Optimal Interface Conditions
for Domain Decomposition Methods

F. Nataf*, F. Rogier** and E. de Sturler***

**CMAP, CNRS URA756, Ecole Polytechnique*
91128 Palaiseau Cedex, France
E-mail address: nataf@cmapx.polytechnique.fr

***Division Calcul parallèle, ONERA, 29 Av. de la Division Leclerc*
92322 Châtillon, France
E-mail address: rogier@onera.fr

****CSCS, ETH, Via Cantonale*
6928 Manno, Switzerland
E-mail address: sturler@cscs.ch

Abstract

We define optimal interface conditions for the additive Schwarz method (ASM) in the sense that convergence is achieved in a number of steps equals to the number of subdomains. Since these boundary conditions are difficult to use, we approximate them by partial differential operators that are easier to use. We present numerical results using these approximate interface conditions for the ASM and Schur type methods (substructuring). We also give a new result of convergence for BiCG which is then used for BiCGSTAB.

1 Introduction

The rate of convergence of Schwarz or Schur (substructuring) type algorithms is very sensitive to the choice of the interface conditions. The original Schwarz method is based on the use of Dirichlet boundary conditions. In order to increase the efficiency of the algorithm, it has been proposed to replace the Dirichlet boundary conditions with more general boundary conditions, see [14]. In the usual Schur method, Dirichlet and Neumann boundary conditions are used. In [11], they are replaced with artificial boundary conditions. More generally, it has been remarked that absorbing (or artificial) boundary conditions are a good choice (see, [11], [1], [17], [8] where such boundary conditions are used). In this report, we try to clarify the question of the interface conditions.

In § 2, we specify the optimal interface conditions for the Schwarz method applied to a domain decomposed into strips. As an example we discuss the Helmholtz equation in some detail. Then, we show that these interface conditions are also very efficient for Schur type algorithms. In § 3, we explain very briefly how to approximate these optimal interface conditions by absorbing (artificial) boundary conditions which are partial differential operators (for more details see [18], [19]). We are then no more restricted to a domain decomposed into strips. In § 4, we show some numerical results for the convection-diffusion equation.

2 Optimal interface conditions

Remark 2.1 *This section is formal. For instance we do not give any functional framework. Any space of functions from a set* Γ *to* $\mathbf{R}$ *will be denoted by* $L(\Gamma)$*. It is implicitly assumed that all the boundary value problems (BVP) are well posed. In this section, we give, formally, interface conditions for an arbitrary second order elliptic partial differential operator* $\mathcal{L}$*, such that the Schwarz algorithm converges in a minimum number of steps. A Schur type method based on the same interface conditions (see § 2.2) will also converge in the minimum number of steps. As we shall see, these interface conditions are difficult to use and will, therefore, be approximated (see § 3). This is the reason why we keep this section formal.*

2.1 Optimal interface conditions for the Schwarz algorithm

The outline of this section is the following. We first define the problem to be solved and the decomposition of the domain into vertical strips. After that, we define the interface conditions used in the Schwarz algorithm, we then prove its convergence in a number of steps equal to the number of subdomains. Finally, we discuss the optimality of the interface conditions.

Let Ω be a connected open subset of $\mathbf{R}^2$. Let $\mathcal{L}$ be a second order partial differential operator and $\mathcal{C}$ be a partial differential operator. We want to solve:

$$\begin{aligned} \mathcal{L}(u) &= f \text{ in } \Omega \\ \mathcal{C}(u) &= g \text{ on } \partial\Omega \end{aligned} \tag{1}$$

where f and g are given functions.

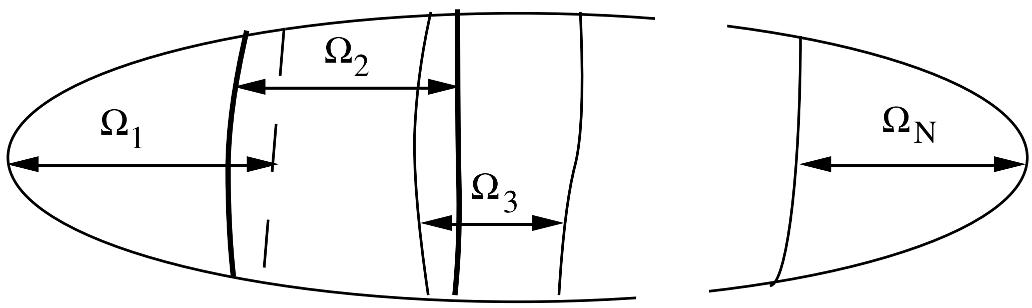

FIGURE 1 - Decomposition into vertical strips

The set Ω is decomposed into N vertical strips Ω_i, $1 \leq i \leq N$ ($\bar{\Omega} = \cup_{1 \leq i \leq N} \bar{\Omega}_i$) (see fig. 1).

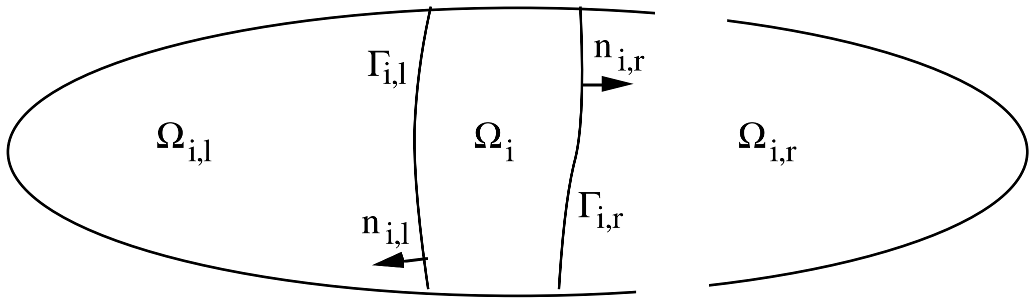

FIGURE 2 - Partition of $\Omega - \bar{\Omega}_i$

For each i, $\Omega - \bar{\Omega}_i$ is written as the disjoint union of two open subsets $\Omega_{i,l}$ and $\Omega_{i,r}$ where $\Omega_{i,l}$ is on the left of Ω_i and $\Omega_{i,r}$ on its right. $\partial\Omega_i - \partial\Omega$ is written as the disjoint union of $\Gamma_{i,l}$ and $\Gamma_{i,r}$ where $\Gamma_{i,l}$ is on the left of Ω_i and $\Gamma_{i,r}$ is on its right ($\Omega_{1,l} = \emptyset$ and $\Omega_{N,r} = \emptyset$) (see fig. 2). The outward normal from Ω_i on $\Gamma_{i,l}$ (resp. $\Gamma_{i,r}$) is denoted by $\vec{n}_{i,l}$ (resp. $\vec{n}_{i,r}$).

In order to define the interface conditions, we introduce the following Steklov-Poincaré operators.

