight of the possibilities of the method. Then we introduce some terminology and explain the way that the Chimera method can be implemented as an iteration-by-subdomain DD technique. In particular, we build a Chimera method based on overlapping Dirichlet/Dirichlet and Dirichlet/Neumann couplings.

5.5.1 Motivation

We want to set up a simple strategy to solve a fluid problem on a given geometry, including the possibility of easily adding, removing and modifying some components, without the need for remeshing the global mesh. The DD algorithm to account for these requirements is based on the Chimera method. A background mesh is first defined. It can contain some objects whose geometries and positions should not change with time, and for which the grid can be easily generated. Then, separate grids are generated for the components to be patched onto the background mesh. This defines a global geometry on which the relative positions of the objects can be changed easily. From the geometrical coupling, an iterative strategy is set up to exchange transmission conditions as in the case of iteration-by-subdomain DD methods, in order to obtain a global solution.

The Chimera method was first envisaged as a tool for simplifying the mesh generation [95, 108, 109]. Independent meshes are generated for each component of the computational domain, enabling a flexibility on the choice of the type of element as well as on their orientation that could not be possible when meshing complex three dimensional geometries [110, 111]. See for example [112] for the application of the Chimera method to the investigation of the aerodynamics of the Space Shuttle launch vehicle. As a direct application, the Chimera method has also been used as a mesh refinement technique [113]. In addition, if it is implemented efficiently, it is a very efficient tool to treat flows with moving components [114]. This issue will be addressed later on in Section 5.6.

5.5.2 Geometrical coupling and terminology

For the sake of clarity, we assume that the flow we solve only involves one object. The generalization to multi-component flows is straightforward. We first define a background mesh containing all the computational domain, preferably structured. We also generate an independent mesh around the object and dispose it onto the background mesh. This is the patch mesh. The set of all the overset grids is called the *composite grid*, or composite mesh. The idea of the Chimera method is to remove some elements of the background located inside the patch in order to define an apparent interface, this task is called *hole cutting*. The domain decomposition method therefore will consist in exchanging suitable transmission conditions between the outer boundary of the patch and the apparent interface just defined.

Some definitions

We introduce the following definitions to identify the nodes of the background mesh (see Figure 5.18 (Left)):

- *guest node*: node of the background having a host element in the patch;
- *lost node*: node of the background having *no* host element in the patch;

- *object node*: lost node of the background located inside an object of the patch;
- *interpolation node*: guest node participating to the DD coupling;
- *overlapping node*: guest node explicitly freed to define a desired overlap, i.e. a minimum distance (geometrical distance or in terms of elements) between the outer boundary of the patch and the apparent interface of the background;
- *free node*: node containing free degrees of freedom to be updated by the flow solver;
- *fringe node*: interpolation node forming part of the apparent interface of the background subdomain.

A *hole* is defined as the set of elements which contain only object and interpolation nodes. By definition, the boundary of the hole, i.e. the apparent interface of the background, is composed of fringe nodes. When dealing with a steady subdomain, only the fringe nodes of the hole participate in the DD coupling. Therefore, all the object nodes as well as the non-fringe interpolation nodes can be eliminated from the solution process. However, in the case of moving subdomains, information at these nodes must be saved at each time step as a hole node can become an interpolation node, for which values at the previous time step are required; in the same way, an interpolation node can become a free node, for which values at the previous time step are also required. Figure 5.18 (Left) shows a simple example which illustrates all the definition introduced above.

The overlap

We now introduce the overlap. In Chapter 3, we showed the explicit dependence of the convergence of overlapping methods upon the overlapping length of two adjacent subdomains; it is therefore interesting to be able to control the geometrical length between interfaces. The algorithm used to ensure a minimum overlap is trivial: if the distance of a guest node to the patch interface is lower than the overlapping length desired, then state it as an overlapping node.

In addition, a certain number of elements of overlap may be required according to the type of transmission conditions to be used in the DD coupling. The first reason for this is that for overlapping methods like the Dirichlet/Dirichlet method, at least one element of overlap is needed on each subdomain. This is a sufficient condition to ensure not only continuity of the velocity but also of its derivatives. The second reason is that to achieve a second order Dirichlet/Neumann method, we need at least one element of the background mesh on each side of the Neumann interface.

In order to ensure a layer of one-element overlap in the patch mesh, we proceed as follows. We define the first layer of elements of the patch as the set of elements of the patch mesh connected to its own outer boundary, i.e. the interface. Then, all the interpolation nodes of the background mesh having a host element belinging to the first layer are freed. Figure 5.18 (Left) illustrates a Chimera coupling with a one-element overlap in the patch mesh, as indicated by the elements painted in grey.

When interpolating the same variable on the patch and on the background mesh (like the D/D coupling), we must ensure that we have at least one layer of element of overlap in the background mesh in addition to the overlap achieved with respect to the patch mesh. This is also necessary when we want to achieve a second order scheme in space for the D/N coupling, or at least to expect it to be of second order. This is done by letting free some of the interpolation nodes of the background mesh. We define the *shadow* of the patch interface the set of elements of the background mesh having both interpolation and free nodes. All the nodes belonging to the shadow are freed and stated as overlapping nodes, as illustrated by Figure 5.18 (Right).

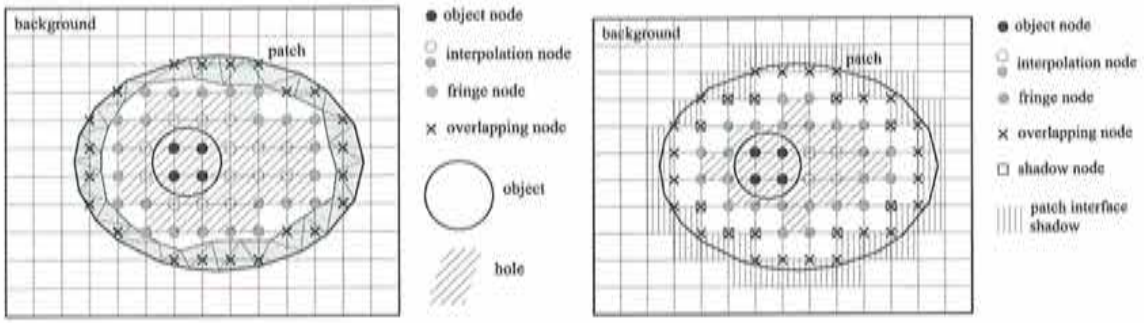


Figure 5.18: (Left) Overlap in the patch mesh. (Right) Overlap in the background mesh.

The three operations to construct the overlap must be carried out in a precise order. In fact, we must start by freeing the interpolation nodes located in the shadow of the interface. Then, free the remaining interpolation nodes located in the first layer of element of the patch mesh. Finally, proceed to the geometric overlap.

Node identification algorithm

The procedure to identify all the nodes of the background subdomain is shown in Algorithm 5.7. We first define the bounding box of the patch mesh and perform an element search strategy for the nodes `ipoin` of the background mesh located inside this bounding box. If `ipoin` has a host element, it can be an interpolation or an overlapping node. In the first case, the node will participate to the DD coupling; in the second case, it must be freed. If `ipoin` has no host element, it means that the node can be outside the patch or inside an object of the patch. A simple way to make short work of the problem is to perform the following test. Let P be the orthogonal projection of `ipoin` onto the object boundary of the patch mesh, and let $\boldsymbol{n}$ be the outward unit vector normal to the patch at P, as illustrated by Figure 5.19.

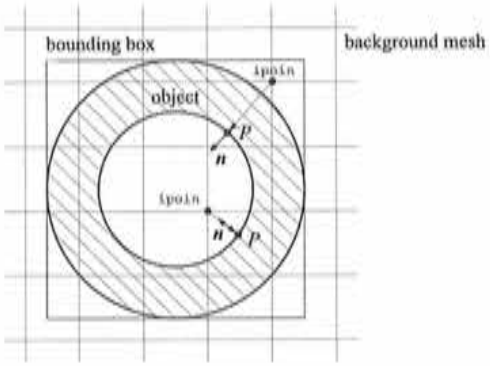


Figure 5.19: Check if a lost node is inside an object.

If the scalar product $\overrightarrow{(\texttt{ipoin}, P)} \cdot \boldsymbol{n} \leq 0$, then `ipoin` is an object node and can be eliminated from the solution process. Otherwise, `ipoin` is outside the patch computational domain and must be freed. This operation consists in identifying the hole and is a simple alternative to the classical (but quite heavy) hole cutting technique, as presented in [115].

Algorithm 5.7 Chimera Method. Identify the actors of the background mesh

```
Define the bounding box of the patch mesh
Set a minimum overlapping length δ
for all nodes ipoin of the background mesh inside the bounding box do
  Perform an element search for ipoin
  if ipoin has a host element then
    If necessary check if ipoin is inside the shadow of the patch interface
    if ipoin is inside an interface shadow element then
      ipoin is an overlapping node and is freed
    else
      ipoin is an interpolation node
    end if
    Find to which layer of elements of the patch ipoin belongs
    If necessary check if ipoin is inside the first layer of element of the patch
    if ipoin belongs to the first layer of element of the patch then
      ipoin is an overlapping node and is freed
    else
      ipoin is an interpolation node
    end if
    Compute the distance between ipoin and the patch interface
    if the distance is lower than δ then
      ipoin is an overlapping node and is freed
    else
      ipoin is an interpolation node
    end if
  else if ipoin is lost then
    Check if ipoin is an object node or a free node
  end if
end for
```

5.5.3 Transmission conditions

In the preceding chapters, we studied some overlapping versions of current mixed DD methods. We now generalize the overlapping Dirichlet/Neumann method applied to the Chimera method, and propose a new Chimera/Neumann coupling (C/N). We also propose to study the classical Chimera method, referred to here as Chimera/Dirichlet coupling (C/D). The background mesh is the "Chimera" subdomain for which the primary variable of the problem is interpolated at the fringe nodes; the patch mesh is either assigned a Dirichlet or a Neumann transmission condition on its outer boundary. "Chimera" is not actually an appropriate term to define an interface type as it generally defines a complete DD method in the scientific literature, but we hope its use in the present context is clear. The C/D and C/N couplings are illustrated in Figure 5.20.

Some special attention must be paid to the pre-process part of the algorithm, which will determine the convergence of the iterative procedure as well as its accuracy. The construction of the C/D coupling requires special care, as a minimum overlap is required to avoid nodes from coinciding. If this becomes the case, the interpolated variable would be frozen at its initial value on the coinciding nodes. In addition, as mentioned in Section 5.5.2, a minimum overlap of one layer of element is needed on each mesh participating in the C/D coupling.

This is not the case in the C/N method because the variables interpolated at the interpolation nodes are different from those interpolated at the interface nodes. However, we saw in Section

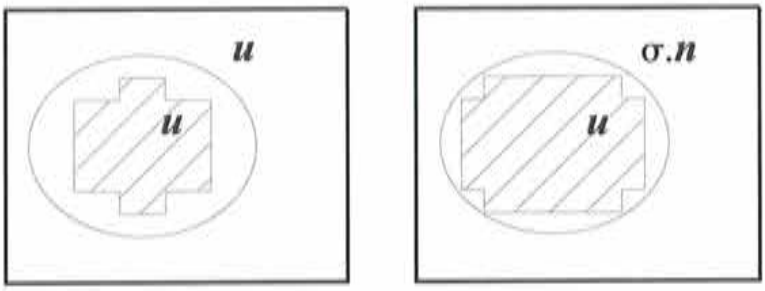


Figure 5.20: Chimera method. Hole and variables transmitted. (Left) Chimera/Dirichlet. (Right) Chimera/Neumann.

(5.3.2) that the overlapping Dirichlet/Neumann needs at least one element on each side of the Neumann interface to expect the least-square smoothing of the derivatives to lead to a second order scheme.

5.5.4 The algorithm

The Chimera/Dirichlet and Chimera/Neumman methods fit perfectly into the framework of the Master/Slave-coupling described at the beginning of this chapter. When dealing with various unconnected patch grids, the solution on each of these subdomains can be obtained in parallel, while keeping the sequential coupling with the background. The Chimera based iteration-by-subdomain method is illustrated by Algorithm 5.8 for the Master's point of view, while the Slave point of view is the same as that presented by Algorithm 5.3.

Algorithm 5.8 Chimera method. Master's point of view

```
Find the host elements of all the interface nodes of the patches
Identify the actors of the background as shown in Algorithm 5.7
while stopping criterion not reached do
  Run background
  Import solution from background
  Interpolate and compute transmission conditions for patches
  Export transmission conditions to patches
  Run patches in parallel
  Wait for all patch processes to be done
  Import solutions from patches
  Interpolate and compute transmission condition at fringe nodes for background
  Export transmission conditions to background¡
end while
```

5.5.5 Example: flow past two cylinders

We now illustrate the Chimera method through a simple example. We consider a two-dimensional flow past two cylinders of diameter $D = 1$. The bounding box of the computational domain is $(0, 16) \times (-6, 6)$ and the center of the cylinders are located at $(4, -1.5)$ and $(4, 1.5)$. The inflow velocity is $U = 1$, and the Reynolds number is

$$\mathrm{Re} = \frac{UD}{\nu} = 1.$$

We perform three simulations. First we solve the problem without domain decomposition and using a mesh of 4414 P1/P1 elements, shown in Figure 5.21 (Top). We now decompose the computational domain into three subdomains. The first domain is the background subdomain and contains 2400 Q1/Q1 elements. Then we define patch subdomains for each cylinder. The outer boundary of the top subdomain is a circle of diameter 1 and the associated mesh is composed of 518 P1/P1 elements; Figure 5.21 (Top) shows the composite mesh around it. The outer boundary of the bottom subdomain is a circle of diameter 2 and its associated mesh as 2400 P1/P1 elements. Figure 5.21 (Bot.) shows the composite mesh.

From this partition we perform two DD simulation with one element-layer overlapping, using both the Chimera/Dirichlet method with mass conservation constraining (which will be presented in the following section) and the Chimera/Neumann method with least-square smoothing. Figure 5.22 (Top) (Left) shows a zoom of the composite mesh of the top cylinder and Figure 5.22 (Top) (Right) shows the resulting hole cutting. Figures 5.22 (Bot.) presents the outline of the holes.

Figure 5.23 presents the solution obtained for the one-domain solution as well as for the C/D and C/N methods. As the pressure of the patch subdomain is unique up to an additive constant (they are confined) when using the C/D, the pressures in the background and patch subdomains have to be calibrated. This is done by adjusting the level of pressure using the difference of pressure at one node of the patch and its corresponding value in the background.

The results of the C/D are far from convincing. The pressure exhibits some strong discontinuities at the interface of the top cylinder; note that for the sake of clarity the maximum value of the pressure contour fill is limited to the maximum value obtained for the one-domain solution. The results obtained around the bottom cylinder, whose interface is farther to the cylinder wall than the top cylinder does, are much better. This is attributed to the fact that Dirichlet conditions are much stiffer than Neumann conditions, i.e. a small error on Dirichlet conditions has much more influence than a small error on the Neumann conditions. This is confirmed by the results of the C/N method which are in very good agreement with those of the one-domain solution.

Geometrical DD methods are not exact if the nodes don't coincide. This example shows however that the C/N seems to be more accurate than the C/D method.

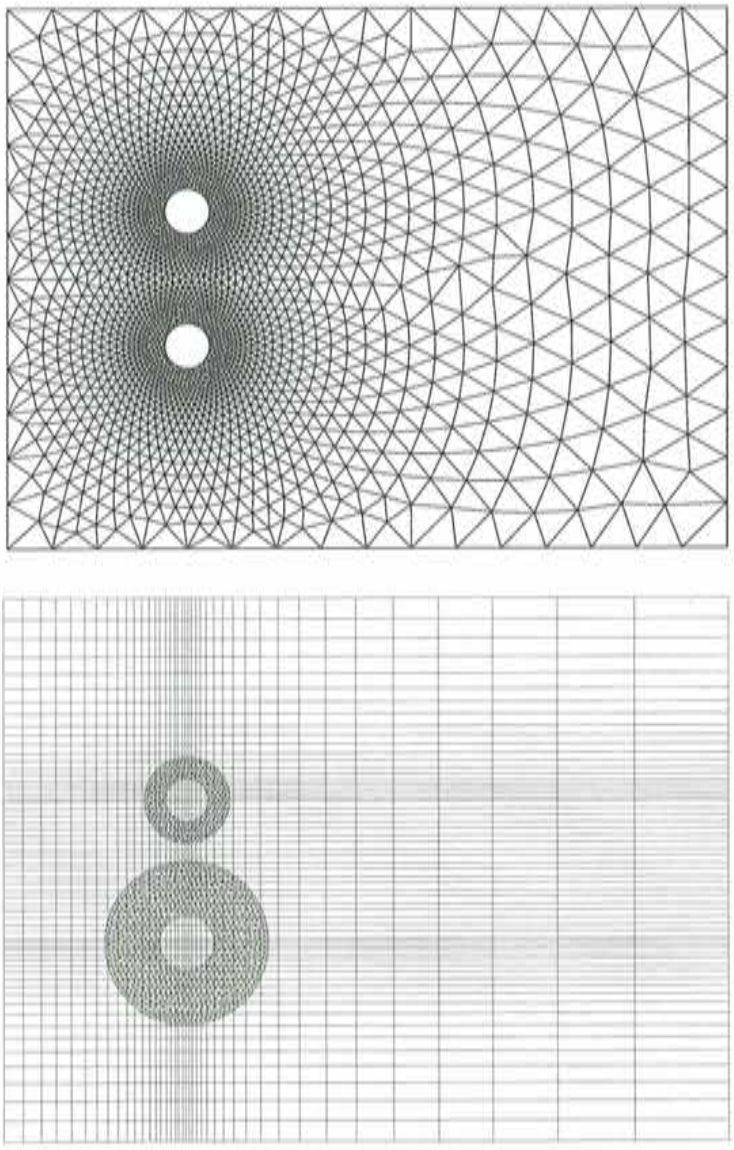


Figure 5.21: Flow past two cylinders. (Top) Mesh of the one-domain solution. (Bot.) Composite mesh resulting from the Chimera partitioning.

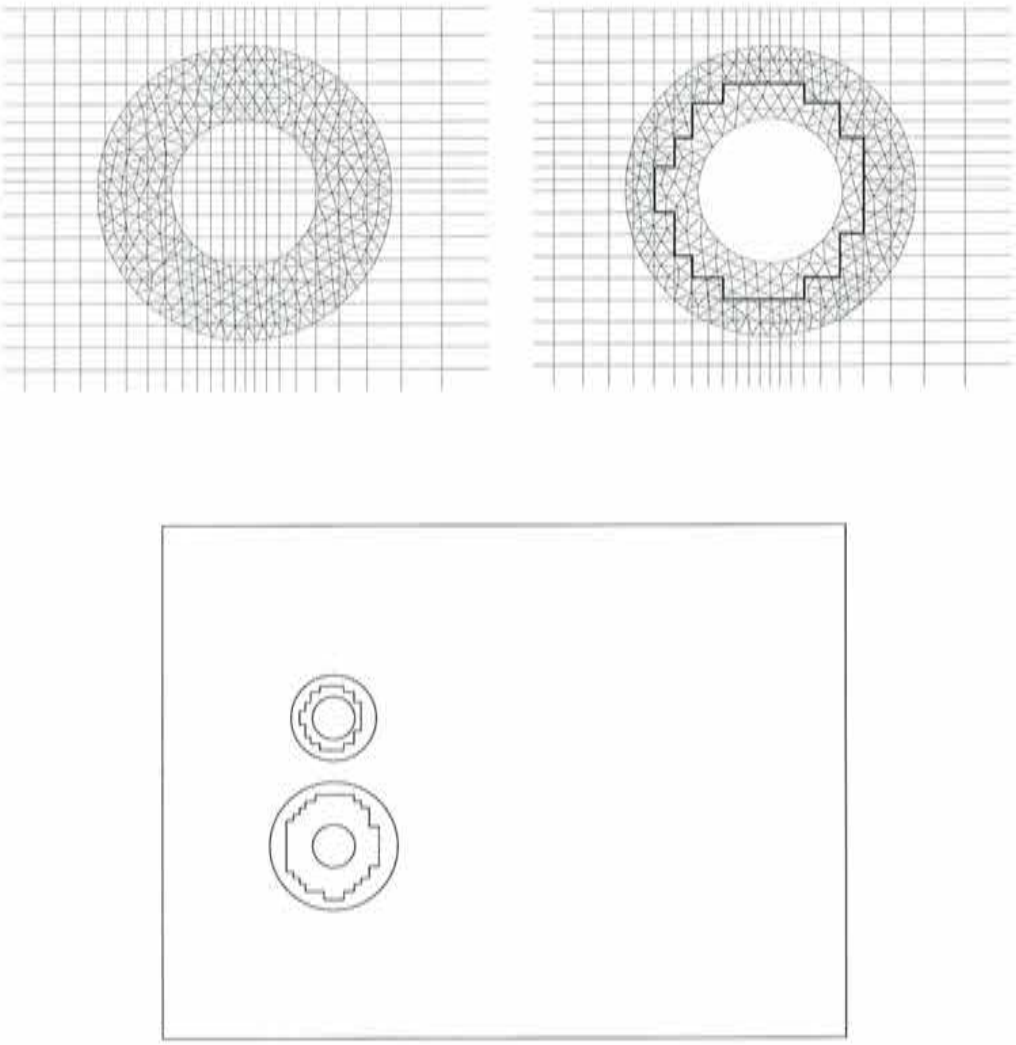


Figure 5.22: Flow past two cylinders. Hole cutting. (Top) (Left) Top cylinder before hole cutting. (Top) (Right) Top cylinder after hole cutting and apparent interface. (Bot.) Interfaces and hole.

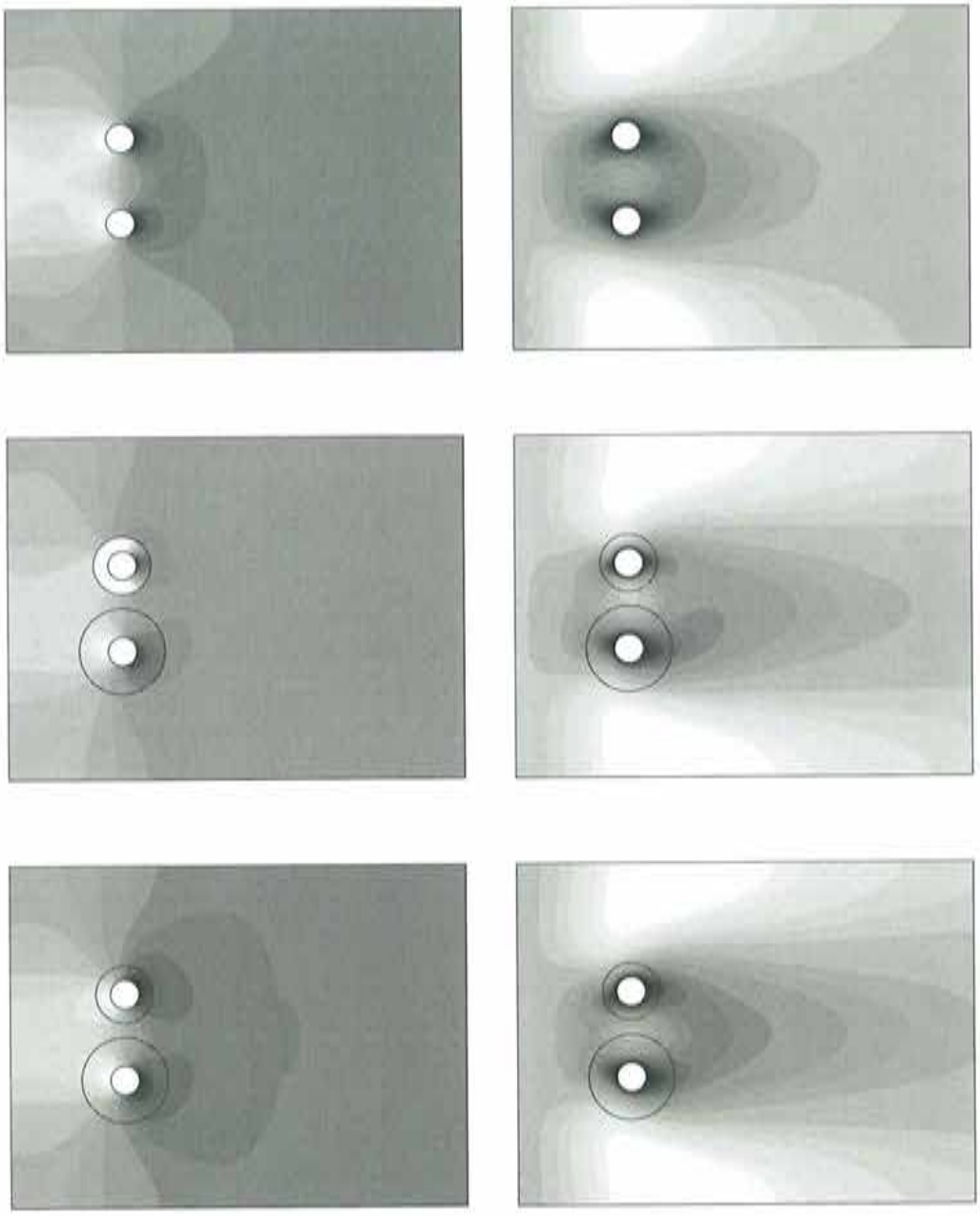


Figure 5.23: Flow past two cylinders. (Top) One-domain solution. (Mid.) C/D solution. (Bot.) C/N solution. (Left) Pressure. (Right) Velocity.

5.5.6 Summary

In this section, we have presented a classical Chimera method (Chimera/Dirichlet), as well as the extension of the Dirichlet/Neumann method in the Chimera context that we named the Chimera/Neumann method. The simple example outlined the deficiencies of the classical method and the possible advantages of the new method.

5.6 Moving subdomains

In this section, we apply the iteration-by-subdomain method and the Chimera method to the study of flows involving moving components [116, 117, 118]. We first overview the existing methods for tracking the relative movements of the different components of a mesh. These variables are only the position, the velocity and the strain rates involved in Dirichlet and Neumann transmission condition, as a scalar variable transforms into the same scalar. In particular, when the trajectories of the components are not known a-priori at any time, a second order integration scheme is derived to track the rotation of the frame of reference.

5.6.1 Overview

When one wants to simulate flows with moving bodies and when there is no possible way of prescribing simple boundary conditions in any frame of reference, four main alternatives to track the body motions are possible:

- the arbitrary-Lagragian-Eulerian (ALE) method together with an automatic remeshing technique of the computational domain adapts the fluid mesh to the spatial configuration in time;
- the fictitious domain method tracks moving solid boundaries inside a background mesh;
- the sliding mesh technique couples different meshes which are allowed to slide along their common interfaces;
- the Chimera method couples the individual meshes of each moving component.

These techniques are illustrated in Figure 5.24.

When using the ALE description of the flow together with automatic remeshing, the mesh accommodates the boundary displacements inside the computational domain (see for example [119, 120, 121]). On the one hand, if the displacements are small, only nodal displacement may be sufficient, and the nodal connectivity of the mesh remains unchanged; on the other hand, if the displacements are large, a complete remeshing is necessary. The main drawback of the method is that the geometric parameters have to be computed at each time iteration. See [122] for an example of application to the simulation of a mixed-flow pump. The ALE technique has also been used for following free surfaces [123, 124] and to simulate fluid/structure interactions [125].

