The Best Interface Conditions for Domain Decomposition Methods: Absorbing Boundary Conditions

C. Japhet[*]and Frédéric Nataf[†]

January 27, 2003

Abstract

We present an iterative domain decomposition method for solving the convection-diffusion equation. In order to have very fast convergence, we use differential interface conditions of order 1 in the normal direction and of order 2 in the tangential direction to the interface, which are optimized approximations of Absorbing Boundary Conditions. These approximations are different from the ones which are used when truncating an unbounded domain. Numerical tests illustrate this approach.

1 Introduction

Domain decomposition methods are a collection of techniques based on the principle of Divide and Conquer. These methods have been developed for solving partial differential equations over regions in 2D and 3D. They are naturally parallel and adapted to multiprocessing technology. They are therefore getting more and more popular. Their main advantage is the dramatic increase in the size of the problems that can be treated by the combination of the Divide and Conquer approach and parallel computing. Moreover, these algorithms are usually faster even when they are implemented on monoprocessor computers.
The basic idea is to decompose the computational domain into small subregions. The original problem is then reformulated as follows: Find functions in each subdomain such that:

- the partial differential equation is satisfied in each subdomain
- the functions match on the interfaces between the subdomains.

[*] Laboratoire d'Analyse, Géométrie et Applications, Université Paris 13, Avenue J-B Clément, 93430 Villetaneuse, France

[†] CMAP, CNRS UMR7641, Ecole Polytechnique, 91128 Palaiseau, France. nataf@cmapx.polytechnique.fr, www.cmap.polytechnique.fr/~nataf

Usually, the first requirement is met by using direct solvers whereas the second one is enforced iteratively. There are many ways to write matching conditions on an interface. For instance, for a second order scalar partial differential equation, the set of jump conditions $[u] = [\partial u/\partial n] = 0$ (where n is a normal on an interface) is equivalent to $[u] = [\partial u/\partial n + \alpha u] = 0$ for any α. The jump conditions will also be called interface conditions.

It turns out that the speed of convergence is very much sensitive on the interface conditions. More precisely, Exact Absorbing Boundary Conditions (although not used for truncating a domain) are optimal in terms of iteration counts. On the other hand, they are non local operators which which are difficult to implement. As for the truncation of unbounded domains, approximations have to be devised. It turns out that the *approximations* which are good for truncating domains are not suited to domain decomposition methods and vice-versa.

The chapter is organized as follows. In § 2, we recall two basic domain decomposition methods. In § 3,we elaborate on these algorithms and show that the Exact Absorbing Boundary Conditions (EABC) are the best interface conditions in terms of iteration counts. In § 4.1, we analyze two approximation procedures of these EABC's: 1) one of the classical ones for truncating domains (i.e. the approach initiated in [EM77]) and 2) the OO2 (Optimized of Order 2) method which is more relevant to domain decomposition methods. In § 5, numerical tests are given for various interface conditions. In § 6, we conclude.

2 Schwarz Methods for the Laplace Operator

2.1 The original Schwarz method

The first domain decomposition method was developed at the end of the 19th century by the mathematician H. Schwarz. His goal was to study the Laplace operator and not at all to develop numerical methods. At that time, the main tool for this purpose was Fourier analysis and more generally the use of special functions. Geometries of the domain were essentially restricted to simple geometries: rectangles and disks. His idea was to study the case of domains that are the union of simple domains. Far example, let $\Omega = \Omega_1 \cup \Omega_2$ with $\Omega_1 \cap \Omega_2 \neq \emptyset$. We want to solve

$$\begin{array}{l} -\Delta(u) = f \quad \text{in } \Omega \\ u = 0 \quad \text{on } \partial\Omega. \end{array}$$

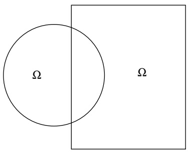

Fig. 1. Overlapping domain decomposition

H. Schwarz proposed the following algorithm (Multiplicative Schwarz Method, **MSM**):

Let (u_1^n, u_2^n) be an approximation to $(u_{|\Omega_1}, u_{|\Omega_2})$ at step n of the algorithm, (u_1^{n+1}, u_2^{n+1}) with

$$\begin{array}{ll} -\Delta(u_1^{n+1}) = f \quad \text{in } \Omega_1 & -\Delta(u_2^{n+1}) = f \quad \text{in } \Omega_2 \\ u_1^{n+1} = 0 \text{ on } \partial\Omega_1 \cap \partial\Omega & u_2^{n+1} = 0 \text{ on } \partial\Omega_2 \cap \partial\Omega \\ u_1^{n+1} = u_2^n \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2}. & u_2^{n+1} = u_1^{n+1} \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}. \end{array}$$

Problem in domain Ω_1 has to be solved before problem in domain Ω_2. This algorithm is sequential.
A slight modification of the algorithm is the additive Schwarz method (**ASM**)

$$\begin{array}{ll} -\Delta(u_1^{n+1}) = f \quad \text{in } \Omega_1 & -\Delta(u_2^{n+1}) = f \quad \text{in } \Omega_2 \\ u_1^{n+1} = 0 \text{ on } \partial\Omega_1 \cap \partial\Omega & u_2^{n+1} = 0 \text{ on } \partial\Omega_2 \cap \partial\Omega \\ u_1^{n+1} = u_2^n \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2}. & u_2^{n+1} = u_1^n \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}. \end{array}$$

Problems in domains Ω_1 and Ω_2 may be solved concurrently. The ASM is a parallel algorithm and is adapted to parallel computers. H. Schwarz proved the linear convergence (u_1^n, u_2^n) to $(u_{|\Omega_1}, u_{|\Omega_2})$ as n tends to infinity.
The benefit of these algorithms is the saving in memory requirements. Indeed, if the problems are solved by direct methods, the the cost of the storage is non linear with respect to the number of unknowns. By dividing the original problem into smaller pieces the amount of storage can be significantly reduced.

2.2 Towards faster Schwarz methods

As far as CPU is concerned, the original algorithms ASM and MSM are very slow. Another weakness is the need of overlapping subdomains. In order to remedy to these drawbacks, it has been proposed [Lio90] to replace the Dirichlet interface conditions on $\partial\Omega_i \backslash \partial\Omega$, $i = 1, 2$ by Robin interface conditions ($\partial_{n_i} + \alpha$, where n is the outward normal to subdomain Ω_i). For example, the modified ASM reads

$$\begin{array}{l} -\Delta(u_1^{n+1}) = f \quad \text{in } \Omega_1, \\ u_1^{n+1} = 0 \quad \text{on } \partial\Omega_1 \cap \partial\Omega, \\ (\frac{\partial}{\partial n_1} + \alpha)(u_1^{n+1}) = (-\frac{\partial}{\partial n_2} + \alpha)(u_2^n) \quad \text{on } \partial\Omega_1 \cap \overline{\Omega_2}, \\ \\ -\Delta(u_2^{n+1}) = f \quad \text{in } \Omega_2, \\ u_2^{n+1} = 0 \quad \text{on } \partial\Omega_2 \cap \partial\Omega \\ (\frac{\partial}{\partial n_2} + \alpha)(u_2^{n+1}) = (-\frac{\partial}{\partial n_1} + \alpha)(u_1^n) \quad \text{on } \partial\Omega_2 \cap \overline{\Omega_1}. \end{array}$$

Note that the normals n_1 and n_2 are opposite.
A good choice of the parameter α yields a much better convergence and the overlap between subdomains is optional. The boundary conditions imposed on $\partial\Omega_i \backslash \partial\Omega$, $i = 1, 2$ are called interface (or matching) conditions.
Another major improvement to Schwarz methods comes from the use of Krylov type methods (e.g. CG, GMRES, BICG, QMR, ...) in order to enforce the continuity relations on the interfaces. These methods come in replacement of the relaxation procedure used in ASM or MSM.

