On the Use of Open Boundary Conditions in Block Gauss-Seidel Methods for the Convection-Diffusion Equation

F. NATAF

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

Abstract

In the context of convection-diffusion equation, we consider the use of open boundary conditions (also called radiation boundary conditons) in Block Gauss-Seidel algorithms. Theoretical results and numerical tests show that the convergence is thus accelerated.

1 INTRODUCTION

The discretization of convection-diffusion equation leads to non symmetric systems of equations. When the diffusion is large, algorithms designed for symmetric operators may be used. For instance, Conjugate Gradient or multigrid methods are efficient. However, when the convection is dominant (small diffusion) these methods are less efficient and there are less theoretical results. We are interested here in block Gauss-Seidel iterative methods. In [6], the solving of Navier-Stokes equations is performed by a block Gauss-Seidel iterative method. In [10], flow directed Gauss-Seidel iterative methods for the convection-diffusion equation are analyzed. The effect of overlapping is considered. These methods may be viewed as the discretization of a domain decomposition method defined at the continuous level. Let us exemplify this. We consider the one-dimensional convection-diffusion model problem:

$$a\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f, \quad 0 < x < 1 \tag{1}$$

$$u(0) = u^0;\ u(1) = 0 \tag{2}$$

[0]1991 Mathematics Subject Classification. Primary 65N55, 47A68, 76R05

where a is the velocity ($a > 0$), ν is the viscosity, f and u^0 are given data. Problem (1) is discretized by the standard upwind scheme:

$$a\frac{u_i - u_{i-1}}{\Delta x} - \nu\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = f(i\Delta x), \ 2 \le i \le NX - 1$$
$$u_1 = u^0 \text{ and } u_{NX} = 0$$

where $\Delta x = \frac{1}{NX-1}$ and u_i is an approximation of $u(i\Delta x)$. In order to write the Block Gauss-Seidel iterative method (cf. [10]), we introduce the overlapping covering B_1, B_2 of the set of indices $\{1, \dots, NX\}$ defined by $B_1 = \{1, \dots, N_1\}$ and $B_2 = \{N_2, \dots, NX\}$ where N_1 and N_2 are some integers satisfying $N_2 < N_1$. We now seek solutions $U^{n,m} = \{u_i^{n,m}\}$ defined on B_m for $m = 1$ or 2 as follows:

$$a\frac{u_i^{n,m} - u_{i-1}^{n,m}}{\Delta x} - \nu\frac{u_{i+1}^{n,m} - 2u_i^{n,m} + u_{i-1}^{n,m}}{\Delta x^2} = f(i\Delta x), \ i \in B_m^{int} \tag{3}$$
$$u_{N_1}^{n,1} = u_{N_1}^{n-1,2}, \ u_{N_2}^{n,2} = u_{N_2}^{n,1}$$

where $B_1^{int} = \{2, \dots, N_1 - 1\}$ and $B_2^{int} = \{N_2 + 1, \dots, NX\}$. Let $L_1 = (N_1 - 1)\Delta x$, $l_2 = (N_2 - 1)\Delta x$, $\Omega_1 = [0, L_1]$ and $\Omega_2 = [l_2, 1]$. Algorithm (3) may be viewed as a discretization of the following iterative continuous domain decomposition method:

$$a\frac{\partial u^{n,m}}{\partial x} - \nu\frac{\partial^2 u^{n,m}}{\partial x^2} = f \text{ in } \Omega_m, \ m = 1, 2 \tag{4}$$
$$u^{n,1}(L_1) = u^{n-1,2}(L_1), \ u^{n,2}(l_2) = u^{n,1}(l_2)$$

It is clear from the continuous algorithm that other boundary conditions than Dirichlet boundary conditions may be used at L_1 and l_2.

In this paper, we analyze the use of open boundary conditions (OBC) (also called absorbing boundary conditions, artificial boundary conditions, radiation boundary conditions, see e.g. [4], [5], [8]) at the boundaries of the subdomains. Open boundary conditions are used when some physical phenomena take place in unbounded domains. For numerical computations, it is necessary to bound the domain by an artificial boundary. The issue is then to design a boundary condition at this boundary such that the solution in the bounded domain is as close as possible to the solution of the problem set in the unbounded domain. Here, this notion is used in order to solve linear equations (see also [7], [1]).

We want to solve the model problem

$$\mathcal{L}(u) = \frac{u}{\Delta t} + a(x,y)\,\frac{\partial u}{\partial x} + b(x,y)\,\frac{\partial u}{\partial y} - \nu\Delta u = F + \frac{u(t - \Delta t)}{\Delta t} = f \tag{5}$$

on a vertical strip $]0, L[\times\mathbf{R}$. This equation arises from an implicit discretization in time of a time-dependent convection-diffusion problem and has to be solved at each time step.

The paper is organized as follows: in § 2, we compare different transmission conditions in the 1-D case where all computations are analytic. In section 3.1, by using the notion of Open Boundary Conditions (OBC), we are able to generalize this situation of superconvergence to the 2-D case. Unfortunately, the OBC are non local and are thus difficult to use. This is the reason why, in section 3.2, they are approximated by local operators. This leads to a modified method which has the advantage compared to the previous one to involve only local operators. The superconvergence is then lost and in § 4, we prove some convergence results. In § 5 the numerical implementation is discussed and numerical results are shown. We also compare with the use of standard boundary conditions. In § 6 we conclude.

2 1-D CASE

We consider the following steady-state convection-diffusion equation on a segment of line :

$$\mathcal{L}(u) = a\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f \qquad \text{on }]0, L[\tag{6}$$

$$u(0) = u(L) = 0, \quad a = C^t > 0 \tag{7}$$

Let L_1, l_2 be such that $0 < l_2 \leq L_1 < L$. We consider the following algorithm to solve (6):

$$\mathcal{L}(u_1^{n+1}) = f,\ \forall x \in]0, L_1[,\ \ u_1^{n+1}(0) = 0, B^+(u_1^{n+1})(L_1) = B^+(u_2^n)(L_1) \tag{8}$$

$$\text{then, } \mathcal{L}(u_2^{n+1}) = f,\ \forall x \in]l_2, L[,\ \ u_2^{n+1}(L) = 0,\ B^-(u_2^{n+1})(l_2) = B^-(u_1^{n+1})(l_2) \tag{9}$$

where B^+, B^- are two operators to be chosen. We analyse the convergence for different transmission conditions B^+ and B^-. We shall take Dirichlet and/or Neumann boundary conditions and also

$$\begin{aligned} B^- &= \partial_x - \tfrac{a}{\nu} \\ B^+ &= \partial_x \end{aligned} \tag{10}$$

which corresponds to exact Open Boundary Conditions (OBC, see next §) for the convection-diffusion operator $\mathcal{L}$. In order to examine the convergence, we introduce the auxilliary unknowns $e_1^n = u_1^n - u, e_2^n = u_2^n - u$ (u is the exact solution of (6)-(7)). Due to the fact that $a = C^t$, we easily find the general solution of $\mathcal{L}(e_1^{n+1}) = 0, e_1^{n+1}(0) = 0$ and $\mathcal{L}(e_2^{n+1}) = 0, e_2^{n+1}(L) = 0$:

$$e_1^{n+1} = r_1^{n+1}(e^{\frac{a}{\nu}x} - 1),\ \ e_2^{n+1} = r_2^{n+1}(e^{\frac{a}{\nu}(x-L)} - 1)$$

The coefficients r_1^n, r_2^n are obtained by application of the boundary conditions $B^+(e_1^{n+1}) = B^+(e_2^n),\ B^-(e_2^{n+1}) = B^-(e_1^{n+1})$.
The convergence rate ρ is defined as :

$$\rho = \frac{r_2^{n+1}}{r_2^n}$$

The following table gives the values of ρ for differents choice of operators B^+, B^-.

Boundary conditions	ρ
$B^+u = B^-u = u$	$\approx e^{-\frac{a}{\nu}(L_1-l_2)}$
$B^+u = \dfrac{\partial u}{\partial x},\ B^-u = u$	$e^{-\frac{a}{\nu}L} * \phi(l_2)$
$B^+u = \dfrac{\partial u}{\partial x}\ ,\ B^-u = \dfrac{\partial u}{\partial x} - \dfrac{a}{\nu}u$	$e^{-\frac{a}{\nu}L}$

Table 1 - convergence rate vs. transmission conditions

where $\phi(l_2) = \frac{e^{\frac{a}{\nu}l_2}-1}{e^{\frac{a}{\nu}(l_2-L)}-1} \to -\infty$, as $l_2 \to L$. The use of Dirichlet boundary conditions lead to a convergence rate which depends on the thickness of the overlapping zone. The second choice of boundary conditions corresponds to the use of Dirichlet boundary condition at inflow and of "characteristic" BC at outflow. The position of l_2 is then crucial to obtain a stable scheme. If l_2 is close to L, the domain decomposition method does not converge. The last boundary conditions yield a convergence rate which is exponentially small with respect to the size of the segment of line and is independent of l_2 and L_1. This motivates, in the 1-D case, the use of OBC as transmission conditions. In the next section, we extend the last algorithm the two-dimensional case.

3 Extension to the 2-D case

We precise the notion of OBC since it will be used in the sequel.

3.1 Open Boundary Conditions

We follow the strategy explained in [4], [5] or [8]. We want to solve $\mathcal{L}(u) = f$ on the plane $\mathbf{R}^2$ with f compactly supported in the left half plane $\mathbf{R}^2_-$. Suppose we want to bound the domain in the direction of positive x by introducing an artificial boundary at $x = 0$. The cut domain is now $\mathbf{R}^2_-$ which contains the support of f. To close the BVP set on $\mathbf{R}^2_-$ we have to add a boundary condition on the artificial boundary. A boundary condition such that the solution obtained in the cut domain, $\mathbf{R}^2_-$, is the restriction of the solution of equation (5) set in the whole plane will be referred to as an exact OBC. We shall see below how it can be designed. This exact OBC is non local in space and is usually approximated for convenience. Such approximations will be referred to as OBC. To design the exact OBC, consider the Dirichlet to Neumann operator of the right half plane:

$$\Lambda^- :\ g \longrightarrow \frac{\partial u}{\partial x}(0,.)$$

where u solves

$$\mathcal{L}(u) = 0 \text{ for } x > 0$$

$$u = g \text{ at } x = 0$$
$$u \text{ bounded at } \infty$$

The boundary condition
$$(\partial_x - \Lambda^-)(v) = 0 \tag{11}$$
is an exact boundary condition. Indeed, the solution u of (5) in the whole plane satisfies (11) since f has compact support in the left half plane. Since (11) leads to a well-posed BVP, (11) is an exact OBC. The coefficients of the operator $\mathcal{L}$ are constant. The Fourier symbol of Λ^- may be written explicitly by performing a Fourier transform of (5) w.r.t y. The dual variable of y is denoted by k and the Fourier transform of u by $\hat{u}$. We obtain:
$$\widehat{\mathcal{L}(u)} = (\frac{1}{\Delta t} + a\,\frac{\partial}{\partial x} - i\,b\,k - \nu\frac{\partial^2}{\partial x^2} + \nu k^2)(\hat{u}) = 0, \text{ for } \quad x \geq 0$$
For a fixed k, this equation is an ODE in x whose general solution is
$$\hat{u}(x,k) = \alpha(k)\, e^{\lambda^-(k)x} + \beta(k)\, e^{\lambda^+(k)x}$$
where
$$\begin{aligned} \lambda^+(k) &= \frac{a + \sqrt{a^2 + \frac{4\nu}{\Delta t} - 4i\nu bk + 4k^2\nu^2}}{2\nu}, \; Re(\lambda^+) > 0 \\ \lambda^-(k) &= \frac{a - \sqrt{a^2 + \frac{4\nu}{\Delta t} - 4i\nu bk + 4k^2\nu^2}}{2\nu}, \; Re(\lambda^-) < 0 \end{aligned} \tag{12}$$