In the fictitious domain method [94, 126], a fixed mesh occupies the whole volume including that occupied by the body. The method consists in including the boundary condition at the body boundary into the set of flow equations for the whole volume by the way of Lagrange multipliers. In the particular case of Dirichlet conditions imposed on the body, the Lagrange multiplier represents the jump in traction obtained at the fluid-solid interface. This method enables one to use simple (structured) background meshes on which fast solvers can be implemented. In [127], a fictitious domain method is presented to simulate two and three-dimensional flow problems with moving boundaries. The authors apply the method to the solution of a Couette problem and a helical

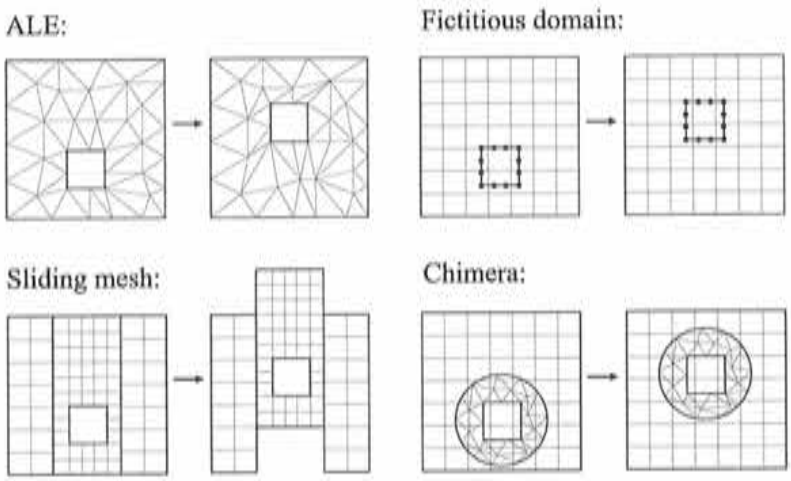


Figure 5.24: Illustration of the most common methods to simulate flows around moving components.

ribbon mixer; in [128], the fictitious method is applied to the solution of the flow around a moving disk. In fictitious domain methods, the motion of the object needs not to be known a-priori, and aerodynamic forces can be taken into account to couple the fluid dynamics and the kinematics of the rigid body. Pan [129] predicts the path of a ball falling in a viscous fluid (at low Reynolds numbers); in [130], the authors solve the two-dimensional flow around an airfoil that is free to rotate around its center of mass, the sedimentation of particles in a box, and a three-dimensional case involving two spherical particles. Using the same method, Suárez [131] simulates the sedimentation of an elliptic body in a two-dimensional viscous fluid. This fictitious domain method is also well-suited for shape optimization problems [132]. Although it has been shown that this method can efficiently solve the flow over moving objects, it presents a serious drawback: at large Reynolds numbers, we have no simple way to refine the mesh near the boundary without dropping the nice characteristic of the method.

The sliding mesh technique regroups DD methods for which two adjacent subdomains are allowed to slide along their common interface. In this work, we generalize this technique to any DD method involving possibly moving overlapping subdomains for which the interface topologies do not change with time. As it can be a hard task to ensure that the nodes of two adjacent sliding meshes coincide at each time step, the sliding mesh method is generally used as a direct application of the mortar method [93, 133] to moving subdomains [134]. The mortar element method is a non-conforming domain decomposition method for coupling non-matching grids. Instead of considering the continuity of the transmission variables point by point by using a simple interpolation technique, the mortar element method performs an interface L_2-projection of the transmission conditions. When the mortar method is used together with a sliding mesh technique, the subdomains are allowed to slide along their common interface. They are therefore necessarily disjoint. See [135] for the application of the sliding mesh technique to the simulation of stirred reactors. See [136] for the simulation of a two-dimensional rotor-stator interactions in a centrifugal pump. We also mention the Shear-Slip Mesh Update Method [137] (SSMUM) where regions in relative straight line translation or rotation are glued by the way of intermediate layers of elements, and where the connecting nodes coincide. In order to avoid remeshing of the regions, only the elements of intermediate layer are allowed to be deformed and its computational domain to be remeshed when necessary. The advantage of this method is that it is conservative as the composite mesh is always

conforming. The main drawback is that arbitrary motions are not possible.

The Chimera method appears to be the most flexible method to treat flow problems with moving bodies [138, 115, 139]. In addition, it fits perfectly within the Chimera based iteration-by-subdomain method introduced along this chapter. Each moving body of the domain is assigned a particular mesh. These meshes are allowed to move independently inside a background mesh, while the coupling can be performed using the Chimera/Dirichlet or Chimera/Neumann method. The main drawback of the Chimera method is its lack of conservation.

All these methods have their advantages and drawbacks that may be important or not according to the type of application we want to carry on. For example, conservation aspects are in general much more important in compressible fluid dynamics than in incompressible fluid dynamics (although we saw that mass conservation is crucial for the numerical scheme used in this work). If we only work on domains in relative rotational motion, the sliding mesh technique may be the most appropriate. However, the reliability (accuracy and robustness) of all these techniques is subjected to crucial choices. In the fictitious domain method, one has to choose the number of mesh points to discretize the rigid boundary; in the mortar element method, one has to find accurate quadrature rules to compute the integral matching condition [140]; in the Chimera method, conservative interpolation techniques may be expensive [141].

In both the sliding mesh technique and the Chimera method applied to moving subdomains, each subdomain solves the governing equations in its own frame of reference: these methods fall within the family of multiple frames of reference (MFR) techniques. They require tensorial transformation when updating the transmission conditions. We now address this point.

5.6.2 Tensorial transformations

If subdomains i and j are in relative motion, tensorial transformations must be performed each time a variable is obtained in i from j and when a host element is to be found. Denote $\boldsymbol{E}_k$ the basis vector in the k-th direction of an absolute frame of reference and $\boldsymbol{X}$ the coordinate vector of a point measured in it. Assume we know or we have a way to calculate the translation vector $\boldsymbol{T}_i$ and the rotation matrix $\boldsymbol{\Theta}_i$ of subdomain i as well as those of subdomain j ($\boldsymbol{T}_j$ and $\boldsymbol{\Theta}_j$ respectively), as shown in Figure 5.25.

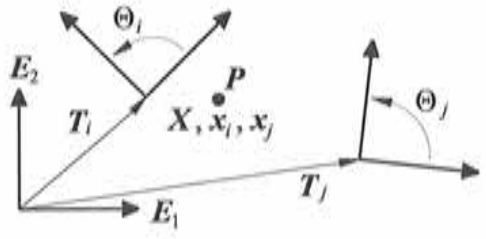


Figure 5.25: Two moving frames of reference in the absolute one.

Expressions for the position, velocity and strain rates

We want to express the position, the velocity and the strain rates in subdomain i in terms of the variables measured in j. We have

$$\begin{aligned} \boldsymbol{x}_j &= \Theta_j(\boldsymbol{X} - \boldsymbol{T}_j), \\ \boldsymbol{x}_i &= \Theta_i(\boldsymbol{X} - \boldsymbol{T}_i). \end{aligned}$$

Knowing that the rotation matrix is orthogonal, we easily get $\boldsymbol{x}_i$ in terms of $\boldsymbol{x}_j$:

$$\boldsymbol{x}_i = \Theta_i(\Theta_j^t \boldsymbol{x}_j + \boldsymbol{T}_j - \boldsymbol{T}_i). \tag{5.20}$$

Differentiating this equation with respect to time, we find the velocity $\boldsymbol{u}_i$ in terms of the velocity $\boldsymbol{u}_j$ measured in j:

$$\boldsymbol{u}_i = \dot{\boldsymbol{x}}_i = \dot{\Theta}_i(\Theta_j^t \boldsymbol{x}_j + \boldsymbol{T}_j - \boldsymbol{T}_i) + \Theta_i(\dot{\Theta}_j^t \boldsymbol{x}_j + \Theta_j^t \boldsymbol{u}_j + \dot{\boldsymbol{T}}_j - \dot{\boldsymbol{T}}_i), \tag{5.21}$$

where $\dot{(\cdot)} = d(\cdot)/dt$ and $\dot{\boldsymbol{T}}_j$ and $\dot{\boldsymbol{T}}_i$ are the velocities of subdomains j and i measured in the absolute frame of reference $\boldsymbol{E}_k$.

We now derive the transformation of the strain rates. We have

$$\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i} = \frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_j} \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_i}.$$

By substituting Equation (5.21), and knowing that

$$\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_i} = \Theta_j \Theta_i^t,$$

we obtain

$$\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i} = \dot{\Theta}_i \Theta_j^t \Theta_j \Theta_i^t + \Theta_i(\dot{\Theta}_j^t \Theta_j)\Theta_i^t + \Theta_i \Theta_j^t \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{x}_j} \Theta_j \Theta_i^t.$$

Due to the orthogonality of the rotation matrices, $\Theta_j^t \Theta_j = \boldsymbol{I}$ so the last equation gives

$$\begin{aligned} \frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i} &= \dot{\Theta}_i \Theta_i^t + \Theta_i(\dot{\Theta}_j^t \Theta_j)\Theta_i^t + \Theta_i \Theta_j^t \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{x}_j} \Theta_j \Theta_i^t, \qquad \text{and} \\ \left(\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i}\right)^t &= \Theta_i \dot{\Theta}_i^t + \Theta_i(\Theta_j^t \dot{\Theta}_j)\Theta_i^t + \Theta_i \Theta_j^t \left(\frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{x}_j}\right)^t \Theta_j \Theta_i^t. \end{aligned}$$

Now we add up the latter two equations and divide the result by two to obtain the equation for the strain rate:

$$\begin{aligned} \varepsilon(\boldsymbol{u}_i) &= \frac{1}{2}\left[\left(\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i}\right) + \left(\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i}\right)^t\right] \\ &= \frac{d}{dt}(\Theta_i \Theta_i^t) + \Theta_i \frac{d}{dt}(\Theta_j^t \Theta_j)\Theta_i^t + (\Theta_i \Theta_j^t)\varepsilon(\boldsymbol{u}_j)(\Theta_i \Theta_j^t)^t \\ &= \frac{d\boldsymbol{I}}{dt} + \Theta_i \frac{d\boldsymbol{I}}{dt}\Theta_i^t + (\Theta_i \Theta_j^t)\varepsilon(\boldsymbol{u}_j)(\Theta_i \Theta_j^t)^t. \end{aligned}$$

The first two terms are zero so we finally find that the velocity strain rate tensor transforms like:

$$\varepsilon(\boldsymbol{u}_i) = (\Theta_i \Theta_j^t)\varepsilon(\boldsymbol{u}_j)(\Theta_i \Theta_j^t)^t. \tag{5.22}$$

We observe that the velocity strain rate undergoes a rotation but no scaling, contrary to the velocity. We can also check that this expression is symmetric.

A second order scheme in time

In order to close the transformation of the position, velocity and strain rates expressed by Equations (5.20), (5.21) and (5.22), we need to compute the rotation matrices Θ_j and Θ_i at each time. If they are known, then we are done. If for example we only know the angular velocity (which is generally the case) and if in addition no analytical expression for the rotation matrix can be obtained at each time, then we may need to proceed. In some few cases we could obtain the rotation matrix, as illustrated by a simple example at the end of this subsection. Let us denote by Θ the rotation matrix of a frame of reference of basis vectors $\boldsymbol{e}_k$'s with respect to the absolute frame of reference such that

$$\Theta = \begin{bmatrix} \boldsymbol{e}_1^t \\ \boldsymbol{e}_2^t \\ \boldsymbol{e}_3^t \end{bmatrix}, \tag{5.23}$$

and assume we know the angular velocity vector $\boldsymbol{\omega}$ of the frame, which is a function of time. At each instant we have

$$\dot{\boldsymbol{e}}_k(t) = \boldsymbol{\omega}(t) \times \boldsymbol{e}_k(t) \tag{5.24}$$

$$= \boldsymbol{W}(t)\, \boldsymbol{e}_k(t), \tag{5.25}$$

where $W_{pq} = -\varepsilon_{pqr}\omega_r$ is given in the basis $\boldsymbol{E}_k$; ε_{pqr} is the permutation (alternating) tensor with value zero if two indices are repeated, and with value 1 or -1 if p, q, r are in cyclic order or not, respectively. Equation (5.24) has been already derived in Section 1.2.2; in the following, we use the matrix form given by Equation (5.25). Let us consider a partition $0 = t^0 < t^1 < \cdots < t^N = T$ of the time interval $[0, T]$ of interest. In order to integrate Equation (5.25), we propose the following approximation:

$$\dot{\tilde{\boldsymbol{e}}}_k(t) = (\boldsymbol{W}^n + \frac{1}{2}\dot{\boldsymbol{W}}^n \delta t)\tilde{\boldsymbol{e}}_k(t), \qquad \text{for } t^n \leq t \leq t^{n+1}, \tag{5.26}$$

where superscript n denotes variables considered at time t^n, the tilde indicates that the solution is approximated and $\delta t = t^{n+1} - t^n$. We are now going to show that the approximation given by Equation (5.26) is of second order in time. By direct integration of Equation (5.26), we find that

$$\tilde{\boldsymbol{e}}_k(t^{n+1}) = \exp(\boldsymbol{W}^n \delta t + \frac{1}{2}\dot{\boldsymbol{W}}^n \delta t^2)\tilde{\boldsymbol{e}}_k(t^n), \tag{5.27}$$

Let us develop the exact solution of equation (5.25) in Taylor series around time t^n:

$$\begin{aligned} \boldsymbol{e}_k(t^{n+1}) &= \boldsymbol{e}_k(t^n) + \dot{\boldsymbol{e}}_k(t^n)\delta t + \frac{1}{2}\ddot{\boldsymbol{e}}_k(t^n)\delta t^2 + \mathcal{O}(\delta t^3) \\ &= [\boldsymbol{I} + \boldsymbol{W}^n \delta t + \frac{1}{2}\dot{\boldsymbol{W}}^n \delta t^2 + \frac{1}{2}(\boldsymbol{W}^n)^2 \delta t^2]\boldsymbol{e}_k(t^n) + \mathcal{O}(\delta t^3). \end{aligned} \tag{5.28}$$

Performing the same expansion for the approximate solution given by equation (5.27), we get

$$\tilde{\boldsymbol{e}}_k(t^{n+1}) = [\boldsymbol{I} + \boldsymbol{W}^n\delta t + \frac{1}{2}\dot{\boldsymbol{W}}^n\delta t^2 + \frac{1}{2}(\boldsymbol{W}^n)^2\delta t^2 + \mathcal{O}(\delta t^3)]\tilde{\boldsymbol{e}}_k(t^n) + \mathcal{O}(\delta t^3). \tag{5.29}$$

Let

$$\boldsymbol{A}^n = \boldsymbol{I} + \boldsymbol{W}^n\delta t + \frac{1}{2}\dot{\boldsymbol{W}}^n\delta t^2 + \frac{1}{2}(\boldsymbol{W}^n)^2\delta t^2.$$

From equations (5.29) and (5.28), we have

$$\begin{aligned}\boldsymbol{e}_k(t^{n+1}) &= \boldsymbol{A}^n\boldsymbol{e}_k(t^n) + \mathcal{O}(\delta t^3),\\ \tilde{\boldsymbol{e}}_k(t^{n+1}) &= \boldsymbol{A}^n\tilde{\boldsymbol{e}}_k(t^n) + \mathcal{O}(\delta t^3).\end{aligned}$$

Therefore

$$\begin{aligned}\tilde{\boldsymbol{e}}_k(t^{n+1}) - \boldsymbol{e}_k(t^{n+1}) &= \boldsymbol{A}^n(\tilde{\boldsymbol{e}}_k(t^n) - \boldsymbol{e}_k(t^n)) + \mathcal{O}(\delta t^3)\\ &= \boldsymbol{A}^n\boldsymbol{A}^{n-1}(\tilde{\boldsymbol{e}}_k(t^{n-1}) - \boldsymbol{e}_k(t^{n-1})) + \mathcal{O}(\delta t^3) + \mathcal{O}(\delta t^3)\\ &\vdots\\ &= \boldsymbol{A}^n\boldsymbol{A}^{n-1}\cdots\boldsymbol{A}^0(\tilde{\boldsymbol{e}}_k(t^0) - \boldsymbol{e}_k(t^0)) + \mathcal{O}(\delta t^2).\end{aligned}$$

Assuming the basis vectors are given at $t = 0$, we have $\tilde{\boldsymbol{e}}_k(t^0) - \boldsymbol{e}_k(t^0) = 0$. Therefore, we have that $\tilde{\boldsymbol{e}}_k(t^{n+1}) - \boldsymbol{e}_k(t^{n+1}) = \mathcal{O}(\delta t^2)$.

In order to find the $\tilde{\boldsymbol{e}}_k$'s at t^{n+1}, we apply equation (5.27) recursively:

$$\tilde{\boldsymbol{e}}_k(t^{n+1}) = \exp\left[\sum_{m=0}^{n}\left(\boldsymbol{W}^m\delta t + \frac{1}{2}\dot{\boldsymbol{W}}^m\delta t^2\right)\right]\boldsymbol{E}_k \tag{5.30}$$

$$= (\boldsymbol{\Theta}^n)^t\boldsymbol{E}_k \qquad \text{(by definition)}, \tag{5.31}$$

where we have assumed that $\tilde{\boldsymbol{e}}_k^0 = \boldsymbol{E}_k$. The last expression for the rotation matrix is not convenient, so we try to derive a nicer equation for the coefficients of $\boldsymbol{\Theta}^n$ at time t^n. By definition, we have

$$\begin{aligned}W_{pq}^m &= -\varepsilon_{pqr}\omega_r^m,\\ \dot{W}_{pq}^m &= -\varepsilon_{pqr}\dot{\omega}_r^m.\end{aligned}$$

Let $\boldsymbol{B}$ be the argument matrix of the exponential function of Equation (5.30), that is,

$$\boldsymbol{B} = \sum_{m=0}^{n}\left(\boldsymbol{W}^m\delta t + \frac{1}{2}\dot{\boldsymbol{W}}^m\delta t^2\right),$$

whose coefficients are

$$B_{pq} = -\varepsilon_{pqr}\sum_{m=0}^{n}\left(\omega_r^m\delta t + \frac{1}{2}\dot{\omega}_r^m\delta t^2\right).$$

We define the vector $\boldsymbol{r}^n$ and the unit vector $\hat{\boldsymbol{r}}^n$ as

$$\begin{aligned} \boldsymbol{r}^n &= \sum_{m=0}^{n} \left(\omega^m \delta t + \frac{1}{2} \dot{\omega}^m \delta t^2 \right), \\ \hat{\boldsymbol{r}}^n &= \frac{\boldsymbol{r}^n}{|\boldsymbol{r}^n|}, \end{aligned}$$

so the coefficients of matrix $\boldsymbol{B}$ become

$$B_{pq} = -\varepsilon_{pqr} \hat{r}_r^n |\boldsymbol{r}^n|.$$

Let us introduce a matrix $\boldsymbol{C}$ and a scalar θ such that

$$\begin{aligned} C_{pq} &= -\varepsilon_{pqr} \hat{r}_r^n, \\ \theta &= |\boldsymbol{r}^n|. \end{aligned} \tag{5.32}$$

According to these definitions, Equation (5.30) can be re-written as

$$\bar{\boldsymbol{e}}_k(t^{n+1}) = \exp(\theta \boldsymbol{C}) \boldsymbol{E}_k,$$

In addition, it can be shown that for a matrix $\boldsymbol{C}$ given by (5.32) with $\hat{\boldsymbol{r}}^n$ being a unit vector, we have

$$\begin{aligned} \left(\exp(\theta \boldsymbol{C})\right)_{pq} &= \hat{r}_p^n \hat{r}_q^n + (\delta_{pq} - \hat{r}_p^n \hat{r}_q^n) \cos\theta + C_{pq} \sin\theta \\ &= \hat{r}_p^n \hat{r}_q^n + (\delta_{pq} - \hat{r}_p^n \hat{r}_q^n) \cos|\boldsymbol{r}^n| - \varepsilon_{pqr} \hat{r}_r^n \sin|\boldsymbol{r}^n|. \end{aligned}$$

By definition of the rotation matrix $\boldsymbol{\Theta}^n$ we have

$$(\boldsymbol{\Theta}^n) = \left(\exp(\theta \boldsymbol{C})\right)^t,$$

so the coefficients of the previous exponential form are given by:

$$\Theta_{pq}^n = \hat{r}_p^n \hat{r}_q^n + (\delta_{pq} - \hat{r}_p^n \hat{r}_q^n) \cos|\boldsymbol{r}^n| + \varepsilon_{pqr} \hat{r}_r^n \sin|\boldsymbol{r}^n|. \tag{5.33}$$

We recognize here the well-known expression for the matrix coefficient of a rotation through an angle $|\boldsymbol{r}^n|$ about an axis whose direction is given by the unit vector $\hat{\boldsymbol{r}}$. It is interesting to note that the time step contributions to the rotation add up in such a simple way...

The derivative of the rotation function is given by

$$\dot{\boldsymbol{\Theta}}^n = \begin{bmatrix} \dot{\bar{\boldsymbol{e}}}_1^t(t^n) \\ \dot{\bar{\boldsymbol{e}}}_2^t(t^n) \\ \dot{\bar{\boldsymbol{e}}}_3^t(t^n) \end{bmatrix}.$$

Using (5.26) evaluated at time t^{n+1} and (5.31) it is found that

$$\dot{\bar{\boldsymbol{e}}}_k(t^{n+1}) = \left(\boldsymbol{W}^n + \frac{1}{2} \dot{\boldsymbol{W}}^n \delta t \right) (\boldsymbol{\Theta}^n)^t \boldsymbol{E}_k,$$

from where it follows that

$$\dot{\Theta}^{n+1} = \Theta^n \left(W^n + \frac{1}{2} \dot{W}^n \delta t \right)^t. \tag{5.34}$$

A simple two-dimensional example

In order to get an insight on the meaning of the tensorial formulation for the position and the velocity as expressed by Equations (5.20) and (5.21), we consider a simple two-dimensional case. Let us assume that subdomain i is rotating with a constant angular velocity $\omega = [0, 0, \omega]^t$ while subdomain j is fixed, and that their axes initially coincide. A simple calculation gives at each time t the following transformation for the position,

$$x_i = \Theta_i x_j = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix} x_j.$$

From Equation (5.21) and deriving the last equation with respect to time, we obtain that the velocity transforms like

$$\begin{aligned} u_i &= \dot{\Theta}_i x_j + \Theta_i u_j \\ &= \omega \begin{bmatrix} -\sin(\omega t) & \cos(\omega t) \\ -\cos(\omega t) & -\sin(\omega t) \end{bmatrix} x_j + \Theta_i u_j \\ &= -\omega \times x_i + \Theta_i u_j. \end{aligned}$$

The first term of the last equation is the scaling due to the relative rotations, while the second term is the velocity measured in j, and rotated as seen in subdomain i.

The algorithm

Algorithm 5.9 gives the steps to follow when updating the Dirichlet transmission condition of subdomain i, in relative motion with subdomain j. The dots hold for the usual calculations to be performed by the Master, and are not repeated here for the sake of clarity. Note that if a conservation algorithm is to be used, then we only have to replace $I^{ij}(u_j)$ by its conservative form, using Equation (5.15) for the constrained interpolation or Equation (5.19) for the NTI operator.

Algorithm 5.10 gives the steps to follow when updating the Neumann transmission condition of subdomain i, in relative motion with subdomain j. As in the case of Dirichlet transmission conditions, the interpolation of the strain rates and the pressure can be constrained.

5.6.3 Examples

We now present two examples of applications. The first example is a classical iteration-by-subdomain DD method applied to moving grids while the second example illustrates the Chimera method applied to moving grids. The first example studies a section of a chemical reactor and the second one studies the flow past a cylinder submitted to rotational and translational velocities.

Before going on to the examples, it should be pointed out that when coupling two subdomains, one steady and the other in an accelerated frame of reference, we must include all the acceleration terms of the Navier-Stokes equations; i.e. we cannot couple the subdomains using the Stokes equations. This is due to the fact that the non-inertial effects accounted for in the accelerated

Algorithm 5.9 Update of a Dirichlet transmission condition on a moving subdomain

for all time steps **do**
  Compute Θ_i, Θ_j, T_i, T_j and their time derivatives, using Equations (5.33) and (5.34)
  ...
  for all interface nodes `ipoin` **do**
    Find the host element `jelem` of `ipoin` of coordinates x_i in Ω_i and x_j in Ω_j:

$$x_j = \Theta_j(\Theta_i^t x_i + T_j - T_i)$$

    Compute the interpolated velocity $I^{ij}(u_j)$ at x_j
    Obtain the velocity at `ipoin` as:

$$u_i = \dot{\Theta}_i(\Theta_j^t x_j + T_j - T_i) + \Theta_i(\dot{\Theta}_j^t x_j + \Theta_j^t I^{ij}(u_j) + \dot{T}_j - \dot{T}_i)$$

  end for
  ...
end for

Algorithm 5.10 Update of Neumann transmission condition on a moving subdomain

for all time steps **do**
  Compute Θ_i, Θ_j and their time derivatives, using Equations (5.33) and (5.34)
  ...
  for all boundary elements `iboun` **do**
    for all integration points `igaub` **do**
      Find the host element `jelem` of `igaub` of coordinates x_i in Ω_i and x_j in Ω_j:

$$x_j = \Theta_j(\Theta_i^t x_i + T_j - T_i)$$

      Compute the velocity strain rates $I^{ij}(\varepsilon(u_j))$ at `igaub` using either the classical or least-square interpolation
      Compute the interpolated pressure $I^{ij}(p_j)$ at `igaub`
      Obtain the pressure and strain rates at `ipoin` as:

$$p_i = I^{ij}(p_j)$$
$$\varepsilon(u_i) = (\Theta_j\Theta_i^t) I^{ij}(\varepsilon(u_j))(\Theta_j\Theta_i^t)^t$$

      ...
    end for
  end for
  ...
end for

frame of reference come from the total derivative appearing in Galilean frame of reference. In each subdomain, there may be some dominant terms, but once we want to couple them, all the terms have to be considered to avoid any physical inconsistency.

As an illustration, consider the domain shown in Figure 5.6 rotating with angular velocity $\omega = [0, 0, \omega]^t$, and with $\nu = 10^5$. We first solve the problem in the rotating frame of reference and

look for the solution of zero velocity $[u, v]^t$ across the domain and centrifugal pressure, by imposing a zero velocity on the contour. In an excess of confidence, we neglect the convective terms; the solution is exactly the same with and without the convective terms. In fact, both the Stokes and Navier-Stokes equations in the Cartesian frame of reference give (with evident notation)

$$\frac{\partial p}{\partial x} = \omega^2 x,$$
$$\frac{\partial p}{\partial y} = \omega^2 y,$$

so that we have $p = \frac{1}{2}\omega^2(x^2 + y^2)$. We now want to reproduce the same flow in a steady frame of reference, i.e. we look for a velocity $-\boldsymbol{\omega} \times \boldsymbol{x} = [-\omega y, \omega x]^t$, where $\boldsymbol{x} = [x, y]^t$ is the position vector. The steady Stokes equations yield

$$\frac{\partial p}{\partial x} = 0,$$
$$\frac{\partial p}{\partial y} = 0,$$

which solution is constant pressure! Where the centrifugal force effects have disappeared? Consider now the full Navier-Stokes equations; we have

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} = 0,$$
$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial x} + \frac{\partial p}{\partial y} = 0,$$

which upon substitution of $[u, v]^t = [-\omega y, \omega x]^t$ gives the correct answer for the pressure. We now present two numerical examples.