Definition 2.2 *For each $2 \leq i \leq N$, let $\Lambda_{i,l} : \; L(\Gamma_{i,l}) \longrightarrow L(\Gamma_{i,l})$ be such that $\Lambda_{i,l}(v_0) = \frac{\partial v}{\partial \vec{n}_{i,l}}$ where v solves the following BVP:*

$$\begin{aligned} \mathcal{L}(v) &= 0 \;\; in \; \Omega_{i,l} \\ v &= v_0 \;\; on \; \Gamma_{i,l} \\ \mathcal{C}(v) &= 0 \;\; on \; \partial\Omega \cap \partial\Omega_{i,l} \end{aligned}$$

Similarly, for each $1 \leq i \leq N-1$, let $\Lambda_{i,r} : \; L(\Gamma_{i,r}) \longrightarrow L(\Gamma_{i,r})$ be such that $\Lambda_{i,r}(v_0) = \frac{\partial v}{\partial \vec{n}_{i,r}}$ where v solves the following BVP:

$$\begin{aligned} \mathcal{L}(v) &= 0 \;\; in \; \Omega_{i,r} \\ v &= v_0 \;\; on \; \Gamma_{i,r} \\ \mathcal{C}(v) &= 0 \;\; on \; \partial\Omega \cap \partial\Omega_{i,r} \end{aligned}$$

The operators

$$\frac{\partial}{\partial \vec{n}_{i,r \text{ or } l}} - \Lambda_{i,r \text{ or } l} \tag{2}$$

are used as transmission conditions for the Schwarz algorithm.

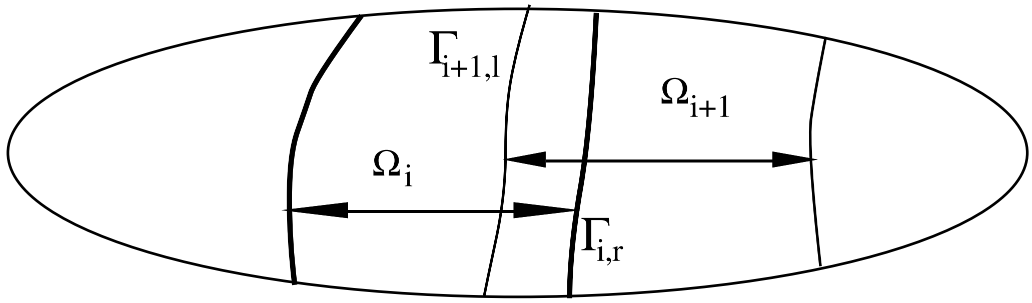

FIGURE 3 - Ω_i

For example, the operator $\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r}$ will be applied to two kinds of functions

1) functions from Ω_i to $\mathbf{R}$.
2) functions from Ω_{i+1} to $\mathbf{R}$.

In any case, the result is a function from $\Gamma_{i,r}$ to $\mathbf{R}$. We explain now how to apply this operator to a function v from Ω_i or Ω_{i+1} to $\mathbf{R}$. The computation of $\frac{\partial v}{\partial \vec{n}_{i,r}}$ is made on $\Gamma_{i,r}$. As for $\Lambda_{i,r}(v)$, one has to take the trace of v on $\Gamma_{i,r}$ and to apply definition 2.2 ($v_0 = v_{|\Gamma_{i,r}}$).

Remark 2.3 *If $\Omega = \mathbf{R}^2$, these operators are exact absorbing boundary conditions, also called artificial boundary conditions, radiation boundary conditions, open boundary conditions, outflow boundary conditions, ... (see e.g. [4], [5]).*

We now define the Schwarz algorithm. Let u_i^0 be an initial approximation to the solution u to (1) and let u_i^{n+1} be the value of the approximated solution to (1) satisfying:

$$
\begin{aligned}
\mathcal{L}(u_i^{n+1}) &= f \text{ in } \Omega_i \\
(\tfrac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u_i^{n+1}) &= (\tfrac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u_{i-1}^{n}) \text{ on } \Gamma_{i,l} \ (2 \le i \le N) \\
(\tfrac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u_i^{n+1}) &= (\tfrac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u_{i+1}^{n}) \text{ on } \Gamma_{i,r} \ (1 \le i \le N-1) \\
\mathcal{C}(u_i^{n+1}) &= g \text{ on } \partial\Omega \cap \partial\Omega_i
\end{aligned}
\tag{3}
$$

Proposition 2.4 *The Schwarz algorithm (3) achieves convergence in N iterations, where N is the number of subdomains.*

We give two proofs. The first one is direct. The second one is based on an interpretation of (3) as an algorithm for unknowns defined on the boundaries of the subdomains. It is an introduction to the Schur method.

first proof The equations are linear. In order to prove the convergence, we can consider the homogeneous case $f = 0$ and $g = 0$. We only have to prove the convergence to 0 of u_i^n. We shall use two propositions:

Proposition 2.5 *a) Let $u : \Omega_i \to \mathbf{R}$ $(2 \le i \le N-1)$ satisfy $\mathcal{L}(u) = 0$ in Ω_i, $\mathcal{C}(u) = 0$ on $\partial\Omega \cap \partial\Omega_{i,r}$ and $(\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u) = 0$ on $\Gamma_{i,r}$*

Then, $(\frac{\partial}{\partial \vec{n}_{i-1,r}} - \Lambda_{i-1,r})(u) = 0$ on $\Gamma_{i-1,r}$.

b) Let $u : \Omega_N \to \mathbf{R}$ *satisfy* $\mathcal{L}(u) = 0$ *in* Ω_N *and* $\mathcal{C}(u) = 0$ *on* $\partial\Omega \cap \partial\Omega_N$.
Then, $(\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(u) = 0$ *on* $\Gamma_{N-1,r}$.

and

Proposition 2.6 *a) Let* $u : \Omega_i \to \mathbf{R}$ $(2 \leq i \leq N-1)$ *satisfy* $\mathcal{L}(u) = 0$ *in* Ω_i, $\mathcal{C}(u) = 0$ *on* $\partial\Omega \cap \partial\Omega_i$ *and* $(\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u) = 0$ *on* $\Gamma_{i,l}$
Then, $(\frac{\partial}{\partial \vec{n}_{i+1,l}} - \Lambda_{i+1,l})(u) = 0$ *on* $\Gamma_{i+1,r}$.
b) Let $u : \Omega_1 \to \mathbf{R}$ *satisfy* $\mathcal{L}(u) = 0$ *in* Ω_1 *and* $\mathcal{C}(u) = 0$ *on* $\partial\Omega \cap \partial\Omega_1$.
Then, $(\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l})(u) = 0$ *on* $\Gamma_{2,l}$.

proof of proposition 2.5 Let u be as in proposition 2.5 a). We introduce the function $v : \Omega_{i-1,r} \to \mathbf{R}$ to be solution of

$$\begin{cases} \mathcal{L}(v) = 0 \text{ in } \Omega_{i-1,r} \\ \mathcal{C}(v) = 0 \text{ on } \partial\Omega \cap \partial\Omega_{i-1,r} \\ v = u \text{ on } \Gamma_{i-1,r} \end{cases}$$

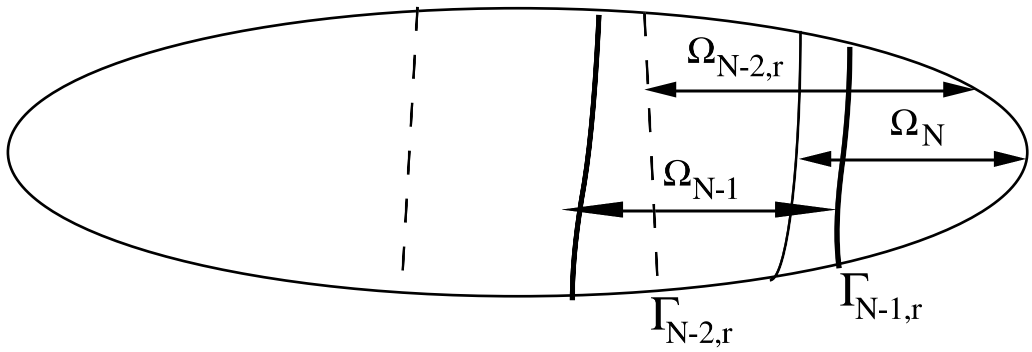

FIGURE 4

By definition of $\Lambda_{i-1,r}$, we have

$$(\frac{\partial}{\partial \vec{n}_{i-1,r}} - \Lambda_{i-1,r})(v) = 0 \text{ on } \Gamma_{i-1,r}. \tag{4}$$