We want u to be bounded at infinity, hence β is zero. Taking into account the Dirichlet boundary condition, we have
$$\hat{u}(x,k) = \hat{g}(k)\, e^{\lambda^-(k)x}$$
Thus,
$$\frac{\partial \hat{u}}{\partial x} = \lambda^-(k)\,\hat{g}\, e^{\lambda^-(k)x} = \lambda^-(k)\hat{u}(x,k) \tag{13}$$
In particular, $\frac{\partial \hat{u}}{\partial x}(0,k) = \lambda^-(k)\,\hat{g}$ and Λ^- has for symbol λ^-. In fact from (13) we see that
$$(\partial_x - \Lambda^-)(u) = 0 \text{ for any } x \geq 0 \text{ and } y \in \mathbf{R} \tag{14}$$
This relation will be very useful in the sequel
We have considered the case of the right half plane. It is of course natural to consider the similar problem in the left half plane. We introduce the Dirichlet to Neumann operator of the left half plane:
$$\Lambda^+ : \; g \longrightarrow \frac{\partial u}{\partial x}(0,.)$$
where u solves
$$\mathcal{L}(u) = 0 \text{ for } x < 0$$
$$u = g \text{ at } x = 0$$
$$u \text{ bounded at } \infty$$

The boundary condition

$$(\partial_x - \Lambda^+)(v) = 0 \tag{15}$$

is an exact boundary condition if f has a support in $\mathbf{R}^2_+$ and the symbol of Λ^+ is λ^+. We also have the important relation satisfied by u

$$(\partial_x - \Lambda^+)(u) = 0 \text{ for any } x \leq 0 \text{ and } y \in \mathbf{R} \tag{16}$$

In the next section, we write superconvergent methods using the exact OBC.

3.2 Superconvergent methods in 2-D

We want to solve

$$\begin{gathered} \mathcal{L}(u) = f \text{ in } \Omega =]0, L[\times\mathbf{R} \\ B(u) = g_+ \text{ at } x = 0 \\ (\partial_x - \Lambda^-)(u) = g_- \text{ at } x = L \end{gathered}$$

where B is a linear operator. To write the method, we decompose $\bar{\Omega}$ as the union of N vertical strips $\bar{\Omega}_i$ where $\Omega_i =]l_i, L_i[\times\mathbf{R}$ $(i = 1, \ldots, N)$ with possible overlap of size δ. Similarly to 1-D case, we write the following algorithm:

Let u_i^n be an estimate of the solution in domain i, u_i^{n+1} is computed by a double sweep over the domains. We first compute $u_i^{n+1/2}$ by solving successively, beginning by domain N and ending at domain 1, the following problems:
right to left sweep

$$\begin{aligned} \mathcal{L}(u_i^{n+1/2}) &= f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1)\ \ C(u_i^{n+1/2}) &= C(u_{i-1}^n) \\ \text{at } x = L_i,\ (i \neq N)\ \ (\frac{\partial}{\partial x} - \Lambda^-)(u_i^{n+1/2}) &= (\frac{\partial}{\partial x} - \Lambda^-)(u_{i+1}^{n+1/2}) \end{aligned} \tag{17}$$

C is any linear operator leading to a well-posed boundary value problem. At $x = 0$ and $x = L$, we impose the boundary conditions of the initial problem. Then, we set $u_1^{n+1} = u_1^{n+1/2}$ and u_i^{n+1} is obtained by solving successively, beginning by domain 2 and ending at domain N, the following problems:
left to right sweep

$$\begin{aligned} \mathcal{L}(u_i^{n+1}) &= f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1)\ \ C(u_i^{n+1}) &= C(u_{i-1}^{n+1}) \\ \text{at } x = L_i,\ (i \neq N)\ \ (\frac{\partial}{\partial x} - \Lambda^-)(u_i^{n+1}) &= (\frac{\partial}{\partial x} - \Lambda^-)(u_{i+1}^{n+1/2}) \end{aligned} \tag{18}$$

(at $x = L$ we impose the boundary conditions of the initial problem).

It is important to note that we have here an interesting property.

Result 3.1 *If we solve the following problem*

$$\mathcal{L}(u) = f \tag{19}$$
$$B(u) = g_+ \ \textit{at}\ x = 0$$
$$(\frac{\partial}{\partial x} - \Lambda^-)(u) = g_- \ \textit{at}\ x = L$$

with algorithm (17)-(18), we have convergence in one double sweep.

proof. The equations being linear we only have to prove the convergence to 0 when $f = 0$ and $g_+ = g_- = 0$. It is useful to introduce the following notation: $w_i^{n+1/2} = (\frac{\partial}{\partial x} - \Lambda^-)(u_i^{n+1/2})$. Let u_i^0 be the initial estimate in domain i, we will prove that $u_i^1 = 0$ for any i. We first prove by induction that $w_i^{1/2} = 0$ in domain i. We begin the induction with $i = N$. Indeed, we have in domain N:

$$\begin{cases} \mathcal{L}(u_N^{1/2}) = 0 \\ C(u_N^{1/2}) = C(u_{N-1}^0) \text{ at } x = l_N \\ (\frac{\partial}{\partial x} - \Lambda^-)(u_N^{1/2}) = 0 \text{ at } x = L_N = L \end{cases}$$

Recall that $\partial_x - \Lambda^-$ is an exact OBC. As a consequence, $u_N^{1/2}$ may be seen as the restriction to Ω_N of a function u satisfying

$$\mathcal{L}(u) = 0, \quad \text{for } \ l_N \leq x \leq \infty, \ y \in \mathbf{R}$$

Hence, from (14) $w_N^{1/2}$ is equal to zero in domain N.
To complete the induction we have to prove that $w_i^{1/2} = 0 \Rightarrow w_{i-1}^{1/2} = 0$. Indeed, by (17)

$$\begin{cases} \mathcal{L}(u_{i-1}^{1/2}) = 0 \text{ in domain } i-1 \\ \text{at } x = L_{i-1}, \ (\frac{\partial}{\partial x} - \Lambda^-)(u_{i-1}^{1/2}) = (\frac{\partial}{\partial x} - \Lambda^-)(u_i^{1/2}) = w_i^{1/2} \end{cases}$$

We have assumed that $w_i^{1/2} = 0$ in domain i. Hence in the same manner we have proved that $w_N^{1/2}$is zero, we can prove that $w_{i-1}^{1/2} = 0$ in domain $i-1$.
We prove now by induction that $u_i^1 = 0$ for any i. We begin the induction with $i = 1$. Let us recall that $u_1^1 = u_1^{1/2}$. We have by (17):

$$\begin{aligned} &\mathcal{L}(u_1^{1/2}) = 0 \text{ in domain } 1 \\ &\text{at } x = 0, \ \ B(u_1^{1/2}) = 0 \\ &\text{at } x = L_1, \ \ (\tfrac{\partial}{\partial x} - \Lambda^-)(u_1^{1/2}) = (\tfrac{\partial}{\partial x} - \Lambda^-)(u_2^{1/2}) = w_2^{1/2} \end{aligned}$$

We have proved that $w_2^{1/2} = 0$ and this shows that $u_1^{1/2} = 0$.
To complete the induction, we have to prove that $u_i^1 = 0 \Rightarrow u_{i+1}^1 = 0$. Indeed, by (18) u_{i+1}^1 satisfies

$$\begin{aligned} &\mathcal{L}(u_{i+1}^1) = 0 \text{ in domain } i+1 \\ &\text{at } x = l_{i+1}, \ \ C(u_{i+1}^1) = C(u_i^1) = 0 \text{ (by assumption of induction)} \\ &\text{at } x = L_{i+1}, \ \ (\tfrac{\partial}{\partial x} - \Lambda^-)(u_{i+1}^1) = (\tfrac{\partial}{\partial x} - \Lambda^-)(u_{i+2}^{1/2}) = w_{i+2}^{1/2} = 0 \end{aligned}$$

This proves that u_{i+1}^1 is equal to zero.

To write algorithm (17)-(18), we have used only one of the exact OBC, namely $\partial_x - \Lambda^-$. By using the other exact OBC $\partial_x - \Lambda^+$, we can propose an algorithm similar to (17)-(18):

Let u_i^n be an estimate of the solution in domain i, u_i^{n+1} is computed by a double sweep over the domains. We first compute $u_i^{n+1/2}$ by solving successively, beginning by domain 1 and ending at domain N, the following problems:
left to right sweep

$$\begin{array}{c} \mathcal{L}(u_i^{n+1/2}) = f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1)\ \ (\frac{\partial}{\partial x} - \Lambda^+)(u_i^{n+1/2}) = (\frac{\partial}{\partial x} - \Lambda^+)(u_{i-1}^{n+1/2}) \\ \text{at } x = L_i,\ (i \neq N)\ \ C(u_i^{n+1/2}) = C(u_{i+1}^n) \end{array} \tag{20}$$

at $x = 0$ and $x = L$, we impose the boundary conditions of the initial problem. C is any linear operator leading to a well-posed boundary value problem. Then, we set $u_N^{n+1} = u_N^{n+1/2}$ and u_i^{n+1} is obtained by solving successively, beginning by domain $N-1$ and ending at domain 1, the following problems:
right to left sweep

$$\begin{array}{c} \mathcal{L}(u_i^{n+1}) = f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1)\ \ (\frac{\partial}{\partial x} - \Lambda^+)(u_i^{n+1}) = (\frac{\partial}{\partial x} - \Lambda^+)(u_{i-1}^{n+1/2}) \\ \text{at } x = L_i,\ \ C(u_i^{n+1}) = C(u_{i+1}^{n+1}) \end{array} \tag{21}$$

(at $x = 0$ we impose the boundary conditions of the initial problem).

In the same way that for algorithm (17)-(18), we may prove the

Result 3.2 *If we solve the following problem*

$$\mathcal{L}(u) = f \tag{22}$$

$$(\frac{\partial}{\partial x} - \Lambda^+)(u) = g_+ \ \textit{at}\ x = 0 \tag{23}$$

$$B(u) = g_- \ \textit{at}\ x = L$$

with algorithm (20)-(21), we have convergence in one double sweep.