In order to use the various Krylov methods, it is necessary to identify the linear system of equations that correspond to the interface conditions. We consider this question in a more general setting than the Laplace operator. Let Ω be an open set of $\mathbb{R}^d$, $\mathcal{L}$ be a linear second order scalar partial differential operator with real coefficients and whose principal symbol is elliptic. Let $f : \Omega \longrightarrow \mathbb{R}$, we consider the following boundary value problem: find u such that

$$\begin{array}{l} \mathcal{L}(u) = f \quad \text{ in } \Omega, \\ u = 0 \quad \text{ on } \partial\Omega. \end{array}$$

The domain is decomposed into two subdomains Ω_1 and Ω_2 such that $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$ (with or without overlaps between the subdomains). The interface conditions on $\Gamma_i = \partial\Omega_i \backslash \partial\Omega$, $i = 1, 2$ are denoted by $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively. They don't need to be specified in this section. The additive Schwarz method (ASM) reads

$$\begin{array}{l} \mathcal{L}(u_i^{n+1}) = f \quad \text{ in } \Omega_i \\ u_i^{n+1} = 0 \text{ on } \partial\Omega_i \cap \partial\Omega \\ \mathcal{B}_i(u_i^{n+1}) = \mathcal{B}_i(u_j^n) \quad \text{ on } \Gamma_i,\ 1 \leq i \neq j \leq 2. \end{array} \tag{2.1}$$

We first identify the above ASM to a Jacobi method applied to interfacial quantities. Indeed, we denote by $u_i(f, \mu_i)$, $i = 1, 2$ the solution to

$$\begin{array}{l} \mathcal{L}(u_i) = f \quad \text{ in } \Omega_i \\ u_i = 0 \text{ on } \partial\Omega_i \cap \partial\Omega \\ \mathcal{B}_i(u_i) = \mu_i \quad \text{ on } \Gamma_i. \end{array}$$

With these notations, the ASM is equivalent to

$$\begin{array}{l} u_1^{n+1} = u_1(f, \mathcal{B}_1(u_2^n)) \\ u_2^{n+1} = u_2(f, \mathcal{B}_2(u_1^n)) \end{array} \tag{2.2}$$

Let us define interfacial quantities

$$\mu_1^n \equiv \mathcal{B}_1(u_2^n) \text{ and } \mu_2^n \equiv \mathcal{B}_2(u_1^n).$$

From (2.2) we have

$$\begin{array}{l} \mu_1^{n+1} \equiv \mathcal{B}_1(u_2^{n+1}) = \mathcal{B}_1(u_2(f, \mathcal{B}_2(u_1^n))) \\ \mu_2^{n+1} \equiv \mathcal{B}_2(u_1^{n+1}) = \mathcal{B}_2(u_1(f, \mathcal{B}_1(u_2^n))) \end{array}$$

or by linearity of the operators,

$$\left(\begin{array}{c} \mu_1^{n+1} \\ \mu_2^{n+1} \end{array} \right) - \left(\begin{array}{c} \mathcal{B}_1(u_2(0, \mathcal{B}_2(u_1^n))) \\ \mathcal{B}_2(u_1(0, \mathcal{B}_1(u_2^n))) \end{array} \right) = \left(\begin{array}{c} \mathcal{B}_1(u_2(f, 0) \\ \mathcal{B}_2(u_1(f, 0) \end{array} \right).$$

We have just proved:

Lemma 2.1 *The ASM (2.1) is a Jacobi method applied to the linear system of equations*

$$A \left(\begin{array}{c} \mu_1 \\ \mu_2 \end{array} \right) = b \tag{2.3}$$

where A is a linear operator defined by

$$A\begin{pmatrix}\mu_1\\\mu_2\end{pmatrix}=\left[Id-\begin{pmatrix}0 & \mathcal{B}_1(u_2(0,.))\\\mathcal{B}_2(u_1(0,.)) & 0\end{pmatrix}\right]\begin{pmatrix}\mu_1\\\mu_2\end{pmatrix}\tag{2.4}$$

and

$$b=\begin{pmatrix}\mathcal{B}_1(u_2(f,0)\\\mathcal{B}_2(u_1(f,0)\end{pmatrix}.$$

We shall call A the substructured operator associated to the ASM. When using a Krylov type method for solving the above system of equations, a matrix vector product corresponds to solving one boundary value problem in each subdomain. These BVP's are independent and the method is thus parallel.

3 The Best Interface Conditions for Domain Decomposition Methods: ABC's

Our point in this section is to show that using Dirichlet to Neumann maps (**DtN**) in the interface conditions is optimal for the convergence of additive and multiplicative Schwarz methods.
More precisely, let Ω be an open set of $\mathbb{R}^n$, $n \geq 1$. Let $\mathcal{L}$ be a second order partial differential equation whose principal symbol is elliptic (e.g. Laplace eq., convection-diffusion eq., Helmholtz eq.,...). We consider the following boundary value problem:
Find $u:\ \Omega \to \mathbb{R}$ such that

$$\begin{array}{l}\mathcal{L}(u)=f \quad \text{in } \Omega\\ u=0 \quad \text{on } \partial\Omega\end{array}\tag{3.1}$$

where f is a given right hand-side.
In order to write a domain decomposition method for this problem, the domain Ω is decomposed into 2 subdomains Ω_1 and Ω_2 such that $\bar{\Omega}=\bar{\Omega}_1\cup\bar{\Omega}_2$ (with or without overlaps between the subdomains). The interface conditions on $\Gamma_i = \partial\Omega_i\backslash\partial\Omega$, $i=1,2$ are denoted by $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively. They will be chosen later. The outward normal derivative on Γ_i is denoted by n_i. The additive Schwarz method reads

$$\begin{array}{l}\mathcal{L}(u_i^{n+1})=f \quad \text{in } \Omega_i\\ u_i^{n+1}=0 \text{ on } \partial\Omega_i\cap\partial\Omega\\ \mathcal{B}_i(u_i^{n+1})=\mathcal{B}_i(u_j^n) \quad \text{on } \Gamma_i,\ 1\leq i\neq j\leq 2.\end{array}\tag{3.2}$$

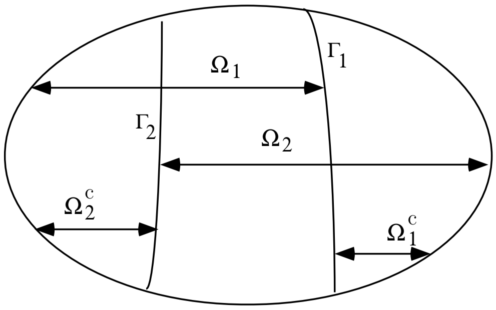

Fig. 2.

Remark 3.1 *When $\mathcal{B}_1 = \mathcal{B}_2 = Id$, (3.2) is the classical Schwarz method. When $\mathcal{B}_i = \partial_{n_i} + \alpha$, $i = 1, 2$, algorithm (3.2) has been proposed in [Lio90].*

We now investigate the use of DtN maps in the interface conditions on Γ_i, $i = 1, 2$. Let us denote the interior of the complementary set of Ω_i by Ω_i^c, $i = 1, 2$ ($\Omega_i^c = \Omega \backslash \bar{\Omega}_i$). Let DtN_i^c be the map defined by:

$$\begin{array}{l} \text{Let } u_0 : \Gamma_i \to \mathbb{R} \\ DtN_i^c(u_0) \equiv \frac{\partial}{\partial n_i^c}(v)_{|\Gamma_i} \end{array} \tag{3.3}$$

where n_i^c is the outward normal to Ω_i^c and v satisfies the following boundary value problem

$$\begin{array}{l} \mathcal{L}(v) = 0 \text{ in } \Omega_i^c \\ v = 0 \text{ on } \partial\Omega_i^c \backslash \Gamma_i \\ v = u_0 \text{ on } \Gamma_i. \end{array}$$

The definition of $\mathcal{B}_1$ and $\mathcal{B}_2$ is

$$\mathcal{B}_i = \frac{\partial}{\partial n_i} + DtN_i^c, \ i = 1, 2 \tag{3.4}$$

Let us notice that $\mathcal{B}_i$ is also an exact ABC (Absorbing boundary condition). Indeed, if $Supp\, f \subset \Omega_i$, the solution to

$$\begin{array}{l} \mathcal{L}(v) = f \text{ in } \Omega_i \\ v = 0 \text{ on } \partial\Omega_i \backslash \Gamma_i \\ \mathcal{B}_i(v) = 0 \text{ on } \Gamma_i. \end{array}$$

is the restriction to Ω_i of the solution u to (3.1).
Surprisingly enough, the use of $\mathcal{B}_i$ as an interface condition for a Schwarz method is optimal.

Result 3.1

- *The additive Schwarz method (3.2) with the interface conditions $\mathcal{B}_i$ defined by (3.4) converges in two iterations*
- *This is optimal in the sense that no additive Schwarz method defined by (3.2) can converge in less than two iterations.*

Proof Let us prove the convergence in two steps. By linearity, it is sufficient to consider the homogeneous case $f = 0$ and to prove that $u_i^2 = 0$, $i = 1, 2$. First we prove that whatever u_2^0 is, we have $\mathcal{B}_2(u_1^1) = 0$. Since $\Omega_2^c \subset \Omega_1$, we have by definition of DtN_2^c that $DtN_2^c(u_{1|\Gamma_2}^1) = \frac{\partial u_1^1}{\partial n_2^c}$, i.e. $-\frac{\partial u_1^1}{\partial n_2^c} + DtN_2^c(u_{1|\Gamma_2}^1) = 0$. On Γ_2, we have $n_2^c = -n_2$ so that we have proved that

$$\mathcal{B}_2(u_1^1) = \frac{\partial u_1^1}{\partial n_2} + DtN_2^c(u_{1|\Gamma_2}^1) = 0.$$

Similarly, we have $\mathcal{B}_1(u_2^1) = 0$.
It follows that $u_i^2 = 0$, $i = 1, 2$. Indeed, from (3.2) we have that u_i^2 satisfies a homogeneous boundary value problem whose solution is zero.