Example 1: stirred tank

We solve a two-dimensional section of a stirred tank, used in the chemical industry, and is made of six blades of width H_1 and radius R_3. The inner subdomain is shown in Figure 5.26 (Left) and the outer subdomain is shown in Figure 5.26 (Right). It includes four baffles of width H_2. The dimensions of the problem are

$$R_1 = 22.5,\ R_2 = 44.5,\ R_3 = 60.0,\ H_1 = 2.0,$$
$$R_4 = 54.0,\ R_5 = 88.5,\ H_3 = 4.0.$$

We solve the transient Navier-Stokes equations with the ASGS model, the time integration being carried out using the backward Euler scheme with $\delta t = 0.5$. The inner subdomain is meshed with 1575 P1/P1 elements, and the outer subdomain with 2105 P1/P1 elements. We choose $|\boldsymbol{\omega}| = 0.1$ and $\nu = 50$ so that the Reynolds and Ekman numbers are

$$\mathrm{Re} = \frac{|\boldsymbol{\omega}| R_2^2}{\nu} = 3.96, \qquad \mathrm{Ek} = \frac{\nu}{2|\boldsymbol{\omega}| R_2^2} = 0.13,$$

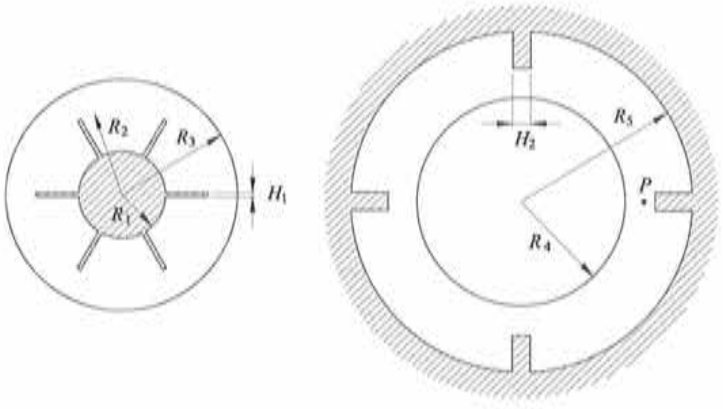


Figure 5.26: Stirred tank. (Left) Inner subdomain. (Right) Outer subdomain.

where we have taken as characteristic velocity the tip velocity i.e. $|\omega|R_2$. The inner subdomain is solved in the non-inertial frame of reference, of angular velocity $\omega = [0, 0, 0.1]^t$, while the outer subdomain is solved in a fixed frame of reference. The subdomains are coupled using both the Schwarz and the Dirichlet/Neumann methods with one element layer of overlap. In the case of the Dirichlet/Neumann method, the inner subdomain is assigned the Neumann condition calculated with the least-square interplation scheme. The interfaces of Dirichlet type are constrained by the mass conservation equation.

As initial conditions, we perform a steady state calculation using the MFR method, i.e. we perform the tensorial transformation but keep the subdomains at their original position. Figure 5.27 (Left) shows the vertical velocity of point P located near a baffle; we first observe that the pseudo steady state solutions of the two DD methods are different.

Figure 5.27 (Right) presents the power spectrum of a fast Fourier transform performed on the time signal. The spectra are normalized so that their integrals are both unity. Both solutions exhibit a peak around the blade passing frequency, i.e. 10.47 time units. We observe that the D/N gives a more sharp peak, and less high frequencies. Finally, Figure 5.28 shows a detail of the mesh, the pressure distribution in the whole domain, as well as the velocity vector near point P for different times $t = 5, 7.5, 10, 12.5$, obtained by the D/N method. Note that the velocity vectors are shown in the fixed frame of reference.

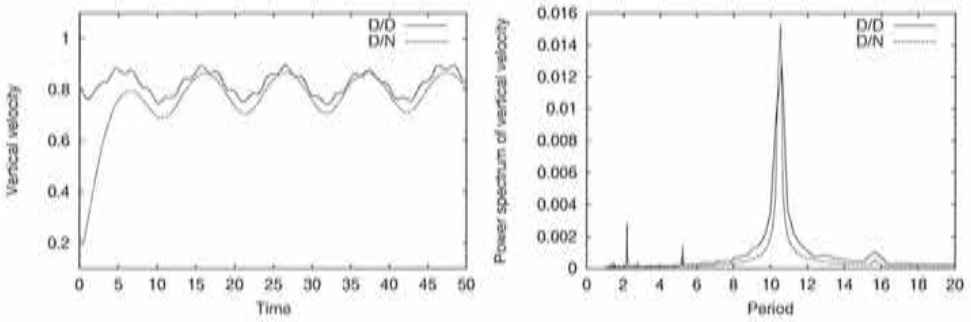


Figure 5.27: Stirred tank. (Left) Vertical velocity of point P. (Right) Power spectrum.

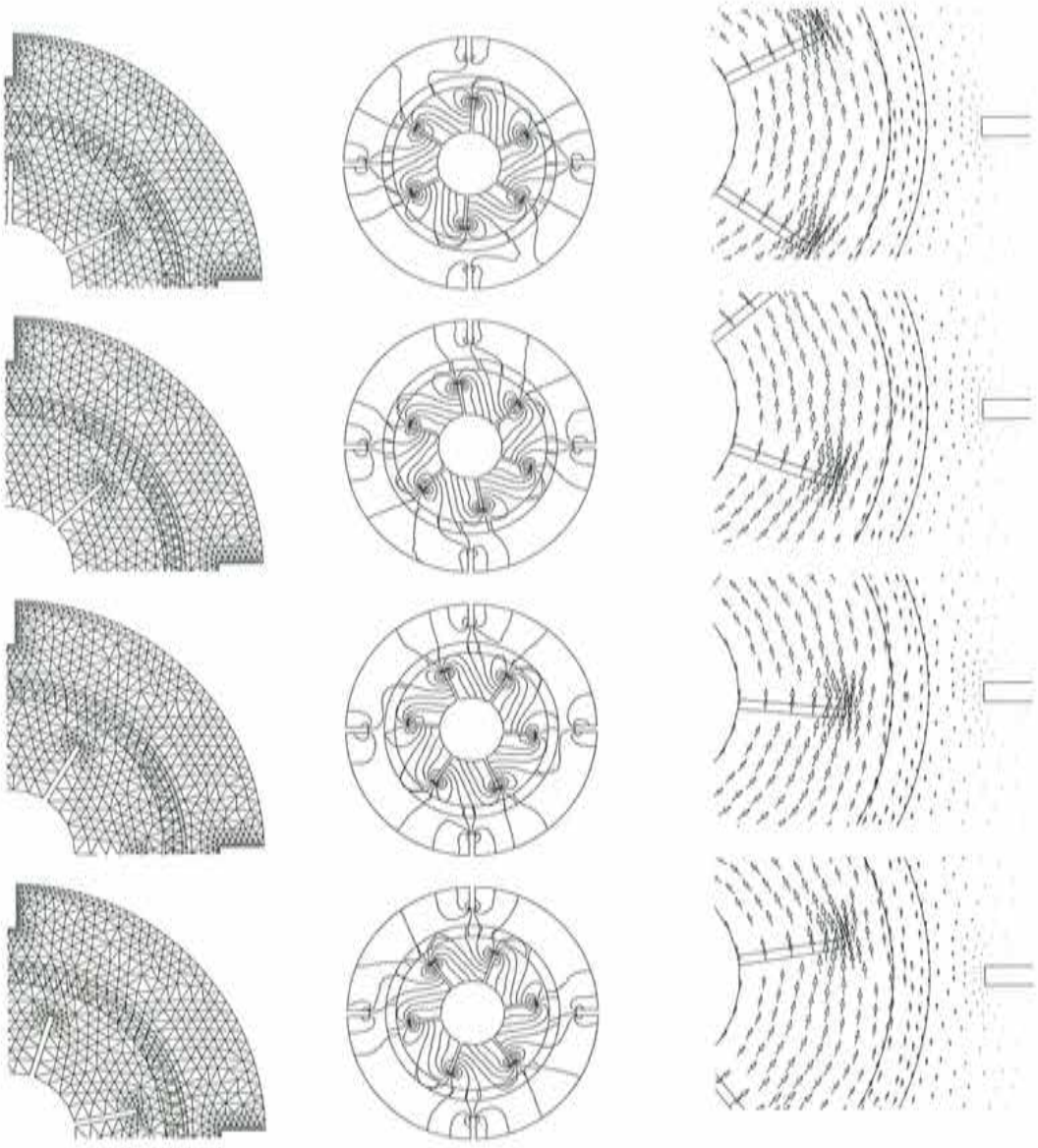


Figure 5.28: Stirred tank. (Left) Detail of the composite mesh. (Mid.) Pressure contours. (Right) Velocity. From top to bottom, $t = 5.0$, $t = 7.5$, $t = 10.0$, $t = 12.5$.

Example 2: moving disk

We simulate the example presented in [128]. It is a two-dimensional flow confined in a square domain $\Omega = (-0.35, 0.9) \times (-0.5, 0.5)$. A circle of radius 0.125 is moving with a trajectory

$$x = \frac{1}{4}(1 - \cos(\pi t/2)),$$

$$y = -0.1 \sin(\pi(1 - \cos(\pi t/2))),$$

and is rotating counterclockwise with an angular velocity 2π. The center of the circle is traveling back and forth from $x = 0$ to $x = 0.5$ and oscillates around the position $y = 0$. Knowing its trajectory in time, we can calculate its linear and angular velocity as well as its linear acceleration exactly. The background mesh is meshed with 1600 Q1/Q1 elements while the patch mesh is meshed with 400 Q1/Q1 elements. Note that in our case the background mesh is much coarser than the numerical reference as their coarsest mesh has 20480 elements. This flow is solved using the Chimera/Neumann method with least-square smoothing and the Euler time integration scheme. As boundary conditions, we impose a no-slip condition on the cylinder and zero velocity on the boundary of the rectangle. Figure 5.29 shows the solution obtained at $t = 5$, $t = 6$, $t = 7$ and $t = 8$. The left hand side column shows the hole and the interface of the patch mesh.

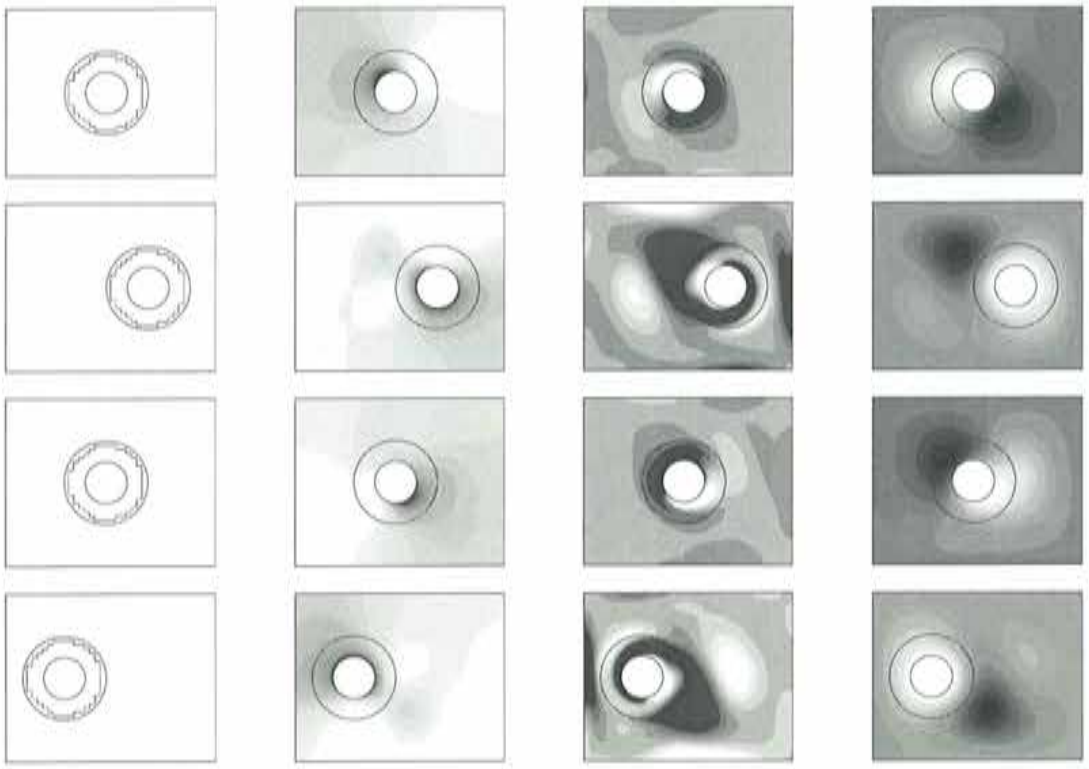


Figure 5.29: Moving disk. From left to right, hole and interfaces, pressure, vorticity, streamlines. From top to bottom, $t = 5$, $t = 6$, $t = 7$, $t = 8$.

Chapter 6

Numerical Applications

In this chapter we present numerical applications of mixed methods on overlapping subdomains. In the first example, we compare the performances of some Chimera methods, namely the C/D, C/N and C/R methods, by solving the vortex shedding behind a cylinder. In all the following examples we drop the study of the C/R and C/D methods and only consider the C/N method. In addition, the Neumann transmission conditions are calculated using the least-square smoothing. In the second example, we solve a turbulent backward facing step to show the good convergence of the D/N iteration-by-subdomain algorithm. The interfaces are deliberately located inside the recirculation zone so that the interfaces have inflow and outflow parts. In the third example we solve the flow around a moving missile using the C/N method: the results are compared to those obtained with an ALE approach. In the fourth example we solve a transient and turbulent flow in a two-dimensional section of a centrifugal fan. The last two examples are three-dimensional transient applications. In the fifth one, we obtain the solution of the laminar flow in an axial stirred tank. The last example is an axial fan.

6.1 Vortex shedding behind a cylinder

This example involves the flow past a cylinder, a widely solved benchmark problem. A circular cylinder is immersed in a viscous fluid. The Reynolds number is based on the cylinder diameter D and the prescribed uniform inflow velocity U. The geometry and boundary conditions are shown in Figure 6.1. The exterior domain is a rectangle $(0, 16) \times (0, 8)$ and we set $U = 1$ and $D = 1$.

For Re approximately less than 40, two symmetrical eddies develop behind the cylinder. These eddies become unstable at higher Reynolds numbers and periodic vortex shedding occurs, leading to the so-called Von Karman vortex street. We first consider the stationary state at Re = 30. As a reference solution, we solve the steady laminar flow on a relatively fine mesh composed of 5400 Q1/Q1 elements, shown in Figure 6.2 (Left). We want to compare here the results obtained with three Chimera methods, the C/D, C/N and C/R methods. As a background mesh, we use a structured mesh composed of 1600 Q1/Q1 elements. The patch mesh contains the cylinder. Its outer boundary, i.e. the interface of the DD method, is a circle of diameter 3. Its mesh is composed of 400 Q1/Q1 elements. The resulting composite mesh is shown in Figure 6.2 (Right). Figures 6.3 shows a close up of the composite mesh in the cylinder region and the results of the hole cutting operation. The middle figure shows the hole created to obtain a one element overlap one each subdomain. This composite mesh is used for the C/D method as well as for the C/N and C/R methods in order to achieve a second order method. The right figure shows the hole created with

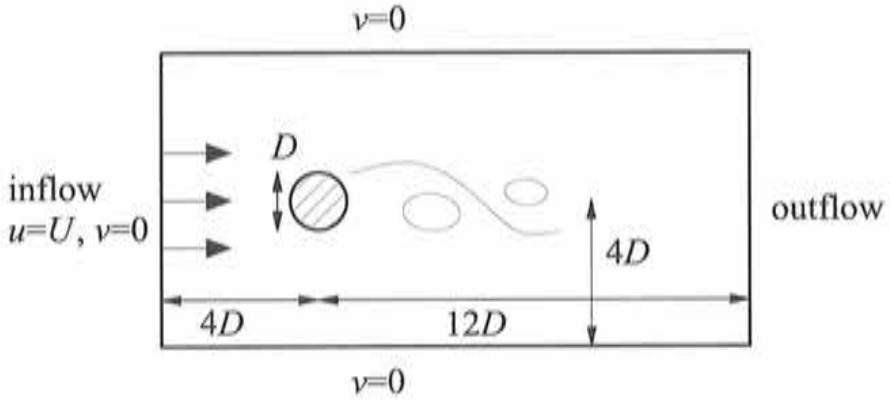


Figure 6.1: Vortex shedding. Geometry.

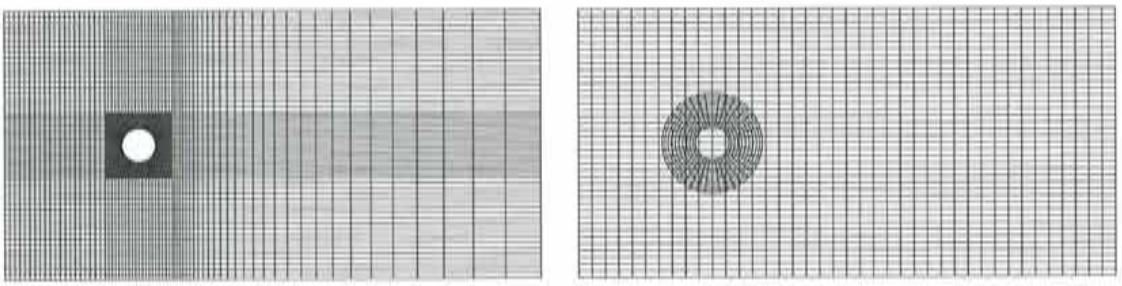


Figure 6.2: Vortex shedding. Meshes. (Left) Fine mesh used for one-domain solution. (Right) Composite mesh of the Chimera method.

a zero overlap, usefull for the C/R and C/N methods together with the classical interpolation. To solve the stationary problem, we employ the Chimera method with one element overlap on

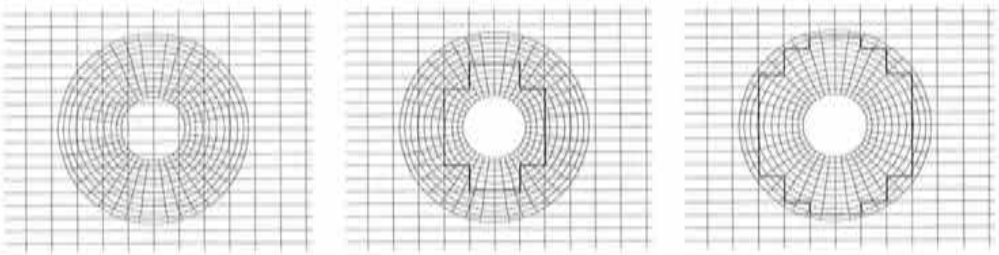


Figure 6.3: Vortex shedding. Zoom of meshes. (Left) Composite mesh. (Mid.) Hole cutting for one element overlap. (Right) Hole cutting for zero overlap.

each subdomain. The Chimera method with zero overlap will be used only for the transient case. Note that when considering the C/D method, the patch subdomain is confined. Therefore in order to have a well-posed problem on the patch subdomain at each iteration, we apply the interface

constraining of the mass conservation [102].

The test we now carry out consists in determining the range of relaxation parameters for which the algorithm converges. To do so, we vary the relaxation parameters of both transmission conditions from 0.1 to 2, using a increment step of 0.1. The C/D turns out to be the most robust method, i.e. the method for which we have the greatest amplitude in the choice of relaxation parameters to achieve convergence. The C/N does not converge at all, at least for the range of parameters tested. The C/R method converges but for a restricted area in the relaxation space, as shown in Figure 6.4 (Left), where θ_D refers to the relaxation parameter of the Dirichlet condition and θ_R refers to that of the Robin condition. Figure 6.4 (Right) compares the convergence histories obtained with the C/D and C/R methods. For the C/D method no relaxation is used while for the C/R method, we use $\theta_D = \theta_R = 0.2$. The figure shows that the convergence of the C/D method looks like monotone while that of the C/R is more unstable. However, the residuals of the Dirichlet data obtained with both methods are of the same order after 30 iterations.

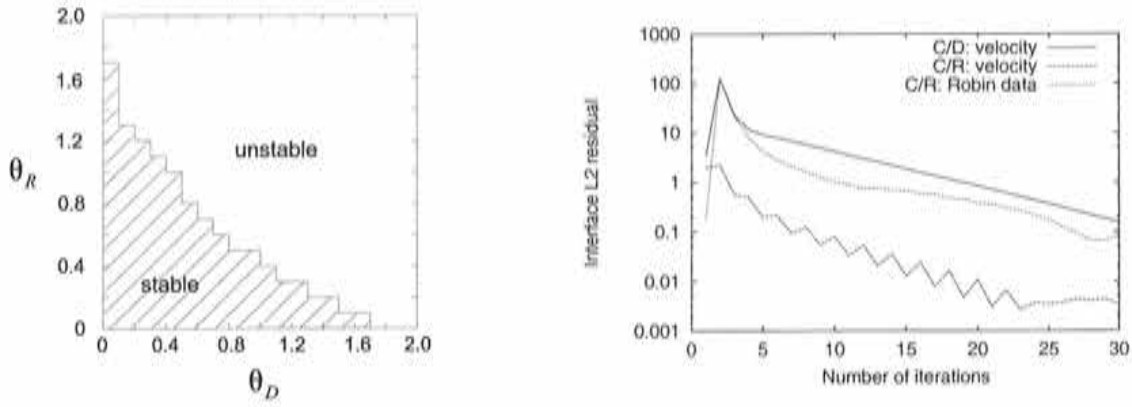


Figure 6.4: Vortex shedding. (Left) Stability curve of the C/R method. (Right) Convergence histories of C/D without relaxation and C/R with $\theta_D = \theta_R = 0.2$.

Now, we have previously mentioned in Section 3.7 that a reaction type term in the ADR equation can help mixed DD method to be stable; so maybe we have a way to make the C/N method converge. Let us solve the transient problem. In order to control as few parameters as possible in the iterative process, we couple the time, linearization and DD loops. Figures 6.5 (Left) and (Right) shows the convergence histories obtained with the C/N method for different relaxation parameters and time steps. We observe a very good convergence of the algorithm and a quite large flexibility in the choice of the parameters (relaxation and time step) to control the convergence of the DD method.

Figures 6.6 compares the streamwise velocity and pressure profiles along a horizontal cut and a vertical cut, both passing by the center of the cylinder. We observe good agreements of the mixed methods with the reference solution; on the contrary, the solution of the C/D method differs notably from that of the reference solution.

We now go on to the transient case and set $\mathrm{Re} = 100$. Although the flow is unstable at this Reynolds number, one can obtain a steady solution. This solution is used as initial condition of the transient simulation, on which we superimpose a small vortex near the cylinder. This is sufficient to trigger the unsteady state. The time integration is carried out with the backward Euler scheme and $\delta t = 0.1$.

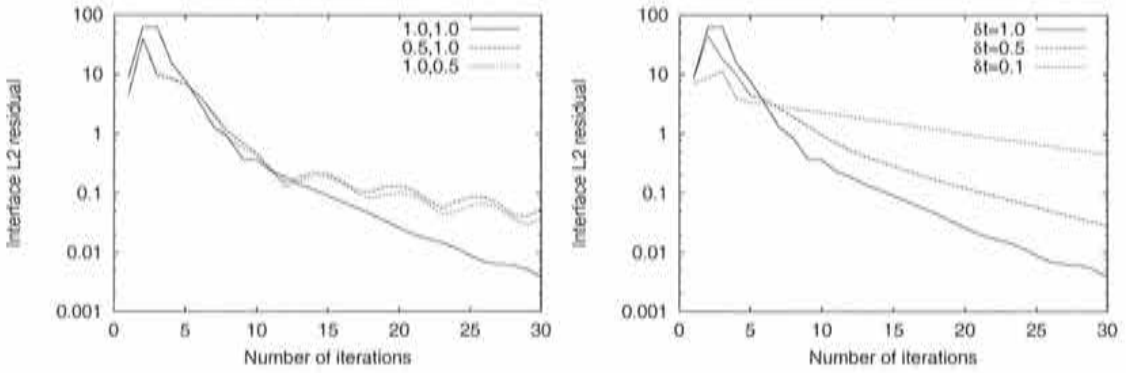


Figure 6.5: Vortex shedding. Velocity convergence history of C/N. (Left) For different relaxation parameters. (Right) For different time steps.

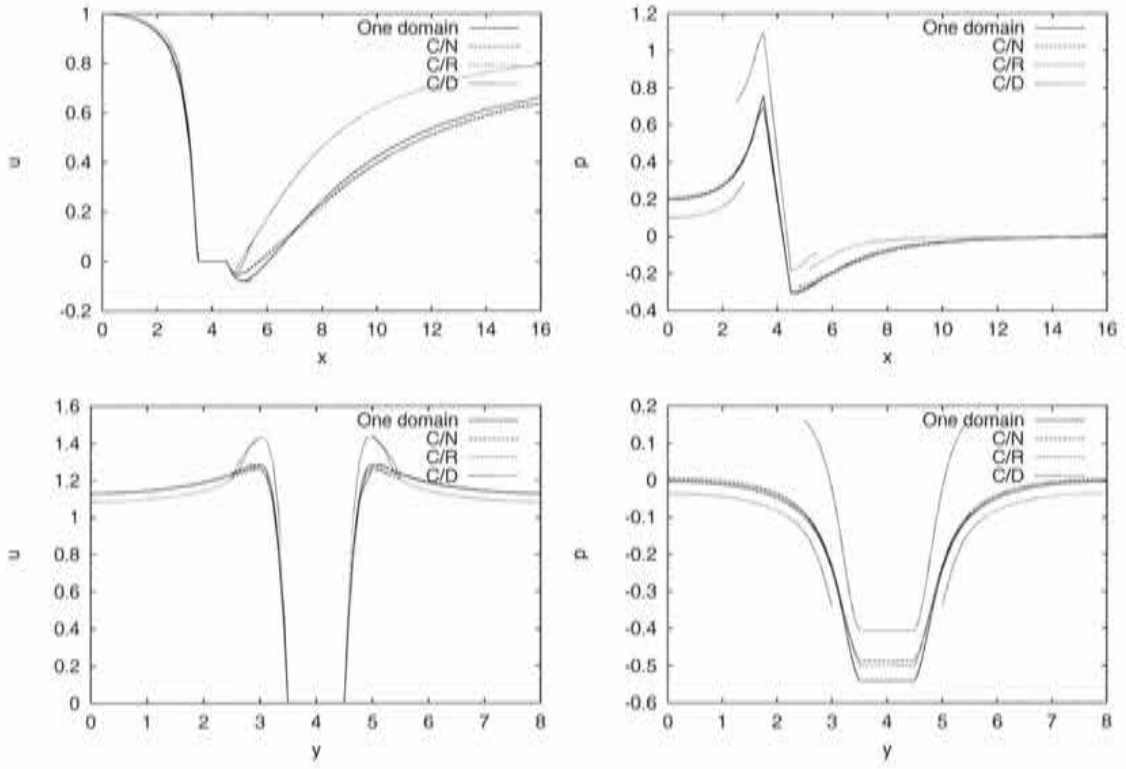


Figure 6.6: Vortex shedding. (Top) Horizontal cut. (Bot.) Vertical cut. (Left) Velocity. (Right) Pressure.