These results prove that the choice of the boundary conditions is optimal. Indeed, since the operator $\mathcal{L}$ is a second order elliptic operator, the solution of (5) at any interior point depends on the values of g_+, g_- and f. Convergence cannot be achieved before each point is "informed" of the value g_+, g_- and f. Thus, a block Gauss-Seidel algorithm cannot converge in less than one double-sweep. It should be noted that using both exact OBC has no influence on the convergence.
In flow oriented Gauss-Seidel methods (see [9], [10]), only sweeps in the direction of the flow are considered. It is interesting to look at the number of iterations needed to

achieve convergence when using the exact OBC as transmission conditions instead of Dirichlet BC in flow oriented sweeps. When a is positive, it means that only left to right sweeps are made. When using the exact OBC $(\partial_x - \Lambda^-)$ to solve problem (19), the algorithm is:
Let u_i^n be an estimate of the solution in domain i, u_i^{n+1} is computed by a sweep over the domains. We compute u_i^{n+1} by solving successively, beginning by domain 1 and ending at domain N, the following problems:

$$\begin{aligned}
\mathcal{L}(u_i^{n+1}) &= f \text{ in domain } i \\
\text{at } x = l_i,\ (i \neq 1)\ \ C(u_i^{n+1}) &= C(u_{i-1}^{n+1}) \\
\text{at } x = L_i,\ (i \neq N)\ \ (\frac{\partial}{\partial x} - \Lambda^-)(u_i^{n+1}) &= (\frac{\partial}{\partial x} - \Lambda^-)(u_{i+1}^{n})
\end{aligned} \tag{24}$$

at $x = 0$ and $x = L$, we impose the boundary conditions of the initial problem. Taking $f = 0$ and $g_+ = g_- = 0$ and reasoning as above, it is easy to prove that $(\partial_x - \Lambda^-)(u_i^n) = 0$ for $i \geq N - n + 1$ and that at step N, $u_i^N = 0$ for any i. We have thus convergence in N sweeps.
Had we taken the exact OBC $(\partial_x - \Lambda^+)$, the solving of problem (23) by the algorithm

$$\begin{aligned}
\mathcal{L}(u_i^{n+1}) &= f \text{ in domain } i \\
\text{at } x = l_i,\ (i \neq 1)\ \ (\frac{\partial}{\partial x} - \Lambda^+)(u_i^{n+1}) &= (\frac{\partial}{\partial x} - \Lambda^+)(u_{i-1}^{n+1}) \\
\text{at } x = L_i,\ (i \neq N)\ \ C(u_i^{n+1}) &= C(u_{i+1}^{n})
\end{aligned} \tag{25}$$

would require N sweeps also. Indeed, it is possible to prove that $(\partial_x - \Lambda^+)(u_i^1)$ is zero for any i and thus that u_i^n is zero for $i \geq N - n + 1$.

When using the exact OBC, both left to right sweeps and double sweeps give rise to convergence in a finite number of steps with a theoretical advantage to double sweeps over flow oriented sweeps. In fact, numerical results show that there is no significant differences between the two algorithms. Unfortunately, these algorithms cannot be used for practical purposes for two reasons. The first one is that the boundary conditions are non local. The second one is that for variable coefficients a and b, the explicit form of the exact OBC is not known. This is why we shall go into OBC involving local operators which can be generalized to variable coefficients.

3.3 Local Open Boundary conditions

We follow the strategy used in [4], [8]. Approximating the operators $\Lambda^\pm$ by local operators is equivalent to approximate their symbols $\lambda^\pm(k)$ by polynomials in k. Since $\lambda^\pm(k)$ as functions of k have a polynomial behavior only for small wavenumbers k, it is reasonable to consider Taylor expansions in the vicinity of $k = 0$. We are restricted to at most third order approximations since for higher order approximations, the sign of the approximations of λ^+ (resp. λ^-) is negative (resp. positive) for high wave numbers.

The corresponding BVP would be ill-posed. We do not consider neither the approximation of order three since it yields a boundary condition of order three in y. The approximations, we shall consider, are thus,

$$\lambda_0^+(k) = \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} \text{ and } \lambda_0^-(k) = \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} \tag{26}$$

$$\begin{aligned}
\lambda_1^+(k) &= \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} - ik\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}} \\
&\text{and} \\
\lambda_1^-(k) &= \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} + ik\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}
\end{aligned} \tag{27}$$

$$\begin{aligned}
\lambda_2^+(k) &= \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} - ik\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}} + \frac{\nu}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\left(1+\frac{b^2}{a^2+\frac{4\nu}{\Delta t}}\right)k^2 \\
&\text{and} \\
\lambda_2^-(k) &= \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} + ik\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}} - \frac{\nu}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\left(1+\frac{b^2}{a^2+\frac{4\nu}{\Delta t}}\right)k^2
\end{aligned} \tag{28}$$

These approximations are also valid for a small viscosity ν. Nevertheless, they are different from Taylor approximations of λ^+ and λ^- with respect to ν because of the term $\frac{\nu}{\Delta t}$.

In the physical space, the approximations of the exact OBC read as follows (with obvious notations)

$$\partial_x - \Lambda_0^+ = \partial_x - \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu},\ \partial_x - \Lambda_0^- = \partial_x - \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} \tag{29}$$

or

$$\begin{aligned}
\partial_x - \Lambda_1^+ &= \partial_x - \tfrac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} - \tfrac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\partial_y \\
\partial_x - \Lambda_1^- &= \partial_x - \tfrac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} + \tfrac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\partial_y
\end{aligned} \tag{30}$$

or

$$\begin{aligned}
\partial_x - \Lambda_2^+ &= \partial_x - \tfrac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} - \tfrac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\partial_y + \tfrac{\nu}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}(1+\tfrac{b^2}{a^2+\frac{4\nu}{\Delta t}})\partial_{yy} \\
\partial_x - \Lambda_2^- &= \partial_x - \tfrac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu} + \tfrac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\partial_y - \tfrac{\nu}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}(1+\tfrac{b^2}{a^2+\frac{4\nu}{\Delta t}})\partial_{yy}
\end{aligned} \tag{31}$$

In the case of *variable* coefficients, we shall use the same OBC.

It is natural to modify algorithms (17)-(18), (20)-(21), (24) or (25) by substituting the local OBC (29), (30) or (31) for the exact OBC. Convergence in a finite number of steps is then lost. It is thus wiser to consider algorithms where both of the local OBC are used. Depending on whether double sweeps or left to right sweeps are used, we have now two types of algorithms. We write them.

According to the order of approximation j (j =0, 1 or 2), the "double sweep" algorithm reads
<u>left to right sweep</u>

$$\begin{array}{rl} & \mathcal{L}(u_i^{n+1/2}) = f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1) & (\frac{\partial}{\partial x} - \Lambda_j^+)(u_i^{n+1/2}) = (\frac{\partial}{\partial x} - \Lambda_j^+)(u_{i-1}^{n+1/2}) \\ \text{at } x = L_i,\ (i \neq N) & (\frac{\partial}{\partial x} - \Lambda_j^-)(u_i^{n+1/2}) = (\frac{\partial}{\partial x} - \Lambda_j^-)(u_{i+1}^{n}) \end{array} \tag{32}$$

At $x = 0$ and $x = L$, we impose the boundary conditions of the initial problem. Then, we set $u_N^{n+1} = u_N^{n+1/2}$ and u_i^{n+1} is obtained by solving successively, beginning by domain $N-1$ and ending at domain 1, the following problems:
<u>right to left sweep</u>

$$\begin{array}{rl} & \mathcal{L}(u_i^{n+1}) = f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1) & (\frac{\partial}{\partial x} - \Lambda_j^+)(u_i^{n+1}) = (\frac{\partial}{\partial x} - \Lambda_j^+)(u_{i-1}^{n+1/2}) \\ \text{at } x = L_i,\ (i \neq N) & (\frac{\partial}{\partial x} - \Lambda_j^-)(u_i^{n+1}) = (\frac{\partial}{\partial x} - \Lambda_j^-)(u_{i+1}^{n+1}) \end{array} \tag{33}$$

(at $x = 0$ we impose the boundary conditions of the initial problem).

The other type of algorithm is (j =0, 1 or 2):
We compute u_i^{n+1} by solving successively, beginning by domain 1 and ending at domain N, the following problems:

$$\begin{array}{rl} & \mathcal{L}(u_i^{n+1}) = f \text{ in domain } i \\ \text{at } x = l_i,\ (i \neq 1) & (\frac{\partial}{\partial x} - \Lambda_j^+)(u_i^{n+1}) = (\frac{\partial}{\partial x} - \Lambda_j^+)(u_{i-1}^{n+1}) \\ \text{at } x = L_i,\ (i \neq N) & (\frac{\partial}{\partial x} - \Lambda_j^-)(u_i^{n+1}) = (\frac{\partial}{\partial x} - \Lambda_j^-)(u_{i+1}^{n}) \end{array} \tag{34}$$

at $x = 0$ and $x = L$, we impose the boundary conditions of the initial problem.

Remark. At this point, it is clear that the same type of work can be made in a 3-D situation. One should perform a Fourier transform w.r.t y and z and consider vertical slabs as subdomains.

4 Convergence proofs

We have not been able to prove convergence results for the double sweep algorithm. All the convergence results proved here concern only algorithm (34).

For two subdomains, we explicit the convergence rate and study the influence of the wavenumber in the direction tangential to the boundaries of the subdomains. This case is not very realistic since we have in mind small subdomains. However it has the advantage to show explicitely the dependance of the convergence rate on the wavenumber and on the size of the overlap. For an arbitrary number of subdomains, we begin by proving convergence when there is no overlap between the subdomains and then, that the convergence is geometric when the subdomains overlap. In § 4.3, we extend some of these results to the case of the operator with variable coefficients.

4.1 Constant coefficients - two subdomains

We carry out a Fourier analysis of the convergence of the method. The exact OBC are used as boundary conditions at $x=0$ and $x=L$. They are: $(\frac{\partial}{\partial x}-\Lambda^{+})(u)=g_{+}$ at $x=0$ and $(\frac{\partial}{\partial x}-\Lambda^{-})(u)=g_{-}$ at $x=L$. The error u_2^n-u (resp. u_1^n-u) at the left (resp. right) boundary of domain 2 (resp. 1) at step n is denoted by e_2^n (resp. e_1^n). The convergence rate $\rho_j^i(k)$ (j denotes the order of the approximation factorization used in (34)) in the Fourier space is defined by:

$$\hat{e}_i^{n+1}(k)=\rho_j^i(k)\hat{e}_i^n(k), \quad n\geq 1$$

It turns out that $\rho_j^i(k)$ is independent of the subdomain i and we thus omit the superscript. A straight-forward computation gives

$$\rho_j(k,\delta)=\frac{\lambda^-(k)-\lambda_j^-(k)}{\lambda^+(k)-\lambda_j^-(k)}\;\frac{\lambda^+(k)-\lambda_j^+(k)}{\lambda^-(k)-\lambda_j^+(k)}\; e^{(\lambda^-(k)-\lambda^+(k))\delta} \tag{35}$$

We use the fact that $\lambda^-(k)+\lambda^+(k)=a/\nu$ and that the same holds for the approximations (i.e. $\lambda_j^-(k)+\lambda_j^+(k)=a/\nu$) to simplify (35) as:

$$\rho_j(k,\delta)=\left(\frac{\lambda^-(k)-\lambda_j^-(k)}{\lambda^+(k)-\lambda_j^-(k)}\right)^2\; e^{-\frac{\sqrt{a^2+\frac{4\nu}{\Delta t}-4i\nu bk+4k^2\nu^2}}{\nu}\delta} \tag{36}$$

For any $\delta\geq 0$, and for small wave numbers, ρ_j tends to zero. For large wave numbers, ρ_j tends to 1 if $\delta=0$, and to zero as soon as $\delta\neq 0$ (see fig. 1).