We prove now that v and u coincide on $\Omega_{i-1} \cap \Omega_i$. Since $\Omega_{i,r} \subset \Omega_{i-1,r}$, we have

$$\begin{cases} \mathcal{L}(v) = 0 \text{ in } \Omega_{i,r} \\ \mathcal{C}(v) = 0 \text{ on } \partial\Omega \cap \partial\Omega_{i,r} \end{cases}$$

so that by definition of $\Lambda_{i,r}$ we have $(\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(v) = 0$ on $\Gamma_{i,r}$.
Thus, u and v solve the same BVP set on $\Omega_{i-1,r} \cap \Omega_i$ i.e.

$$\begin{cases} \mathcal{L}(v) = \mathcal{L}(u) = 0 \text{ in } \Omega_{i-1,r} \cap \Omega_i \\ \mathcal{C}(v) = \mathcal{C}(u) = 0 \text{ on } \partial\Omega \cap \partial(\Omega_{i-1,r} \cap \Omega_i) \\ (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(v) = (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u) = 0 \text{ on } \Gamma_{i,r} \\ v = u \text{ on } \Gamma_{i-1,r} \end{cases}$$

We have assumed that the different boundary value problems are well-posed and we have thus $v = u$. We have a similar proof for the second part of the lemma and for lemma 2.6.$\Box$

Thanks to these propositions, we have that

$$(\frac{\partial}{\partial \vec{n}_{N-j,r}} - \Lambda_{N-j,r})(u^n_{N-j+1}) = 0 \text{ on } \Gamma_{N-j,r} \text{ for all } 1 \leq j \leq N-1 \text{ and } n \geq j \tag{5}$$

Let us prove (5) for $j = 1$. From (3), u^n_N satisfies

$$\begin{cases} \mathcal{L}(u^n_N) = 0 \text{ in } \Omega_N \\ \mathcal{C}(u^n_N) = 0 \text{ on } \partial\Omega \cap \partial\Omega_N \end{cases}$$

so that by proposition 2.5 b), we have $(\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(u^n_N) = 0$ on $\Gamma_{N-1,r}$.
Let us prove now (5) for $j = 2$. From (3), u^n_{N-1} satisfies for $n \geq 2$

$$\begin{cases} \mathcal{L}(u^n_{N-1}) = 0 \text{ in } \Omega_{N-1} \\ \mathcal{C}(u^n_{N-1}) = 0 \text{ on } \partial\Omega \cap \partial\Omega_{N-1} \\ (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(u^n_{N-1}) = (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(u^{n-1}_N) = 0 \text{ on } \Gamma_{N-1,r} \text{ (cf. above)} \end{cases}$$

By proposition 2.5 a), we have that $(\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r})(u^n_{N-1}) = 0$ on $\Gamma_{N-2,r}$.
A similar proof can be constructed for $j = 3, \ldots, N-1$.

By using proposition 2.6 a similar proof can be made to prove that

$$(\frac{\partial}{\partial \vec{n}_{j+1,l}} - \Lambda_{j+1,l})(u^n_j) = 0 \text{ on } \Gamma_{N-j,l} \text{ for all } 1 \leq j \leq N-1 \text{ and } n \geq j \tag{6}$$

It is now easy to prove that $u^N_i = 0$ for every i. Indeed, from (3), (5) and (6) we see that the right hand sides of the BVP defining u^N_i are zero and thus $u^N_i = 0$ for every i.$\Box$

second proof of proposition (2.4) We drop the requirement that f and g be zero. Let $h^n_{i,r \text{ or } l} = (\frac{\partial}{\partial \vec{n}_{i,r \text{ or } l}} - \Lambda_{i,r \text{ or } l})(u^n_i)$ on $\Gamma_{i,r \text{ or } l}$ $(2 \leq i \leq N-1)$, $h^n_{1,r} = (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r})(u^n_1)$ on $\Gamma_{1,r}$ and $h^n_{N,l} = (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l})(u^n_N)$ on $\Gamma_{N,l}$. We will show that $h^N_{i,r \text{ or } l}$ is equal to

$$h_{i,r \text{ or } l} = (\frac{\partial}{\partial \vec{n}_{i,r \text{ or } l}} - \Lambda_{i,r \text{ or } l})(u_i) \tag{7}$$

This will prove the convergence of (3) in N steps. In order to specify the algorithm for the computation of $h^n_{i,r \text{ or } l}$, we introduce some operators. Let $S_i : L(\Gamma_{i,l}) \times L(\Gamma_{i,r}) \times L(\Omega_i) \times L(\partial\Omega \cap \partial\Omega_i) \longrightarrow L(\Omega_i)$ $(2 \leq i \leq N)$ be such that $S_i(h_l, h_r, f, g) = v$ where v solves the following BVP:

$$\begin{aligned} \mathcal{L}(v) = f \text{ in } \Omega_i \\ (\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(v) = h_l \text{ on } \Gamma_{i,l} \ (2 \leq i \leq N) \\ (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(v) = h_r \text{ on } \Gamma_{i,r} \ (1 \leq i \leq N-1) \\ \mathcal{C}(v) = g \text{ on } \partial\Omega \cap \partial\Omega_i \end{aligned} \tag{8}$$

The domains $\Omega_{1,l}$ and $\Omega_{N,r}$ are empty. We consider $S_1: L(\Gamma_{1,r}) \times L(\Omega_1) \times L(\partial\Omega \cap \partial\Omega_1) \longrightarrow L(\Omega_1)$ and $S_N: L(\Gamma_{N,l}) \times L(\Omega_N) \times L(\partial\Omega \cap \partial\Omega_N) \longrightarrow L(\Omega_N)$ as operators of only three arguments but defined in a similar way to (8).

From (3), we have for $n \geq 1$

$$\begin{array}{ll}
h_{2,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l})(S_1(h_{1,r}^n, 0, 0) + S_1(0, f, g)) \\
h_{3,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l})(S_2(h_{2,l}^n, 0, 0, 0) + S_2(0, h_{2,r}^n, 0, 0) + S_2(0, 0, f, g)) \\
 & \vdots \\
h_{N,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l})(S_{N-1}(h_{N-1,l}^n, 0, 0, 0) + S_{N-1}(0, h_{N-1,r}^n, 0, 0) + S_{N-1}(0, 0, f, g)) \\
h_{N-1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(S_N(h_{N,l}^n, 0, 0) + S_N(0, f, g)) \\
h_{N-2,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r})(S_{N-1}(h_{N-1,l}^n, 0, 0) + S_{N-1}(0, h_{N-1,r}^n, 0, 0) + S_{N-1}(0, 0, f, g)) \\
 & \vdots \\
h_{1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r})(S_2(h_{2,l}^n, 0, 0) + S_2(0, h_{2,r}^n, 0, 0) + S_2(0, 0, f, g))
\end{array} \quad (9)$$