As comparison criterions, we calculate the period and amplitude of the vertical pressure force acting on the cylinder. Numerical references report values of the period between 5.6 and 6.0. See for example [142]. We test the C/D method using the one element overlap and the mixed methods using the one element and zero overlaps. The values of the amplitudes and frequencies are reported in table 6.1. As a reference, we also indicate the results obtained with the fine mesh.

The C/R method gives the best results. For this very coarse mesh, the C/N method does not perform so well, and does not converge at all when using only the first order interpolation with overlap. This is due to the fact that at high Reynolds number, the weak continuity of the velocity derivatives looses weight with respect to the weak continuity of the pressure when imposing a natural (Neumann) condition. The discontinuity in velocity is corrected by the Robin condition which provides additional weight to the continuity of the velocity components, in the form of a weak Dirichlet condition. Remember that for Re = 30, we observed that the C/N and C/R methods gives similar solutions. The C/D is much more diffusive than the mixed methods in amplitude, while the time frequency is slightly better.

			With overlap				No overlap	
			least-square		classical		classical	
	One-domain	C/D	C/N	C/R	C/N	C/R	C/N	C/R
Amplitude	0.101	0.029	0.060	0.074	-	0.074	0.075	0.074
Period	5.822	6.044	6.444	6.400	-	6.400	6.400	6.356

Table 6.1: Vortex shedding. Amplitude and period of the vertical pressure force.

6.2 Backward facing step

We solve the turbulent backward facing step already presented in Section 1.4.2, but this time using the overlapping D/N method. For the notation, refer to this section. We decompose vertically the domain into two overlapping subdomains, which interfaces fall inside the recirculation zone. The meshes of each subdomain are such that they approximately mimic the mesh used to compute the one domain solution, namely Mesh 2. The left-hand side subdomain is meshed with 800 Q1/Q1 elements and the right-hand side subdomain with 1600 Q1/Q1 elements. Remember that Mesh 2 has 2000 Q1/Q1 elements. A zoom around the step corner of the composite mesh and Mesh 2 are shown in Figures 6.8 (Top) (Left) and (Top) (Right), respectively. The problem is solved with the following data: $\hat{y}/H = 4\%$, $\nu_{t,\infty}/\nu = 100$ and zero velocity at the step corner. This corresponds to the third line of Table 1.2.

The interface of the left subdomain is of Neumann type while that of the right subdomain is of Dirichlet type. Figures 6.7 (Left) and (Right) show the convergence history obtained for two combinations of relaxation parameters. Parameter θ_{D} is the relaxation factor of the Dirichlet conditions, used to update the velocity and the eddy-viscosity, while θ_{N} is that of the Neumann condition, used to update the traction and flux of eddy-viscosity. Both combinations lead to convergence of the D/N method. Figure 6.8 compares the contours of velocity module and eddy-viscosity obtained with the D/N method and on the one domain solution. The profiles are identical and confirm the convergence of the D/N method.

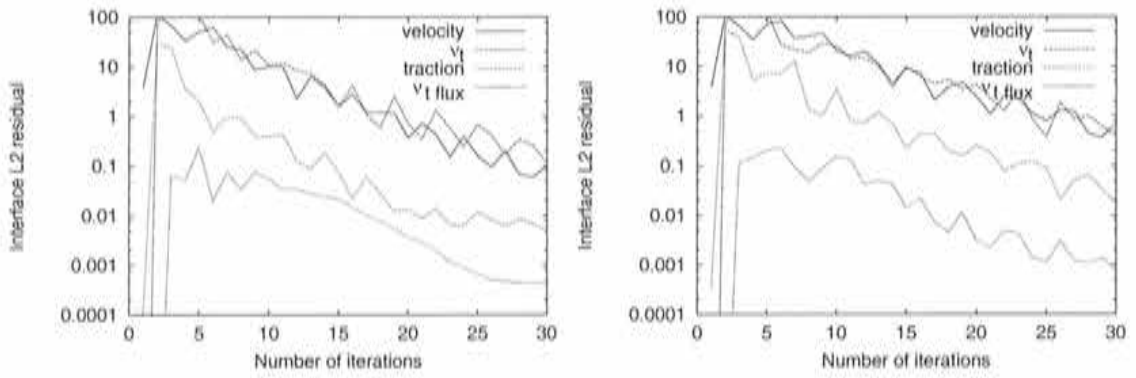


Figure 6.7: Backward facing step. Convergence history. (Left) $\theta_D = 1.0$, $\theta_N = 0.3$. (Right) $\theta_D = 1.0$, $\theta_N = 1.0$.

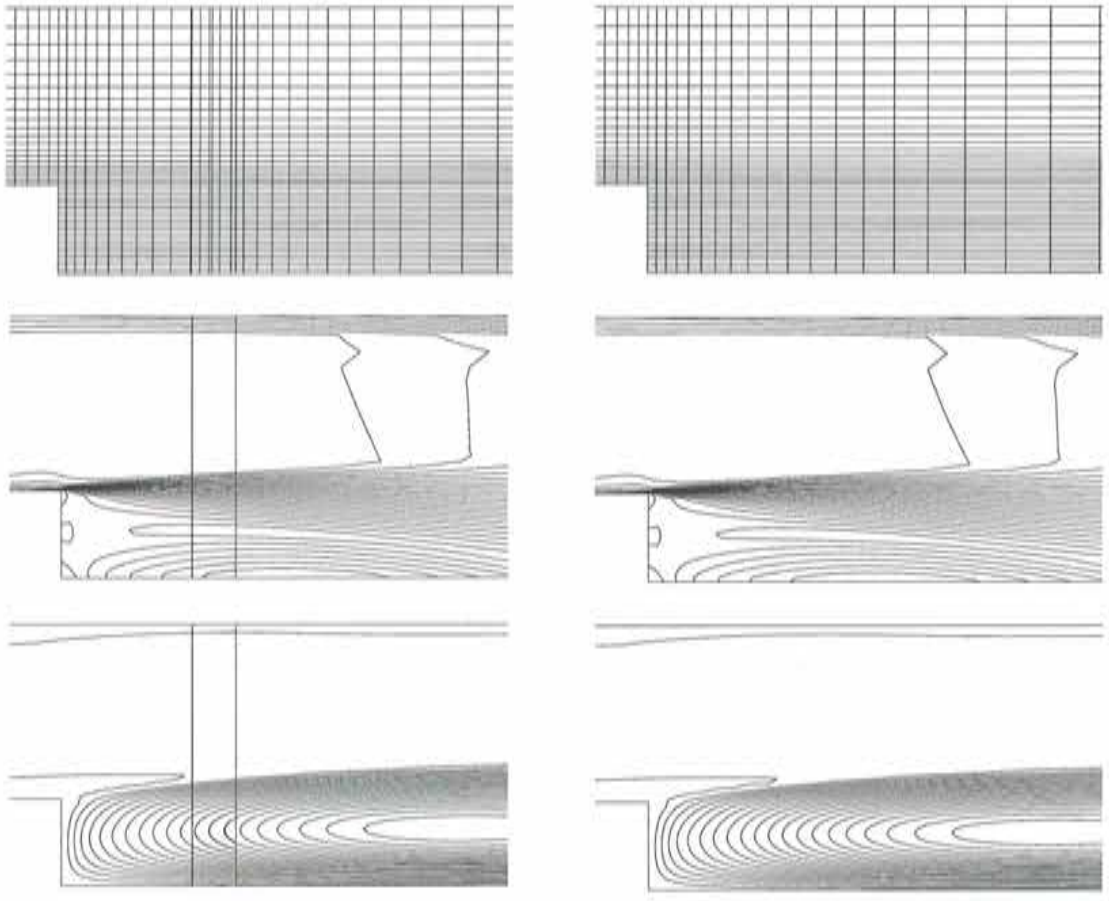


Figure 6.8: Backward facing step. (Left) D/N method. (Right) One domain. (Top) Mesh. (Mid.) Velocity. (Bot.) Eddy-viscosity.

6.3 Missile launch from a submarine

In this example we propose to solve the transient and laminar flow around a moving missile [119], using the Chimera D/N method. The missile is moving upward with a constant velocity U. The Reynolds number based on the length H of the missile is

$$\mathrm{Re} = \frac{UH}{\nu} = 1\,000.$$

We are going to compare our results to that of Folch [121], obtained with an ALE approach and an explicit flow solver described in [143]. The geometry is shown in Figure 6.9 (Left). As a background mesh we use a structured mesh of 4500 Q1/Q1 elements, shown in Figure 6.9 (Right), while the patch mesh is composed of 4841 P1/P1 elements. According to the Chimera D/N method described in the preceding chapter, the velocity of the background mesh is prescribed at the interpolation nodes, while the patch mesh is assigned a Neumann transmission condition on its outer boundary. Note that the missile subdomain does not contain any information on its own velocity as the force imposed as transmission condition would be the same in any Galilean frame of reference; all the information on the velocity of the missile is passed through the transmission conditions imposed on the background subdomain.

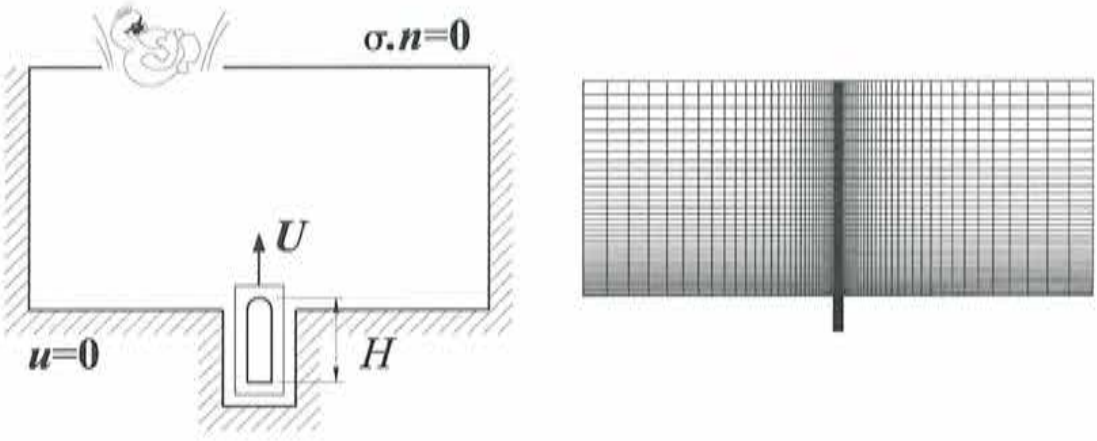


Figure 6.9: Missile launch. (Left) Geometry and boundary conditions. (Right) Background Mesh.

The transient simulation is carried out using the backward Euler scheme with a time step $\delta t = 2.2 \times 10^{-2}\, H/U$. We perform 15 domain decomposition iterations at each time step and use as relaxation parameters $\theta_{\mathrm{D}} = 0.5$ and $\theta_{\mathrm{N}} = 0.4$. Each problem is solved using a direct solver: the total time used by the Master is 2.5 % of the total CPU time used to solve this problem. The sum up of the computation time used by the Master to perform its different tasks is shown in Table 6.2. The update of the Neumann transmission conditions, which requires a least-square smoothing at each iteration, is the most consuming operation, while the communication between the Master and the Slaves is almost negligible.

The convergence of the problem is shown in Figure 6.10.

Figure 6.11 shows the composite mesh at time $t = 0.22\, H/U$, near the missile bottom right corner and at the submarine exit corner.

Element search	Dirichlet updates	Neumann updates	Import/export	Other
13 %	2 %	84 %	0.2 %	0.8 %

Table 6.2: Missile launch. CPU time dedication of the Master.

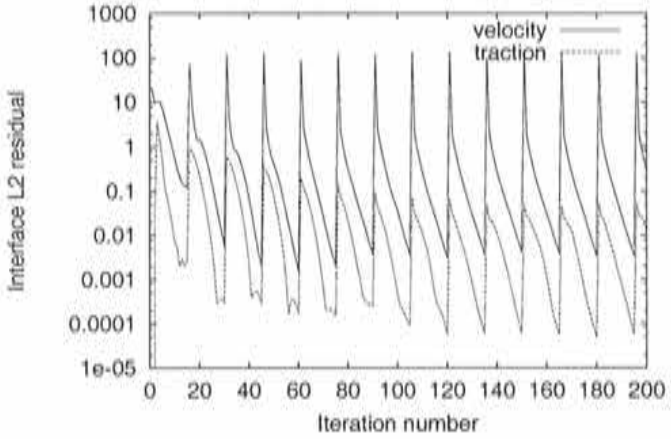


Figure 6.10: Missile launch. Convergence history.

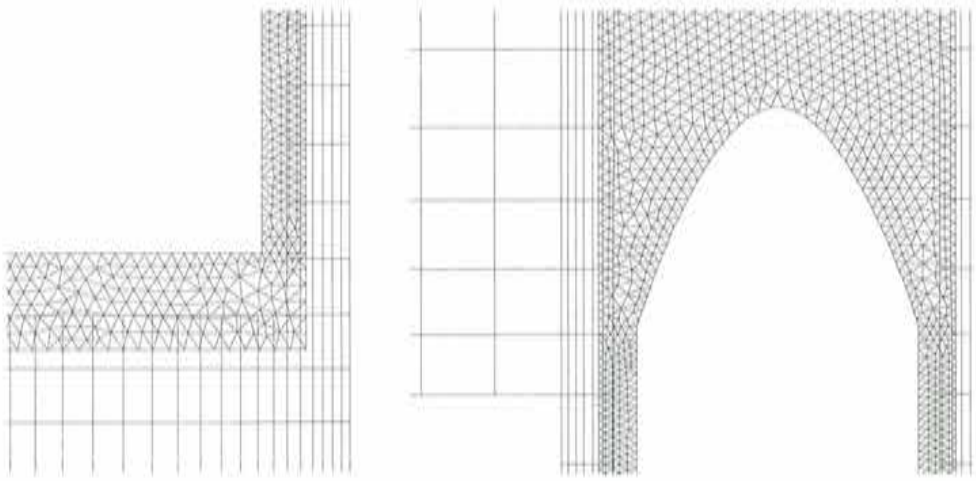


Figure 6.11: Missile launch. Composite mesh at $t = 0.22\,H/U$. (Left) Bottom right corner of missile. (Right) Submarine corner.

Figure 6.12 presents the velocity vectors obtained at different time steps. They show the development of the vortices created by the suction of air from both sides of the missile.

Figures 6.13 and 6.14 compare the results of the present simulation to that of Folch [121] at different time steps. The first figure shows the streamlines while the next figure shows the pressure. We notice that both methods give very similar profiles.

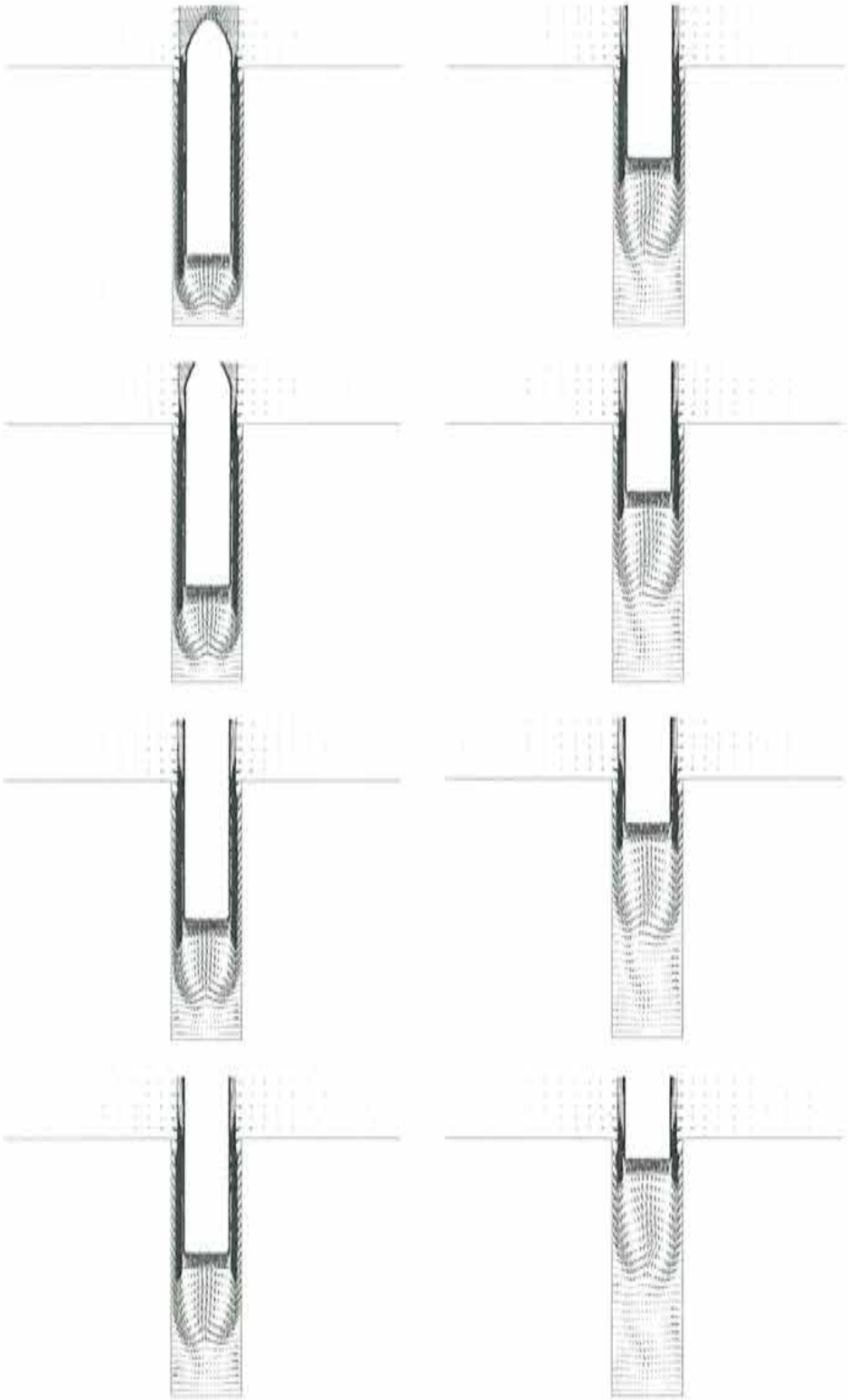


Figure 6.12: Missile launch. Velocity vectors. From top to bottom and left to right: $t = 0.22\,H/U$, $t = 0.33\,H/U$, $t = 0.44\,H/U$, $t = 0.55\,H/U$, $t = 0.66\,H/U$, $t = 0.77\,H/U$, $t = 0.88\,H/U$, $t = 0.99\,H/U$.

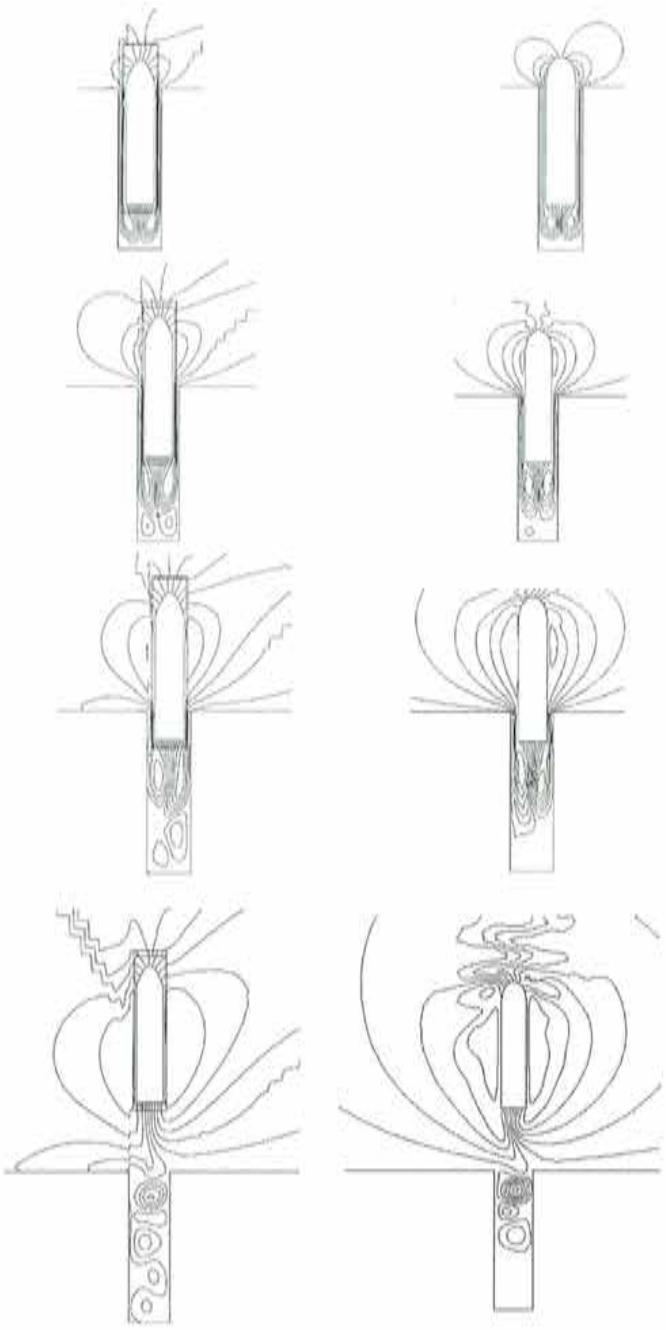


Figure 6.13: Missile launch. Streamlines. (Left) Present simulation. (Right) Folch's results [121]. From top to bottom, $t = 0.22\,H/U$, $t = 0.55\,H/U$, $t = 0.88\,H/U$, $t = 1.65\,H/U$.

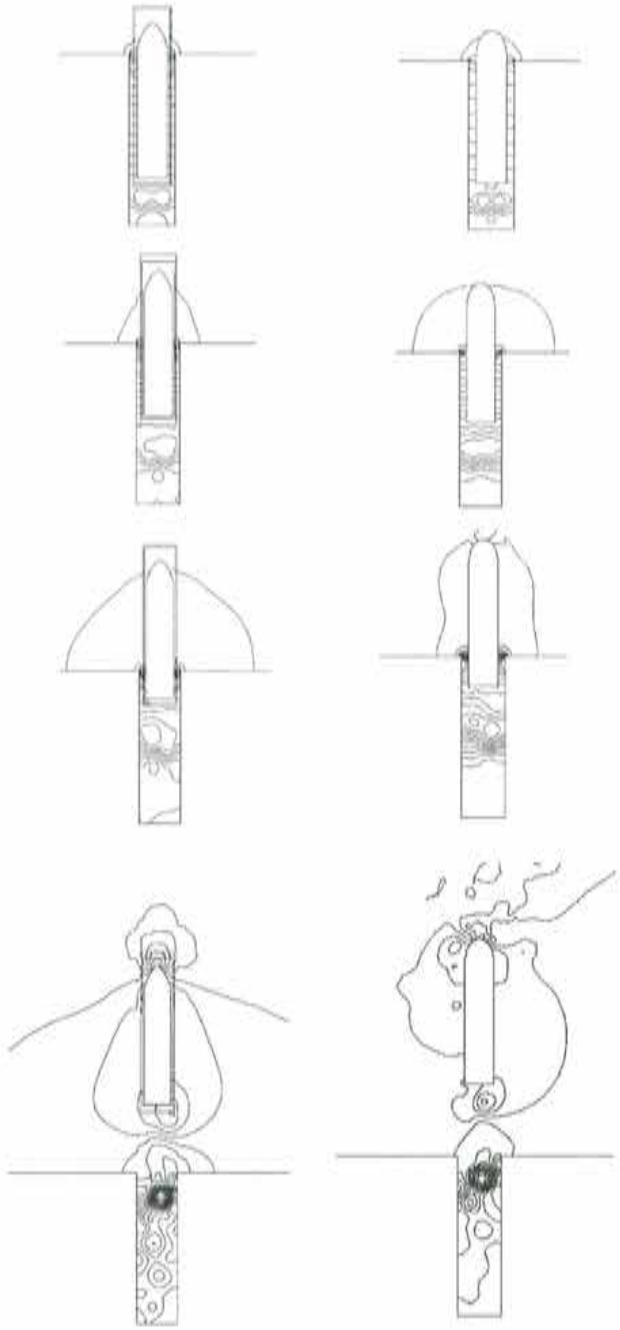


Figure 6.14: Missile launch. Pressure. (Left) Present simulation. (Right) Folch's results [121]. From top to bottom, $t = 0.22\,H/U$, $t = 0.55\,H/U$, $t = 0.88\,H/U$, $t = 1.65\,H/U$.

6.4 Centrifugal fan

We propose to solve a two-dimensional section of a domestic centrifugal fan. The geometry as well as the data are based on the CK-40 fan of Soler-i-Palau, shown in Figure 6.15 (Left). The

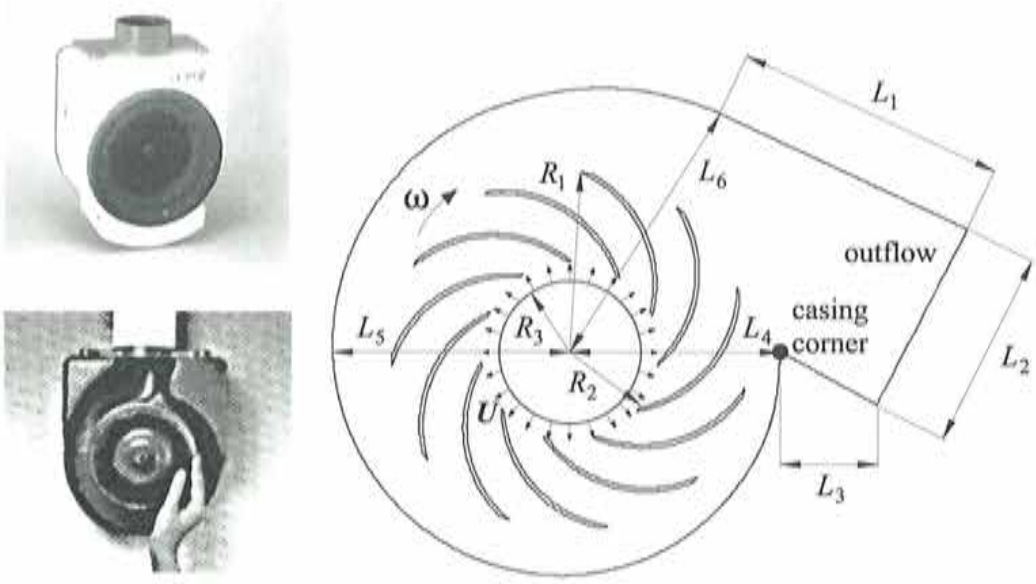


Figure 6.15: CK-40 fan. (Left) Pictures of the fan. (Right) Two dimensional section.

dimensions of the impeller and the casing of the fan are

$$R_1 = 80\,\text{mm}, \ , R_2 = 40\,\text{mm}, \ R_3 = 31\,\text{mm},$$

$$L_1 = 126\,\text{mm}, \ L_2 = 88\,\text{mm}, \ L_3 = 50\,\text{mm}, \ L_4 = 94\,\text{mm}, \ L_5 = 106\,\text{mm}, \ L_6 = 125\,\text{mm}.$$

The role of a rotodynamic device is to transform the mechanical energy of its driving force into mechanical energy available for the fluid which traverses the rotor. This energy is distributed into two components: the static pressure and the kinetic energy. According to the user requirement, the function of the device will be to provide high static pressure rise or high volume flow rate. The reasons for increasing the pressure of fluid are various and are proper to pumps. Fans, as well as some types of pumps, can rather provide a high volume flow rate. The fan under study was designed for domestic purpose and works as an extractor to eliminate smokes, bad smells and greases from kitchens. The high flow rate together with the high pressure it delivers enables to expel contaminated air outside, even if the fan is connected to long and narrow conducts, where pressure is lost by friction.