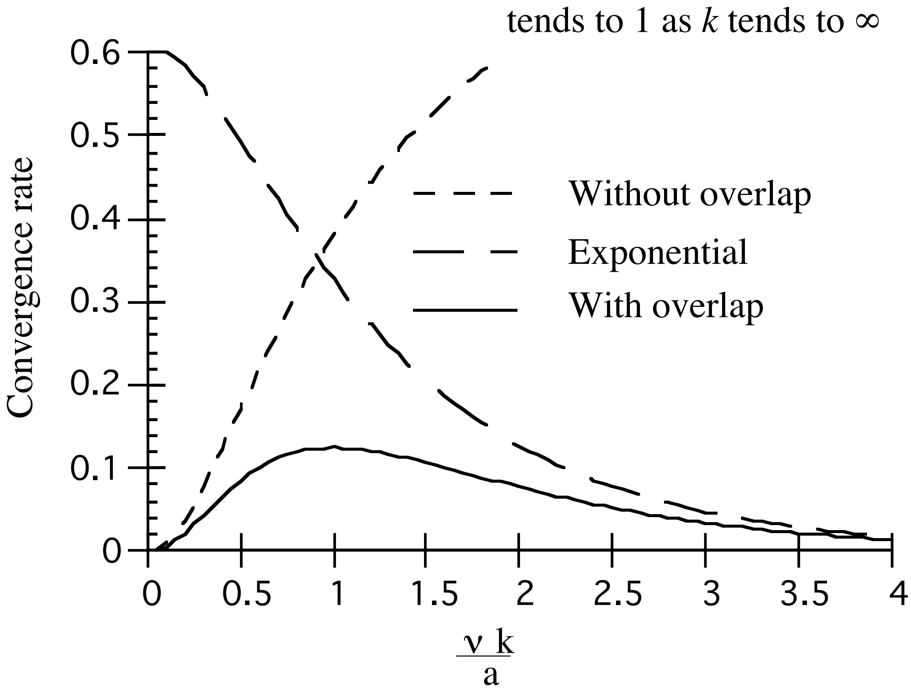

FIGURE 1 - Convergence rate

This shows the importance of the overlap which ensures a geometric convergence of the algorithm since we shall prove that $|\rho_j(k,0)| < 1$:

Let $\mathbf{C}$ be the set of complex numbers, $\bar{\mathbf{C}} = \mathbf{C} \cup \{\infty\}$ and $B_0(0,1) = \{z \in \bar{\mathbf{C}} / \ |z| < 1\}$. The map

$$f: \ \bar{\mathbf{C}} \longrightarrow \bar{\mathbf{C}}$$
$$z \longrightarrow \frac{\lambda^- - z}{\frac{a}{\nu} - \lambda^- - z}$$

is one-to-one and

$$f^{-1}: \ \bar{\mathbf{C}} \longrightarrow \bar{\mathbf{C}}$$
$$z \longrightarrow (\lambda^- - \frac{a}{2\nu})\frac{1+z}{1-z} + \frac{a}{2\nu}.$$

Let g be the map

$$g: \ \bar{\mathbf{C}} \longrightarrow \bar{\mathbf{C}}$$
$$z \longrightarrow \frac{1+z}{1-z}.$$

Then, $|\rho_j(k,0)| < 1 \Leftrightarrow \{\lambda_j^-\} \subset f^{-1}(B_0(0,1)) \Leftrightarrow \{\frac{\lambda_j^- - \frac{a}{2\nu}}{\lambda^- - \frac{a}{2\nu}}\} \subset g(B_0(0,1))$. We have $g(B_0(0,1)) = \{z \in \mathbf{C}/Re(z) > 0\} \cup \{\infty\}$. From (12), (26), (27) and (28), it is easy to see that $Re(\lambda^- - \frac{a}{2\nu}) < 0$, $Re(\lambda_j^- - \frac{a}{2\nu}) < 0$ and $sgn(Im(\lambda^- - \frac{a}{2\nu}) = sgn(Im(\lambda_j^- - \frac{a}{2\nu})$. Thus, $Re(\frac{\lambda_j^- - \frac{a}{2\nu}}{\lambda^- - \frac{a}{2\nu}}) > 0$.

4.2 Constant coefficients - arbitrary number of subdomains

We consider the problem set in the strip $]0, L[\times \mathbf{R}$:

$$\left\{\begin{array}{c} \mathcal{L}(u) = f \\ \text{At } x = 0,\ (\frac{\partial}{\partial x} - \Lambda_j^+)(u) = g_+ \\ \text{and at } x = L,\ (\frac{\partial}{\partial x} - \Lambda_j^-)(u) = g_- \end{array}\right. \tag{37}$$

The rectangle $[0, L] \times \mathbf{R}$ is decomposed in the union $\cup_{i=1}^N \bar{\Omega}_i$ where $\Omega_i =]l_i, L_i[\times \mathbf{R}$ with $0 \le l_i < L_i \le L$, $l_{i+1} = L_i$, $1 \le i \le N$ (non-overlapping domains) and $\Gamma_{-,i} = \{L_i\} \times \mathbf{R}$, $\Gamma_{+,i} = \{l_i\} \times]0, h[$.

The proof is based on the following energy estimate

Lemma 4.1 *Let $L > 0$ and $\Omega =]0, L[\times \mathbf{R}$. Let $u \in H^2(\Omega)$ such that $u(0, .)$ and $u(L, .)$ belong to $H^2(\mathbf{R})$ and satisfy*

$$\mathcal{L}(u) = 0 \quad in\ \Omega \tag{38}$$

Then, we have the following estimate for j =0, 1 or 2:

$$\begin{array}{l} \frac{1}{\nu \Delta t}\sqrt{a^2 + \frac{4\nu}{\Delta t}} \int_0^L \int_{\mathbf{R}} u^2 \\ +\sqrt{a^2 + \frac{4\nu}{\Delta t}} \int_0^L \int_{\mathbf{R}} (\frac{\partial u}{\partial y})^2 + \frac{1}{2}\sqrt{a^2 + \frac{4\nu}{\Delta t}} \int_0^L \int_{\mathbf{R}} (\frac{\partial u}{\partial x})^2 \\ +\nu \int_{x=0} (\frac{\partial u}{\partial x} - \Lambda_j^-(u))^2\, dy + \nu \int_{x=L} (\frac{\partial u}{\partial x} - \Lambda_j^+(u))^2\, dy \\ \le\ \nu \int_{x=0} (\frac{\partial u}{\partial x} - \Lambda_j^+(u))^2\, dy + \nu \int_{x=L} (\frac{\partial u}{\partial x} - \Lambda_j^-(u))^2\, dy \end{array} \tag{39}$$

proof (,) will denote the scalar product in $L^2(\Omega)$. Let $j = 0$, 1 or 2. The proof is based on the approximate factorization of $\mathcal{L}$ (recall that $\Lambda_j^+ + \Lambda_j^- = \frac{a}{\nu}$):

$$\begin{aligned} \mathcal{L}(u) &= -\nu(\partial_x - \Lambda_j^+)\,(\partial_x - \Lambda_j^-)(u) + \nu(\Lambda_j^+\Lambda_j^- - \Lambda^+\Lambda^-)(u) \\ &= -\nu(\partial_x - \Lambda_j^-)\,(\partial_x - \Lambda_j^+)(u) + \nu(\Lambda_j^+\Lambda_j^- - \Lambda^+\Lambda^-)(u) \end{aligned} \tag{40}$$

We multiply (40) by $(\partial_x - \Lambda_j^-)(u)$ and we integrate by parts on Ω. This yields

$$\begin{array}{c} -\frac{\nu}{2}\int_{\mathbf{R}} \left[(\partial_x - \Lambda_j^-)(u)^2\right]_0^L + \nu(\Lambda_j^+(\partial_x - \Lambda_j^-)(u), (\partial_x - \Lambda_j^-)(u)) \\ +\nu((\Lambda_j^+\Lambda_j^- - \Lambda^+\Lambda^-)(u), (\partial_x - \Lambda_j^-)(u)) = 0 \end{array}$$

Similarly, multiplying (40) by $-(\partial_x - \Lambda_j^+)(u)$ yields

$$\begin{array}{c} \frac{\nu}{2}\int_{\mathbf{R}}\left[(\partial_x - \Lambda_j^+)(u)^2\right]_0^L - \nu(\Lambda_j^-(\partial_x - \Lambda_j^+)(u), (\partial_x - \Lambda_j^+)(u)) \\ -\nu((\Lambda_j^+\Lambda_j^- - \Lambda^+\Lambda^-)(u), (\partial_x - \Lambda_j^+)(u)) = 0 \end{array}$$

After the summation and the simplification of these equalities, we obtain

$$\begin{array}{r} -\frac{\nu}{2}\int\left[(\partial_x - \Lambda_j^-)(u)^2\right]_0^L + \frac{\nu}{2}\int\left[(\partial_x - \Lambda_j^+)(u)^2\right]_0^L + \nu((\Lambda_j^+ - \Lambda_j^-)(\frac{\partial u}{\partial x}), \frac{\partial u}{\partial x}) \\ -\nu(\Lambda^+\Lambda^-(u), (\Lambda_j^+ - \Lambda_j^-)(u) + \nu(({}^t\Lambda_j^+\Lambda_j^- - {}^t\Lambda_j^-\Lambda_j^+)(\frac{\partial u}{\partial x}), u) = 0 \end{array}$$

The use of the relation $\Lambda_j^+ + \Lambda_j^- = \frac{a}{\nu}$ simplifies ${}^t\Lambda_j^+\Lambda_j^- - {}^t\Lambda_j^-\Lambda_j^+$ as $\frac{a}{\nu}({}^t\Lambda_j^+ - \Lambda_j^+)$. We have thus

$$\begin{array}{r} -\frac{\nu}{2}\int\left[(\partial_x - \Lambda_j^-)(u)^2\right]_0^L \\ +\frac{\nu}{2}\int\left[(\partial_x - \Lambda_j^+)(u)^2\right]_0^L + \nu((\Lambda_j^+ - \Lambda_j^-)(\frac{\partial u}{\partial x}), \frac{\partial u}{\partial x}) \\ -\nu(\Lambda^+\Lambda^-(u), (\Lambda_j^+ - \Lambda_j^-)(u) + (a({}^t\Lambda_j^+ - \Lambda_j^+)(\frac{\partial u}{\partial x}), u) = 0 \end{array} \tag{41}$$

Since the operator Λ_j^+ is positive and the operator Λ_j^- is negative, when the operator Λ_j^+ is self-adjoint, equality (41) is an energy estimate. In the general case, the only term which remains to be estimated is $(a({}^t\Lambda_j^+ - \Lambda_j^+)(\frac{\partial u}{\partial x}), u)$. Then, we shall use the explicit forms of the operators. We have to distinguish according to the value of j.