We prove now the optimality of the interface conditions (3.4). Here, f is arbitrary. At step 1, the value of u_i^1 $(i = 1, 2)$ does not depend at all on the value of f in Ω_i^c. Convergence in one step would mean that the solution in Ω_i does not depend on the value of f in Ω_i^c. This contradicts the assumption of ellipticity on the principal symbol of $\mathcal{L}$. ■

Remark 3.2 *This type of result is also valid for a decomposition of Ω into more than two subdomains. In [NRdS95], it is proved that for a decomposition into N strips it is possible to define interface conditions that yield a convergence in N steps and convergence into less than N steps is impossible.*
In [Nie98], results and conjectures (negative) are given for an arbitrary decomposition of Ω into subdomains.

4 Optimized Interface Conditions

We have seen in the previous section that, in terms of iteration counts, the optimal interface conditions are the exact ABC (3.4). For the same reasons that for artificial boundary conditions, we consider approximations that will be easier to implement. Indeed, the Dirichlet-to-Neumann (DtN) operator arising from an elliptic operator is a non local operator. As a result, the Interface Conditions (3.4) are difficult to implement since it is necessary to use Fourier transform (or more generally decompositions on special functions) on the interfaces. Moreover, an explicit form of the DtN is known only in special cases (constant coefficient operators, straight or circled interfaces).
A first possibility developed in § 4.1 is to use "off the shelf" approximate ABC as interface conditions, i.e. the ones which are used on the boundary of a truncated domain, see [EM77], [Hal82]. A second idea, see § 4.2, is to design interface conditions related more closely to the domain decomposition method. We choose them so that they are easy to implement and lead to the best possible convergence rate. Both approaches are compared on a theoretical point of view in § 4.2.3.
The methodology that is developed in this section is general and has been successfully applied to Fluid Dynamics [Jap97], sound propagation [CN98], electromag-

netic waves, porous media flows [WFNS98], ... As an example relevant to fluid dynamics we consider the convection-diffusion equation.

4.1 Approximate ABC as Interface Conditions

The model equation is the convection-diffusion equation:

$$\frac{\partial u}{\partial t} + \boldsymbol{a}.\nabla u - \nu \Delta u = f \tag{4.1}$$

This equation is important in itself in engineering or environnemental sciences for instance. It models the transport and diffusion of species (pollutant in air or water, electrons in semiconductor devices, . . .) in a given flow (with velocity field $\boldsymbol{a}$) . It is also a key ingredient in Navier-Stokes equations. An implicit scheme in time will demand at each time step the solving of

$$\mathcal{L}(u) \equiv \frac{u}{\Delta t} + \boldsymbol{a}.\nabla u - \nu \Delta u = f \tag{4.2}$$

Far sake of simplicity, we consider the plane $\mathbb{R}^2$ decomposed into two half-planes $\Omega_1 =]-\infty, \delta[\times\mathbb{R}$ and $\Omega_2 =]0, \infty[\times\mathbb{R}$, $\delta \geq 0$.
Our results are based on Fourier analysis. We denote the Fourier transform $\hat{f}(k)$ of $f(y) : \mathbb{R} \longrightarrow \mathbb{R}$ by

$$\hat{f}(k) = \mathcal{F}_y(f)(k) := \int_{-\infty}^{\infty} e^{-Iky} f(y) dy$$

($I^2 = -1$) and the inverse Fourier transform of $\hat{f}(k)$ by

$$f(y) = \mathcal{F}_y^{-1}(\hat{f})(y) := \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{Iky} \hat{f}(k) dk.$$

We shall use also the Fourier transform of a convolution operator of kernel h:

$$\Lambda(u)(y) := \int_{-\infty}^{\infty} h(y-z) u(z) dz.$$

By taking the Fourier transform of the above definition, we get

$$\mathcal{F}_y(\Lambda(u))(k) = \hat{h}(k)\,\hat{u}(k)$$

or equivalently with $\hat{\Lambda}(k) := \hat{h}(k)$,

$$\Lambda(u) = \mathcal{F}_y^{-1}(\hat{\Lambda}(k)\hat{u}(k)).$$

The function $\hat{\Lambda}(k)$ is called the (Fourier) symbol of the operator Λ. For example, the symbol of the operator $-\partial_{yy}$ is the polynomial k^2. More generally, the symbol of any constant coefficient differential operator is a polynomial in the Fourier variable k and vice-versa.

The symbol of the composition of two convolution operators is the product of their symbols since

$$\Lambda_1 \circ \Lambda_2(u) = \mathcal{F}_y^{-1}(\hat{\Lambda}_1(k)\hat{\Lambda}_2(k)\hat{u}(k)).$$

This will be denoted shortly by $\mathcal{F}_y(\Lambda_1\Lambda_2) = \hat{\Lambda}_1\hat{\Lambda}_2$.
According to § 3, we compute DtN_i^c, $i = 1, 2$ given by (3.3). By definition, we have $\Omega_1^c =]\delta, \infty[\times\mathbb{R}$. Let $u_0 : \mathbb{R} \to \mathbb{R}$ be an arbitrary function, then $DtN_1^c(u_0)(y) = -\frac{\partial v}{\partial x}(\delta, y)$ where v satisfies

$$\mathcal{L}(v) = 0 \text{ in } \Omega_1^c \tag{4.3}$$

$$v \to 0 \text{ as } x \to \infty \tag{4.4}$$

$$v(\delta, y) = u_0(y) \ \ \forall y \in \mathbb{R}. \tag{4.5}$$

We perform a Fourier transform of (4.3), in the y direction and get

$$\frac{\hat{v}}{\Delta t} + a_x\frac{\partial \hat{v}}{\partial x} + a_y Ik\hat{v} - \nu(\frac{\partial^2 \hat{v}}{\partial x^2} - k^2\hat{v}) = 0.$$

For a given wavenumber k, $\hat{v}$ solves an ODE whose general solution is

$$\hat{v}(x, k) = \alpha(k)e^{\lambda_1(x-\delta)} + \beta(k)e^{-\lambda_2(x-\delta)}$$

where

$$\lambda_1(k) = \frac{a_x + \sqrt{\frac{4\nu}{\Delta t} + a_x^2 + 4Ia_y k\nu + 4k^2\nu^2}}{2\nu} \tag{4.6}$$

and

$$\lambda_2(k) = \frac{-a_x + \sqrt{\frac{4\nu}{\Delta t} + a_x^2 + 4Ia_y k\nu + 4k^2\nu^2}}{2\nu}. \tag{4.7}$$

The solution is bounded at infinity so that $\alpha(k) = 0$. The boundary condition (4.5) yields

$$\hat{v}(x, k) = \hat{u}_0(k)e^{-\lambda_2(x-\delta)}.$$

Finally, we have

$$DtN_1^c(u_0) = \mathcal{F}_y^{-1}(\lambda_2(k)\hat{u}_0(k))$$

i.e. the symbol of the operator DtN_1^c is λ_2. Similarly, we have

$$DtN_2^c(u_0) = \mathcal{F}_y^{-1}(\lambda_1(k)\hat{u}_0(k))$$

i.e. the symbol of the operator DtN_2^c is λ_1. The symbols are not polynomial in k so that the DtN maps are not differential operators.

The optimal interface conditions (3.4) are pseudo-differential operators that would be very difficult to use in practice. Following [EM77], [Hal82] the "classical" approximations to (3.4) are obtained by taking a Taylor expansion of the symbols in the vicinity of $k = 0$. In the physical space, we have at order 0:

$$\frac{\partial}{\partial n_1} + DtN_1^c \simeq \partial_{n_1} + \frac{-a_x + \sqrt{a_x^2 + \frac{4\nu}{\Delta t}}}{2\nu}, \quad \frac{\partial}{\partial n_2} + DtN_2^c \simeq \partial_{n_2} + \frac{a_x + \sqrt{a_x^2 + \frac{4\nu}{\Delta t}}}{2\nu} \tag{4.8}$$

at order 1, we have

$$\begin{aligned} \frac{\partial}{\partial n_1} + DtN_1^c &\simeq \partial_{n_1} + \frac{-a_x+\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}{2\nu} + \frac{a_y}{\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}\partial_y \\ \frac{\partial}{\partial n_2} + DtN_2^c &\simeq \partial_{n_2} + \frac{a_x+\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}{2\nu} + \frac{a_y}{\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}\partial_y \end{aligned} \tag{4.9}$$

and at order 2,

$$\begin{aligned} \frac{\partial}{\partial n_1} + DtN_1^c &\simeq \partial_{n_1} + \frac{-a_x+\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}{2\nu} + \frac{a_y}{\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}\partial_y - \frac{\nu}{\sqrt{a_x^2-\frac{4\nu}{\Delta t}}}(1+\frac{a_y^2}{a_x^2+\frac{4\nu}{\Delta t}})\partial_{yy} \\ \frac{\partial}{\partial n_2} + DtN_2^c &\simeq \partial_{n_2} + \frac{a_x+\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}{2\nu} + \frac{a_y}{\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}\partial_y - \frac{\nu}{\sqrt{a_x^2+\frac{4\nu}{\Delta t}}}(1+\frac{a_y^2}{a_x^2+\frac{4\nu}{\Delta t}})\partial_{yy} \end{aligned} \tag{4.10}$$

These interface conditions will be referred to as Taylor of order 0, 1 and 2 interface conditions. The new domain decomposition methods are obtained by taking the above boundary conditions as interface conditions $\mathcal{B}_1$ and $\mathcal{B}_2$ in (2.1). Numerical tests, see below § 5, show that when the flow is normal to the interface the domain decomposition method based on the Taylor interface conditions has a fast convergence. When the flow is tangential to the interface, results are poor. This is due to the fact that the above interface conditions are not related to the convergence rate of the domain decomposition method. It is shown in the next section that it is possible to have much better convergence rates if other interface conditions are used.