In brief, centrifugal fans operate as follows. The fluid is forced outward in the radial direction, by way of a radial cascade of vanes. It is therefore accelerated by the centrifugal force and attains its maximum velocity at the impeller vane tips. Note that the potential energy furnished by the rotation is available for the mechanical energy, i.e. both pressure and kinetic energies. When the fluid leaves the tips of the vanes, it moves tangentially along the walls of the casing. Due to the

increasing cross-sectional area of the casing along the flow passage, the fluid undergoes an expansion as it flows towards the outlet. This expansion is accompanied by a reduction of its velocity and an increase of its static pressure. This fact is confirmed by the present simulation, and in particular in Figure 6.19 (Left) which show the pressure contours obtained at different time steps. For information on fans and pumps see e.g. [144, 145, 146].

Now let us study how we can simulate the flow through our centrifugal fan. Fans usually operate at a fixed rotation speed calculated to furnish optimally a given pressure (at low cost), so that the air has sufficient energy to flow through the conducts of the set up. According to this pressure rise, the fan delivers a given flow rate. The curve which expresses the relation between the pressure rise and mass flow rate is called the *characteristic curve* of the fan. The characteristic curve of the CK-40 is shown in Figure 6.16. In open conditions, the fan delivers maximum flow rate whereas

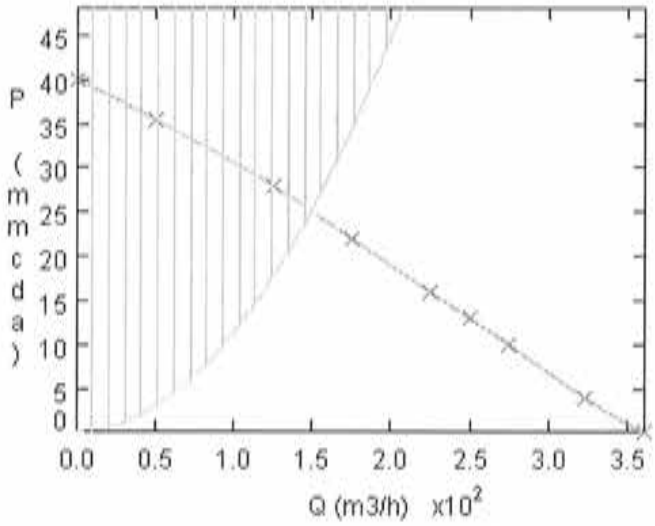


Figure 6.16: CK-40 fan. Characteristic curve (Soler i Palau).

when pressure is lost due to the presence of conducts for example, the flow rate diminishes. Now in the context of this simulation, instead of imposing the pressure both at the inlet and outlet, we impose the mass flow rate at the inflow, through the specification of the velocity, and zero traction at the outflow, i.e. zero pressure if the flow is fully developed. The inflow velocity $\boldsymbol{U}$ is imposed normal to the circular inlet of radius R_3, as sketched in Figure 6.15 (Right). A similar approach was used in [136] for the simulation of the rotor-stator interactions in a centrifugal pump. The Reynolds number based on the inflow velocity $|\boldsymbol{U}| = 1.97 \times 10^3\,\mathrm{mm/s}$ and the length of the inlet $D = 2\pi R_3 = 201.06\,\mathrm{mm}$ is

$$\mathrm{Re} = \frac{|\boldsymbol{U}|D}{\nu} = 2.65 \times 10^4,$$

ν being the kinematic viscosity of air, $\nu = 15\,\mathrm{mm}^2/\mathrm{s}$. This Reynolds number corresponds approximately to a flow rate located at the middle of the characteristic curve of the fan. In this example, the rotation is 2350 revolution per minute (r.p.m.) which corresponds to an angular

speed $|\omega| = 246.09\,\text{rad/s}$, and an Eckman number

$$\text{Ek} = \frac{\nu}{2|\omega|(D)^2} = 7.54 \times 10^{-7}.$$

Due to the high Reynolds number, the flow is solved using the Spalart-Allmaras turbulence model together with the wall function approach. The inflow turbulence viscosity is $\nu_{t,\infty} = 100\nu$ and the distance from the computational wall to the real wall is set to $\hat{y}/D \approx 2.1\%$ for the inner subdomain and to $\hat{y}/D \approx 0.6\%$ for the outer subdomain. Later on, the results will show that this is a reasonable choice.

The impeller domain is meshed with 12782 P1/P1 elements and the casing subdomain is meshed with 7345 P1/P1 elements. Figures 6.18 (Left) show a zoom of the composite mesh near the casing corner at some time steps.

As for the domain decomposition problem, we assign the impeller a Neumann transmission condition while the casing interface is of Dirichlet type. Choosing an overlapping of one element layer and using the least-square smoothing interpolation to compute the Neumann transmission condition, we expect the method to be of second order in space. An overlapping of one-element layer enables the method to be of second order in space. The time integration is carried out by the backward Euler scheme, with $\delta t = 2.32 \times 10^{-4}\,\text{s}$ so that we impose approximately 10 times steps between two blade passings. We set both the relaxation parameters of the Dirichlet and Neumann conditions to 0.3 and perform 20 iterations per time step. As initial conditions, the inner subdomain is solved with zero traction and zero eddy-viscosity flux. Then the outer subdomain is calculated by interpolating Dirichlet conditions from the solution on the inner subdomain. Each problem is solved using a direct solver: the total time used by the Master is 3.7 % of the total CPU time used to solve this problem. Figure 6.17 (Top) (Left) shows the good convergence of the problem.

Before presenting any result, let us check that the y^+ along the walls has reasonable values. Figures 6.17 (Mid.) (Left) and (Mid.) (Right) show the distribution of y^+ along one blade of the impeller and along the casing wall, obtained at time $t = 1.44 \times 10^{-2}\,\text{s}$, i.e. once the "periodic" regime is achieved.

Figure 6.17 (Top) (Right) gives the variation of the pressure coefficient c_p along the casing wall, at time $t = 1.44 \times 10^{-2}\,\text{s}$ where

$$c_p = \frac{2p}{\rho|\boldsymbol{U}|^2}.$$

The starting point of the curve is the casing corner, while the upper left part of the curve is the outflow where the pressure is "weakly" zero. The figure shows the static pressure expansion undergone by the fluid as it flows along the casing wall to the outlet.

As an indication, we compare the total pressure rise (called fan total pressure) obtained by the simulation to the pressure rise indicated by the characteristic curve. This comparison is only qualitative as we are only simulating a two-dimensional section of the fan with approximate geometry and data. By definition, the pressure rise P of a fan is the difference of total pressures p_{total} between the outflow and inflow

$$P = p_{\text{total,outflow}} - p_{\text{total,inflow}},$$

where the total pressure p_{total} is defined as

$$p_{\text{total}} = p_{\text{static}} + p_{\text{dynamic}}.$$

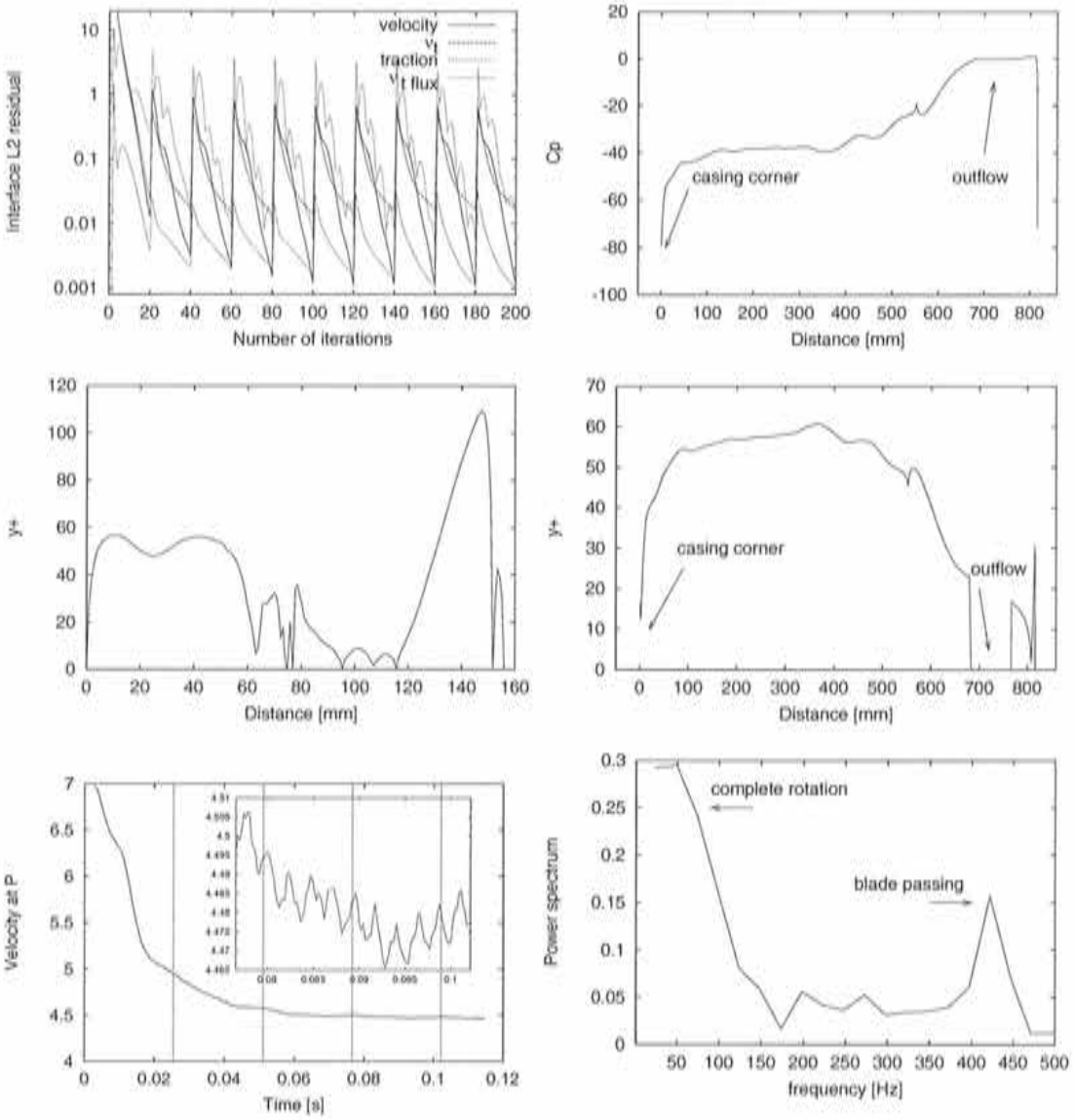


Figure 6.17: CK-40 fan. (Top) (Left) Convergence history. (Top) (Right) Pressure along casing wall. (Mid.) (Left) y^+ along one blade. (Mid.) (Right) y^+ along the casing wall. (Bot.) (Left) Velocity at P. (Bot.) (Right) Power spectrum of the velocity at P.

As when the flow discharges in the atmosphere, the static pressure is the atmospheric pressure and the dynamic pressure is zero, the pressure rise is given by

$$P = p_{\text{atmospheric}} - p_{\text{total,inflow}}.$$

The present calculation gives a total pressure rise between the inlet and outlet of approximately

$100\,\mathrm{g/m\,s^2}$. In terms of water columns, this gives

$$P = \frac{100}{\rho_{\mathrm{water}} g} = 7.2\,\mathrm{mm\ w.c.},$$

where g is the gravitational acceleration and ρ_{water} is the density of water. The experimental results shown in Figure 6.16 give a pressure rise of approximately 19 mm w.c.. It should be pointed out that this result is only qualitative for two main reasons. Firstly, the geometry as well as the data used in this simulation are only *based* on the real geometry of the fan. Secondly, we must be aware of the fact that the very coarse mesh used in the simulation makes the results overdiffusive, and therefore the pressure gradient is likely to be underestimated. We can only conclude that the order of magnitude is good!

Figure 6.17 (Bot.) (Left) shows the velocity obtained at a point P located near the casing corner. The vertical dotted lines stand for each complete rotation of the impeller. We observe that the periodic regime is obtained after two rotations. Figure 6.17 (Bot.) (Right) shows the power spectrum obtained from the velocity at P. Two main frequencies are obtained: the blade passing frequency and the complete rotation frequency.

Figures 6.18 and 6.19 give some results obtained at different times steps: the velocity vectors near the corner of the casing, the pressure contours and the eddy-viscosity contours. Finally, Figure 6.20 presents the vorticity contours. We observe that when the vorticity contours almost mimic the eddy-viscosity: in fact, vorticity is the essence of eddy-viscosity production.

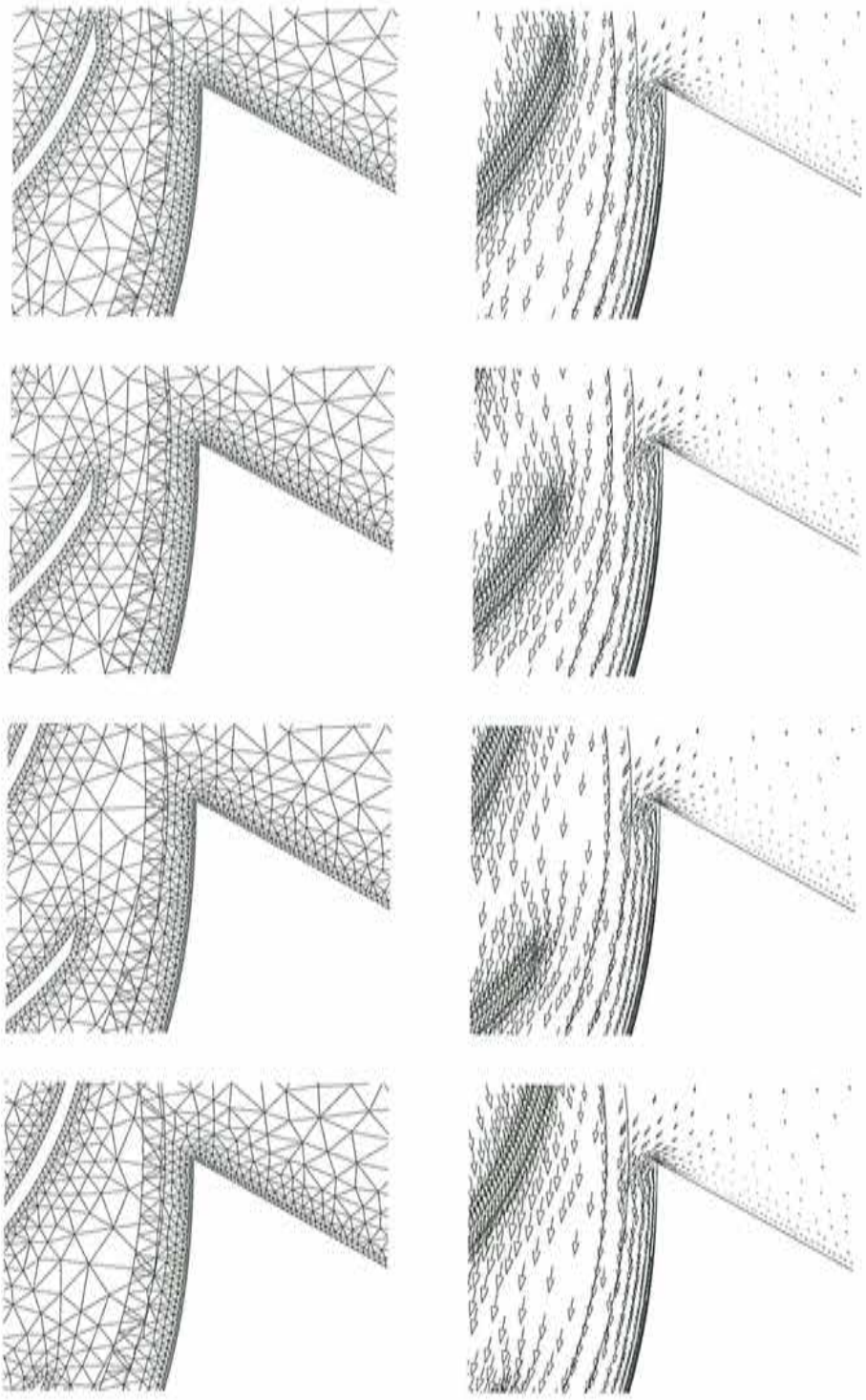


Figure 6.18: CK-40 fan. (Left) Composite mesh. (Right) Velocity vectors near casing corner. From top to bottom: $t = 5.75 \times 10^{-2}$ s, $t = 5.82 \times 10^{-2}$ s, $t = 5.88 \times 10^{-2}$ s, $t = 5.95 \times 10^{-2}$ s.

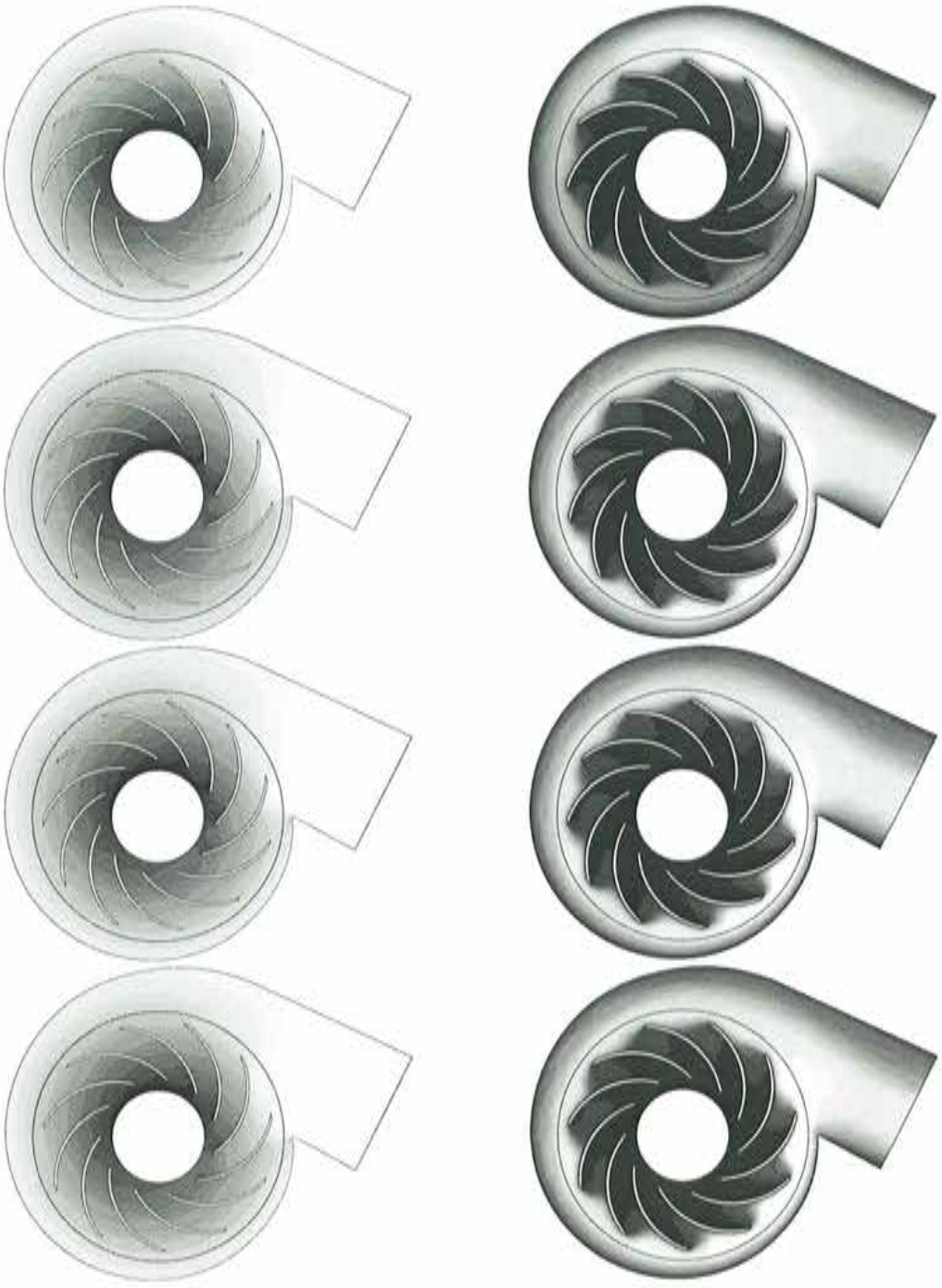


Figure 6.19: CK-40 fan. (Left) Pressure contours. (Right) Eddy-viscosity contours. From top to bottom: $t = 5.75 \times 10^{-2}$ s, $t = 5.82 \times 10^{-2}$ s, $t = 5.88 \times 10^{-2}$ s, $t = 5.95 \times 10^{-2}$ s.

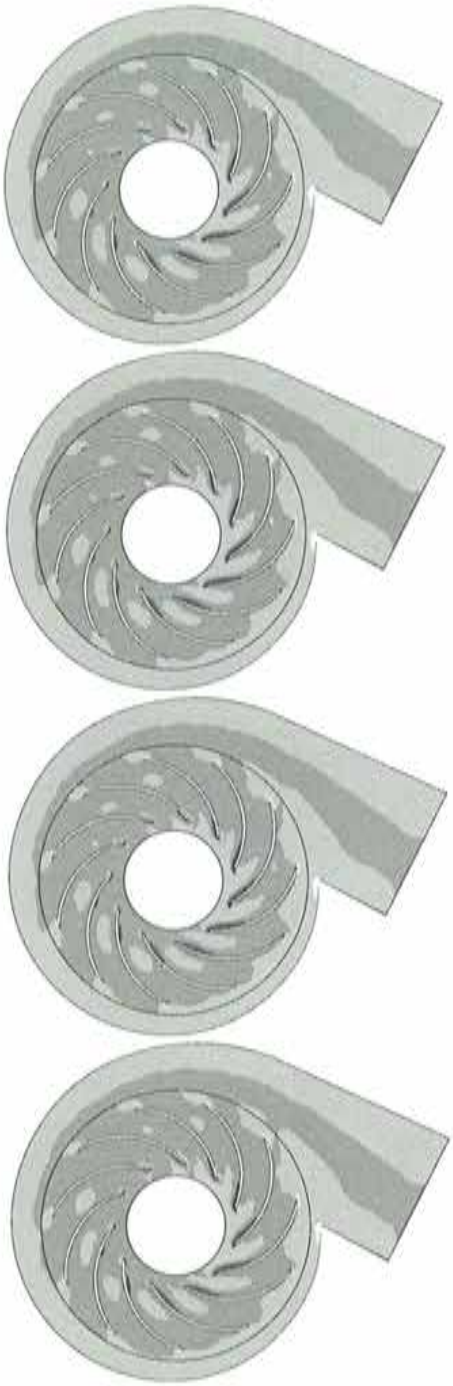


Figure 6.20: CK-40 fan. Vorticity. From top to bottom: $t = 5.75 \times 10^{-2}$ s, $t = 5.82 \times 10^{-2}$ s, $t = 5.88 \times 10^{-2}$ s, $t = 5.95 \times 10^{-2}$ s.

6.5 Stirred tank

In this example, we apply the Dirichlet/Neumann method to the solution of a stirred tank. Stirred tanks are frequently used in industrial processes for the preparation of plastic, rubber, pharmaceutical products, food etc. They can act simply as a blending device of miscible liquids, as for example for the blending of petroleum products, but can also be required to produce chemical reactions; in this case, the main task of the chemical reactor is to provide sufficient blending so that all the reagents meet and the reaction can occur. It is therefore important to know and study the thermodynamic and kinetic data of the chemical process as well as the mass and thermodynamic transfers. Whether the tank is required for blending only or for providing a suitable medium for chemical reactions, the hydrodynamic characteristic of the flow is of primary importance. This is what we propose to study in this example.

The stirred tank we consider is made of an axial flow impeller and four wall-welded baffles in the tank. The impeller has four pitched blades at a 45° angle designed to draw in the liquid from above and direct it downwards to the bottom of the tank, as shown in Figure 6.21 (Left). Actually, the flow is discharged both axially and radially depending on the angle and Reynolds number; for example at low Reynolds numbers the flow is principally radial, as will show the simulations. They are in general very efficient for blending miscible materials and solids suspension. As an example of other type of impellers, let us mention the radial flow impellers, which work as follows: they discharge the flow radially before the fluid leaves the blade tips upwards and downwards in similar proportions, as shown in Figure 6.21 (Right). Radial flow impellers, such as the Rushton turbine, are less efficient for mixing but generate more shear at the blade tips and are therefore suitable for gas-liquid applications. For example, they are used in the generation of bioproducts via gasification, through bubble break up in the region of high shear.

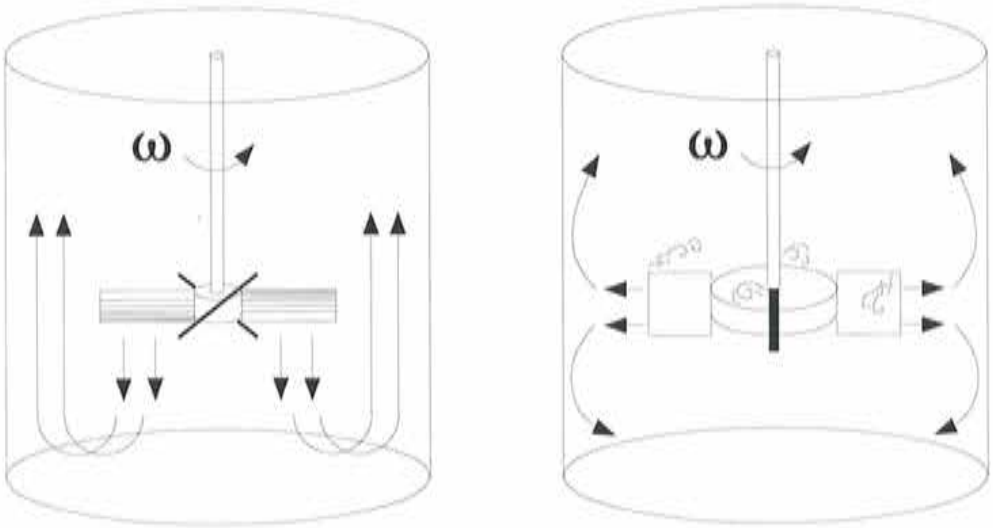


Figure 6.21: Stirred tank. (Left) Axial flow impeller. (Right) Radial flow impeller.