If $j = 0$, ${}^t\Lambda_j^+ - \Lambda_j^+ = 0$ and $\Lambda_j^+ - \Lambda_j^- = \frac{\sqrt{a^2+\frac{4\nu}{\Delta t}}}{\nu}$. Since

$$-\Lambda^+\Lambda^- = \frac{1}{\nu\Delta t} + \frac{b}{\nu}\frac{\partial}{\partial y} - \frac{\partial^2}{\partial y^2},$$

it is easy to see by integrating by parts in (41) that (39) holds.

If $j = 1$, ${}^t\Lambda_j^+ - \Lambda_j^+ = -2\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\frac{\partial}{\partial y}$ and $\Lambda_j^+ - \Lambda_j^- = \frac{\sqrt{a^2+\frac{4\nu}{\Delta t}}}{\nu} + 2\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\frac{\partial}{\partial y}$. Integrating by parts in (41) yields

$$\begin{array}{r} -\frac{\nu}{2}\int\left[(\partial_x - \Lambda_j^-)(u)^2\right]_0^L + \frac{\nu}{2}\int\left[(\partial_x - \Lambda_j^+)(u)^2\right]_0^L + \sqrt{a^2 + \frac{4\nu}{\Delta t}}(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial x}) \\ +\frac{\sqrt{a^2 + \frac{4\nu}{\Delta t}}}{\nu\Delta t}(u, u) + \sqrt{a^2 + \frac{4\nu}{\Delta t}}(\frac{\partial u}{\partial y}, \frac{\partial u}{\partial y}) \\ +2\frac{b^2}{\sqrt{a^2 + \frac{4\nu}{\Delta t}}}(\frac{\partial u}{\partial y}, \frac{\partial u}{\partial y}) + \frac{2ab}{\sqrt{a^2 + \frac{4\nu}{\Delta t}}}(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = 0 \end{array}$$

by using $-\frac{2ab}{\sqrt{a^2+\frac{4\nu}{\Delta t}}} \leq \frac{a^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}} + \frac{2b^2}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}$, it is clear that (39) holds for $j = 1$.

If $j=2$, ${}^t\Lambda_j^+ - \Lambda_j^+ = -2a\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\frac{\partial}{\partial y}$ and $\Lambda_j^+ - \Lambda_j^- = \frac{\sqrt{a^2+\frac{4\nu}{\Delta t}}}{\nu} + 2\frac{b}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}\frac{\partial}{\partial y} - \frac{2\nu}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}(1+\frac{b^2}{a^2+\frac{4\nu}{\Delta t}})\frac{\partial^2}{\partial y^2}$. Integrating by parts in (41) yields

$$
\begin{aligned}
&-\frac{\nu}{2}\int\left[(\partial_x-\Lambda_j^-)(u)^2\right]_0^L + \frac{\nu}{2}\int\left[(\partial_x-\Lambda_j^+)(u)^2\right]_0^L + \sqrt{a^2+\frac{4\nu}{\Delta t}}(\frac{\partial u}{\partial x},\frac{\partial u}{\partial x}) \\
&+\frac{2\nu^2}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}(1+\frac{b^2}{a^2+\frac{4\nu}{\Delta t}})(\frac{\partial^2 u}{\partial y^2},\frac{\partial^2 u}{\partial y^2}) \\
&+2\frac{\nu^2}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}(1+\frac{b^2}{a^2+\frac{4\nu}{\Delta t}})(\frac{\partial^2 u}{\partial x\partial y},\frac{\partial^2 u}{\partial x\partial y}) + \frac{\sqrt{a^2+\frac{4\nu}{\Delta t}}}{\nu\Delta t}(u,u) \\
&+(\sqrt{a^2+\frac{4\nu}{\Delta t}} + 2\frac{b^2}{\sqrt{a^2+\frac{4\nu}{\Delta t}}} + \frac{2\nu}{\Delta t\sqrt{a^2+\frac{4\nu}{\Delta t}}}(1+\frac{b^2}{a^2+\frac{4\nu}{\Delta t}}))(\frac{\partial u}{\partial y},\frac{\partial u}{\partial y}) \\
&+\frac{2ab}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}) = 0
\end{aligned}
$$

again, by using $-\frac{2ab}{\sqrt{a^2+\frac{4\nu}{\Delta t}}} \le \frac{a^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}} + \frac{2b^2}{\sqrt{a^2+\frac{4\nu}{\Delta t}}}$ it is clear that (39) holds.

We are now able to prove the

Theorem 4.2 *When solving (37) by algorithm (34) with j =0, 1 or 2, we have convergence in the sense that $\lim_{n\to\infty} \| u_i - u_i^n \|_{H^1} \longrightarrow 0$, for $i=1,\ldots,N$.*

The equations are linear, we can take $f=0$ and $g_+ = g_- = 0$. We prove that u_i^n tends to zero in the H^1 norm as $n\to\infty$. Let us define

$x\in[l_m,L_m]$, $A_x^{n,m} = \int_{\mathbf{R}}[(\partial_x-\Lambda_j^-)(u_m^n)]^2(x,y)\,dy$ and $B_x^{n,m} = \int_{\mathbf{R}}[(\partial_x-\Lambda_j^+)(u_m^n)]^2(x,y)\,dy$

and

$$E^{n,m} = \int_{\mathbf{R}}\int_{l_m}^{L_m} \frac{1}{\nu\Delta t}\sqrt{a^2+\frac{4\nu}{\Delta t}}{u_m^n}^2 + \frac{1}{2}\sqrt{a^2+\frac{4\nu}{\Delta t}}(\frac{\partial u_m^n}{\partial x})^2 + \sqrt{a^2+\frac{4\nu}{\Delta t}}(\frac{\partial u_m^n}{\partial y})^2$$

The energy estimates of lemma 4.1 gives:

$$E^{n,m} + A_{l_m}^{n,m} + B_{L_m}^{n,m} \le A_{L_m}^{n,m} + B_{l_m}^{n,m}$$

By summing over m and using the following relations (resulting from algorithm (34))

$$
\begin{aligned}
A_{L_m}^{n,m} &= A_{L_m}^{n-1,m+1} = A_{l_{m+1}}^{n-1,m+1},\ m\le N-1 \\
B_{l_m}^{n,m} &= B_{l_m}^{n,m-1} = B_{L_{m-1}}^{n,m-1},\ 2\le m \\
A_{L_N}^{n,N} &= 0 \text{ and } B_{l_1}^{n,1} = 0
\end{aligned}
$$

we obtain

$$\sum_m E^{n,m} + \sum_m A_{l_m}^{n,m} + \sum_m B_{L_m}^{n,m} \le \sum_m A_{l_m}^{n-1,m} + \sum_m B_{L_m}^{n,m}$$

We set

$$\mathcal{E}^n = \sum_m E^{n,m}, \quad \mathcal{A}^n = \sum_m A_{l_m}^{n,m} \text{ and } \mathcal{B}^n = \sum_m B_{L_m}^{n,m}$$

so that summing over n yields

$$\sum_{n=1}^{\infty} \mathcal{E}^n \le \mathcal{A}^0$$

This ends the proof of theorem 4.2.

We suppose now that the subdomains Ω_i have an overlap of size $\delta > 0$ (i.e. $L_i - l_{i+1} = \delta$, for $1 \le i \le N-1$). To prove the geometric convergence we need the

Lemma 4.3 *Let $a \in \mathbf{R}$, $b \in C_b^1([0,L] \times \mathbf{R})$ and u satisfies*

$$\frac{u}{\Delta t} + a\frac{\partial u}{\partial x} + b(x,y)\frac{\partial u}{\partial y} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0$$

in $]0,L[\times\mathbf{R}$. If $\frac{1}{\Delta t} > \frac{1}{2}\sup(b_y)$,
the function $G(x) = \int_{\mathbf{R}} u^2(x,y)\,dy$ satisfies the following estimate:

$$G(x) \le G(0)\left(\frac{e^{\frac{aL}{\nu}} - e^{\frac{ax}{\nu}}}{e^{\frac{aL}{\nu}} - 1}\right) + G(L)\left(\frac{e^{\frac{ax}{\nu}} - 1}{e^{\frac{aL}{\nu}} - 1}\right) \tag{42}$$

Here b is not constant because this lemma will be useful also in the next section.
proof The proof is based on the maximum principle. Let us compute the derivatives of G:

$$G' = 2\int_{\mathbf{R}} u\frac{\partial u}{\partial x}\,dy \text{ and } G'' = 2\int_{\mathbf{R}} u\frac{\partial^2 u}{\partial x^2} + (\frac{\partial u}{\partial x})^2\,dy$$

Hence,

$$\begin{aligned} \frac{G}{\Delta t} + \frac{a}{2}G' - \frac{\nu}{2}G'' &= \textstyle\int_{\mathbf{R}} u\,(\frac{u}{\Delta t} + a\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2})\,dy - \nu\int_{\mathbf{R}}(\frac{\partial u}{\partial x})^2\,dy \\ &= \textstyle\int_{\mathbf{R}} u\,(-b\frac{\partial u}{\partial y} + \nu\frac{\partial^2 u}{\partial y^2})\,dy - \nu\int_{\mathbf{R}}(\frac{\partial u}{\partial x})^2\,dy \end{aligned}$$

Integrating by parts we obtain

$$G(\frac{1}{\Delta t} - \frac{1}{2}\sup(b_y)) + \frac{a}{2}G' - \frac{\nu}{2}G'' \le 0 \tag{43}$$

We introduce H solution of

$$\begin{cases} aH' - \nu H'' = 0 \\ H(0) = G(0) \\ H(L) = G(L) \end{cases}$$

We have

$$H(x) = G(0)\left(\frac{e^{\frac{aL}{\nu}} - e^{\frac{ax}{\nu}}}{e^{\frac{aL}{\nu}} - 1}\right) + G(L)\left(\frac{e^{\frac{ax}{\nu}} - 1}{e^{\frac{aL}{\nu}} - 1}\right).$$

Then, we have that $G(x) \leq H(x)$. Indeed,

$$\begin{aligned}(G-H)(\tfrac{1}{\Delta t}-\tfrac{1}{2}\sup b_y)+\tfrac{a}{2}(G-H)'-\tfrac{\nu}{2}(G-H)'' = \\ \leq -(H(\tfrac{1}{\Delta t}-\tfrac{1}{2}\sup b_y)+\tfrac{a}{2}H'-\tfrac{\nu}{2}H'')\leq 0\end{aligned} \quad (44)$$

By maximum principle we conclude that $G(x) \leq H(x)$ and lemma 4.3 is proved.
We are now able to prove the geometric convergence. The equations are linear and we may take $g_+ = g_- = 0$ and $f = 0$. We have to prove the convergence to zero.