Thus algorithm (9) can be interpreted as an algorithm to solve the following linear system in $h_{i,r \text{ or } l}$:

$$\begin{array}{ll}
h_{2,l} & = (\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l})(S_1(h_{1,r}, 0, 0) + S_1(0, f, g)) \\
h_{3,l} & = (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l})(S_2(h_{2,l}, 0, 0, 0) + S_2(0, h_{2,r}, 0, 0) + S_2(0, 0, f, g)) \\
 & \vdots \\
h_{N,l} & = (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l})(S_{N-1}(h_{N-1,l}, 0, 0, 0) + S_{N-1}(0, h_{N-1,r}, 0, 0) + S_{N-1}(0, 0, f, g)) \\
h_{N-1,r} & = (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(S_N(h_{N,l}, 0, 0) + S_N(0, f, g)) \\
h_{N-2,r} & = (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r})(S_{N-1}(h_{N-1,l}, 0, 0) + S_{N-1}(0, h_{N-1,r}, 0, 0) + S_{N-1}(0, 0, f, g)) \\
 & \vdots \\
h_{1,r} & = (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r})(S_2(h_{2,l}, 0, 0) + S_2(0, h_{2,r}, 0, 0) + S_2(0, 0, f, g))
\end{array} \quad (10)$$

We assume that this system has a unique solution. Due to the choice of the interface conditions (2) we have

Proposition 2.7 *For every $h_{r \text{ or } l}$, we have*

$$(\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l})S_1(h_r, 0, 0) = (\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})S_{i-1}(0, h_r, 0, 0) = 0 \ (3 \leq i \leq N) \quad (11)$$

and

$$(\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})S_N(h_l, 0, 0) = (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})S_{i+1}(h_l, 0, 0, 0) = 0 \ (1 \leq i \leq N-2) \quad (12)$$

proof Let us prove for instance that $(\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})S_{i-1}(0, h_r, 0, 0) = 0 \ (3 \leq i \leq N)$. Indeed, $v = S_{i-1}(0, h_r, 0, 0)$ satisfies

$$\left\{ \begin{array}{l}
\mathcal{L}(v) = 0 \text{ in } \Omega_{i-1,r} \\
\mathcal{C}(v) = 0 \text{ on } \partial\Omega \cap \partial\Omega_{i-1,r} \\
(\frac{\partial}{\partial \vec{n}_{i-1,l}} - \Lambda_{i-1,l})(v) = 0 \text{ on } \Gamma_{i-1,l}
\end{array} \right.$$

By proposition 2.6 a), $(\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(v) = 0$. The proofs of the other relations are similar.□

Summarizing, algorithm (9) can be written in a simpler form:

$$
\begin{array}{ll}
h_{2,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l}) S_1(0,f,g) \\
h_{3,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l}) (S_2(h_{2,l}^n,0,0,0) + S_2(0,0,f,g)) \\
 & \vdots \\
h_{N,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l}) (S_{N-1}(h_{N-1,l}^n,0,0,0) + S_{N-1}(0,0,f,g)) \\
h_{N-1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r}) S_N(0,f,g) \\
h_{N-2,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r}) (S_{N-1}(0,h_{N-1,r}^n,0,0) + S_{N-1}(0,0,f,g)) \\
 & \vdots \\
h_{1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r}) (S_2(0,h_{2,r}^n,0,0) + S_2(0,0,f,g))
\end{array}
\tag{13}
$$

The equations on the $h_{i,l}$ and on the $h_{i,r}$ are decoupled. From (13), it can be checked that $h_{2,l}^1 = h_{2,l}^{n+1}$, $n \geq 0$, $h_{N-1,r}^1 = h_{N-1,r}^{n+1}$, $n \geq 0$ and then that $h_{3,l}^2 = h_{3,l}^{n+2}$, $n \geq 0$, $h_{N-2,r}^2 = h_{N-2,r}^{n+2}$, $n \geq 0$ and so on ... After step N algorithm (13) will have converged. At step N, $h_{i,r \text{ or } l}^N$ and $h_{i,r \text{ or } l}$ satisfy the same linear system (10) and by the assumption of well-posedness of this system, we have thus $h_{i,r \text{ or } l}^N = h_{i,r \text{ or } l}$. Then, from (3), we have $u_i^N = u_{|\Omega_i}$ for all i (u is the solution to (1)). This ends the proof that algorithm (3) converges in N steps.□

We have just proved the convergence of the additive Schwarz method (algorithm (3)) in N steps where N is the number of subdomains. This result is optimal in the following sense. Take $\mathcal{L} = \Delta$, the solution in domain 1 depends on the value of the right hand side f in domain N and vice versa. Thus at least $N-1$ exchanges of information between contiguous subdomains are necessary to converge. In the additive Schwarz method, information is exchanged only between contiguous subdomains. Since the initial approximation u_i^0 to the solution in each subdomain does not depend on f and g, at least N steps are needed to converge.

Application. We consider the 1-D Helmholtz equation with one discontinuity. Let c_+ and c_- be two different positive real numbers. Using a domain decomposition method, we want to solve the following problem:

$$
\left\{
\begin{array}{l}
\omega^2 u_- + c_-^2 \Delta u_- = f \quad x < 0 \\
\omega^2 u_+ + c_+^2 \Delta u_+ = f \quad x > 0 \\
u_+ = u_- \text{ at } x = 0 \\
c_-^2 \frac{\partial u_-}{\partial x} = c_+^2 \frac{\partial u_+}{\partial x} \text{ at } x = 0 \\
\frac{\partial u_+}{\partial x} + i \frac{\omega}{c_+} u_+ = 0 \text{ at } x = \infty \\
-\frac{\partial u_-}{\partial x} + i \frac{\omega}{c_-} u_- = 0 \text{ at } x = -\infty
\end{array}
\right.
$$

where f is a given function and $i^2 = -1$. Let $]l, L[\subset \mathbf{R}^-$ be a subdomain. We first write the interface condition to be used at l. We have to introduce:

$$
\begin{array}{rccc}
\Lambda_l : & \mathbf{C} & \longrightarrow & \mathbf{C} \\
 & u_0 & \longrightarrow & -\frac{\partial v}{\partial x}(l)
\end{array}
$$

where v satisfies

$$\begin{array}{l}\omega^2 v + c_-^2 \Delta v = 0 \quad x < l \\ v(l) = u_0 \\ -\frac{\partial v}{\partial x} + i\frac{\omega}{c_-} v = 0 \text{ at } x = -\infty\end{array}$$

It may easily be seen that $\Lambda_l(u_0) = -i\frac{\omega}{c_-}u_0$. The interface condition at $x = l$ is therefore given by: $-\frac{\partial}{\partial x} + i\frac{\omega}{c_-}$.

We now consider the interface condition at $x = L$. We introduce:

$$\begin{array}{rccc}\Lambda_r: & \mathbf{C} & \longrightarrow & \mathbf{C} \\ & u_0 & \longrightarrow & \frac{\partial v}{\partial x}(L)\end{array}$$

where v satisfies

$$\begin{array}{l}\omega^2 v + c_-^2 \Delta v = 0 \quad L < x < 0 \\ \omega^2 v + c_+^2 \Delta v = 0 \quad x > 0 \\ v(0^-) = v(0^+) \\ c_-^2 \frac{\partial v}{\partial x}(0^-) = c_+^2 \frac{\partial v}{\partial x}(0^+) \\ v(L) = u_0 \\ \frac{\partial v}{\partial x} + i\frac{\omega}{c_+} v = 0 \text{ at } x = \infty\end{array}$$

It may easily be seen that

$$v = \alpha e^{i\frac{\omega}{c_-}x} + \beta e^{-i\frac{\omega}{c_-}x} \quad x < 0$$

and

$$v = \gamma e^{-i\frac{\omega}{c_+}x} \quad x > 0$$

where α, β and γ are deduced from the boundary conditions at $x = 0$ and $x = L$. A straight-forward computation gives:

$$\Lambda_r(u_0) = \begin{cases} -i\frac{\omega}{c_-}u_0 \left(\dfrac{1 - \frac{c_- - c_+}{c_- + c_+} e^{i\frac{2\omega}{c_-}L}}{1 + \frac{c_- - c_+}{c_- + c_+} e^{i\frac{2\omega}{c_-}L}} \right), \text{ for } L \neq 0 \\ -i\frac{\omega}{c_-}u_0, \text{ for } L = 0 \end{cases}$$