Only laminar simulations are presented here. Contrary to usual aerodynamic and hydraulic

applications, the laminar state in stirred tank reactors is not exceptional. In fact, the typical fluid in play may be highly viscous (like the case of polymerization reactors), and, in addition, high characteristic shears of turbulent flows may not be desirable when dealing with sensitive materials. The impeller we study here is suitable for achieving high circulation rates in low to medium viscosity liquids.

In order to increase the vertical mixing, break up the circular flow around the tank, and possibly to generate turbulence more rapidly, four baffles are disposed around the tank. The baffles are welded to the wall although off-set baffles may be preferable to avoid stagnation zones in the corners.

The geometry is based on the stirred tank described in [135], and is shown in Figure 6.22. The tank has a diameter $T = 0.3\,\mathrm{m}$, while the impeller diameter is $D = T/3$. The blades have a width $W = D/5$, and the impeller to bottom clearance is $C = T/3$. The four baffles are $B = T/12$ wide.

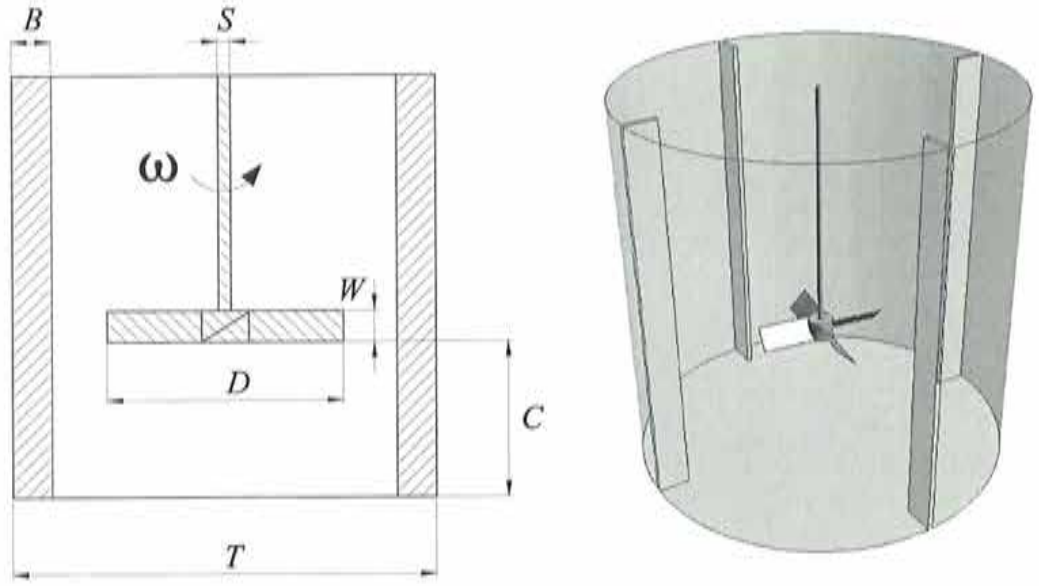


Figure 6.22: Stirred tank. Geometry.

The non-inertial subdomain is attached to the impeller and is assigned a Neumann transmission condition. The fixed subdomain is the tank and is assigned a Dirichlet transmission condition. The impeller subdomain is meshed with 93332 P1/P1 elements and the tank subdomain with 23135 P1/P1 elements; they are shown in Figure 6.23.

The impeller rotational speed is $N = 225\,\mathrm{r.p.m.}$ which corresponds to an angular velocity $|\omega| = 23.6\,\mathrm{rad/s}$. The agitator tip speed is $U = |\omega|D/2 = \pi DN = 1.18\mathrm{m/s}$, providing a low agitation. The Reynolds number is defined as:

$$\mathrm{Re} = \frac{ND^2}{\nu} = 90,$$

while according to this choice the Eckman number is simply $\mathrm{Ek} = 1/2\mathrm{Re}$. Each problem is solved

using an iterative solver (GMRES with diagonal preconditioning): the total time used by the Master is 3.3 % of the total CPU time used to solve this problem.

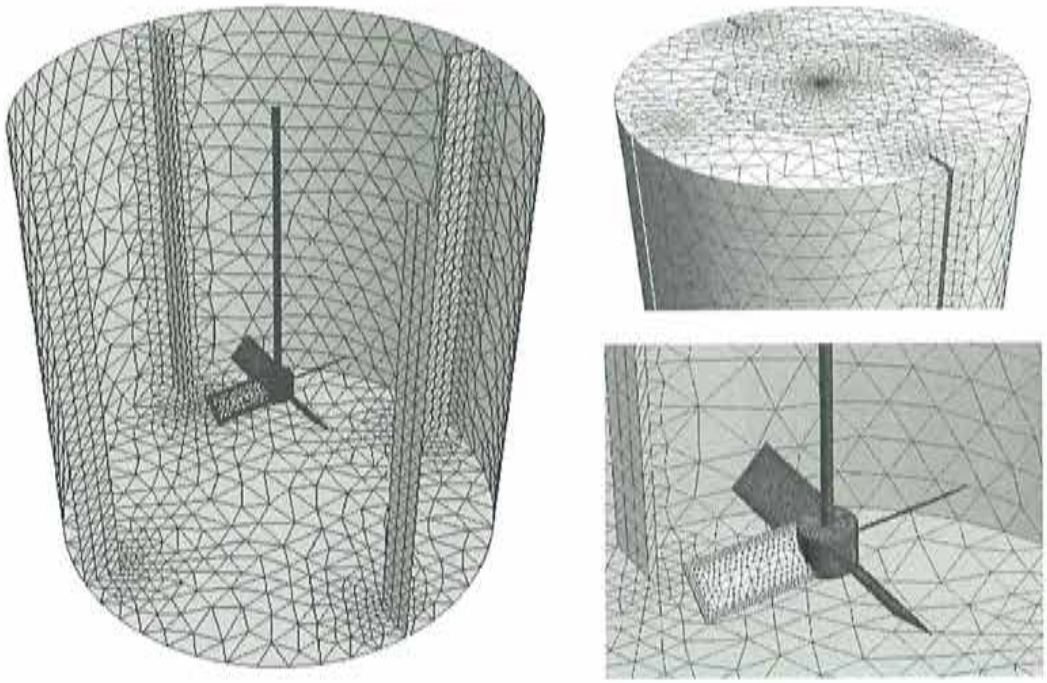


Figure 6.23: Stirred tank. Composite mesh. (Left) Overwhole view. (Top) (Right) Top of the tank. (Bot.) (Right) Impeller and baffle.

Figure 6.24 (Left) shows the convergence of the DD method in the first steps of the simulation. Due to the high viscosity of the fluid, the flow becomes very rapidly periodic. Figure 6.24 (Right) presents the power spectrum of the x-viscous force exerted on the impeller. We recognize the rotation frequency at 3.75 Hz, but we cannot distinguish any other important frequency.

Figure 6.25 shows the pressure contours on the impeller blades. The contours are smooth and confirms the good stabilization of the numerical scheme. On the left-hand side blade the pressure is low: this is the suction face which draws the fluid from above. Figure 6.28 (Top) (Left) shows the pressure contours on a vertical cut outlining the low pressure above the impeller and high pressure below the impeller. On the right hand side blade, the pressure is higher and pushed the flow downwards.

Figure 6.26 shows a cut just below the impeller. The small stains just below the blades indicate downward flow, while the bigger stains indicate upward flow. A wider perspective is given by Figure 6.28 (Bot.) (Right).

Figure 6.27 shows some iso-surfaces of the vertical velocity, around the impeller. The dark contours represent negative values while the light contours represent upward movement of the fluid. The fluid vertical swirl is confirmed by Figure 6.28 (Top) (Right) which shows a vertical cut of the velocity vectors. Figure 6.28 (Bot.) (Left) shows the instantaneous streamlines, winding around the tank from top to bottom and bottom to top.

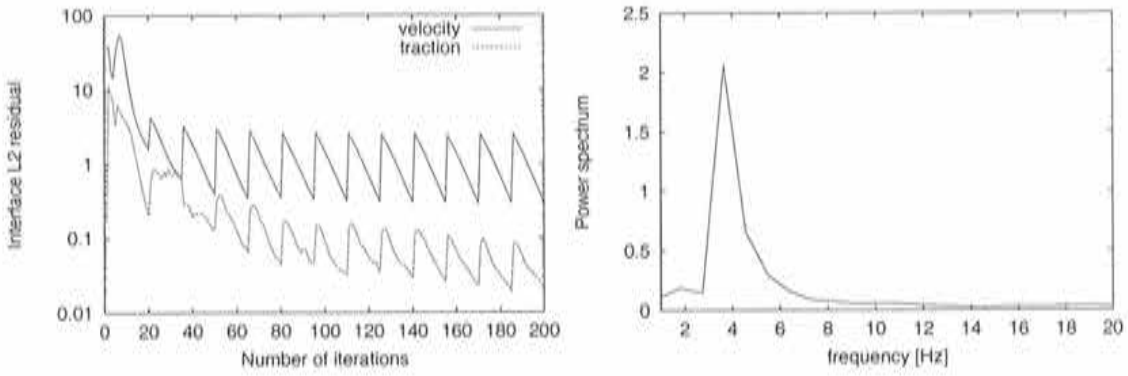


Figure 6.24: Stirred tank. (Left) Convergence history. (Right) Power spectrum of the x-viscous force.

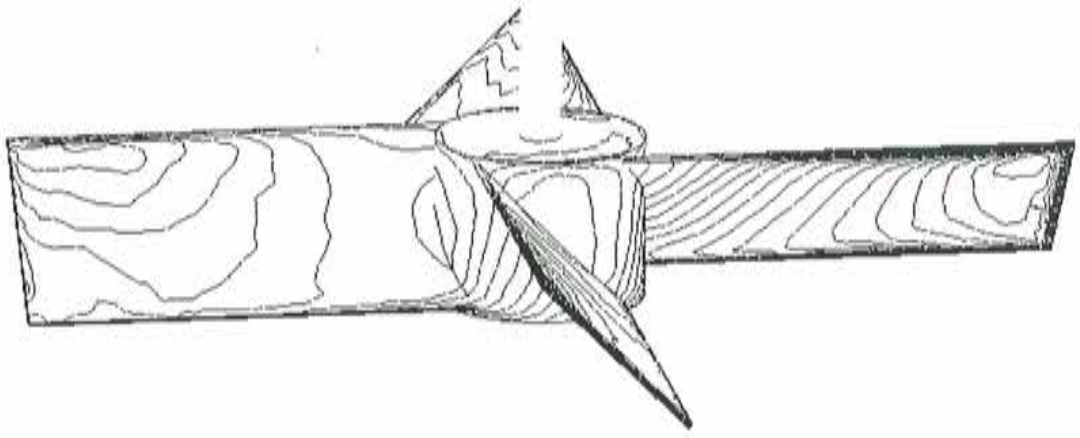


Figure 6.25: Stirred tank. Pressure on impeller. Left-hand blade: low pressure. Right-hand blade: high pressure.

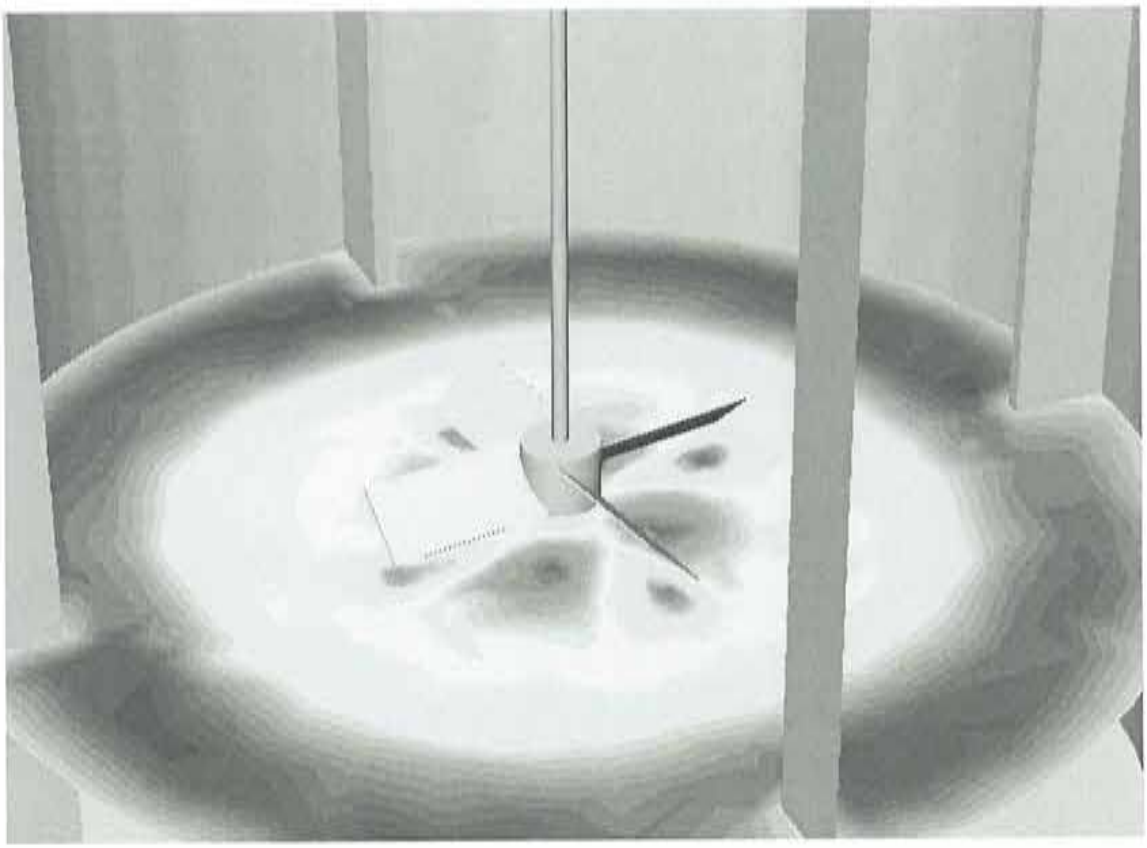


Figure 6.26: Stirred tank. Horizontal cut: vertical velocity.

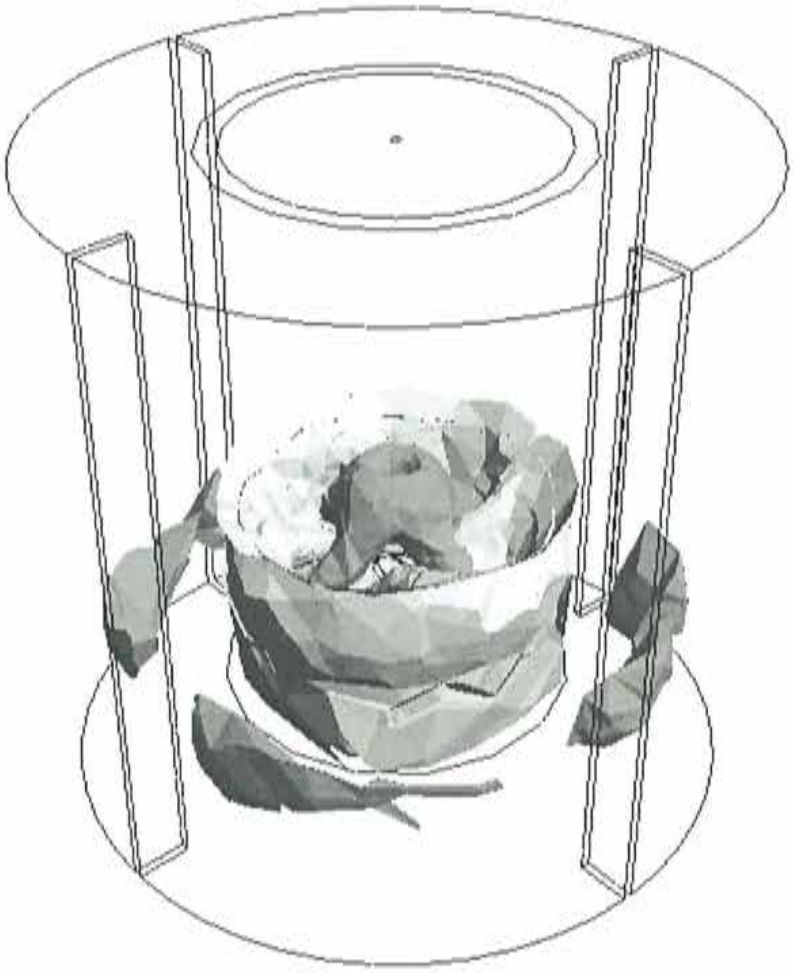


Figure 6.27: Stirred tank. Vertical velocity iso-surfaces.

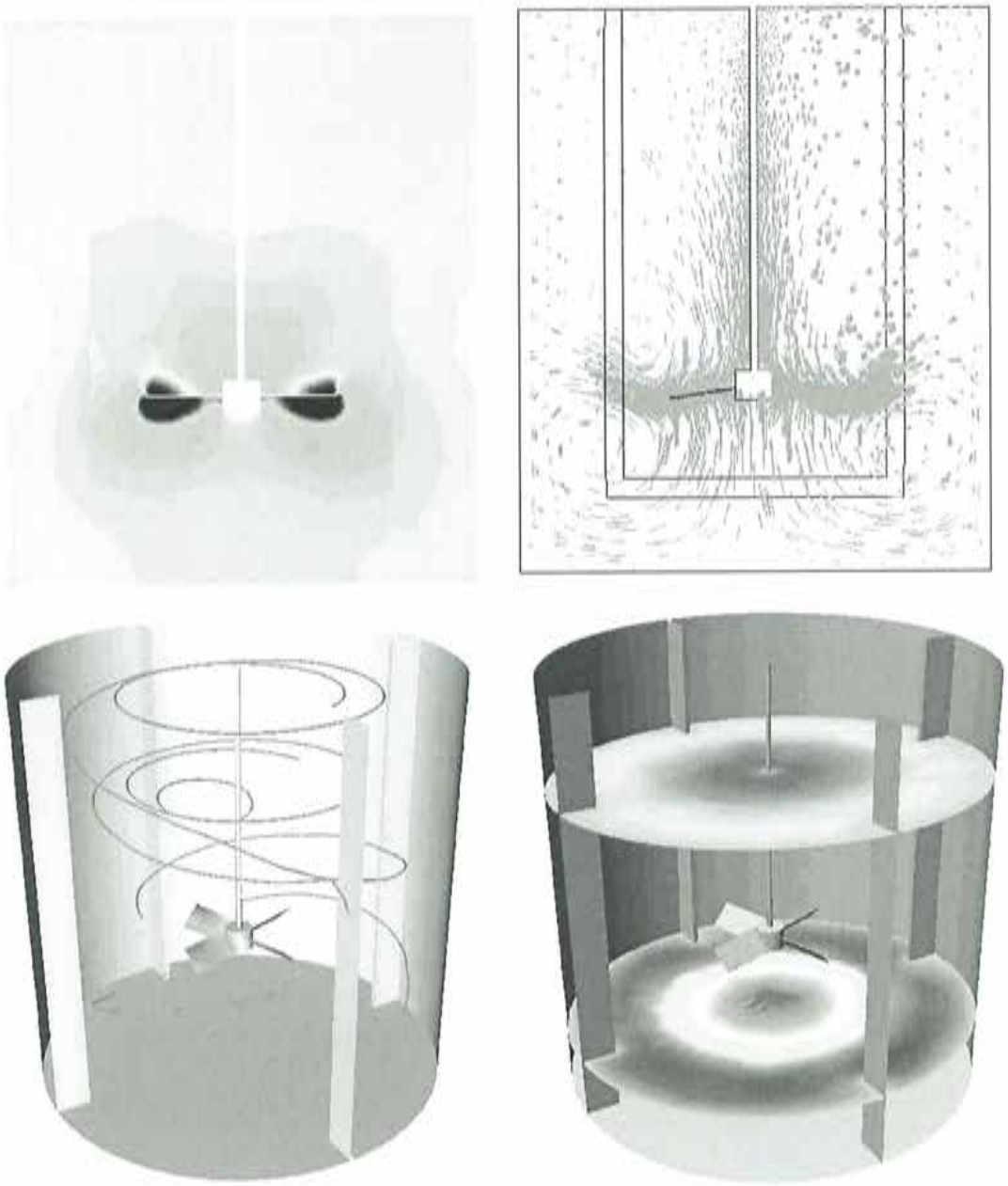


Figure 6.28: Stirred tank. (Top) (Left) Pressure. (Top) (Right) Velocity vectors on vertical cut. (Bot.) (Left) Streamlines. (Bot.) (Right) Vertical velocity.

6.6 Axial fan

We solve an axial fan of the COMPACT series of Soler i Palau, namely the HCFT/4-630/H axial fan, abbreviated here as C-630. This fan operates at 1420 r.p.m. and in open conditions works with a maximum flow rate of 17060 m^3/h. The geometry as well as the data are based on the real geometry of this fan and are shown in Figure 6.29 (Left). The fan consists of a rotor composed of seven blades and a stator composed of height struts. The dimensions of the impeller and the

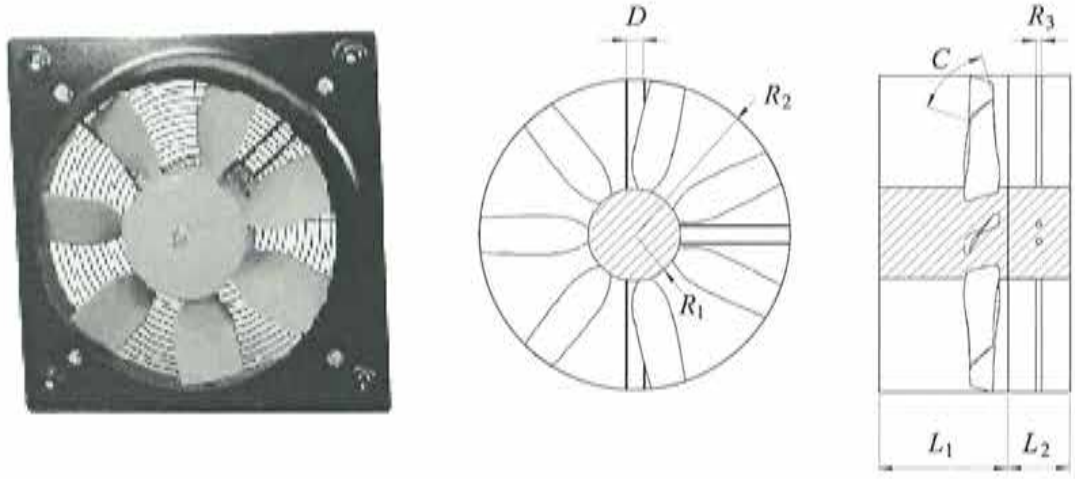


Figure 6.29: C-630 fan. (Left) Picture of the fan. (Right) Geometry.

casing of the fan are

$$R_1 = 92\,\text{mm},\ , R_2 = 310\,\text{mm},\ R_3 = 7\,\text{mm},$$
$$L_1 = 250\,\text{mm},\ L_2 = 100\,\text{mm},\ D = 27\,\text{mm},\ C = 82\,\text{mm}.$$

In Section 6.4 we revised briefly how centrifugal fans operate: the impellers play the role of vanes, guiding the flow outwards and taking advantage of the acceleration due to the centrifugal force; in the case of axial-flow machines, the impellers modify the angular momentum of the fluid as energy exchange. Axial fans work as follows: the fluid is first set in motion by the displacements of the blades. The cascade of blades forces the fluid to follow the inclined path formed by two successive blades, resulting in a net change of angular momentum. Therefore, the fluid on the upper surface of a blade will have to accelerate with respect to the fluid on the lower surface, forming the usual suction/pressure pair of surfaces of an airfoil. As in the case of centrifugal fans, the machine provides a net flow rate as well as an increase of the static pressure.

The rotor is meshed with 41469 P1/P1 elements and the stator is meshed with 71138 P1/P1 elements. The boundary composite mesh is shown in Figure 6.30 at its initial position.

The simulations are laminar, although the flow is clearly turbulent. We do not try to solve the boundary layer and impose a slip condition on the walls. We expect that the meshes are coarse enough to damp any perturbating frequencies so that a "stable" solution can be obtained. Due to the large number of degrees of freedom, the algebraic systems are solved using the GMRES iterative solver with diagonal preconditioner.

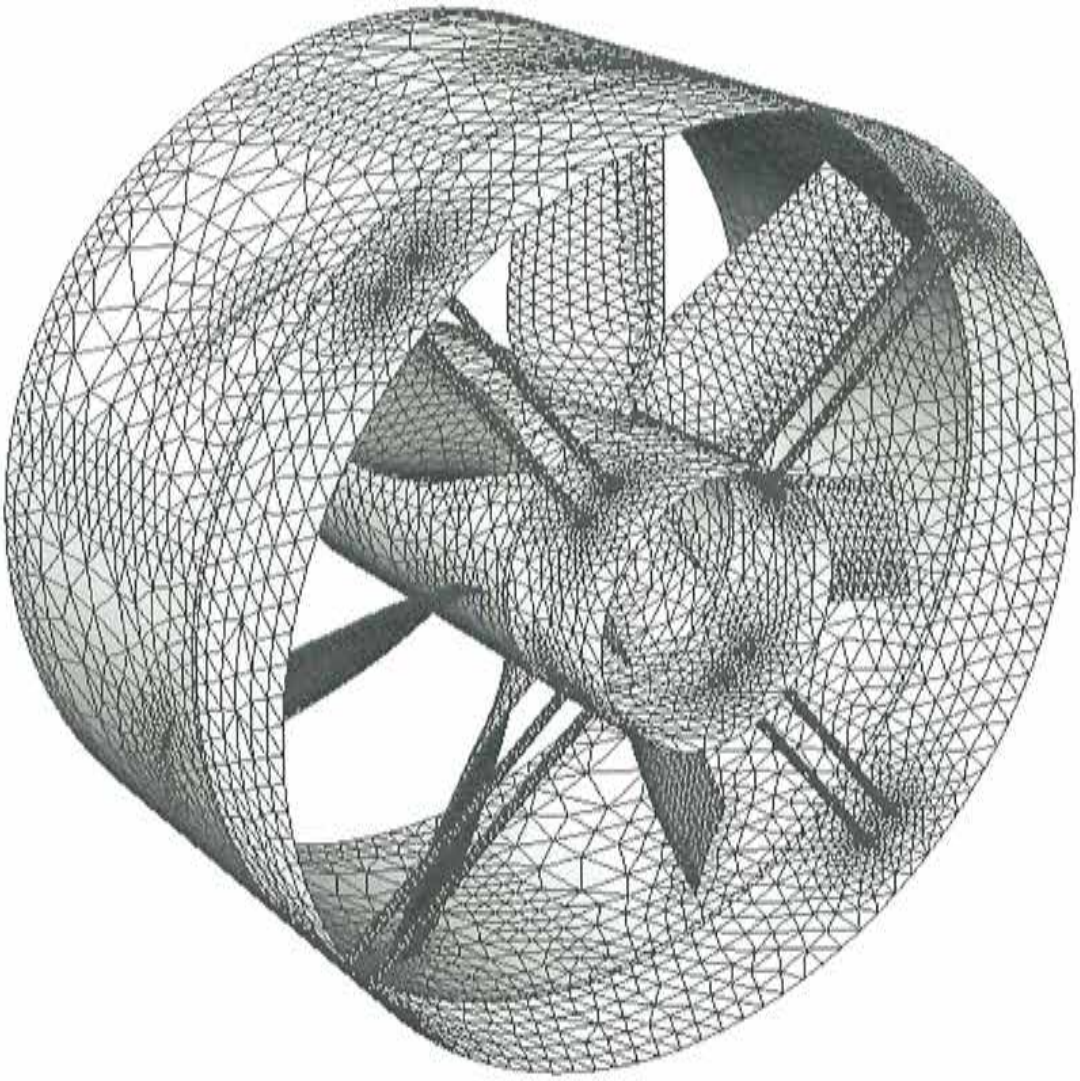


Figure 6.30: C-630 fan. Composite mesh.