Theorem 4.4 *Let a, $b \in \mathbf{R}$, ν, $\Delta t > 0$, $g_+ = g_- = 0$ and $f = 0$. The subdomains overlap with an overlap of size δ ($l_{i+1} - L_i = \delta, 1 \leq i \leq N-1$). Then, algorithm (34) with j =0, 1 or 2 has a geometric convergence in the following sense:*
let $g = \frac{e^{\frac{a\delta}{\nu}}-1}{e^{\frac{aL}{\nu}}-1}$ and $\mathcal{C}^n = \sup_{0\leq i\leq N-2}(\sum_{m=1}^{N} A_{l_m}^{n-i,m})$, we have for $n \geq N-1$

$$\mathcal{C}^{n+N-1} \leq (1-g^{N-1})\mathcal{C}^n$$

proof The estimate of theorem 4.1 gives:

$$E^{n,m} + A_{l_m}^{n,m} + B_{L_m}^{n,m} \leq A_{L_m}^{n,m} + B_{l_m}^{n,m} \quad (45)$$

By definition of the algorithm we have,

$$A_{L_m}^{n,m} = A_{L_m}^{n-1,m+1} \quad \text{and} \quad B_{l_m}^{n,m} = B_{l_m}^{n,m-1} \quad (46)$$

Let $w_{i,j}^n = (\partial_x - \Lambda_j^-)(u_j^n)$ and $z_{i,j}^n = (\partial_x - \Lambda_j^+)(u_j^n)$. Since a and b are constants, $w_{i,j}^n$ and $z_{i,j}^n$ satisfy $\mathcal{L}(w_{i,j}^n) = 0$ and $\mathcal{L}(z_{i,j}^n) = 0$. Hence, we can apply lemma 4.3 to these quantities and we obtain

$$\begin{aligned}A_{L_m}^{n-1,m+1} &\leq (1-g)\,A_{l_{m+1}}^{n-1,m+1} + g\,A_{L_{m+1}}^{n-1,m+1}, \quad 1\leq m\leq N-1\\ B_{l_m}^{n,m-1} &\leq (1-h)\,B_{L_{m-1}}^{n,m-1} + h\,B_{l_{m-1}}^{n,m-1} \quad 2\leq m\leq N\end{aligned} \quad (47)$$

where $h = \frac{e^{\frac{aL}{\nu}}-e^{\frac{a(L-\delta)}{\nu}}}{e^{\frac{aL}{\nu}}-1}$ and g is as defined in theorem 4.4. The boundary conditions of the initial problem give

$$A_{L_N}^{n,N} = 0 \quad \text{and} \quad B_{l_1}^{n,1} = 0 \quad (48)$$

As for theorem 4.2 we define

$$\mathcal{E}^n = \sum_{m=1}^{N} E^{n,m} \quad \text{and} \quad \mathcal{A}^n = \sum_{m=1}^{N} A_{l_m}^{n,m}$$

From relations (45), (46), (47) and (48) we get the following estimate for $0 \leq j \leq N-2$,

$$\begin{aligned}\mathcal{E}^n + \mathcal{A}^n + \sum_{i=0}^{j} h^i B_{L_{N-i}}^{n,N-i} \leq (1-g)\sum_{i=1}^{j+1}\mathcal{A}^{n-i}g^{i-1} + g^{j+1}\sum_{m=j+2}^{N-1} A_{L_m}^{n-2-j,m+1}\\ -h^{j+1}\sum_{m=1}^{N-1-j} B_{L_m}^{n,m} + h^{j+1}\sum_{m=2}^{N-1-j} B_{l_m}^{n,m-1}\end{aligned} \quad (49)$$

We prove (49) by induction starting with $j=0$. We sum (45) over m and use (48) to obtain:

$$\mathcal{E}^n + \mathcal{A}^n + \sum_{m=1}^{N} B_{L_m}^{n,m} \leq \sum_{m=1}^{N-1} A_{L_m}^{n,m} + \sum_{m=2}^{N} B_{l_m}^{n,m}$$

By (46) we have:

$$\mathcal{E}^n + \mathcal{A}^n + \sum_{m=1}^{N} B_{L_m}^{n,m} \leq \sum_{m=1}^{N-1} A_{L_m}^{n-1,m+1} + \sum_{m=2}^{N} B_{l_m}^{n,m-1}$$

Relation (47) yields:

$$\mathcal{E}^n + \mathcal{A}^n + \sum_{m=1}^{N} B_{L_m}^{n,m} \leq \sum_{m=1}^{N-1} (1-g) A_{l_{m+1}}^{n-1,m+1} + g A_{L_{m+1}}^{n-1,m+1} + \sum_{m=2}^{N} (1-h) B_{L_{m-1}}^{n,m-1} + h B_{l_{m-1}}^{n,m-1}$$

After simplification and use of (48) we have,

$$\mathcal{E}^n + \mathcal{A}^n + B_{L_N}^{n,N} \leq (1-g)\mathcal{A}^{n-1} + g \sum_{m=2}^{N-1} A_{L_m}^{n-1,m} - h \sum_{m=1}^{N-1} B_{L_m}^{n,m} + h \sum_{m=2}^{N-1} B_{l_m}^{n,m}$$

By (46) we obtain the desired estimate,

$$\mathcal{E}^n + \mathcal{A}^n + B_{L_N}^{n,N} \leq (1-g)\mathcal{A}^{n-1} + g \sum_{m=2}^{N-1} A_{L_m}^{n-2,m+1} - h \sum_{m=1}^{N-1} B_{L_m}^{n,m} + h \sum_{m=2}^{N-1} B_{l_m}^{n,m-1}$$

We suppose now that (49) holds for some j. We prove that (49) holds for $j+1$. We use (47) in (49) to obtain

$$\begin{aligned}\mathcal{E}^n + \mathcal{A}^n + \sum_{i=0}^{j} h^i B_{L_{N-i}}^{n,N-i} \leq (1-g) \sum_{i=1}^{j+1} \mathcal{A}^{n-i} g^{i-1} + g^{j+1} \sum_{m=j+2}^{N-1} (1-g) A_{l_{m+1}}^{n-2-j,m+1} + g A_{L_{m+1}}^{n-2-j,m+1} \\ -h^{j+1} \sum_{m=1}^{N-1-j} B_{L_m}^{n,m} + h^{j+1} \sum_{m=2}^{N-1-j} (1-h) B_{L_{m-1}}^{n,m-1} + h B_{l_{m-1}}^{n,m-1}\end{aligned}$$

After simplification and use of (48) we obtain:

$$\begin{aligned}\mathcal{E}^n + \mathcal{A}^n + \sum_{i=0}^{j} h^i B_{L_{N-i}}^{n,N-i} \leq (1-g) \sum_{i=1}^{j+2} \mathcal{A}^{n-i} g^{i-1} + g^{j+2} \sum_{m=j+3}^{N-1} A_{L_m}^{n-2-j,m} \\ -h^{j+1} B_{L_{N-1-j}}^{n,N-1-j} - h^{j+2} \sum_{m=1}^{N-2-j} B_{L_m}^{n,m} + h^{j+2} \sum_{m=2}^{N-2-j} B_{l_m}^{n,m}\end{aligned}$$

By using (46), we have:

$$\begin{aligned}\mathcal{E}^n + \mathcal{A}^n + \sum_{i=0}^{j+1} h^i B_{L_{N-i}}^{n,N-i} \leq (1-g) \sum_{i=1}^{j+2} \mathcal{A}^{n-i} g^{i-1} + g^{j+2} \sum_{m=j+3}^{N-1} A_{L_m}^{n-3-j,m+1} \\ -h^{j+2} \sum_{m=1}^{N-2-j} B_{L_m}^{n,m} + h^{j+2} \sum_{m=2}^{N-2-j} B_{l_m}^{n,m-1}\end{aligned}$$

which is (49) for $j+1$.
proof of geometric convergence
We make $j = N-2$ in (49) to obtain:

$$\mathcal{E}^n + \mathcal{A}^n + \sum_{i=0}^{N-1} h^i B^{n,N-i}_{L_{N-i}} \le (1-g) \sum_{i=1}^{N-1} \mathcal{A}^{n-i} g^{i-1} \tag{50}$$

From this relation we shall prove the geometric convergence of the algorithm. Let us define $\mathcal{C}^n = \sup_{0\le i\le N-2} \mathcal{A}^{n-i}$. We first prove that for $n \ge N-1$, $\mathcal{C}^n \le \mathcal{C}^{n-1}$. Indeed, estimate (50) yields:

$$\begin{aligned}\mathcal{A}^n &\le (1-g)\textstyle\sum_{i=1}^{N-1} g^{i-1}\, \mathcal{C}^{n-1} \\ &\le (1-g^{N-1})\mathcal{C}^{n-1} \le \mathcal{C}^{n-1}\end{aligned}$$

and thus $\mathcal{C}^n \le \mathcal{C}^{n-1}$, since we have obviously $\mathcal{A}^{n-i} \le \mathcal{C}^{n-1}$ for $1 \le i \le N-1$.
Next we prove that

$$\mathcal{C}^{n-1+\alpha(N-1)} \le (1-g^{N-1})^\alpha \mathcal{C}^{n-1}$$

Indeed, from (50) we have for $j \ge 0$,

$$\mathcal{A}^{n+j} \le (1-g^{N-1})\,\mathcal{C}^{n-1+j} \le (1-g^{N-1})\,\mathcal{C}^{n-1}$$

Thus,

$$\mathcal{C}^{n+N-2} \le (1-g^{N-1})\mathcal{C}^{n-1}$$

Finally,

$$\mathcal{C}^{n-1+\alpha(N-1)} \le (1-g^{N-1})\mathcal{C}^{n-1+(\alpha-1)(N-1)} \le (1-g^{N-1})^\alpha \mathcal{C}^{n-1}$$

4.3 Variable Coefficients

We consider in this section only the case where the approximations of order zero are used. We shall prove the convergence when there is no overlap and the geometric convergence in the case where a is a constant and b depends only on y. The proofs are based on the analogue to estimate (39):

Lemma 4.5 *Let $L > 0$ and $\Omega =]0, L[\times \mathbf{R}$. Let a, $b \in C^1_b(\bar{\Omega})$ and $u \in H^2(\Omega)$ satisfy*

$$\mathcal{L}(u) = 0 \quad in\ \Omega \tag{51}$$

Then, we have the following estimate:

$$\begin{aligned}&\int\!\!\int u^2\left(\tfrac{1}{\Delta t} - \tfrac{1}{2}\mathrm{div}(a,b)\right) + \nu(\tfrac{\partial u}{\partial x})^2 + \nu(\tfrac{\partial u}{\partial y})^2 \\ &+ \int_{x=L} \frac{\nu^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}}\Big(\frac{\partial u}{\partial x} - \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu}u\Big)^2 + \int_{x=0} \frac{\nu^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}}\Big(\frac{\partial u}{\partial x} - \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu}u\Big)^2 = \\ &\int_{x=0} \frac{\nu^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}}\Big(\frac{\partial u}{\partial x} - \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu}u\Big)^2 + \int_{x=L} \frac{\nu^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}}\Big(\frac{\partial u}{\partial x} - \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu}u\Big)^2\end{aligned} \tag{52}$$

proof. Multiplying $\mathcal{L}(u) = 0$ by

$$u = \frac{\nu}{\sqrt{a^2 + \frac{4\nu}{\Delta t}}}((\partial_x - \Lambda_0^-)(u) - (\partial_x - \Lambda_0^+)(u))$$