For $L < 0$, the optimal interface condition is:

$$\frac{\partial}{\partial x} + i\frac{\omega}{c_-} \left(\frac{1 - \frac{c_- - c_+}{c_- + c_+} e^{i\frac{2\omega}{c_-}L}}{1 + \frac{c_- - c_+}{c_- + c_+} e^{i\frac{2\omega}{c_-}L}} \right)$$

At $x = 0$, $\frac{\partial u}{\partial x}$ is discontinuous, contrarily to the implicit assumption made in § 2.1. Thus, if $L = 0$, a small adaptation is needed. We specify directly the optimal additive Schwarz method for a decomposition of $\mathbf{R}$ into $\mathbf{R}^+$ and $\mathbf{R}^-$. Let u_-^n and u_+^n be the approximations to u_- and u_+ at step n, u_-^{n+1} and u_+^{n+1} satisfy

$$\begin{cases} \omega^2 u_{r \text{ or } l}^{n+1} + c_\pm^2 \Delta u_{r \text{ or } l}^{n+1} = f, \quad x \in \mathbf{R}^\pm \\ c_-^2 \frac{\partial u_-^{n+1}}{\partial x} + i\omega c_+ u_-^{n+1} = c_+^2 \frac{\partial u_+^n}{\partial x} + i\omega c_+ u_+^n \text{ at } x = 0 \\ -c_+^2 \frac{\partial u_+^{n+1}}{\partial x} + i\omega c_- u_+^{n+1} = -c_-^2 \frac{\partial u_-^n}{\partial x} + i\omega c_- u_-^n \text{ at } x = 0 \\ \frac{\partial u_+^{n+1}}{\partial x} + i\frac{\omega}{c_+} u_+^{n+1} = 0 \text{ at } x = \infty \\ -\frac{\partial u_-^{n+1}}{\partial x} + i\frac{\omega}{c_-} u_-^{n+1} = 0 \text{ at } x = -\infty \end{cases}$$

It may easily be checked that we have convergence in two steps.

Open question. We have considered a decomposition of the domain into vertical strips. We have seen that there exist interface conditions which lead to optimal convergence results for the additive Schwarz method. If we consider a decomposition into concentric rings, it may easily be seen that there exist also interface conditions which lead to convergence in a finite number of steps for the additive Schwarz method. A natural question is what happens when the geometry is more complex, e.g. the domain is decomposed into polygons. We guess that for an elliptic operator there are no interface conditions such that the additive Schwarz method converges in a finite number of steps in the case of a general domain decomposition, e.g. a square decomposed into four squares. As far as we know this is still an open question. We note that D. Gottlieb in [10] proposed for the Laplacian on a square divided into four squares a domain decomposition method which converges in a finite number of steps. This result is not in contradiction with our guess since his algorithm is not a Schwarz method.

2.2 A Schur type algorithm

In § 2.1, we defined the system of equations (10) whose unknowns are functions from the boundaries of the subdomains to $\mathbf{R}$. We thus obtain a substructuring formulation which may be solved by a conjugate gradient like method. We refer to the resulting system as a Schur-type algorithm. More precisely, we introduce the following notation in order to write the system of equations (10) in a compact form. Let $\Gamma = \cup_i \Gamma_{i,r \text{ or } l}$ be the set of the interfaces and $L(\Gamma) = L(\Gamma_{2,l}) \times \ldots L(\Gamma_{N,l}) \times L(\Gamma_{1,r}) \ldots L(\Gamma_{N-1,r})$. An element H of $L(\Gamma)$ is denoted by a $2(N-1)$-tuple $(h_{2,l}, \ldots, h_{N,l}, h_{1,r}, \ldots, h_{N-1,r})$. Let $\mathcal{T}$ be a map from $L(\Gamma)$ to itself defined by:

$$\mathcal{T}(h_{2,l}, \ldots, h_{N,l}, h_{1,r}, \ldots, h_{N-1,r}) = \begin{bmatrix} 0 \\ (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l}) S_2(h_{2,l}, 0, 0, 0) \\ \vdots \\ (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l}) S_{N-1}(h_{N-1,l}, 0, 0, 0) \\ (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r}) S_2(0, h_{2,r}, 0, 0) \\ \vdots \\ (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r}) S_{N-1}(0, h_{N-1,r}, 0, 0) \\ 0 \end{bmatrix}$$

and $G \in L(\Gamma)$ be defined by:

$$G = \begin{bmatrix} (\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l}) S_1(0, f, g) \\ (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l}) S_2(0, 0, f, g) \\ \vdots \\ (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l}) S_{N-1}(0, 0, f, g) \\ (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r}) S_2(0, 0, f, g) \\ \vdots \\ (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r}) S_{N-1}(0, 0, f, g) \\ (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r}) S_N(0, f, g) \end{bmatrix}$$

Taking into account (11) and (12), system (10) can be written in the form:

$$(Id_{L(\Gamma)} - \mathcal{T})(H) = G \tag{14}$$

Equation (14) defines what we refer to as the Schur-type (or substructuring) formulation of problem (1). Before considering conjugate gradient like methods in order to solve (14), we make a remark concerning the additive Schwarz method. From (13), we see that the additive Schwarz method corresponds to the solution of (14) by a Jacobi algorithm:

$$H^{n+1} = \mathcal{T}(H^n) + G \tag{15}$$

The operator $\mathcal{T}$ can be written in the form of an operator valued matrix

$$\mathcal{T}(H) = \left[\begin{array}{cccccccc}
0 & & & 0 & & & & \\
\times & \ddots & & & & \mathbf{0} & & \\
 & \ddots & \ddots & & & & & \\
0 & & \times & 0 & & & & \\
 & & & & 0 & \times & & 0 \\
 & \mathbf{0} & & & & \ddots & \ddots & \\
 & & & & & & \ddots & \times \\
 & & & & 0 & & & 0
\end{array}\right]
\left[\begin{array}{c} h_{2,l} \\ \vdots \\ \vdots \\ h_{N,l} \\ h_{1,r} \\ \vdots \\ \vdots \\ h_{N-1,r} \end{array}\right]$$

The crosses correspond to non zero operators. From the structure of $\mathcal{T}$, it is clear that $\mathcal{T}^{N-1} = 0$. Therefore, we have

$$H = G + \mathcal{T}G + \mathcal{T}^2 G + \ldots + \mathcal{T}^{N-2} G \tag{16}$$

and algorithm (15) converges in $N-1$ steps in the general case. But, if $H^0 = G$, only $N-2$ steps are needed. This is not in contradiction with § 2.1. Indeed, the computation of $\mathcal{T}$ applied to some vector implies exchanges of information between contiguous subdomains. The computation of G counts for 1 exchange, the $N-2$ iterations of (15) count for $N-2$ exchanges. This corresponds to $N-1$ exchanges as for the additive Schwarz method. From another point of view, in agreement with proposition 2.4, the BVP needs to be solved N times in each subdomains to compute the solution u of (1). The computation of G counts for one solution per subdomain, $N-2$ iterations of (15) count for $N-2$ solutions per subdomain, the computation of u from H counts for one solution per subdomain.