The DD problem is solved using the D/N method with less than one-element layer. The stresses computed at the Neumann boundary are of first order in space (the same approach was used in [147] for the simulation of an axial turbine). The time integration is carried out using the backward Euler scheme with a time step of $\delta t = 6 \times 10^{-4}$ s: this corresponds to approximately 10 time steps between two blade passings. As an initial condition, the rotor is solved using zero traction at the exit. The initial solution of the stator is obtained by solving some few steady iterations using as Dirichlet inflow conditions the velocity obtained from the steady state of the stator plus the rotational component. The initial velocity solution is shown in Figure 6.31.

Figure 6.32 presents the velocity module and pressure contours on three planes, once the periodic regime is obtained. The variables are perfectly continuous across the interface.

Figure 6.33 shows the pressure contours on both faces of the blades. The left-hand side figure

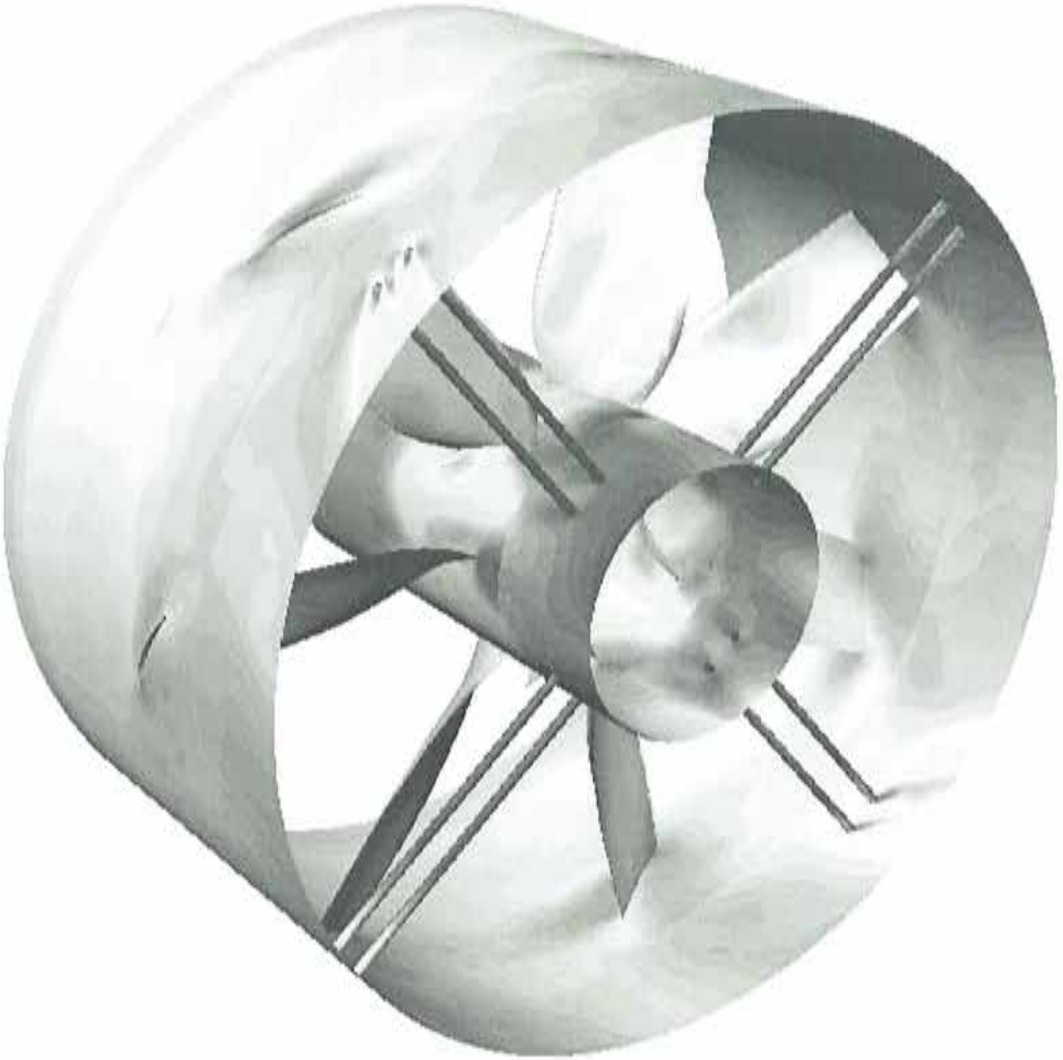


Figure 6.31: C-630 fan. Initial solution.

is the pressure face, located downstream, while the suction face is shown in the right-hand side figure.

Figures 6.34 (Top) and (Bot.) show the velocity module and pressure contours on the boundaries of the fan; remember that a slip condition is imposed on the walls, and therefore the velocity is always tangent to it.

Figures 6.35 and 6.36 show the velocity vectors at some time steps, when a blade passes by a pair of struts.

Figure 6.37 shows some streamlines. They show how the fluid threads between the struts of the stator.

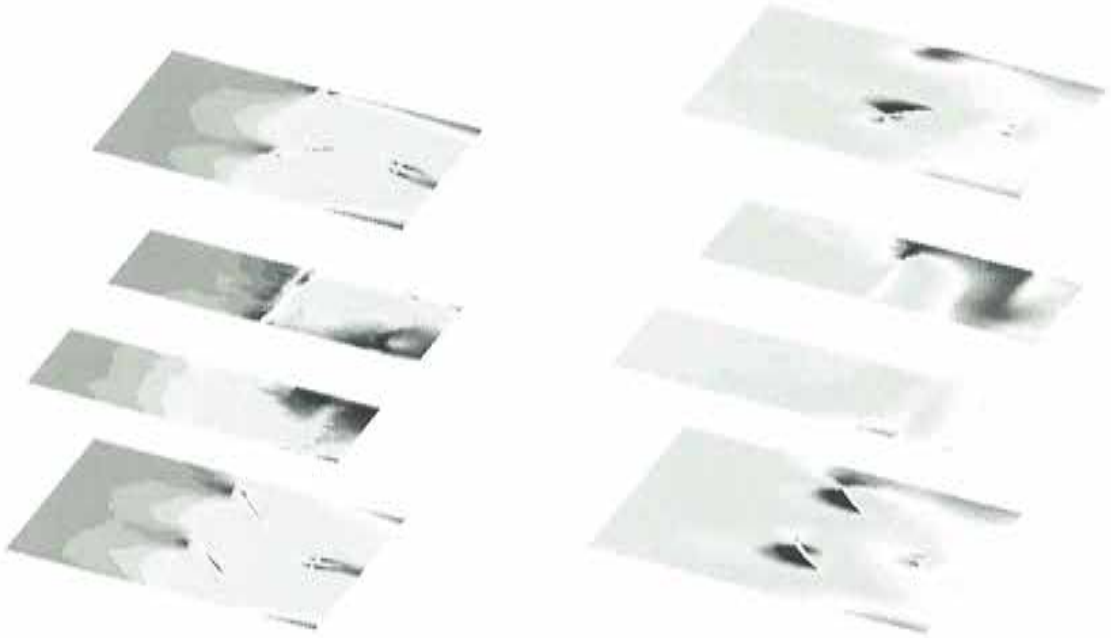


Figure 6.32: C-630 fan. Three cuts across the fan. (Left) Velocity module. (Right) Pressure.

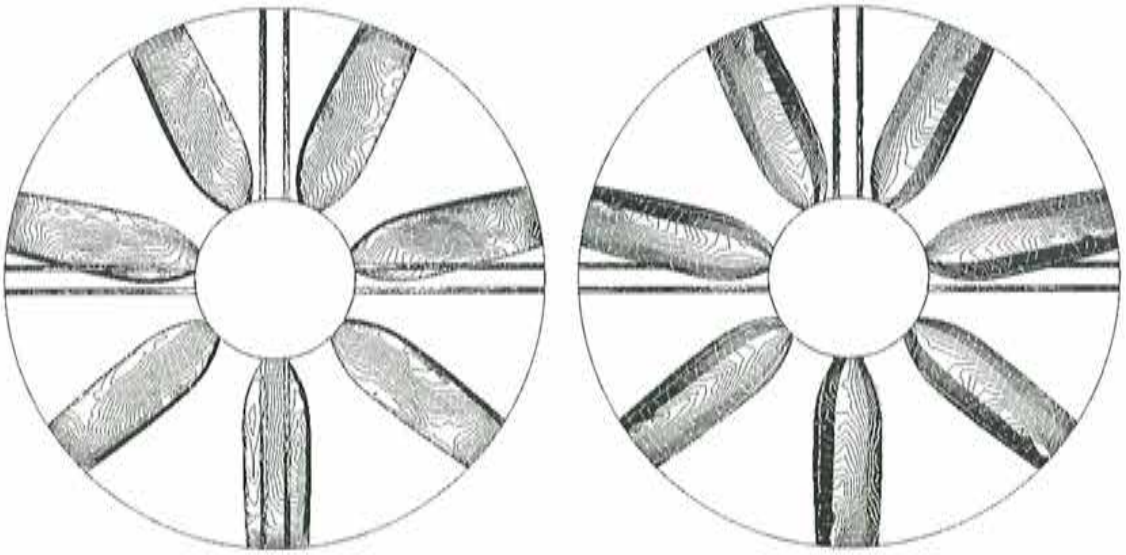


Figure 6.33: C-630 fan. Pressure contours. (Left) Pressure face. (Right) Suction face.

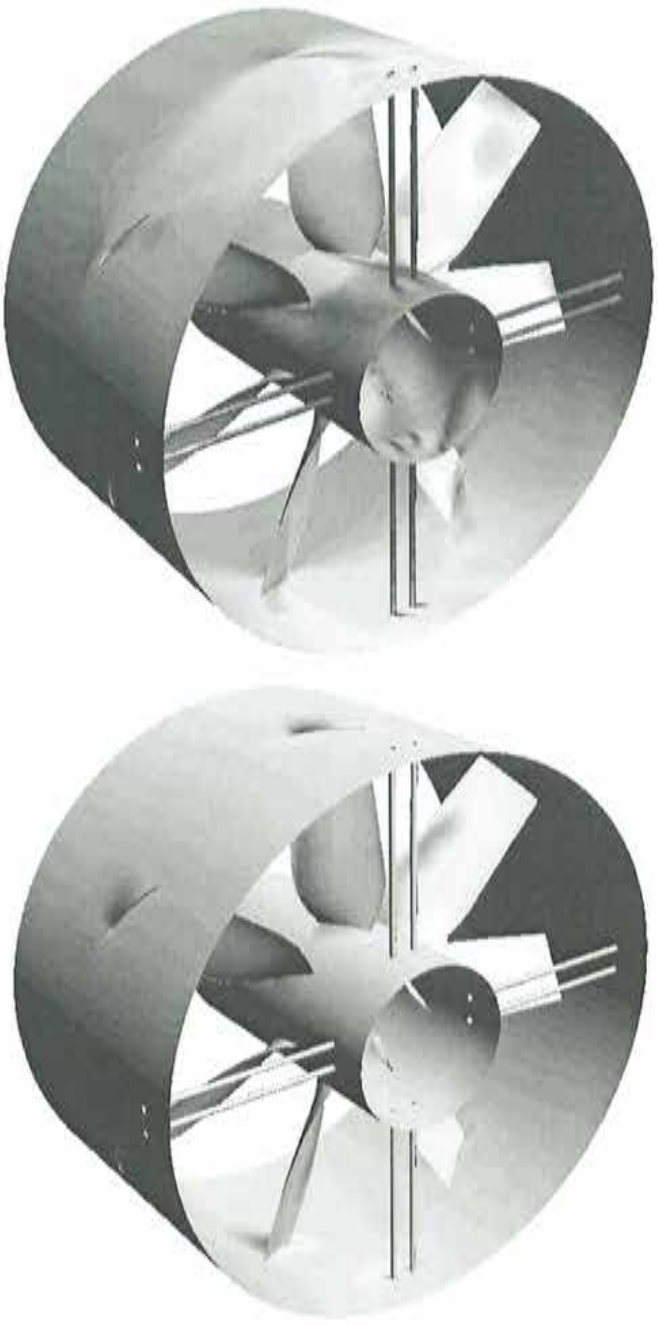


Figure 6.34: C-630 fan. (Top) Velocity module. (Bot.) Pressure.

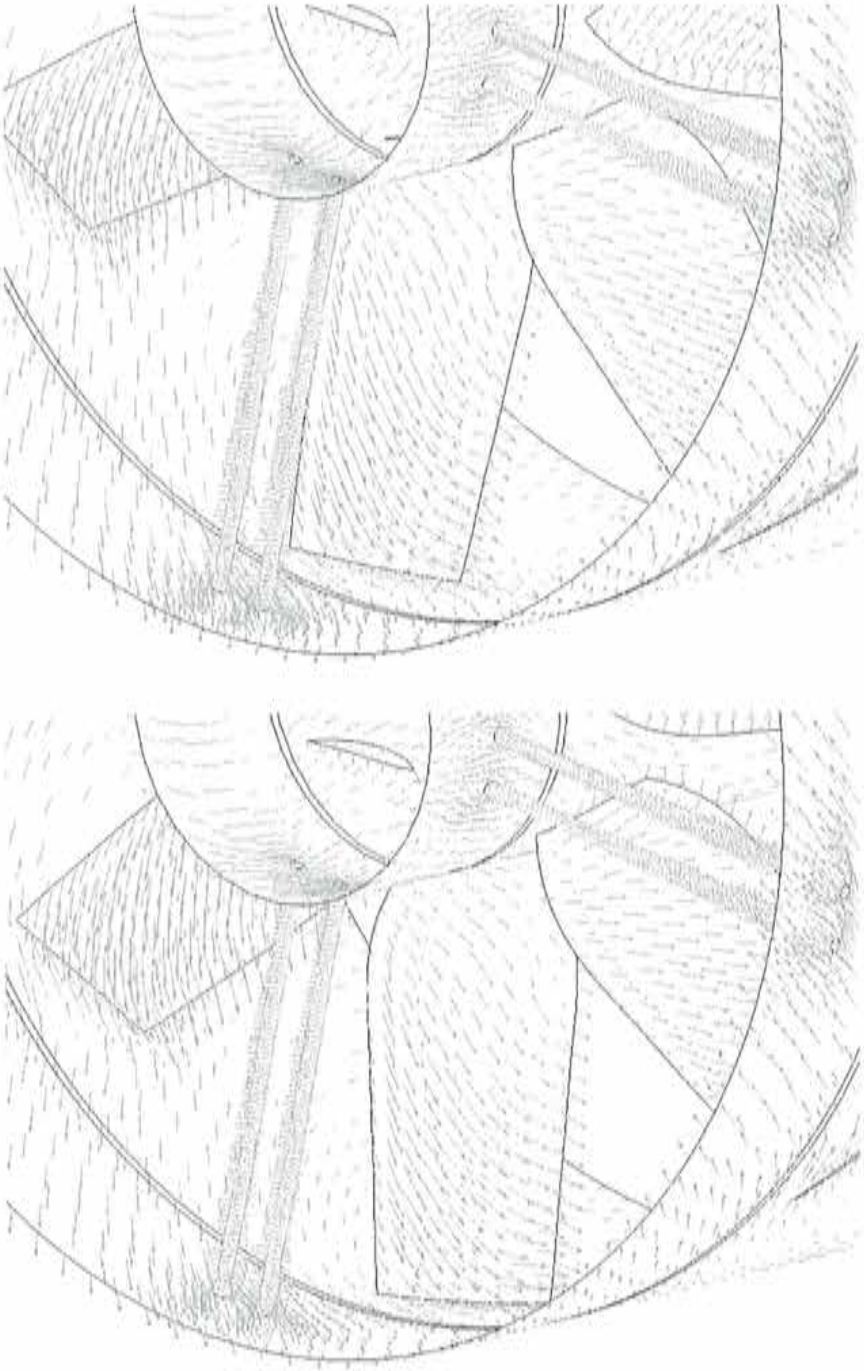


Figure 6.35: C-630 fan. Velocity vectors.

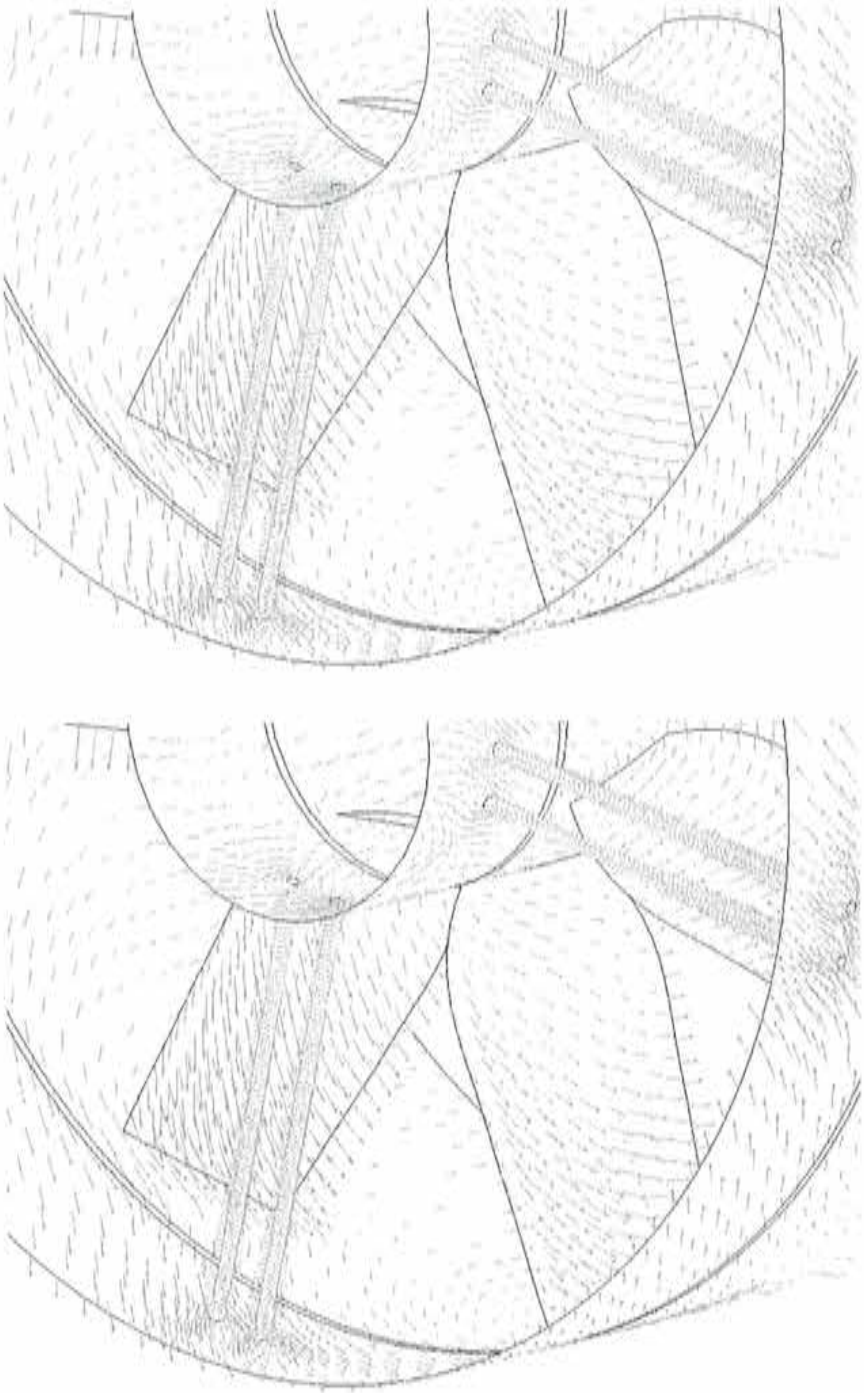


Figure 6.36: C-630 fan. Velocity vectors.

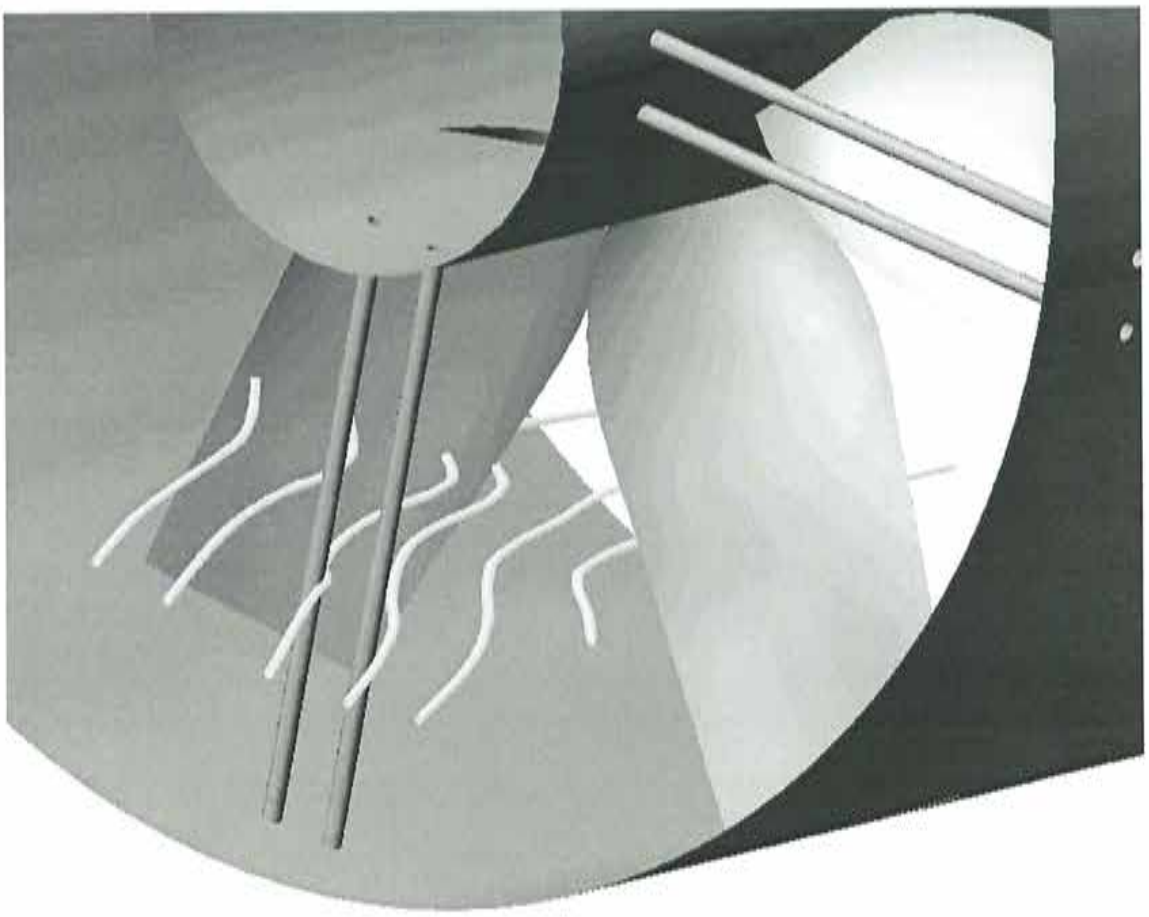


Figure 6.37: C-630 fan. Streamlines.

Conclusion

We have proposed and studied *overlapping mixed iteration-by-subdomain domain decomposition methods*. These methods are extensions of some existing DD methods to the case of overlapping subdomains; the transmission conditions on the interfaces are mixed, i.e. they are of different type on each side of the interfaces; the solutions on the subdomains are coupled iteratively until convergence is achieved.

The study of a one-dimensional scalar advection-diffusion-reaction equation has enabled us to foresee the possible benefits of using overlapping subdomains together with a mixed DD method. We also discussed the importance of the relaxation parameter to gain control on the stability of the DD algorithm. The most important result is that even in the hyperbolic limit, Dirichlet and Neumann (or Robin) conditions can be placed indifferently with respect to the direction of the advection in order to achieve convergence. We concluded that the overlap renders mixed methods more robust.

Then we studied an overlapping Dirichlet/Robin method within a variational framework for a two-subdomain partition. All the analysis was based on an equivalent set of equations for the interface unknowns, involving Steklov-Poincaré like operators. We showed the convergence of the relaxed sequential algorithm. We also considered the possibility of applying the Dirichlet/Neumann method to overlapping subdomains. Using the finite element approximation, we showed that the overlapping methods lead to an algebraic preconditioned Richardson procedure for the interface unknowns. We outlined the relation between the proposed algorithm and the classical Schwarz and mixed methods on disjoint subdomains. Both overlapping DD methods were tested through the solution of three numerical examples. It is well known that reflecting transmission conditions are undesirable in the advection dominated range as they destabilize the iterative algorithm. However, when used on overlapping subdomains, we showed that mixed methods diffuse much more rapidly the error and a considerable gain in convergence can be obtained even with a small geometric overlap. In particular, a notable improvement was obtained using overlapping subdomains when a vortex passes by the interfaces.

In the view of a practical implementation for the solution of the Navier-Stokes equations, we built up a Master/Slave algorithm to couple efficiently the numerical solution obtained on different subdomains. A master code is in charge of controlling the iterative process and performing all the necessary operations to leave the slaves unworried. Therefore, very few modifications of the original finite element solver are required. We then addressed the importance of the way the Neumann data is calculated: we identified the need for using the solution from the underlying mesh on both sides of the Neumann type interface. This is not possible when using disjoint subdomains as the solution is only available on one side. From this remark, we derived a second order scheme in space based on a least-square smoothing of the derivatives. The resulting scheme requires at least a one-element overlap between the subdomains. We also discussed some conservation aspects of the interpolation and proposed two conservative algorithms. The interface constraint of mass

conservation appeared to be indispensable when dealing with confined subdomains. Afterwards, we introduced a Chimera strategy to treat complex geometries. Through an example, we showed than the Dirichlet/Neumann method is more accurate than the Dirichlet/Dirichlet method. Then we applied the iteration-by-subdomain algorithm to the solution of flows around moving objects by deriving tensorial transformations and an accurate time integration algorithm.

Finally, the last chapter presented some examples of applications of the method and showed the robustness of the algorithm.

A lot of work remains to be done on overlapping mixed methods. First of all, we only presented a one-dimensional analytical example as a first test for the new methods. Even though this example provided a good insight on what can happen in multi-dimensional problems, the study of a two-dimensional example could reaveal much more on the characteristics of the methods.

When proving the convergence of the iterative Dirichlet/Robin method, we did not try to derive sharp estimates. In addition, we did not investigate the role of the precise role of the overlap. A closer look at the proof of convergence is therefore needed.

Also, we have only dealt with Dirichlet/Robin and Dirichlet/Neumann couplings. The present method can also be applied to the Robin/Robin method which offers much more flexibility in the choice of the coefficients in play in the transmission conditions.