4.2 Analysis of the domain decomposition method and Optimized Interface Conditions

In order to design better interface conditions, we first analyze the convergence of the domain decomposition method. Then optimized interface conditions will be proposed.

4.2.1 Analysis of the domain decomposition method

The additive Schwarz method (2.1) is applied to a problem set on the plane $\mathbb{R}^2$ decomposed into two half-planes $\Omega_1 =]-\infty, \delta[\times\mathbb{R}$ and $\Omega_2 =]0, \infty[\times\mathbb{R}$, $\delta \geq 0$. We consider the case where the optimal (but integro-differential) interface conditions (3.4) are replaced by partial differential operators

$$\mathcal{B}_i = \frac{\partial}{\partial n_i} + DtN_{i,ap}^c, \quad i = 1, 2 \tag{4.11}$$

to be chosen later. The symbol of $DtN^c_{i,ap}$ is denoted by $\lambda_{i,ap}$, $i = 1, 2$. Both symbols are polynomial since $DtN^c_{i,ap}$ is a differential operator in the y direction. We compute now the convergence rate of the additive Schwarz method (2.1). By linearity, it is sufficient to consider the convergence to zero in the homogeneous case $f = 0$. In domain Ω_1, the general form of $\hat{u}_1^{n+1}(x, k)$ is

$$\hat{u}_1^{n+1}(x,k) = \alpha_1^{n+1}(k)\ e^{\lambda_1(k)x}$$

and in domain Ω_2,

$$\hat{u}_2^{n+1}(x,k) = \alpha_2^{n+1}(k)\ e^{-\lambda_2(k)x}.$$

The interface conditions $\mathcal{B}_{1,2}$ in (2.1) yields respectively

$$\begin{aligned}(\lambda_1 + \lambda_{2,ap})\,\alpha_1^{n+1} &= (-\lambda_2 + \lambda_{2,ap})\,e^{-\delta\lambda_2}\alpha_2^n\\ (\lambda_2 + \lambda_{1,ap})\,\alpha_2^{n+1} &= (-\lambda_1 + \lambda_{1,ap})\,e^{-\delta\lambda_1}\alpha_1^n\end{aligned}$$

Hence, we have

$$\alpha_1^{n+1}(k) = \rho_2(k)\alpha_2^n(k), \quad \alpha_2^{n+1}(k) = \rho_1(k)\alpha_1^n(k)$$

where

$$\rho_i(k) := \left(\frac{-\lambda_i(k) + \lambda_{i,ap}(k)}{\lambda_j(k) + \lambda_{i,ap}(k)}\right) e^{-\delta\lambda_i(k)},\ i = 1, 2,\ i \neq j. \tag{4.12}$$

Thus, we have

Theorem 4.1 *Let the plane $\mathbb{R}^2$ be decomposed into $\Omega_1 =]-\infty, \delta[\times\mathbb{R}$ and $\Omega_2 =]0, \infty[\times\mathbb{R}$, $\delta \geq 0$. Let u_i^{n+1} denote the iterate of the ASM (3.2) with the interface conditions $\mathcal{B}_i$ defined by (4.11).*
Then, we have

$$\begin{aligned}&\max(|\hat{u}_1^n(\delta,k) - \hat{u}_1(\delta,k)|, |\hat{u}_2^n(0,k) - \hat{u}_2(0,k)|)\\ \leq\ &\max(|\rho_1(k)|, |\rho_2(k)|)^n \max(|\hat{u}_1^0(\delta,k) - \hat{u}_1(\delta,k)|, |\hat{u}_2^0(0,k) - \hat{u}_2(0,k)|), \forall k \in \mathbb{R}.\end{aligned}$$

Remark 4.1 *This analysis shows that when the subdomains overlap ($\delta \neq 0$) the convergence rate tends to zero as $k \to \infty$. This is at the expense of extra computations since the solution on the overlapping region is computed twice. Moreover, overlapping decompositions might be more difficult to implement in practice.*
In the non-overlapping case ($\delta = 0$), since $\lambda_i(k)$ is supposed to be a polynomial, the convergence rate tends to one as $k \to \infty$.

Remark 4.2 *With the interface conditions of section 4.1, the convergence rate is zero at $k = 0$. More precisely, with the Taylor of order l interface conditions, we have $|\rho_i(k)| = O(|k|^{l+1})$.*

These remarks don't say anything about a uniform bound (i.e. independent of k) of the convergence rate. In [NN97], it is proved that a simple choice of the interface conditions guarantees convergence:

Lemma 4.1 *Let the interface conditions (4.11) be chosen such that*

$$\lambda_{1,ap} = \frac{\boldsymbol{a}.n_1}{2\nu} + p(k)$$
$$\lambda_{2,ap} = \frac{\boldsymbol{a}.n_2}{2\nu} + p(k)$$

where $p(k)$ satisfies: $\Re(p(k)) > 0$ and $sgn(\Im(p(k)) = sgn(a_y\,k)$. Then, we have

$$\max(|\rho_1(k)|, |\rho_2(k)|) < 1 \quad \forall k \in \mathbb{R}.$$

In the above expressions for λ_i, we have used the fact that $\boldsymbol{a}.n_1 = -\boldsymbol{a}.n_2 = a_x$. This gives a more intrinsic expression to the symbol of the interface conditions. Let us remark that this does not contradict the fact that for $\delta = 0$, the convergence rate of the ASM tends to one as k tends to infinity. Let us notice that with the choice made in this lemma, we have the following simple expression

$$\rho_i(k) = \frac{-\frac{\sqrt{\frac{4\nu}{\Delta t} + a_x^2 + 4Ia_yk\nu + 4k^2\nu^2}}{2\nu} + p(k)}{\frac{\sqrt{\frac{4\nu}{\Delta t} + a_x^2 + 4Ia_yk\nu + 4k^2\nu^2}}{2\nu} + p(k)},\ i = 1, 2.$$

So far in this section, we have only considered the ASM. As noticed in § 2.2, the ASM may be seen as a Jacobi algorithm applied to the substructured problem (2.3). This paves the way to the use of more efficient iterative solvers such as Krylov methods. It is interesting in our case to see the relation between the convergence rate of the ASM and the substructured formulation. It is easy to check the

Lemma 4.2 *The Fourier symbol of the substructured operator A (2.3) is*

$$\hat{A}(k) = \begin{pmatrix} 1 & -\rho_2(k) \\ -\rho_1(k) & 1 \end{pmatrix}$$

4.2.2 OO2 (Optimized of order 2) interface conditions

In this section, we present a procedure for designing optimized interface conditions in the non overlapping case.
For sake of efficiency and simplicity in the implementation, the interface conditions are sought in the form

$$\mathcal{B}_1 = \frac{\partial}{\partial \mathbf{n_1}} + c_1 + c_2\frac{\partial}{\partial \boldsymbol{\tau_1}} + c_3\frac{\partial^2}{\partial \boldsymbol{\tau_1^2}}, \quad \mathcal{B}_2 = \frac{\partial}{\partial \mathbf{n_2}} + c_4 + c_5\frac{\partial}{\partial \boldsymbol{\tau_2}} + c_6\frac{\partial^2}{\partial \boldsymbol{\tau_2^2}} \qquad (4.13)$$

with $c_i \in \mathbb{R}$ and where τ_i is the tangential derivative along the interface $\partial\Omega_i$.
We choose the coefficients c_i:

- First according to Lemma 4.1, we take

$$c_1 = \frac{\boldsymbol{a}.\boldsymbol{n}_1}{2\nu} + c_0, \quad c_4 = \frac{\boldsymbol{a}.\boldsymbol{n}_2}{2\nu} + c_0,\ c_0 > 0,$$

$$c_2 = -c_5, \quad sgn(c_i) = sgn(\boldsymbol{a}.\tau_i), \; i = 2, 5$$

and

$$c_3 = c_6 > 0.$$

This ensures the convergence of the ASM. So we only have to determine c_0, c_2, c_3.