(cf. lemma 4.1) and integrating by parts over Ω yields,

$$\begin{aligned} &\int\int u^2\left(\tfrac{1}{\Delta t} - \tfrac{1}{2}\mathrm{div}(a,b)\right) + \nu(\tfrac{\partial u}{\partial x})^2 + \nu(\tfrac{\partial u}{\partial y})^2 \\ &+ \int_{-\infty}^{\infty}\left[\tfrac{a}{2}u^2(.,y) - \nu u \tfrac{\partial u}{\partial x}\right]_0^L dy = 0 \end{aligned}$$

The boundary term can be rewritten in the form:

$$\left[\frac{a}{2}u^2(.,y) - \nu u\frac{\partial u}{\partial x}\right]_0^L = \left[\frac{\nu^2}{2\sqrt{a^2+\frac{4\nu}{\Delta t}}}\left((\frac{\partial u}{\partial x} - \frac{a+\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu}u)^2 - (\frac{\partial u}{\partial x} - \frac{a-\sqrt{a^2+\frac{4\nu}{\Delta t}}}{2\nu}u)^2\right)\right]_0^L$$

Lemma 4.5 enables to prove the

Theorem 4.6 *Let a, $b \in C_b^1(\bar{\Omega})$, if the following condition is satisfied:*
i) $\frac{1}{\Delta t} > \frac{1}{2}\sup \mathrm{div}(a,b)$,
the algorithm (34) with $j = 0$ converges in the sense that
$\lim_{n\to\infty} \| u_i - u_i^n \|_{H^1} \longrightarrow 0$, *for $i = 1, \ldots, N$.*

The proof is similar to the proof of theorem 4.2 and is not written here.

We suppose now that the subdomains Ω_i have an overlap of size $\delta > 0$ (i.e. $L_i - l_{i+1} = \delta$, for $1 \le i \le N-1$).

Theorem 4.7 *Let $a \in \mathbf{R}$, $b = b(y) \in C_b^1(\mathbf{R})$, ν, $\Delta t > 0$, $g_+ = g_- = 0$ and $f = 0$. The subdomains overlap with an overlap of size δ ($l_{i+1} - L_i = \delta, 1 \le i \le N-1$). If $\frac{1}{\Delta t} > \frac{1}{2}\partial_y(b)$, algorithm (34) with j =0 has a geometric convergence in the following sense: Let $g = \frac{e^{\frac{a\delta}{\nu}}-1}{e^{\frac{aL}{\nu}}-1}$ and $\mathcal{C}^n = \sup_{0\le i\le N-2}(\sum_{m=1}^{N} A_{l_m}^{n-i,m})$, we have for $n \ge N-1$*

$$\mathcal{C}^{n+N-1} \le (1 - g^{N-1})\mathcal{C}^n$$

proof The operators $\mathcal{L}$ and $\partial_x - \frac{a\pm\sqrt{a^2+\frac{4\nu}{\Delta t}}}{\nu}$ commute and it is thus possible to apply lemma 4.3 to the quantities $(\partial_x - \frac{a\pm\sqrt{a^2+\frac{4\nu}{\Delta t}}}{\nu})(u)$. We have the same inequalites than in equation (45). Together with (52) this enables to prove the theorem with the same proof than for theorem 4.4.

5 Implementation and Numerical Results

For the numerical implementation of algorithms (32)-(33) or (34) it is possible to discretize independently the operator $\mathcal{L}$ and the boundary conditions (29), (30) and (31).

The method was designed and studied at the continuous level. It is thus natural to estimate that the conclusions derived for continuous equations should remain valid for any consistent discretization. This is sensible as long as the mesh sizes Δx and Δy are small enough. Nevertheless, for a given computation, it is possible to optimize the discretization of the boundary conditions by taking into account the finite value of the mesh size. For instance, in our computations, the operator $\mathcal{L}$ was discretized by a standard upwind finite difference scheme. To find the discret equivalent of operators $(\partial_x - \Lambda^-_{0,1 or 2})$, we follow the same strategy than in [5]. To ease the computations, we take $b = 0$, $a = C^t$, a uniform grid and we introduce some notations:
Let $U = (u_{i,j})$ be the matrix of the unknowns. We consider the following difference operators:

$$(D_x^+ U)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}, \ (D_x^- U)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

and similar definitions for D_y^+ and D_y^-. The discretization of $\mathcal{L}$ by the upwind scheme is thus

$$\mathcal{L}_{dis} = \frac{1}{\Delta t} + aD_x^- - \nu D_x^+ D_x^- - \nu D_y^+ D_y^-$$

First, we seek the discrete analogue to $(\partial_x - \Lambda^-)$. Suppose we have discretized the right half plane and we want to find $U = (u_{i,j})_{\begin{pmatrix} i \geq 0 \\ j \in \mathbf{Z} \end{pmatrix}}$ such that

$$\mathcal{L}_{dis}(U) = 0, \quad i \geq 0, \, j \in \mathbf{Z} \tag{53}$$

$$u_{0,j} = u_j^0, \ j \in \mathbf{Z} \text{ and } u_{i,j} \text{ is bounded as } i \to \infty \tag{54}$$

for some given vector $U^0 = (u_j^0)_{j \in \mathbf{Z}}$. The operator $(\partial_x - \Lambda^-)$ is used as a boundary condition on the right of a subdomain. It is thus natural to discretize ∂_x by the operator D_x^-. The analogue to the operator Λ^- is thus the discret operator Λ^-_{dis}:

$$(u_j^0)_{j \in \mathbf{Z}} \longrightarrow (D_x^- U)_{i=0, j \in \mathbf{Z}}$$

To solve (53) we separate variables. We write

$$u_j^0 = \sum_k \hat{u}_k^0 e^{Ij\Delta x k}$$

where I is the square root of -1. The solution is sought in the form

$$u_{i,j} = \sum_{k \in \mathbf{Z}} \alpha_k \ e^{Ikj\Delta y} \ e^{z(k) i \Delta x}$$

By inserting this expression in (53), we get the dispersion relation:

$$\frac{1}{\Delta t} + a\frac{1 - e^{-z\Delta x}}{\Delta x} - \nu \frac{e^{z\Delta x} - 2 + e^{-z\Delta x}}{\Delta x^2} - \nu \frac{e^{Ik\Delta y} - 2 + e^{-Ik\Delta y}}{\Delta y^2} \tag{55}$$

For a given k, it may be put in the form of a second order equation in $e^{-z\Delta x}$. By analogy to the continuous case, we must have two solutions z^+ and z^- with $Re(z^+) \geq 0$ and $Re(z^-) \leq 0$. The solution of (53) is thus:

$$u_{i,j} = \sum_{k\in\mathbf{Z}} \hat{u}_k^0 \; e^{Ikj\Delta y} \; e^{z^-(k)i\Delta x}$$

and

$$\Lambda_{dis}^-(U^0) = (D_x^- U)_{i=0,j\in\mathbf{Z}} = \sum_{k\in\mathbf{Z}} \hat{u}_k^0 \; e^{Ikj\Delta y} \; \frac{1-e^{-z^-(k)\Delta x}}{\Delta x}$$

The symbol of Λ_{dis}^- is thus

$$\lambda_{dis}^- = \frac{1-e^{-z^-(k)\Delta x}}{\Delta x}$$

By equation (55), λ_{dis}^- satisfies

$$\frac{1}{\Delta t} + a\lambda_{dis}^- - \nu\frac{-\Delta x\lambda_{dis}^- - 1 + \frac{1}{1-\Delta x\lambda_{dis}^-}}{\Delta x^2} + \nu h$$

where $h = \frac{e^{Ik\Delta y}-2+e^{-Ik\Delta y}}{\Delta y^2}$. Thus, we get

$$\lambda_{dis}^- = \frac{a - \frac{\Delta x}{\Delta t} - \nu h\Delta x - \sqrt{(a+\frac{\Delta x}{\Delta t}+\nu h\Delta x)^2 + 4\nu^2 h + \frac{4\nu}{\Delta t}}}{2(\nu + a\Delta x)}$$

As in the continuous case, the finite difference operator Λ_{dis}^- is not local. To obtain local approximations, we approximate its symbol with respect to h. At order zero we have,

$$\lambda_{dis}^- \simeq \lambda_{dis,0}^- = \frac{a - \frac{\Delta x}{\Delta t} - \sqrt{(a+\frac{\Delta x}{\Delta t})^2 + \frac{4\nu}{\Delta t}}}{2(\nu + a\Delta x)}$$

and at order 1 with respect to h (i.e. order 2 with respect to k),

$$\lambda_{dis,2}^- = \frac{a - \frac{\Delta x}{\Delta t} - \nu h\Delta x - \sqrt{(a+\frac{\Delta x}{\Delta t})^2 + \frac{4\nu}{\Delta t}}\left(1 + \frac{a\nu\Delta x h + \frac{\nu\Delta x^2 h}{\Delta t} + 2\nu^2 h}{(a+\frac{\Delta x}{\Delta t})^2 + \frac{4\nu}{\Delta t}}\right)}{2(\nu + a\Delta x)}$$

The discret analogous to $(\partial_x - \Lambda_0^-)$ is thus

$$D_x^- - \frac{a - \frac{\Delta x}{\Delta t} - \sqrt{(a+\frac{\Delta x}{\Delta t})^2 + \frac{4\nu}{\Delta t}}}{2(\nu + a\Delta x)} \tag{56}$$

and the analogue to the operator $(\partial_x - \Lambda_2^-)$ is

$$D_x^- - \frac{a - \frac{\Delta x}{\Delta t} - \sqrt{(a+\frac{\Delta x}{\Delta t})^2 + \frac{4\nu}{\Delta t}}}{2(\nu + a\Delta x)} - \nu\left(\frac{\Delta x}{2(\nu + a\Delta x)} + \frac{a\Delta x + \frac{\Delta x^2}{\Delta t} + 2\nu}{2(\nu + a\Delta x)\sqrt{(a+\frac{\Delta x}{\Delta t})^2 + \frac{4\nu}{\Delta t}}}\right) D_y^+ D_y^- \tag{57}$$

The same work can be done to find the analogue to $(\partial_x - \Lambda^+_{0\,or\,2})$.

A direct discretization of $(\partial_x - \Lambda^-_{0\,or\,2})$ would consist in making $\Delta x = 0$ in (56) or (57), i.e. for $(\partial_x - \Lambda^-_0)$

$$D^-_x - \frac{a - \sqrt{a^2 + \frac{4\nu}{\Delta t}}}{2\nu} \tag{58}$$

and for $(\partial_x - \Lambda^-_2)$

$$D^-_x - \frac{a - \sqrt{a^2 + \frac{4\nu}{\Delta t}}}{2\nu} - \frac{\nu}{\sqrt{a^2 + \frac{4\nu}{\Delta t}}} D^+_y D^-_y \tag{59}$$

We compared the use of (58) or (59) and of (56) or (57). For some set of parameters, we observed that the convergence was twice as fast with (56) or (57) as with (58) or (59). Of course, for Δx small it does not make change. Boundary conditions (56) or (57) have always led to more efficient algorithms and have been used in the sequel.