Although we carried out some algebraic applications of the methods, the final goal of this work was the development of an iteration-by-subdomain algorithm based on a Chimera method. It would be intertesting to derive a multidomain formulation for the view of parallelization and compare its performance to that of other existing methods.

Finally but not of least importance, a theoretical study on the application of the overlapping Dirichlet/Neumann(Robin) to the solution of the stationary Stokes and Navier-Stokes equations is to be done, as well as the extension to the case of transient flows.

References

[1] P.-L. Lions. On the Schwarz alternating method. II. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia (USA), 1989. SIAM.

[2] R. Dautrey and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology Volume 2: functional and variational methods.* Springer-Verlag, 2000.

[3] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations.* Springer-Verlag, 1994.

[4] R. Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comp. Meth. Appl. Mech. Eng.*, 156:185–210, 1998.

[5] All the articles. *Comp. Meth. Appl. Mech. Eng.*, 166(1-2):1–182, 1998.

[6] F. Brezzi, M.O. Bristeau, L.P. Franca, M. Mallet, and G. Rogé. A relationship between stabilized finite element methods and the Galerkin method with bubble functions. *Comp. Meth. Appl. Mech. Eng.*, 96:117–129, 1992.

[7] O. Pironneau, J. Liou, and T. Tezduyar. Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains. *Comp. Meth. Appl. Mech. Eng.*, 100:117–141, 1992.

[8] J. Donea. A Taylor-Galerkin method for convection transport problems. *Int. J. Num. Meth. Eng.*, 20:101–119, 1984.

[9] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, 1987.

[10] G.S. Baruzzi, W.G. Habashi, and M.M. Hafez. A second order accurate finite element method for the solutions of the Euler and Navier-Stokes equations. In *Proceedings of the 13*th *International Conference on Numerical Methods in Fluid Dynamics*, pages 509–513, Rome (Italy), 1992. Springer-Verlag.

[11] T.J.R. Hughes and A.N. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In T.J.R. Hughes, editor, *Finite Elements Methods for Convection Dominated Flows*, pages 19–35. ASME, New-York (USA), 1979.

[12] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 32:199–259, 1982.

[13] T. J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accomodating equal-order interpolation. *Comp. Meth. Appl. Mech. Eng.*, 59:85–99, 1986.

[14] T. J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/Least-Squares method for advective-diffusive equations. *Comp. Meth. Appl. Mech. Eng.*, 73:173–189, 1989.

[15] J. Douglas and J. Wang. An absolutely stabilized finite element method. *Math. Comput.*, 52:495–508, 1989.

[16] L.P. Franca, S.L. Frey, and T. J.R. Hughes. Stabilized finite element methods: I. application to the advective-diffusive model. *Comp. Meth. Appl. Mech. Eng.*, 95:253–276, 1992.

[17] T. J.R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems. *Comp. Meth. Appl. Mech. Eng.*, 58:305–328, 1986.

[18] T. J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comp. Meth. Appl. Mech. Eng.*, 127:387–401, 1995.

[19] R. Codina. Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Comp. Meth. Appl. Mech. Eng.*, 190:1579–1599, 2000.

[20] R. Codina and J. Blasco. Analysis of a stabilized finite element approximation of the transient convection-diffusion-reaction equation using orthogonal subscales. *Submitted to Computing and Visualization in Science*, 2001.

[21] T. J.R. Hugues, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Comp. Meth. Appl. Mech. Eng.*, 54:341–355, 1986.

[22] R. Codina. A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation. *Comp. Meth. Appl. Mech. Eng.*, 110:325–342, 1993.

[23] R. Codina. A finite element formulation for the numerical simulation of the convection-diffusion equation. CIMNE Monograph 14, 1993.

[24] R. Temam. *Navier-Stokes equations*. North-Hoolland, 1984.

[25] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1970.

[26] P.L. Lions. *Mathematical Topics in Fluid Mechanics. Volume 1. Incompressible Models*. Oxford University Press, 1996.

[27] R. Tagg. The Couette-Taylor problem. *Nonlin. Sci. Tod.*, 4(3), 1994.

[28] R. Codina. A finite element formulation for viscous incompressible flows. CIMNE Monograph 16, 1993.

[29] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991.

[30] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Comp. Meth. Appl. Mech. Eng.*, 190, 2001.

[31] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Comp. Meth. Appl. Mech. Eng.*, 188:61–82, 2000.

[32] R. Codina and O. Soto. Computation of exterior normals and prescription of periodicial conditions in finite element calculation. Technical Report 112, CIMNE, 1997.

[33] Ramon Codina and Orlando Soto. Finite element solution of the stokes problem with dominating Coriolis force. *Comp. Meth. Appl. Mech. Eng.*, 1997.

[34] M.D. Laundau and M. Lifschitz. *Fluid Mechanics.* Pergamon Press, 1959.

[35] C.G. Speziale. Analytical methods for the development of Reynolds stress closures in turbulence. *Ann. Rev. Fluid Mech.*, 23:107–157, 1991.

[36] D.C. Wilcox. *Turbulence modeling for CFD.* DCW Industries, Inc., 1993.

[37] J. Boussinesq. Théorie de l'écoulement tourbillant. In *Mém. Présentés par Divers Savants Acad. Sci. Inst. Fr.*, volume 23, pages 46–50. 1877.

[38] C.G. Speziale. Some interesting properties of two-dimensional turbulence. *Phys. Fluids*, 24(1):1425–1427, 1981.

[39] C.G. Speziale. Closure models for rotating two-dimensional turbulence. *Geophys. Astrophys. Fluid Dyn.*, 23:69–84, 1983.

[40] B. Mohammadi and O. Pironneau. *Analysis of the k-ε Turbulence Model.* Masson, 1994.

[41] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows, 1992. AIAA Paper 92-0439.

[42] H. Grotjans and F.R. Menter. Wall functions for general application CFD codes. In *Proceedings of the Fourth ECCOMAS Computational Fluid Dynamics Conference*, Athens (Greece), 1998.

[43] N.T. Frink. Tetrahedral unstructured Navier-Stokes method for turbulent flows. *AIAA J.*, 36(11), 1998.

[44] P.B. Bradshaw and P.G. Huang. The Law of the Wall in turbulent flows. In *Proc. Roy. Soc. Lond. A*, volume 451, pages 165–188, 1995.

[45] J. Cousteix. *Turbulence et couche limite (aérodynamique).* Cepadues, 1989.

[46] R.M. Halleen and J.P. Johnston. The influence of rotation on flow in a long rectangular channel - an experimental study. Technical Report MD-18, Dept. Mech. Eng., Stanford University, CA, 1967.

[47] N.N. Mansour, J. Kim, and P. Moin. Reynolds stress and dissipation rate budgets in turbulent channel flow. *J. Fluid Mech.*, 194:15–44, 1988.

[48] J. Kim, S.J. Kline, and J.P. Johnston. Investigation of a reattaching turbulent shear layer: Flow over a backward facing step. *J. Fluid Mech.*, 102:302–308, 1980.

[49] O. Soto. *Estabilización de la Solución por Elementos Finitos de Problemas de Flujo Incompresible con Rotación, Turbulencia, Superficie Libre y Temperatura.* PhD thesis, Universitat Politècnica de Catalunya, Barcelona (Spain), 1996.

[50] D.A. Lyn, S. Einav, W. Rodi, and J.-H. Park. A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder. *J. Fluid Mech.*, 304:285–319, 1995.

[51] G. Bosh and W.Rodi. Simulation of vortex shedding past a square cylinder with different turbulence models. *Int. J. Num. Meth. Fluids*, 28:601–616, 1998.

[52] D.F.G. Durao, M.V. Heitor, and J.C.F Pereira. Measurements of turbulent and periodic flows around a square cross-section cylinder. *Exp. Fluids*, 6:298–304, 1984.

[53] M. Kato and B.E. Launder. The modelling of turbulent flow around stationary and vibrating square cylinders. In F. Durst, N. Kasagi, B.E. Launder, and F.W. Schmidt, editors, *Proceedings of the Ninth International Symposium on Turbulent Shear Flows*, volume 9, pages 10–4–1, Kyoto (Japan), 1993.

[54] P.W. Bearman and D.M. Trueman. An investigation of the flow around rectangular cylinders. *Aeronaut. Q.*, 23:1–6, 1971.

[55] P. Le Tallec. Domain decomposition methods in computational mechanics. In J. Tinsley Oden, editor, *Computational Mechanics Advances*, volume 1 (2), pages 121–220. North-Holland, 1994.

[56] P. E. Bjørstad and O. B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23:1097–1120, 1986.

[57] L.D. Marini and A. Quarteroni. An iterative procedure for domain decomposition methods: A finite element approach. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 129–143, Philadelphia (USA), 1988. SIAM.

[58] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford Science Publications, 1999.

[59] D. Funaro, A. Quarteroni, and P. Zanolli. An iterative procedure with interface relaxation for domain decomposition methods. *SIAM J. Numer. Anal.*, 25:1213–1236, 1988.

[60] P.L. Lions. On the Schwarz alternating method III: a variant for nonoverlapping subdomains. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 202–223, Philadelphia (USA), 1990. SIAM.

[61] R. Glowinski and P. LeTallec. Augmented Lagrangian interpretation of the nonoverlapping Schwarz alternating method. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–231, Philadelphia (USA), 1990. SIAM.

[62] F. Gastaldi and L. Gastaldi. On a domain decomposition for the transport equation: Theory and finite element approximation. *IMA J. Numer. Anal.*, 14:111–135, 1993.

[63] C. Carlenzoli and A. Quarteroni. Adaptive domain decomposition methods for advection-diffusion problems. In I. Babuška, editor, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, volume 75 of *IMA Volumes in Mathematics and its Applications*, pages 165–186. Springer Verlag, 1995.

[64] M.-C. Ciccoli. Adaptive domain decomposition algorithms and finite volume/finite element approximation for advection-diffusion equations. *J. Sci. Comput.*, 11(4):229–341, 1996.

[65] F. Gastaldi, L. Gastaldi, and A. Quarteroni. ADN and ARN domain decomposition methods for advection-diffusion equations. In Petter E. Bjørstad, Magne Espedal, and David Keyes, editors, *Ninth international Conference of Domain Decomposition Methods*. ddm.org, 1998. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[66] R.L. Trotta. Multidomain finite elements for advection-diffusion equations. *Applied Numer. Math.*, 21:91–118, 1996.

[67] M.-C. Ciccoli and R.L. Trotta. Multidomain finite elements and finite volumes for advection-diffusion equations. In Petter E. Bjørstad, Magne Espedal, and David Keyes, editors, *Ninth international Conference of Domain Decomposition Methods*. ddm.org, 1998. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[68] A. Auge, A. Kapurkin, G. Lube, and F.-C. Otto. A note on domain decomposition of singularly perturbed elliptic problems. In Petter E. Bjørstad, Magne Espedal, and David Keyes, editors, *Ninth international Conference of Domain Decomposition Methods*. ddm.org, 1998. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[69] F. Nataf and F. Rogier. Factorization of the convection-diffusion operator and the Schwarz algorithm. *Math. Mod. Meth. Appl. S.*, 5(1):67–93, 1995.

[70] F.-C. Otto. *A Non-Overlapping Domain Decomposition Method for Elliptic Equations*. PhD thesis, Universität Göttingen, Germany, 1999.

[71] G. Lube, F.-C. Otto, and H. Müller. A non-overlapping domain decomposition method for parabolic initial-boundary value problems. *Applied Numer. Math.*, 28:359–369, 1998.

[72] G. Lube, L. Müller, and F.-C. Otto. A non-overlapping DDM of Robin-Robin type for parabolic equations. In C-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, editors, *Eleventh international Conference of Domain Decomposition Methods*. ddm.org, 1999.

[73] A. Alonso, R.L. Trotta, and A. Valli. Coercive domain decomposition algorithms for advection-diffusion equations and systems. *J. Comput. Appl. Math.*, 96:51–76, 1998.

[74] J. F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In T.F. Chan, R. Glowinski, J. Périaux, and O.B. Widlund, editors, *Domain Decomposition Methods*, pages 3–16. SIAM, Philadelphia (USA), 1989.

[75] H.A. Schwarz. Über einige abbildungsdufgaben. *Ges. Math. Abh.*, 11:65–83, 1869.

[76] P.-L. Lions. On the Schwarz alternating method I. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, Philadelphia (USA), 1988. SIAM.

[77] T.P. Mathew. Uniform convergence of the Schwarz alterning method for solving singularly perturbed advection-diffusion equations. *SIAM J. Numer. Anal.*, 35(4):1663–1683, 1998.

[78] F. Nataf. On the use of open boundary conditions in block gauss-seidel methods for the convection-diffusion equation. Technical Report RI284, Centre de Mathématiques Appliquées, Ecole Polytechnique, 1993.

[79] F. Nataf and F. Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Num. Math.*, 75:357–377, 1997.

[80] P. Le Tallec and M.D. Tidriri. Convergence analysis of domain decomposition algorithms with full overlapping for the advection diffusion porblems. *Math. Comput.*, 68(226):585–606, 1999.

[81] Y. Achdou, P. Le Tallec, F. Nataf, and M. Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comp. Meth. Appl. Mech. Eng.*, 184:145–170, 2000.

[82] L.C. Berselli and F. Saleri. New substructuring domain decomposition methods for advection-diffusion equations. *J. Comput. Appl. Math.*, 116:201–220, 2000.

[83] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[84] J. Mandel. Balancing domain decomposition. *Commun. Numer. Meth. Engng*, 9:233–241, 1993.

[85] T. F. Chan and T. P. Mathew. *Domain Decomposition Algorithms*, volume 3 of *Acta Numerica*, pages 61–143. Cambridge University Press, Cambridge, 1994.

[86] X. c. Cai and Y. Sadd. Graph decomposition techniques for general sparse matrices. In SIAM, editor, *Proceedings of the sixth SIAM Conference on Parallel Porocessing for Scientific Computaing*, 1993.

[87] T. F. Chan and D. Goovaerts. Schwarz=Schur: Overlapping versus nonoverlapping domain decomposition. Technical Report CAM Report 88-21, UCA, 1988.

[88] L.S. Kang. *Parellel Algorithms and Domain Decomposition.* Wuhan University Press, Wuhan (China), 1987.

[89] A. Quarteroni. Domain decomposition methods for the incompressible Navier-Stokes equations. In *Computational Fluid Dynamics '94*, pages 72–77. John Wiley & Sons Ltd, 1994.

[90] A. Quarteroni. Domain decomposition algorithms for the stokes equations. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia (USA), 1989. SIAM.

[91] G. Lube, L. Müller, and H. Müller. A new non-overlapping domain decomposition method for stabilized finite element methods applied to the non-stationary Navier-Stokes equations. *Numer. Linear Algebra Appl.*, 7:449–472, 2000.

[92] S.H. Lui. On Schwarz alterning methods for the incompresible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 2001.

[93] C. Bernardi, Y. Maday, and A.T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In H. Brezis and J.-L. Lions, editors, *Nonlinear Partial Differential Equations and Their Applications. Collège de France Seminar, Vol. XI*, pages 13–51, 1994.

[94] R. Glowinski, T.-W. Pan, and J. Périaux. Fictitious domain methods for the Dirichlet problem and its generlization to some flow problems. In K. Morgan, E. O nate, J. Périaux, J. Péraire, and O.C. Zienkiewicz, editors, *Finite Element in Fluids, New Trends and Applications*, pages 347–368. Pineridge Press, Barcelona (Spain), 1993.

[95] J.L. Steger, F.C. Dougherty, and J.A. Benek. A Chimera grid scheme. In K. N. Ghia and U. Ghia, editors, *Advances in Grid Generation*, volume ASME FED-5, pages 59–69, 1983.

[96] G. Houzeaux and R. Codina. An iteration-by-subdomain overlapping Dirichlet/Robin domain decomposition method for advection-diffusion problems. *J. Comput. Appl. Math.*, 2001. Submitted.

[97] R. Codina and G. Houzeaux. Un método de descomposición de dominios de tipo Dirichlet–Neumann con solapamiento para la ecuación de convección–difusión. In L. Ferragut y A. Santos, editor, *Actas del XVII CEDYA - VII CMA, publicadas en CD*, Salamanca (Spain), September 2001. Departamento de Matemática Aplicada, Universidad de Salamanca; SEMA.

[98] L.D. Marini and A. Quarteroni. A relaxation procedure for domain decomposition methods using finite elements. *Num. Math.*, 55:575–598, 1989.

[99] A. Toselli. FETI domain decomposition methods for scalar advection-diffusion problems. *Comp. Meth. Appl. Mech. Eng.*, 190:5759–5776, 2001.

[100] J. R. Cebral and R. Löhner. Conservative load projection and tracking for fluid-structure problems. Technical Report 96-0797, AIAA, 1996.

[101] G. Houzeaux and R. Codina. Transmission conditions with constraints in domain decomposition methods for incompressible Navier-Stokes equations. In K.D. Papailiou, D. Tsahalis, J. Périaux, C. Hirsch, and M. Pandolfi, editors, *Computational Fluid Dynamics 98'. Proceedings of the Fourth European Fluid Dynamics Conference*, pages 194–199, Athens (Greece), 7-11 September 1998.

[102] G. Houzeaux and R. Codina. Transmission conditions with constraints in finite element domain decomposition method for flow problems. *Commun. Numer. Meth. Engng*, 17:179–190, 2001.

[103] P. Wesseling. Introduction to Multi - Grid methods. CR - 195045 ICASE 95 - 11, NASA, 1995.

[104] M. Vázquez and R. Codina. Numerical solution of the Navier - Stokes equations using a splitting technique with Multigrid acceleration. In *Proc. 4th World Congress on Computational Mechanics, Buenos Aires, Argentina*, volume Part 2, page 663. International Asociation for Computational Mechanics, 1998.

[105] M. Vázquez, M. Ravachol, and M. Mallet. Multigrid applied to fully implicit fem solver for turbulent incompressible flows. In *ECCOMAS 2001 Computational Fluid Dynamics Conference*, Swansea (UK), 2001.

[106] M. Vázquez, G. Houzeaux, and R. Codina. Chimera type domain decomposition methods applied to fractional step finite element schemes for incompressible flows. In G. Bugeda E. Oñate and B. Suarez, editors, *CD proceedings of the ECCOMAS 2000*, Barcelona (Spain), 2000.

[107] U. Ghia, K.N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multi-grid method. *J. Comput. Phys.*, 48:387–441, 1982.

[108] J.A. Benek, P.G. Buning, and J.L. Steger. A 3–D Chimera grid embedding technique. *AIAA*, 85–1523CP, 1985.

[109] J.L. Steger and J.A. Benek. On the use of composite grid schemes in computational aerodynamics. *Comp. Meth. Appl. Mech. Eng.*, 64:301–320, 1987.

[110] J.A. Benek, T.L. Tonegan, and N.E. Suhs. Extended Chimera grid embedding system with application to viscous flow, 1987. AIAA Paper 87-1126.

[111] E. Guilmineau, J. Piquet, and P. Queutey. Two-dimensional turbulent viscous flow simulation past airfoils at fixed incidence. *Computers & Fluids*, 26(2):135–162, 1997.

[112] J.P. Slotnick, M. Kandula, and P.G. Buning. Navier-Stokes simulation of the space shuttle launch vehicle flight transonic flowfield using a large scale Chimera grid system. In *AIAA 12th Applied Aerodynamics Conference*, number AIAA-94-1860, Colorado Springs (USA), June 1994.

[113] E. Pärt-Enander. *Overlapping Grids and Applications in Gas Dynamics*. PhD thesis, Uppsala Universitet, Sweden, 1995.

[114] J.-J. Chattot and Y. Wang. Improve treatment of interstecting bodies with the Chimera method and validation with a simple and fast flow sovler. *Computers & Fluids*, 27(5-6):721–740, 1998.

[115] Z.J. Wang and V. Parthasarathy. A fully automated Chimera methodology for multiple moving body problems. *Int. J. Num. Meth. Fluids*, 33:919–938, 2000.

[116] G. Houzeaux and R. Codina. A domain decomposition method for the solution of moving subdomains in fluid dynamics. In E. Oñate S.R. Idelsohn and CIMNE E.N. Dvorkin, editors, *Computational Mechanics. New Trends and Applications. Proceedings of the Fourth World Congress on Computational Mechanics*, pages CD proceedings: Part IV (Fluid Mechanics), Section 1 (Computational Fluid Mechanics), Paper 2, Buenos Aires (Argentina), 29 June-2 July 1998.

[117] G. Houzeaux and R. Codina. A domain decomposition method for the solution of the incompressible Navier-Stokes equations applied to rotating flows. In V.V. Kudriavtsev, W. Cheng, C.P Kleijn, S. Kawano, and M. Souli, editors, *Proceedings of the 1st ASME/JSME Pressure Vessels and Piping Conference*, volume 377-1 of *Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications*, pages 145–151, San Diego (USA), 26-30 July 1998.

[118] G. Houzeaux and R. Codina. A finite element method for the solution of incompressible flows in rotodynamic machines. In C.P Kleijn V.V. Kudriavtsev, W. Cheng and S. Kawano, editors, *Proceedings of the 2nd ASME Pressure Vessels and Piping Conference*, volume 397-2 of *Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications*, pages 165–172, Boston (USA), 1-5 August 1999.

[119] A. Masud and T.J.R. Hughes. A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems. *Comp. Meth. Appl. Mech. Eng.*, 146:91–126, 1997.

[120] E. Mestreau, R. Lohner, and S. Aita. TGV tunnel entry simulations using a finite element code with automatic remeshing, 1993. AIAA Paper 93-0890.

[121] A. Folch. *A Numerical Formulation to solve the ALE Navier-Stokes equations applied to the withdrawal of magma chambers.* PhD thesis, Universitat Politècnica de Catalunya, Barcelona (Spain), 2000.

[122] M. Zhu, T. Fusegi, A. Tabbal, E. Mestreau, H. Malanda, D. Vinteler, A. Goto, and M. Nohmi. A tetrahedral finite-element method for 3d numerical modeling and entire simulation of unsteady turbulent flow in a mixed-flow pump. In *ASME Fluids Engineering Division Summer Meeting*, number FEDSM98-4879, Washington (USA), June 1998.

[123] A. Huerta and W.K. Liu. Viscous flow with large free surface motion. *Comp. Meth. Appl. Mech. Eng.*, 69:277–324, 1988.

[124] A. Soulaimani, M. Fortin, D. Dhatt, and Y. Ouellet. Finite elment simulation of two and three dimensional free surface flows. *Comp. Meth. Appl. Mech. Eng.*, 89:265–296, 1991.

[125] J. Donéa, P. Fasoli-Stella, and S. Giuliani. Lagrangian and Eulerian finite element techniques for transient fluid structure interaction problems. In *Transactions of the fourth International conference of structural mechanics in reactor technology, B1/2*, 1977.

[126] R. Glowinski, T.-W. Pan, and J. Périaux. A fictitious domain method for Dirichlet problems and applications. *Comp. Meth. Appl. Mech. Eng.*, 111:203–303, 1994.

[127] F. Bertrand, P.A. Tanguy, and F. Thibault. A three-dimensional fictitious domain method for incompressible fluid flow problems. *Int. J. Num. Meth. Fluids*, 1997.

[128] R. Glowinski, T.-W. Pan, and J. Périaux. On a domain embedding method for flow around moving rigid bodies. In Petter E. Bjørstad, Magne Espedal, and David Keyes, editors, *Ninth international Conference of Domain Decomposition Methods.* ddm.org, 1998. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[129] T.-W. Pan. Numerical simulation of the motion of a ball falling in an incompressible viscous fluid. *C. R. Acad. Sci. Paris*, 327(Série IIb):1035–1038, 1999.

[130] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux. A distributed Lagrange multiplier/fictitious domain method for particulate flows around moving rigid bodies. *Int. J. Num. Meth. Fluids*, 1999.

[131] L.H. Juárez. Numerical simulation of the sedimentation of an elliptic body in an incompressible viscous fluid. *C. R. Acad. Sci. Paris*, 2001.

[132] R. Glowinski, T.-W. Pan, J. Kearsley, and J. Périaux. Numerical simulation and optimal shape for viscous flow by a fictitious domain method. *Int. J. Num. Meth. Fluids*, 20:685–711, 1995.

[133] A. Ben Abdallah and J.-L. Guermond. A fully parallel mortar finite element projection method for the solution of the unsteady Navier-Stokes equations. In *Proceedings of the Third ECCOMAS Computational Fluid Dynamics Conference*, pages 852–858, Paris (France), 1996. John Wiley & Sons Ltd.

[134] G. Anagnostou, Y. Maday, C. Mavriplis, and A.T. Patera. On the mortar element method: Generalizations and implementation. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 157–173, Philadelphia (USA), 1990. SIAM.

[135] A. Bakker, R.D. Laroche, M.H. Wang, and R.V. Calabrese. Sliding mesh simulation of laminar flow in stirred reactors. *Trans. IChemE.*, 74(Part A):42–44, 1997.

[136] F. Longatte and J.-L. Kuen. Analysis of rotor-stator-circuit interactions in a centrifugal pump. In C.P Kleijn V.V. Kudriavtsev, W. Cheng and S. Kawano, editors, *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference*, number FEDSM99-6866 in Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications, San Francisco (USA), July 1999.

[137] M. Behr and T. Tezduyar. The shear-slip mesh update method. *Comp. Meth. Appl. Mech. Eng.*, 174:261–274, 1999.

[138] R.L. Meakin and N.E. Suhs. Unsteady aerodynamic simulation of multiple bodies in relative motion, 1989. AIAA Paper 89-1996-CP.

[139] N.C. Prewitt, D.M. Belk, and W. Shyy. Parallel computing of overset grids for aerodynamic problems with moving objects. *J. Progr. Aero. Sci.*, 36:117–172, 2000.

[140] F.B. Belgacem and Y. Maday. The mortar element method for three dimensional finite elements. *R.A.I.R.O. Modél. Math. Anal. Numér.*, 31(2):289–302, 1997.

[141] Z.J. Wang and N. Hariharan R. Chen. Recent development on the conservation property of Chimera. *Int. J. Comp. Fluid Dyn.*, 2001. Submitted.

[142] M.S. Engelman and M.A. Jamnia. Transient flow past a circular cylinder: A benchmark solution. *Int. J. Num. Meth. Fluids*, 11:985–1000, 1990.

[143] O.C. Zienkiewicz, P. Nithiarasu, R. Codina, M. Vázquez, and P. Ortiz. Characteristic - Based - Split (CBS) algorithm. part 1. the theory and general discussion. In *Proc. 4th European Computational Fluid Dynamics Conference, Athens, Greece*, volume Part 2, pages 4–16. John Wiley & Sons Ltd., 1998.

[144] A. Kováts. *Design and Performance of Centrifugal and Axial Flow Pumps and Compressors.* Pergamon Press, 1964.

[145] C.E. Brennen. *Hydrodynamics of pumps.* Oxford University Press and Concepts ETI, Inc., 1994.

[146] M.W. Volk. *Pump Characteristics and Applications.* Maral Dekker, Inc., 1996.

[147] A. Ruprecht, C. Bauer, C. Gentner, and G. Lein. Parallel computation of stator-rotor interaction in axial turbine. In C.P Kleijn V.V. Kudriavtsev, W. Cheng and S. Kawano, editors, *Proceedings of the 2nd ASME Pressure Vessels and Piping Conference*, volume 397-2 of *Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications*, Boston (USA), 1-5 August 1999.