- Then, we choose $c_0 = \sqrt{a_x^2 + \frac{4\nu}{\Delta t}}\;/2\nu$ so that the interface condition is exact for the lowest wave number (i.e. $\rho_1(0) = \rho_2(0) = 0$).
- Finally, we compute c_2 and c_2 by minimizing the maximum of the convergence rate of the ASM $k \to \rho(k, c_2, c_3)$ on the interval $|k| \leq k_{max}$ where k_{max} is a given constant, $k_{max} > 0$ (in the discrete case, $k_{max} = \frac{\pi}{h}$ where h is the mesh size in y). The details of the optimization procedure can be found in [Jap97]. Let us say that in practice, it is sufficient to solve a monodimensional optimization problem expressed in terms of a wavenumber k_{int} where the convergence rate vanishes (see Figure 3). So, the OO2 conditions are easy to use and not costly.

Remark 4.3 *In the OO2 interface conditions, the minimization problem on c_2 and c_3 is on a set of conditions which satisfy Lemma 4.1, i.e. on a set of conditions which ensures (adding a condition on the sign of c_2) the convergence of the Schwarz algorithm. So an approximate minimization problem on the same set of conditions will also ensure the convergence. In the case of 2 subdomains, the convergence was proved by computing explicitly the convergence ratio. When the domain is decomposed in N subdomains (strips) the convergence ratio is estimated in function of the convergence ratio of the 2 subdomains case and the decomposition geometry. The convergence is proved in [NN97].*

Remark 4.4 *We have considered so far the convergence rate of the ASM. One might argue that when using Krylov type methods the optimization criterium should be the condition number of the substructured operator (2.3). It turns out that both criteria (minimizing the convergence rate of the additive Schwarz method and minimizing the condition number) are identical. Indeed, let us examine the condition number of A. For a given wavenumber k, the eigenvalues of $\hat{A}(k)$ are $1 \pm \sqrt{\rho_1 \rho_2}$. Therefore, the eigenvalues of A restricted to the frequency domain $|k| \leq |k_{\max}|$ (denoted A_R) is the set $\{1 \pm \sqrt{\rho_1 \rho_2},\; |k| \leq |k_{\max}|\}$. When $\lambda_{i,ap}$, $i = 1, 2$ are chosen according to Lemma 4.1, $\rho_1 = \rho_2$ and $|\rho_1| < 1$. It is easy to check that the condition number of A_R is $\kappa(A_R) = \frac{1 + \max_{|k| \leq k_{\max}} |\rho_1(k)|}{1 - \max_{|k| \leq k_{\max}} |\rho_1(k)|}$. Minimizing $\kappa(A_R)$ is thus equivalent to minimizing $\max_k |\rho(k)|$.*

4.2.3 Behaviors of the convergence rates

In this section, we compare the convergence of the ASM endowed with either Taylor interface conditions (4.8)-(4.9)-(4.10) or OO2 interface conditions. Let us first notice that all of these interface conditions satisfy the conditions of Lemma 4.1 so that the convergence rate is smaller than 1. Figure 3 illustrate the behavior of the convergence rate with the different interface conditions.

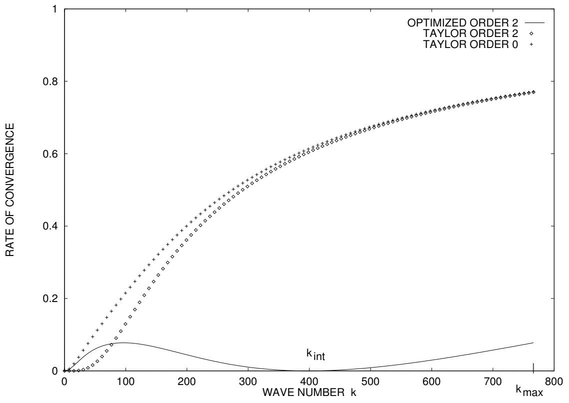

Fig. 3. Rate of convergence versus wave numbers k
$0 \leq k \leq k_{max},\ k_{max} = \frac{\pi}{h}$
$a_x = 1,\ a_y = 1,\ \nu = 0.01,\ \Delta t = \infty,\ h = \frac{1}{240}$

Moreover, when the mesh size tends to 0, the condition number is asymptotically much better for OO2 than for Taylor order 0 or 2 interface conditions :

Theorem 4.2 *Let $\Omega = \mathbb{R}^2$ be decomposed in 2 subdomains $\Omega_1 = \mathbb{R}^- \times \mathbb{R}$ and $\Omega_2 = \mathbb{R}^+ \times \mathbb{R}$. Let $a_x \in \mathbb{R}$, $a_x \neq 0$, $a_y = 0$ and $\Delta t > 0$ in (4.2). Let h be the mesh size, and let $(\rho_{max})_{IC}$ be the maximum of $\rho_1 = \rho_2$ on $0 \leq k \leq \frac{\pi}{h}$ with the interface condition IC. Let $\alpha = 1 + \frac{4\nu}{\Delta t a^2}$. Then, when $h \to 0$:*

$$\begin{aligned}
(\rho_{max})_{Taylor\ order\ 0} &\approx 1 - \tfrac{2}{\pi}\alpha^{\frac{1}{2}}(\tfrac{|a|h}{\nu}) \\
(\rho_{max})_{Taylor\ order\ 2} &\approx 1 - \tfrac{4}{\pi}\alpha^{\frac{1}{2}}(\tfrac{|a|h}{\nu}) \\
(\rho_{max})_{OO2} &\approx 1 - 8\alpha^{\frac{1}{6}}(\tfrac{1}{4\pi}\tfrac{|a|h}{\nu})^{\frac{1}{3}}
\end{aligned}$$

5 Numerical Results

The global domain Ω is decomposed in N non-overlapping subdomains. The interface problem (2.4) is solved by a BICG-STAB algorithm. This involves solving, at each iteration, N independent subproblems (one per subdomain) which can be performed in parallel. Each subproblem is solved by a direct method. We denote by h the mesh size. The CFL number is $\frac{|a|\Delta t}{h}$. In most cases, it will be very large. We compare the results obtained with the OO2 interface conditions, the Taylor order 0 or 2 interface conditions and Dirichlet interface conditions (overlapping subdomains).

Remark 5.1 : *The optimized coefficients in the OO2 method are computed in an initialisation step, that is in the computation of the local matrix. They are not computed again in the iterations of BICG-STAB. Moreover, each iteration of BICG-STAB has the same cost for all the interface conditions (Taylor order 0, Taylor order 2, OO2), because the use of order 2 conditions does not increase the bandwidth of the local matrix. So, in the BICG-STAB algorithm, the CPU time is proportional to the number of iterations.*

The stopping criterion was the maximum error between the converged solution and the iterative solution to be smaller than 10^{-6}.

5.1 Flow in a square

We solve the following problem :

$$\begin{aligned} \mathcal{L}(u) &= 0, \quad 0 \le x \le 1,\ 0 \le y \le 1 \\ u(0,y)=0,\ \frac{\partial u}{\partial x}(1,y) &= 0, \quad 0 \le y \le 1 \\ u(x,0)=1,\ \frac{\partial u}{\partial y}(x,1) &= 0, \quad 0 \le x \le 1 \end{aligned} \tag{5.1}$$

where $\mathcal{L}$ is as in (4.2). We consider a cartesian mesh with constant mesh size h. The unit square is decomposed in $N_x \times N_y$ subdomains, where N_x (resp. N_y) is the number of subdomains in the x (resp. y) direction. We consider two types of convection velocity field : a shear velocity ($a_x = y,\ a_y = 0$) and a rotating velocity ($a_x = -\sin(\pi(y-\frac{1}{2}))\cos(\pi(x-\frac{1}{2})),\ a_y = \cos(\pi(y-\frac{1}{2}))\sin(\pi(x-\frac{1}{2}))$). The isovalues of the solution of problem (5.1) with the shear velocity are represented in figure 4, and with the rotating velocity in figure 5.