In order to illustrate the validity of the method, a 2-D test problem has been performed. The convection-diffusion equation (5) was discretized in space by a standard finite difference upwind scheme. Double precision arithmetic was used. The computational domain is the unit square. On the left and on the bottom boundaries, we used Dirichlet boundary conditions.

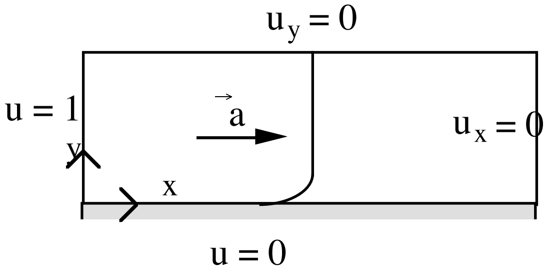

FIGURE 2 - Computational domain

On the other boundaries, homogeneous Neumann boundary conditions were imposed. In addition to the 6 algorithms (6=2 (one sweep/double sweep) $\times$ 3(OBC of order 0,1 or 2)), we also considered other algorithms. They are obtained simply by using other boundary conditions on the boundaries of the subdomains. We considered two other possibilities: firstly, the use of Dirichlet boundary conditions and secondly, the use of "characteristic" boundary conditions. In the first case, the method is a standard Block Gauss-Seidel algorithm. In the second case, we use at inflow a Dirichlet BC and at outflow the boundary operator $\frac{1}{\Delta t} + a\partial_x + b\partial_y$. These BC may be used in a double-sweep or a one sweep (in the direction of positive x) algorithm. We have thus four more algorithms. The ten different algorithms will be denoted as indicated in the following table:

Boundary Cond.	*one sweep*	*double sweep*
OBC of order 0	mobc0os	mobc0ds
OBC of order 1	mobc1os	mobc1ds
OBC of order 2	mobc2os	mobc2ds
Dir/Dir	mddos	mddds
Dir/Char	mdcos	mdcds

Table 2: abbreviated names of the algorithms

The width of a subdomain is denoted npb. The size of the overlap in terms of mesh size is denoted by $nrec$ (see fig. 3). The non overlapping case ($nrec = 0$) could not be considered here since the discretization of the normal derivatives in x have to be done on the same grid points to have an algorithm consistent with our discretization of (5).

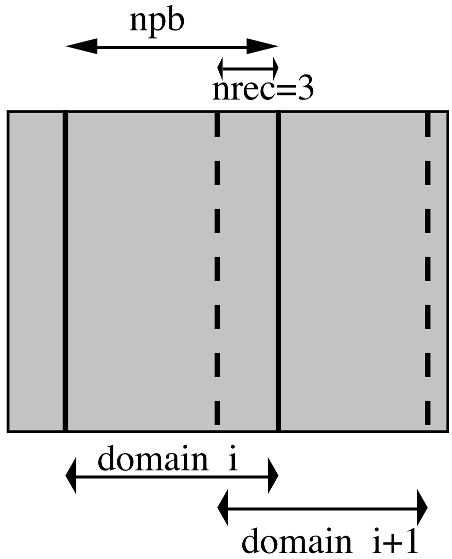

FIGURE 3 - Grid

In order to avoid difficulties in the choice of a stopping criterion, the following method was adopted. We first compute the solution of the problem with a boundary layer (see figure 2). This solution is then used as an initial guess for the solving of the homogeneous equation (whose solution is zero) by the various algorithms. The solution is considered to be close enough to zero when the maximum value of the solution is lower than 10^{-5}. Since the equations are linear, it is equivalent to the solving of the non homogeneous problem with zero everywhere as initial guess.

In tables 3, 4 and 5 we have indicated for different velocity fields (a, b) the number of sweeps. In order to have a fair comparison, the number of sweeps is reported in the tables (i.e. one double sweep counts for two sweeps). The term div means that divergence has occured. A number between brackets means that the convergence was very slow and this number is the maximum value of the solution when the algorithm was stopped. We limited the number of sweeps to 600 (i.e. 300 iterations of the algorithm) for double sweeps methods and to 300 for one-sweep methods.

In all these computations (except for table 4) we took fixed geometric parameters: $nx = 80$, $ny = 30$, $npb = 10$, $nrec = 2$

	$\nu = 10^{-1}$	$\nu = 10^{-2}$	$\nu = 10^{-3}$	$\nu = 10^{-4}$
mobc2ds	22	6	2	2
mobc1ds	24	6	2	2
mobc0ds	56	12	4	2
mdcds	36	8	2	2
mddds	178	22	8	4
mobc2os	15	2	1	1
mobc1os	19	3	1	1
mobc0os	32	4	2	1
mdcos	32	4	1	2
mddos	144	16	4	2

Table 3: Number of sweeps vs. viscosity ($a = y$ and $b = 1.$), $\Delta t = 2.10^{10}$

	$nx = 80$	$nx = 160$	$nx = 240$	$nx = 320$	$nx = 400$
mobc2ds	2	2	2	4	4
mobc1ds	4	4	6	6	6
mobc0ds	4	4	6	6	6
mdcds	4	4	6	8	8
mddds	3	10	14	24	28
mobc2os	1	1	1	1	1
mobc1os	2	3	4	4	6
mobc0os	2	3	4	4	6
mdcos	3	5	6	8	8
mddos	6	33	38	116	108

Table 4: Number of sweeps vs. nx ($a = 1$ and $b = 0$, $\nu = 0.01$), $\Delta t = 2.10^{10}$

	$\nu = 10^{0}$	$\nu = 10^{-1}$	$\nu = 10^{-2}$	$\nu = 10^{-3}$
mobc2ds	92	18	2	2
mobc1ds	214	34	4	2
mobc0ds	214	34	4	2
mdcds	div	div	10	4
mddds	600 (10^{-4})	600 (10^{-3})	248	4
mobc2os	103	34	15	2
mobc1os	210	52	20	4
mobc0os	210	52	20	4
mdcos	div	89	56	4
mddos	300(10^{-2})	300(10^{-2})	300(0.16)	300(10^{-2})

Table 5: Number of sweeps vs. viscosity ($a = 10 \times (x - \frac{1}{3})(x - \frac{2}{3})$ and $b = 0$), $\Delta t = 2.10^{10}$

From these reults, it appears (as expected from table 1) that the use of Dirichlet boundary conditions leads to very slow convergence rates when the Reynolds mesh number

$\dfrac{|a|\Delta x}{\nu}$ is lower than 1 (cf. also [10]). This is cleraly demonstrated by table 4 where only nx (and thus Δx) is changed. The use of characteristic outflow boundary conditions leads sometimes to fast convergence but also to divergence. This choice of transmission conditions is thus not safe. The use of the OBC leads to faster algorithms. From table 1, one sees that higher order OBC lead to more effecient algorithms. Table 4 shows that the number of sweeps is very stable with respect to the mesh size in the x direction. When there is no reverse flow, one sweep algorithms are slightly superior to double sweeps algorithms. But, in presence of reverse flow (table 5), double sweeps algorithms are far superior to one sweep algorithms. It seems thus preferable to use double sweeps algorithms since they do not need any a priori knowledge of the velocity field.

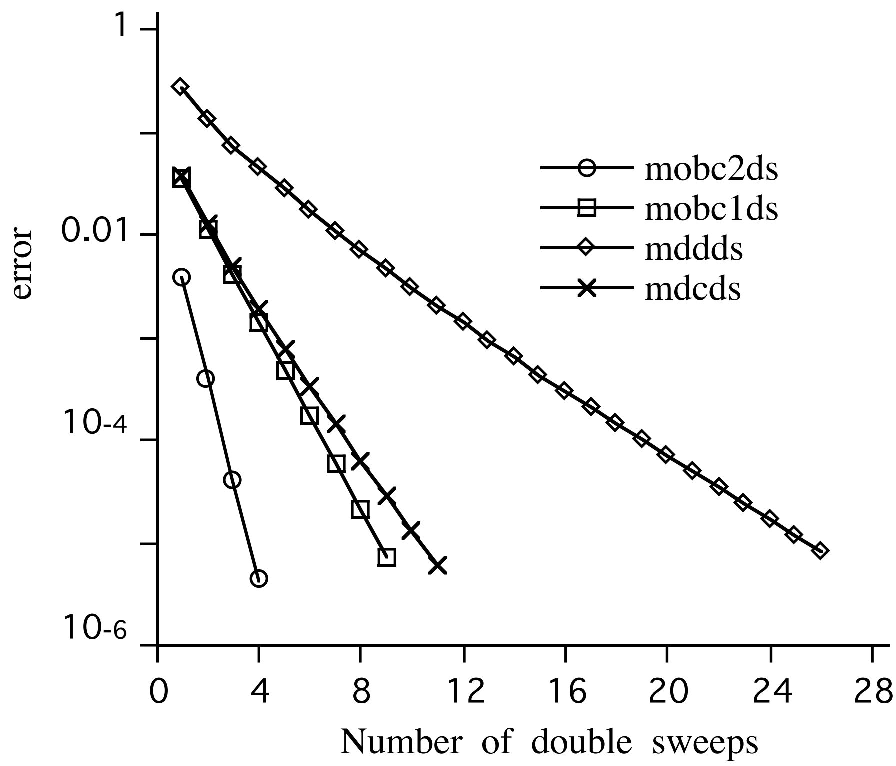

FIGURE 4 - Convergence rate, ($a = y$ and $b = 0$), $\nu = 10^{-2}$, $\Delta t = 2.10^{10}$

6 CONCLUSION

We have considered block Gauss-Seidel algorithms as domain decomposition methods. It is then interesting to compare the use of various boundary conditions as transmission conditions. From the theoretical and numerical results, it appears that the use of open boundary conditions leads to faster convergence rates than the use of Dirichlet boundary conditions (which corresponds to the classical block Gauss-Seidel method). The improvement is significant for low Reynolds mesh numbers.

References

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

[2] —, *Décomposition de domaine et problème de Helmholtz*, C.R. Acad. Sci., Paris, **t. 311, Série I** (1990), 313-316.

[3] B. Despres, P. Joly and J.E. Roberts, International Symposium on Iterative methods in linear algebra, Brussels, Belgium, April 2-4th 1991, 475-484

[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] D'Errico M.A. and Pironneau O., *The Incompressible Navier-Stokes Equations with Extremly Flat Finite Elements in the Boundary Layers*, Int. J. Num. Meth. Heat Fluid Flow, vol. 1, (1991), 171-184.

[7] 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.

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

[9] Han H., Il'in V.P. and Kellog R.B., *Flow Directed Iterations for Convection Dominated Flow*, BAIL V, Boole Press. Conf. Ser., vol 12, 7-17.

[10] Johnson C., *Flow Directed Gauss-Seidel Iterative Methods for Stationnary Convection-Diffusion Problems*, Preprint Dept. of Math. Chalmers Univ. of Technology, 1992:29 (1992).

[11] 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.

[12] J.P. Lohéac, F. Nataf and M. Schatzman, *Parabolic Approximations of the Convection-Diffusion Equation*, to appear in Math. Comp.

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