Let us consider now two conjugate like methods: GMRES and Bi-CGSTAB. Let H^0 be the initial approximation to the solution to (14). Let $r_0 = G - (Id - \mathcal{T})(H^0)$ be the initial residual. We seek for $\tilde{H}$ such that $H = H^0 + \tilde{H}$ i.e. $\tilde{H}$ satisfies:

$$(Id - \mathcal{T})(\tilde{H}) = r_0$$

The GMRES method minimizes the residual norm over the Krylov space $K^n(Id - \mathcal{T}), r_0) \equiv span\{r_0, (Id - \mathcal{T})r_0, \ldots, (Id - \mathcal{T})^{n-1} r_0\}$. Clearly, $\tilde{H} \in K^{N-1}(Id - \mathcal{T}), r_0)$ so that $N-1$ iterations are necessary for the solution of (14). Thus, we have just proved

Proposition 2.8 *The GMRES algorithm applied to (14) converges in at most $N-1$ steps.*

Let us now consider the convergence of Bi-CGSTAB [21] for the solution of the linear system (14). We shall see that

Proposition 2.9 *If there is no breakdown of Bi-CGSTAB, we have convergence of Bi-CGSTAB applied to (14) in at most $N-1$ steps.*

Because Bi-CGSTAB is based on BiCG [7] we will first discuss the convergence of BiCG. We choose some $\tilde{r}_0 \neq 0$, (for example $\tilde{r}_0 = r_0$). Now the BiCG algorithm generates two sequences of polynomials, the residuals $r_i = P_i(Id - \mathcal{T})r_0$:

$$r_0, r_1, r_2, \ldots$$

and $\tilde{r}_i = P_i((Id - \mathcal{T})^T)\tilde{r}_0$:

$$\tilde{r}_0, \tilde{r}_1, \tilde{r}_2, \ldots,$$

where P_i indicates a polynomial of degree i. These sequences satisfy the following relations [7]:

$$r_i^T \tilde{r}_j = 0 \qquad i \neq j \tag{17}$$

$$r_i^T \tilde{r}_i \neq 0 \tag{18}$$

If $r_i^T \tilde{r}_i = 0$ then BiCG would break down, but we will not discuss this problem here. For the residuals we have $r_i = P_i(Id - \mathcal{T})r_0 \in span\{r_0, (Id - \mathcal{T})r_0, (Id - \mathcal{T})^2 r_0, \ldots, (Id - \mathcal{T})^i r_0\} = K^{i+1}(Id - \mathcal{T}, r_0)$, and furthermore we have $K^{i+1}((Id - \mathcal{T}), r_0) = K^{i+1}(\mathcal{T}, r_0)$. Together this gives

$$r_i \in K^{i+1}(\mathcal{T}, r_0) \tag{19}$$

Proposition 2.10 *Let $\{r_0, r_1, \ldots, r_{k-1}\}$ be independent and $r_k \in span\{r_0, r_1, \ldots, r_{k-1}\}$, then $r_k = 0$ and BiCG converges in k steps.*

Although being similar, this property differs from the finite termination properties for BiCG [7], for CG if the operator is a low rank perturbation of the identity, which leads, as in this case, to convergence in a number of steps equal to the rank of the perturbation [9], and for GMRES [20]. In these other cases, the residual is necessarily zero because it is both an element of and orthogonal to the same space, whereas the present property is derived from the residual being an element of one space and orthogonal to another, in principle completely different space.

Proof: For r_k we have the following two relations:

$$r_k \in span\{r_0, r_1, \ldots, r_{k-1}\} \tag{20}$$

$$r_k \perp span\{\tilde{r}_0, \tilde{r}_1, \ldots, \tilde{r}_{k-1}\} \tag{21}$$

So (20) implies $r_k = \sum_{i=0}^{k-1} \alpha_i r_i$, and then (21) gives

$$\forall j : 0 \leq j \leq k-1 : \tilde{r}_j^T \left(\sum_{i=0}^{k-1} \alpha_i r_i \right) = 0 \Leftrightarrow \sum_{i=0}^{k-1} \alpha_i \tilde{r}_j^T r_i = 0$$

Together with (17) this leads to

$$\forall j : 0 \leq j \leq k-1 : \alpha_j \tilde{r}_j^T r_j = 0,$$

which means, using (18), that $\alpha_j = 0, \;\; 0 \leq j \leq k-1$. Therefore we have

$$r_k = 0,$$

and hence BiCG has converged. $\square$

Proposition 2.11 *For the linear system defined in (14) BiCG will converge in at most $N-1$ iterations if there is no breakdown.*

Proof: From (14) we can derive that $K^N(\mathcal{T}, r_0) = K^{N-1}(\mathcal{T}, r_0)$. Together with (19) this leads to $r_{N-1} \in K^{N-1}(\mathcal{T}, r_0)$, so that $r_{N-1} \in span\{r_0, r_1, \ldots, r_{N-2}\}$. Proposition 2.10 then proves that $r_{N-1} = 0$, and therefore BiCG has converged. □

Note that if the set $\{r_0, r_1, \ldots, r_k\}$ becomes dependent before $k = N-1$ BiCG will have converged as well. It is not difficult to see that if the BiCG-residual $r_{N-1} = 0$, then also the Bi-CGSTAB-residual $r^{stab}_{N-1} = 0$. Bi-CGSTAB constructs its residual r^{stab}_i such as to be a polynomial of the form $r^{stab}_i = Q_i(Id - M)P_i(Id - M)r_0$, where $P_i(Id - M)r_0$ is still the BiCG-residual [21]. So that, if the BiCG-residual $r_i = P_i(Id - \mathcal{T})r_0 = 0$, then also $r^{stab}_i = 0$, and Bi-CGSTAB will have converged as well.

Assuming that the set of vectors $\{G, \mathcal{T}G, \mathcal{T}^2G, \ldots, \mathcal{T}^{N-2}G\}$ is independent, the solution of $(Id - \mathcal{T})x = G$ is given by (16), so it can be computed without any extra Krylov method. Assuming that the norm of $\mathcal{T}^{N-2}G$ is sufficiently large, equation (16) also indicates that GMRES cannot solve the set of equations (14) in less iterations than BiCG (however with half the number of matrix vector products).

3 Approximation of the optimal interface conditions

The interface conditions (2) lead to optimal results but only in the case of a decomposition into vertical strips. Even in this case, they are difficult to use in a code. Indeed, operators $\Lambda_{i,r \text{ or } l}$ are not partial differential operators. Moreover, in general, we do not have an explicit form of these operators. Nevertheless, it is usually possible to approximate them by partial differential operators as it is done for approximating exact absorbing boundary conditions (see e.g. [4], [5]). In this section, we explain briefly how the optimal interface conditions are approximated by local operators (i.e. partial differential operators). This enables us to write a Schur type formulation for an arbitrary decomposition of the domain and to remove the restriction of a decomposition into vertical strips. In § 4, this strategy is applied to the convection-diffusion operator and numerical results are shown.

3.1 Design of the approximate optimal interface conditions

Our goal is to approximate at some point $x_0 \in \Gamma_{i,r \text{ or } l}$ the operators $\Lambda_{i,r \text{ or } l}$ by partial differential operators. In order to be able to follow the strategy developed in [4], we make the following assumptions: x_0 is far from $\partial\Omega$ and the interface is flat enough so it can be approximated by its tangent at x_0. As a result, we may approximate $\Omega_{i,r \text{ or } l}$ by a half-plane. We also assume that the coefficients of the operator $\mathcal{L}$ vary slowly so that they can be approximated by their values at x_0 (contrarily to the application of § 2.1). By making use of the Fourier transform with respect to the tangential variable, we obtain an approximation of $\Lambda_{i,r \text{ or } l}$ in the form of a convolution operator. This operator is itself approximated by a partial differential operator by approximating its symbol by a polynomial (for more details, see [19], [18]). In some cases, it is possible to make less restrictive assumptions (see e.g. in the context of absorbing boundary conditions or of paraxial approximations [2], [3], [15], [16], [13]).