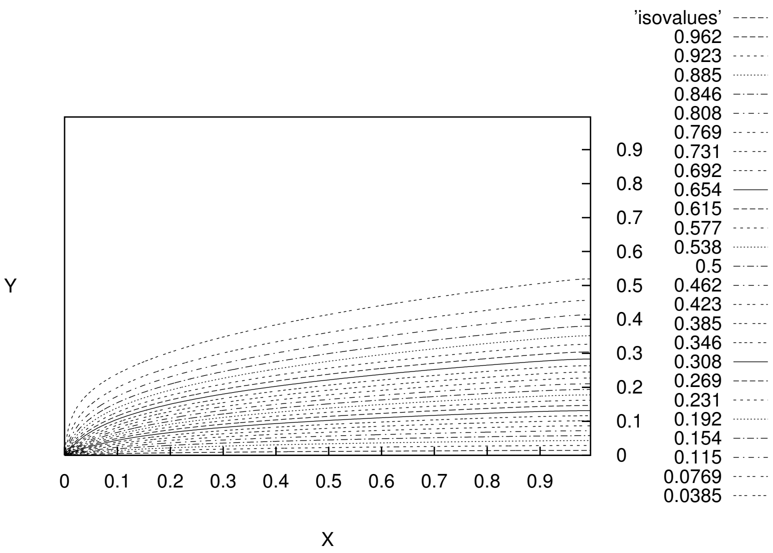

Fig. 4. Isovalues of the solution u, shear velocity
$a_x = y,\ a_y = 0,\ \nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241}$

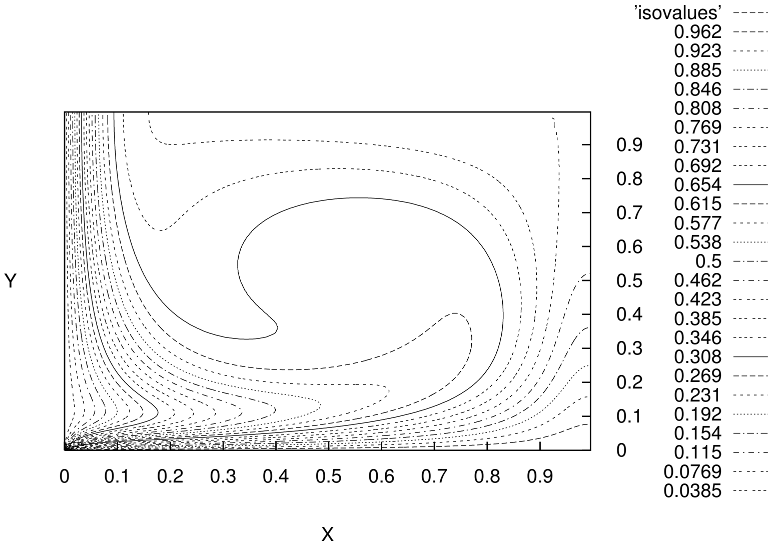

Fig. 5. Isovalues of the solution u, rotating velocity
$a_x = -\sin(\pi(y-\frac{1}{2}))\cos(\pi(x-\frac{1}{2})),\ a_y = \cos(\pi(y-\frac{1}{2}))\sin(\pi(x-\frac{1}{2}))$
$\nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241}$

We first consider an upwind finite difference scheme with a small overlap of size h. The time step is taken very large ($\Delta t = 10^9$) so that it corresponds to a stationary equation. For a decomposition into 16 × 1 subdomains, we give in Table 1 the iteration count for two velocity fields: one normal to the interface and one tangential to the interface.

Nb of iterations	**Dirichlet**	**Taylor 0**	**Taylor 2**	**002**
$a_y = 0$, $a_x = y$	60	137	15	15
$a_x = 0$, $a_y = x$	60	90	265	9

Table 1: Overlapping subdomains ($\delta = h$) - $\nu = 10^{-2}$, $CFL = 1.d9$, $h = 1/241$

We now consider a finite volume discretization with no overlap between the subdomains ($\delta = 0$). The Dirichlet interface conditions cannot be used anymore. In Table 2, we take a decomposition in strips in order to observe the influence on the convergence of the convection velocity angle to the interfaces. We observe that the OO2 interface conditions give a significantly better convergence which is independent of the convection velocity angle to the interfaces. One of the advantages is that for a given number of subdomains, the decomposition of the domain doesn't affect the convergence. Particularly here, for 16 subdomains, the decomposition in strips (table 2) or in squares (figure 6) doesn't affect the convergence.

Decomposition of the domain	OO2	Taylor order 2	Taylor order 0
normal velocity to the interface **16** × **1** subdomains	15	123	141
tangential velocity to the interface **1** × **16** subdomains	21	not convergent	86

Table 2: Number of iterations versus the convection velocity's angle
$a_x = y,\ a_y = 0,\ \nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241},\ \log_{10}(Error) < 1.d-6$

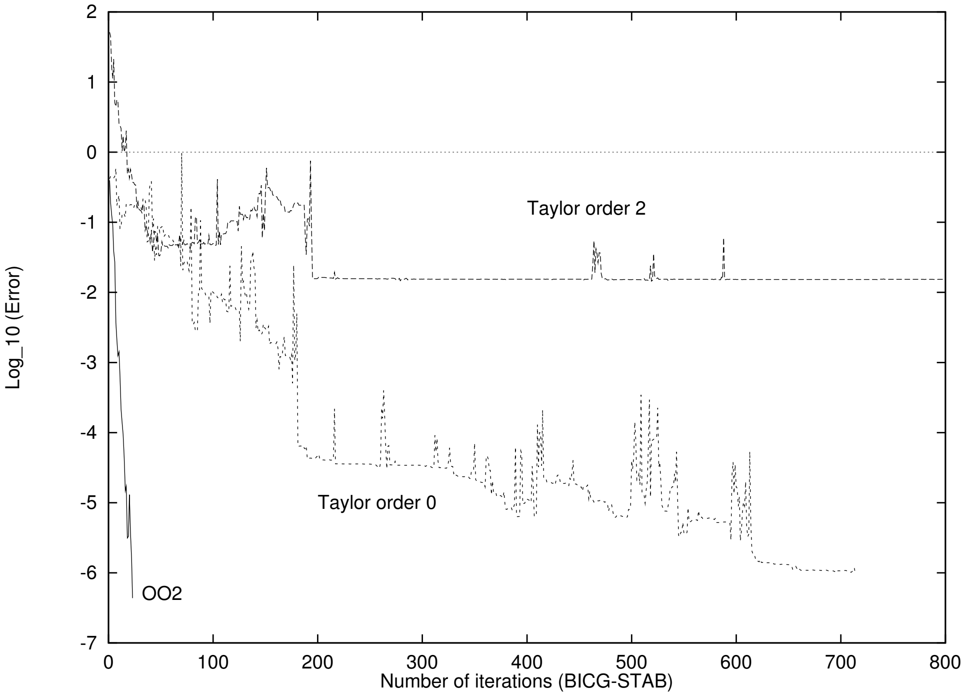

Fig. 6. Error versus the number of iterations

4 × **4** subdomains, shear velocity
$a_x = y,\ a_y = 0,\ \nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241}$

Figure 7 shows that the convergence with the OO2 interface conditions is significantly better for a more general convection velocity (the rotating velocity) and decomposition (in 4 × 8 subdomains).

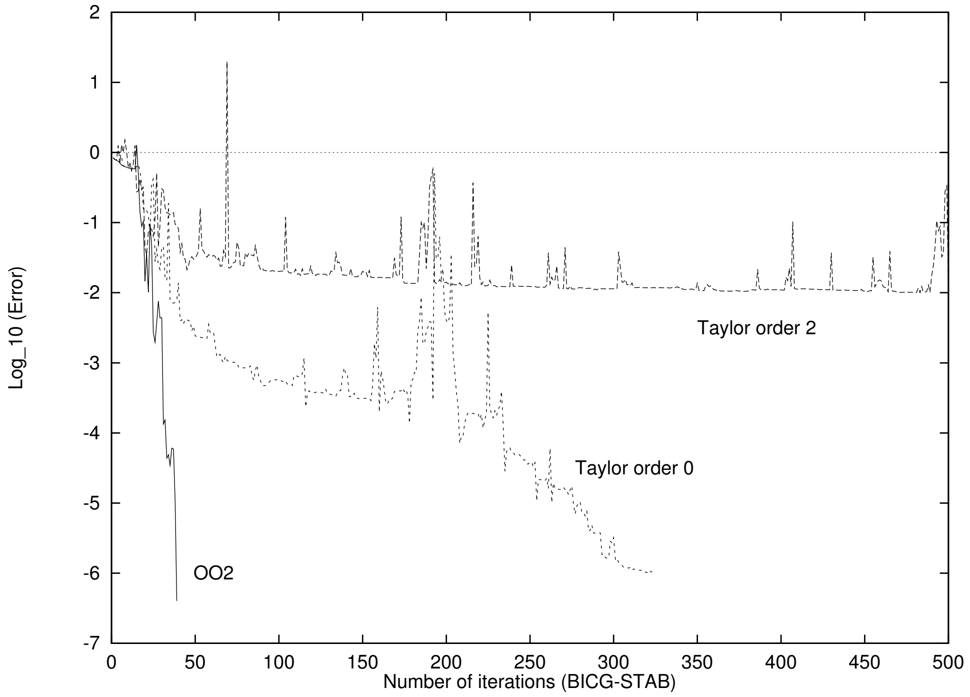

Fig. 7. Error versus the number of iterations
4 × **8** subdomains, rotating velocity
$a_x = -\sin(\pi(y-\frac{1}{2}))\cos(\pi(x-\frac{1}{2})),\ a_y = \cos(\pi(y-\frac{1}{2}))\sin(\pi(x-\frac{1}{2}))$
$\nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241}$

The convergence with the OO2 interface conditions, for the studied numerical cases, is also nearly independent of the mesh size (see table 3). We practically fit to the theoretical estimates of theorem 4.2.

grid	65×65	129×129	241×241
OO2	25	26	30
Taylor order 0	76	130	224

Table 3: Number of iterations versus the mesh size
4 × **4** subdomains, rotating velocity
$a_x = -\sin(\pi(y-\frac{1}{2}))\cos(\pi(x-\frac{1}{2})),\ a_y = \cos(\pi(y-\frac{1}{2}))\sin(\pi(x-\frac{1}{2}))$
$\nu = 1.d-2,\ CFL = 1.d9,\ \log_{10}(Error) < 1.d-6$

The convergence with the OO2 interface conditions is also very little sensible to the variations of the CFL, as it shown on table 4.