3.2 A Schur type algorithm

We want to write a system analogous to system (14) but based on the approximate optimal interface conditions. Since these operators are local, we are not restricted any more to decompositions into vertical strips. We will thus obtain a substructuring formulation which can be solved by conjugate gradient like methods. The resulting algorithm is what we call a Schur type algorithm.

Let Ω be a bounded open set of $\mathbf{R}^2$. Let $\Omega_{i,\,1\le i\le N}$ be a finite sequence of sets embedded in Ω such that $\bar\Omega = \cup_{i=1}^N \bar\Omega_i$. Let $\Gamma = \partial\Omega$, $\Gamma_i = \partial\Omega_i - \Gamma$. The outward normal from Ω_i is $\vec{n}_i$ and $\vec{\tau}_i$ is a tangential unit vector. Let us denote by $\mathcal{B}_{i,\,1\le i\le N}$ the approximations to the optimal interface conditions defined by (2). Since the operators $\mathcal{B}_i$ are local, the subscript r or l is meaningless and will not be used here. We assume the operators $\mathcal{B}_{i,\,1\le i\le N}$ to lead to well posed boundary value problems (see below BVP (22)). We assign to each subdomain i an operator S_i: Let f be a function from Ω_i to $\mathbf{R}$ and h a function from Γ_i to $\mathbf{R}$, $S_i(h,f,g)$ is the solution v of the following boundary value problem:

$$\begin{aligned} \mathcal{L}(v) = f(x), &\qquad x \in \Omega_i \\ \mathcal{B}_i(v) = h(x), &\qquad x \in \Gamma_i \\ \mathcal{C}(v) = g(x), &\qquad x \in \partial\Omega_i \cap \Gamma \end{aligned} \tag{22}$$

In order to take multiple overlaps into account, we introduce a sequence (η_i^j), $1\le i\le N$, $1\le j\le N$, $i\ne j$ of functions defined on the boundaries of the subdomains which satisfy:

$$\begin{aligned} &i)\quad \eta_i^j : \partial\Omega_i \longrightarrow [0,1] \\ &ii)\quad \eta_i^j = 0 \text{ on } \partial\Omega_i - \bar\Omega_j \\ &iii)\quad \textstyle\sum_{j,j\ne i} \eta_i^j(x) = 1, \qquad x \in \partial\Omega_i \end{aligned}$$

Remark 3.1 η_i^j *is zero if* $\partial\Omega_i \cap \bar\Omega_j = \emptyset$.

It is now possible to write a substructuring formulation. Let u be the solution to (1) and $u_i = u_{|\Omega_i}$. We write a system for $\mathcal{B}_i(u_i)$:

$$\begin{aligned} \mathcal{B}_i(u_i) &= \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(u_i) = \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(u_j) \\ &= \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(S_j(\mathcal{B}_j(u_j), f_{|\Omega_j}, g)) \\ &= \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(S_j(0, f_{|\Omega_j}, g)) + \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(S_j(\mathcal{B}_j(u_j), 0, 0)) \end{aligned}$$

Thus, $(\mathcal{B}_i(u_i))_{1\le i\le N}$ solves the following linear system:

$$\mathcal{B}_i(u_i) - \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(S_j(\mathcal{B}_j(u_j),0,0)) = \sum_{j,j\ne i} \eta_i^j \mathcal{B}_i(S_j(0, f_{|\Omega_j}, g)),\ 1\le i\le N \tag{23}$$

Let $H = (H_i)_{1\le i\le N}$ and $G = (G_i)_{1\le i\le N}$ be the vectors

$$H = \begin{bmatrix} \mathcal{B}_1(u_1) \\ \vdots \\ \mathcal{B}_N(u_N) \end{bmatrix} \text{ and } G = \begin{bmatrix} \sum_{j,j\ne 1} \eta_1^j \mathcal{B}_1(S_j(0, f_{|\Omega_j}, g)) \\ \vdots \\ \sum_{j,j\ne N} \eta_N^j \mathcal{B}_N(S_j(0, f_{|\Omega_j}, g)) \end{bmatrix}$$

and $\mathcal{T}$ be the linear operator defined by

$$\mathcal{T}(H) = \left[\begin{array}{c} \sum_{j,j\neq 1} \eta_1^j \mathcal{B}_1(S_j(\mathcal{B}_j(u_j),0,0)) \\ \vdots \\ \sum_{j,j\neq N} \eta_N^j \mathcal{B}_N(S_j(\mathcal{B}_j(u_j),0,0)) \end{array} \right]$$

System (23) may now be written in the following compact form:

$$(Id - \mathcal{T})(H) = G \tag{24}$$

As in § 2.2, we consider three three algorithms for the solution of (24), GMRES, BiCGSTAB and Jacobi :

$$H^{n+1} = \mathcal{T}(H^n) + G$$

The last algorithm corresponds to the additive Schwarz method. Since the operator $\mathcal{T}$ is no longer nilpotent, the Schwarz method should not converge in a finite number of steps. GMRES and BiCGSTAB (except if breakdown occurs) always converge in a finite number of steps (ignoring round-off errors) for a finite dimensional problem.

4 Numerical results for the convection-diffusion equation

We apply the strategy explained above to the convection-diffusion equation. Let

$$\mathcal{L} = \frac{1}{\Delta t} + a(x,y)\frac{\partial}{\partial x} + b(x,y)\frac{\partial}{\partial y} - \nu\Delta \tag{25}$$

where $\vec{a} = (a,b)$ is the velocity field, ν is the viscosity. Δt is a constant which could correspond for instance to a time step for a backward-Euler scheme for the time dependent convection-diffusion equation.
For a subdomain Ω_i, the approximations to the optimal interface conditions obtained using the method outlined in § 3.1 read as follows ($\vec{a}$ is the velocity field (a,b), $\vec{n_i}$ is the outward normal from Ω_i and $\vec{\tau_i}$ is a tangential unit vector on $\partial\Omega_i$):

$$\mathcal{B}_i^0 = \frac{\partial}{\partial \vec{n_i}} - \frac{\vec{a}.\vec{n_i} - \sqrt{(\vec{a}.\vec{n_i})^2 + \frac{4\nu}{\Delta t}}}{2\nu} \tag{26}$$

or

$$\begin{aligned} \mathcal{B}_i^2 = \frac{\partial}{\partial \vec{n_i}} - \frac{\vec{a}.\vec{n_i} - \sqrt{(\vec{a}.\vec{n_i})^2 + \frac{4\nu}{\Delta t}}}{2\nu} + \frac{\vec{a}.\vec{\tau_i}}{\sqrt{(\vec{a}.\vec{n_i})^2 + \frac{4\nu}{\Delta t}}}\frac{\partial}{\partial \vec{\tau_i}} \\ - \frac{\nu}{\sqrt{(\vec{a}.\vec{n_i})^2 + \frac{4\nu}{\Delta t}}}\left(1 + \frac{(\vec{a}.\vec{\tau_i})^2}{(\vec{a}.\vec{n_i})^2 + \frac{4\nu}{\Delta t}}\right)\frac{\partial^2}{\partial \vec{\tau_i}^2} \end{aligned}$$

where the superscript denotes the order of the approximation, for more details see [19], [18]. The boundary conditions $\mathcal{B}_i^{0 \text{ or } 2}$ are far field boundary conditions (also called Outflow B.C., Absorbing B.C., Artificial B.C., Radiation B.C.,..., see [4], [12]).