	$CFL = 1.d0$	$CFL = 1.d3$	$CFL = 1.d5$	$CFL = 1.d9$
OO2	3	12	15	15
Taylor ordre 2	2	21	58	123
Taylor ordre 0	3	18	48	141

Table 4: Number of iterations versus the CFL, $\mathbf{16} \times \mathbf{1}$ subdomains
$a_x = y,\ a_y = 0,\ \nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241},\ \log_{10}(Error) < 1.d-6$

Figure 8 shows the *speed-up* $= \frac{CPU\ time\ (1\ domain)}{CPU\ time\ (N\ subdomains)}$ of the method. Let i_{max} (resp. j_{max}) be the number of grid points in the x (resp. y) direction, for the global domain. We note N_{it} the number of BICG-STAB iterations. For a decomposition of the domain in $N_x \times N_y$ subdomains, the total cost can be estimated by : $\alpha_1 (\frac{i_{max}}{N_x})^3 \frac{j_{max}}{N_y} + \alpha_2 N_{it} (\frac{i_{max}}{N_x})^2 \frac{j_{max}}{N_y}$, where α_1 and α_2 are constants. Figure 8 shows that for a small number of subdomains, the first term (arising from the LU factorization of the local matrix) is predominant. Then, the second term (arising from the BICG-STAB algorithm) become predominant. After 32 subdomains, the estimate is no more valid, because of the communication costs which can not be neglected.

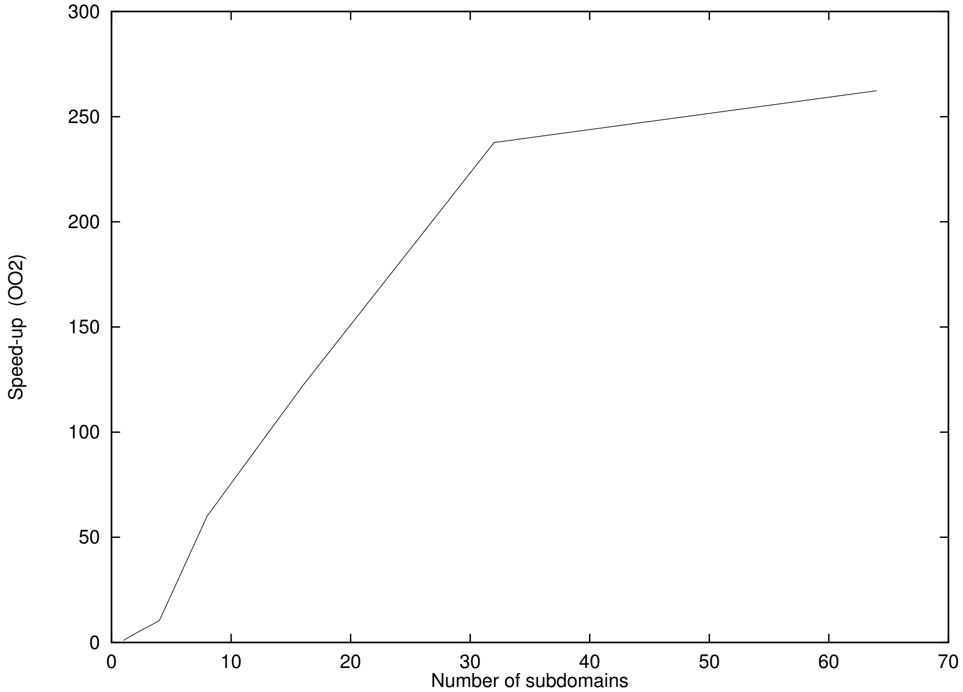

Fig. 8. Speed-up versus the number of subdomains, for decompositions in 2×1, 2×2, 4×2, 4×4, 8×4 and 8×8 subdomains
shear velocity
$a_x = y,\ a_y = 0,\ \nu = 1.d-2,\ CFL = 1.d9,\ h = \frac{1}{241},\ Residual < 1.d-9$

5.2 Flow around a cylinder, issued from a Navier-Stokes computation

The convection velocity field is a Navier-Stokes incompressible flow, with Reynolds number $Re = 10000$, around a cylinder. This velocity field is from a computation performed with the AEROLOG software of the aerodynamic department at Matra BAe Dynamics France. The domain is defined by $\Omega = \{(x,y) = (r\cos(\theta), r\sin(\theta)),\ 1 \leq r \leq R,\ 0 \leq \theta \leq 2\pi\}$ with $R > 1$ given. We solve the following problem :

$$\begin{aligned} L(u) &= 0 \quad \text{in } \Omega \\ u &= 1 \quad \text{on } \{(x,y) = (\cos(\theta), \sin(\theta)),\ 0 \leq \theta \leq 2\pi\} \\ u &= 0 \quad \text{on } \{(x,y) = (R\cos(\theta), R\sin(\theta)),\ 0 \leq \theta \leq 2\pi\} \end{aligned} \tag{5.2}$$

The grid is $\{(x_i, y_j) = (r_i \cos(\theta_j), r_i \sin(\theta_j)),\ 1 \leq\ i,j\ \leq\ 65\}$, and is refined around the cylinder and in the direction of the flow (see figure 9). The isovalues of the solution of problem (5.2) are represented in figure 10 (without the grid) and in figure 11 (with the grid). We note N_{max} the number of points on the boundary of a subdomain multiply by the number of subdomains.

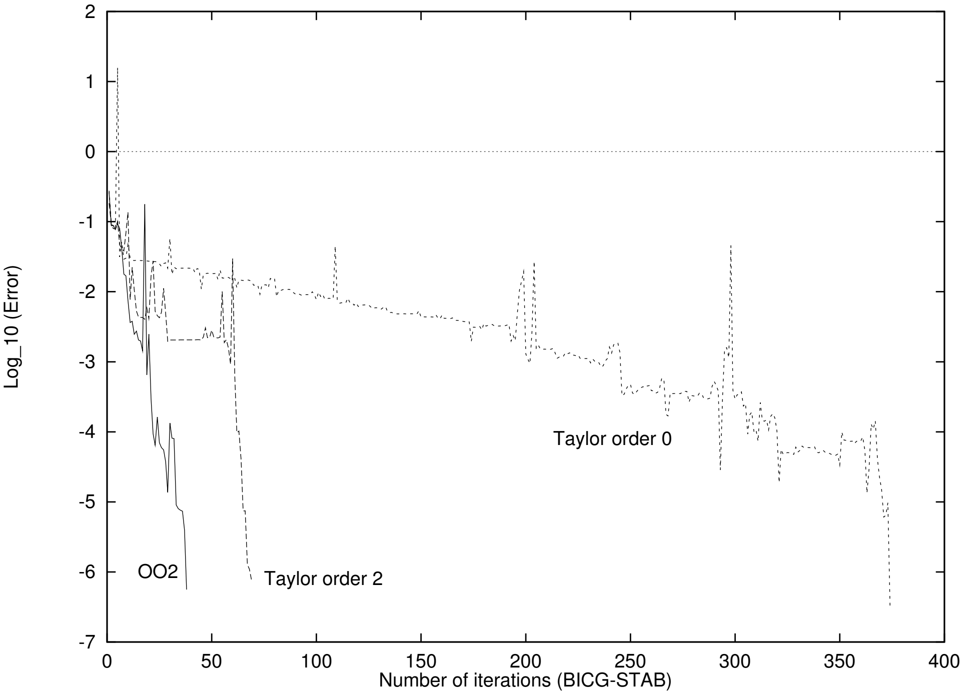

Fig. 9. Error versus the number of iterations

$\mathbf{4 \times 2}$ subdomains, Navier-Stokes flow velocity, $\nu = 1.d-4,\ CFL = 1.d9$