We use a two-dimensional test problem to illustrate the validity of the method. We solve the following problem:

$$\begin{cases} \mathcal{L}(u) = \frac{u}{\Delta t} + a(x,y)\,\frac{\partial u}{\partial x} + b(x,y)\,\frac{\partial u}{\partial y} - \nu \Delta u = 0, \quad 0 \le x \le 1,\ 0 \le y \le 1 \\ u(0,y) = 1, \quad 0 < y < 1 \\ \frac{\partial u}{\partial y}(x,1) = 0, \quad 0 < x < 1 \\ \frac{\partial u}{\partial x}(1,y) = 0, \quad 0 < y < 1 \\ u(x,0) = 0, \quad 0 < x < 1 \end{cases}$$

The operator $\mathcal{L}$ is discretized by a standard upwind finite difference scheme of order 1 (see [6]) and $\mathcal{B}_{i,\,1\le i\le N}$ by a finite difference approximation. We used a rectangular finite difference grid. The mesh size is denoted by h. The unit square is decomposed into overlapping rectangles. The resulting discretization of system (24) is denoted by:

$$(Id - \mathcal{T}_h)(H_h) = G_h \tag{27}$$

The test problem has been implemented at ONERA on an IPSC860.

Remark 4.1 *Any other discretization could be used as well.*

From the definition of $\mathcal{T}_h$, we see that the computation of $\mathcal{T}_h$ applied to some vector H_h amounts to the solution of N independent boundary value subproblems (one subproblem in each subdomain) which can be solved in parallel. We have considered three algorithms in order to solve (27): GMRES(∞), Bi-CGSTAB and a Jacobi algorithm (cf. § 2.2):

$$H_h^{n+1} = \mathcal{T}_h(H_h^n) + G_h \tag{28}$$

which corresponds to an additive Schwarz method (ASM) whose convergence in the continuous case has been studied in [19] for outflow boundary conditions.

In tables 1 and 2, we give the number of subproblems solved so that the maximum of the error is smaller than 10^{-6}. One iteration of GMRES(∞) or of ASM counts for computing the solution for each subdomain once and one iteration of BiCGSTAB counts for computing the solution for each subdomain twice. In the tables, Id corresponds to the use of Id as interface condition (Dirichlet problems). The tests include the case $\mathcal{B}_i = Id$ since it corresponds to the classical Schwarz method when the Jacobi algorithm is used.

The results in Table 1 were obtained using the following parameters:
8×1 subdomains, 21×120 points in each subdomain, overlap $= 2h$, $\nu = 0.1$, $\Delta t = 10^{40}$, $a = y$, $b = 0$.

Boundary Cond.	ASM	Bi-CGSTAB	GMRES
Id	844	88	61
$\mathcal{B}_0$	86	38	33
$\mathcal{B}_2$	46	28	24

Table 1: Computational cost vs. interface conditions and solvers

The results in Table 2 were obtained using the following parameters:
4×4 subdomains, 35×35 points in each subdomain, overlap $= 2h$, $\nu = 0.1$, $\Delta t = 1$, $a = y$, $b = 0$.

Boundary Cond.	ASM	Bi-CGSTAB	GMRES
Id	479	64	50
$\mathcal{B}_0$	27	22	19
$\mathcal{B}_2$	18	16	16

Table 2: Computational cost vs. interface conditions and solvers

The use of outflow boundary conditions leads to a significant improvement whatever iterative solver is used. Bi-CGSTAB and GMRES give similar results with an advantage to GMRES in terms of computational cost and to BiCGSTAB in terms of storage requirements, since only two directions have to be stored.

References

[1] B. Despres, *Domain Decomposition Method and the Helmholtz Problem*, Mathematical and Numerical aspects of wave propagation phenomena, SIAM (1991), 44-52.

[2] A. Bamberger, B. Engquist, L. Halpern and P. Joly, *Parabolic wave equation approximations in heterogenous media*, SIAM J. Appl. Math., **48 No 1**, (1988) 99-128.

[3] A. Bamberger, B. Engquist, L. Halpern and P. Joly, *Higher order paraxial wave approximations in heterogeneous media*, SIAM J. Appl. Math., **48 No 1**, (1988) 129-154.

[4] B. Engquist and A. Majda, *Absorbing Boundary Conditions for the Numerical Simulation of Waves*, Math. Comp. **31** (139), (1977) 629-651.

[5] B. Engquist and A. Majda, *Radiation Boundary Conditions for Acoustic and Elastic Wave Calculations*, Comm. on Pure and Appl. Math., vol XXXII, (1979), 313-357.

[6] C.A.J. Fletcher, *Computational Techniques for Fluid Dynamics*, Springer Series in Computational Physics

[7] R. Fletcher. Conjugate gradient methods for indefinite systems. In G.A. Watson, editor, *Numerical Analysis Dundee 1975, Lecture Notes in Mathematics 506*, pages 73–89, Berlin, Heidelberg, New York, 1976. Springer-Verlag.

[8] E. Givois, *Etude et implémentation de deux méthodes de décomposition de domaines. Une approche monodimensionnelle pour l'initiation de la détonique à l'échelle moléculaire.* Thèse de l'Université Paris Dauphine (1992).

[9] Golub, G. H. and Van Loan C. F. ,*Matrix Computations (2nd edition)* , John Hopkins University Press, 1989.

[10] D. Gottlieb and Hirsh R.S., *Parallel Pseudospectral Domain Decomposition Techniques* J. Sci. Comput. **4**, No 4, (1989), 309-325.

[11] T. Hagstrom, R.P. Tewarson and A. Jazcilevich, *Numerical Experiments on a Domain Decomposition Algorithm for Nonlinear Elliptic Boundary Value Problems*, Appl. Math. Lett., **1, No 3** (1988), 299-302.

[12] L. Halpern, *Artificial Boundary Conditions for the Advection-Diffusion Equations*, Math. Comp., vol 174, 1986, 425-438.

[13] L. Halpern and J. Rauch, *Absorbing Boundary Conditions for Diffusion Equations*, Prépublications Mathématiques de l'Université de Paris Nord, 93-05 (1993).

[14] P.L. Lions, *On the Schwarz Alternating Method III: A Variant for Nonoverlapping Subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM (1989), 202-223.

[15] J. P. Lohéac, *An Artificial Boundary Condition for an Advection-Diffusion Equation*, Math. Meth. in the Appl. Sci., **14**, (1991) 155-175.

[16] J.P. Lohéac, F. Nataf and M. Schatzman, *Parabolic Approximations of the Convection-Diffusion Equation*, Math. of Comp., **60** (202), p. 515-530, 1993.

[17] F. Nataf, *Méthodes de Schur généralisées pour l'équation d'advection-diffusion (Generalized Schur methods for the advection-diffusion equation)*, C.R.Acad. Sci. Paris, t. 314, Série I, p. 419-422, 1992.

[18] F. Nataf, *On the Use of Open Boundary Conditions in Block Gauss-Seidel Methods for the Convection-Diffusion Equations*, Rapport interne CMAP n^0 284, Octobre 1993.

[19] F. Nataf and F. Rogier, *Factorization of the Convection-Diffusion Operator and the Schwarz Algorithm*, to appear in M^{3}AS.

[20] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[21] H.A. Van der Vorst. *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems.* SIAM J. Sci. Statist. Comput., 13:631–644, 1992.