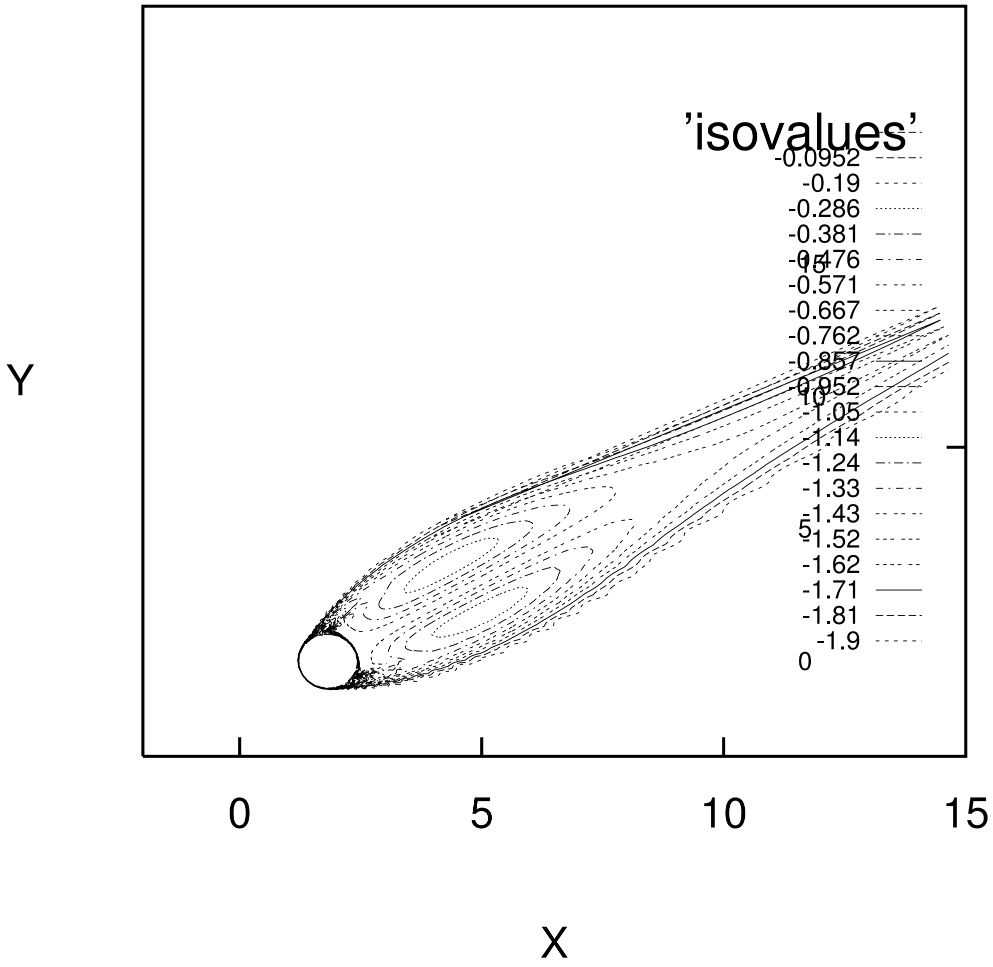

Fig. 10. Isovalues of the solution u
Navier-Stokes flow velocity, $\nu = 1.d - 4,\ CFL = 1.d9$

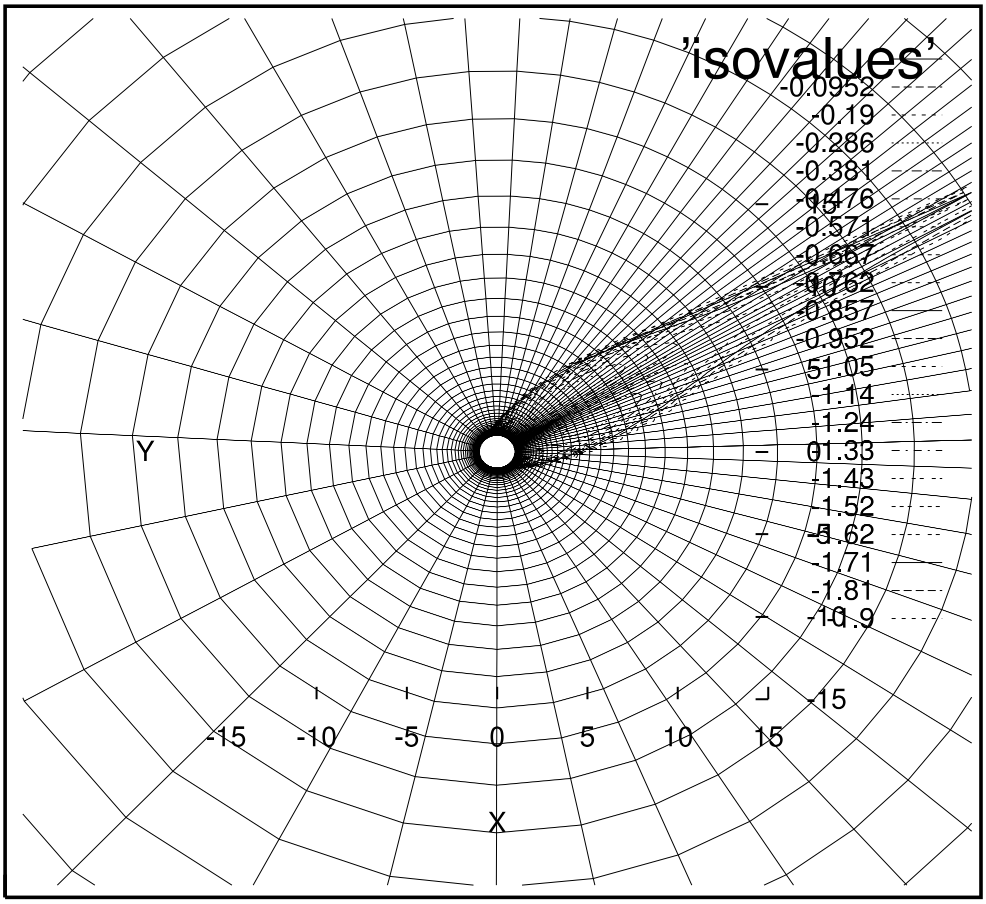

Fig. 11. Grid and isovalues of the solution u
Navier-Stokes flow velocity, $\nu = 1.d-4,\ CFL = 1.d9$

The OO2 interface conditions give also significantly better convergence in that case (figure 9). We observe in table 5 that the convergence is practically independent of the viscosity ν.

	OO2	Taylor order 2	Taylor order 0
$\nu = 1.d-5$	56	41	119
$\nu = 1.d-4$	43	121	374
$\nu = 1.d-3$	32	$N_{max} = 768$ $log_{10}(Error) = -5.52$	$N_{max} = 768$ $log_{10}(Error) = -2.44$

Table 5: Number of iterations versus the viscosity
4 × **2** subdomains, Navier-Stokes flow velocity
$\nu = 1.d-4,\ CFL = 1.d9,\ \log_{10}(Error) < 1.d-6$

Remark 5.2 *Numerically, the convergence ratio of the method is nearly linear upon the number of subdomains. So it is necessarily to send global information between subdomains, in order to have a convergence ratio independent of the number* *of subdomains. To tackle this problem, in [JNR98], a "low wave number" preconditioner is applied to the OO2 method.*

6 Conclusion

We have presented the 002 method for the convection-diffusion equation. Let us mention that the OO2 approach is versatile and has been applied to various type of equations and physical situations ([CN98] [WFNS98]). An interesting perspective is the extension of this approach to systems of equations.

References

[CN98] Philippe Chevalier and Frédéric Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.

[EM77] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31(139):629–651, 1977.

[Hal82] L. Halpern. Absorbing boundary conditions for the discretization schemes of the one-dimensional wave equation. *Mathematics of Computation*, 38(158):415–429, April 1982.

[Jap97] C. Japhet. Conditions aux limites artificielles et décomposition de domaine: Méthode oo2 (optimisé d'ordre 2). application à la résolution de problèmes en mécanique des fluides. Technical Report 373, CMAP (Ecole Polytechnique), 1997.

[JNR98] Caroline Japhet, Frédéric Nataf, and François-Xavier Roux. Extension of a coarse grid preconditioner to non-symmetric problems. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 279–286. Amer. Math. Soc., Providence, RI, 1998.

[Lio90] Pierre Louis Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.

[Nie98] F. Nier. Remarques sur les algorithmes de décomposition de domaine. In *Séminaire Equations aux dérivées partielles du Centre de Mathématiques de l'Ecole Polytechnique*, 1998.

[NN97] F. Nataf and F. Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numer. Math.*, 75(3):357–377, 1997.

[NRdS95] F. Nataf, F. Rogier, and E. de Sturler. Domain decomposition methods for fluid dynamics, Navier-Stokes equations and related nonlinear analysis. *Edited by A. Sequeira, Plenum Press Corporation*, pages 367–376, 1995.

[WFNS98] F Willien, I Faille, F Nataf, and F Schneider. Domain decomposition methods for fluid flow in porous medium. In *6th European Conference on the Mathematics of Oil Recovery*, sep 1998.