

## A NEW DOMAIN DECOMPOSITION METHOD FOR THE COMPRESSIBLE EULER EQUATIONS

VICTORITA DOLEAN<sup>1</sup> AND FRÉDÉRIC NATAF<sup>2</sup>

**Abstract.** In this work we design a new domain decomposition method for the Euler equations in 2 dimensions. The starting point is the equivalence with a third order scalar equation to whom we can apply an algorithm inspired from the Robin-Robin preconditioner for the convection-diffusion equation [Achdou and Nataf, *C. R. Acad. Sci. Paris Sér. I* **325** (1997) 1211–1216]. Afterwards we translate it into an algorithm for the initial system and prove that at the continuous level and for a decomposition into 2 sub-domains, it converges in 2 iterations. This property cannot be conserved strictly at discrete level and for arbitrary domain decompositions but we still have numerical results which confirm a very good stability with respect to the various parameters of the problem (mesh size, Mach number, ...).

**Mathematics Subject Classification.** 35M20, 65M55.

Received: March 31, 2005. Revised: May 16, 2006.

### 1. INTRODUCTION

The need of using domain decomposition methods when solving partial differential equations is nowadays more and more obvious. The challenge is now the acceleration of these methods. Different possibilities were studied such as the use of optimized interface conditions on the artificial boundaries between subdomains or the preconditioning of a substructured system defined at the interface. The former were widely studied and analyzed for scalar problems such as elliptic equations in [12, 26], for the Helmholtz equation in [3, 7, 15, 23] convection-diffusion problems in [22]. For time dependent problems and local times steps, see for instance [13, 14]. The preconditioning methods have also known a wide development in the last decade. The Neumann-Neumann algorithms for symmetric second order problems [5, 27, 30] have been the subject of numerous works, see [31] and references therein. An extension of these algorithms to non-symmetric scalar problems (the so called Robin-Robin algorithms) has been done in [2, 20] for advection-diffusion problems.

As far as optimized interface conditions are concerned, when dealing with supersonic flows, whatever the space dimension is, imposing the appropriate characteristic variables as interface conditions leads to a convergence of the algorithm which is optimal with regards to the number of subdomains. This property is generally lost for subsonic flows except for the case of one-dimensional problems, when the optimality is expressed by the fact that the number of iterations is equal to the number of subdomains (see Bjørhus [4] and Quarteroni [28] for more details). In the subsonic case and in two or three dimensions, we can find a formulation with classical (natural)

---

*Keywords and phrases.* Smith factorization, domain decomposition method, Euler equations.

<sup>1</sup> Laboratoire de Mathématiques J.A. Dieudonné, Université de Nice Sophia-Antipolis, Parc Valrose, 06108 Nice Cedex 02, France. [dolean@math.unice.fr](mailto:dolean@math.unice.fr)

<sup>2</sup> Laboratoire J.L. Lions, CNRS UMR 7598, Université Pierre et Marie Curie, Paris 75005, France. [nataf@ann.jussieu.fr](mailto:nataf@ann.jussieu.fr)

transmission conditions in [6, 28, 29] or with more general interface conditions in [8] and optimized transmission conditions in [9]. The analysis of such algorithms applied to systems proved to be very different from the scalar case, see [10, 11]. The generalization of the above domain decomposition methods to the system of the Euler equations is difficult in the subsonic case in dimensions equal or higher to two. As far as preconditioning methods are concerned, to our knowledge, no extension of the Neumann-Neumann, FETI [24] or BDDC [25] methods to the Euler equations was done.

In this paper, we consider a preconditioning technique for the system of the compressible Euler equations in the subsonic case. The paper is organized as follows: in Section 2 we will first show the equivalence between the 2D Euler equations and a third order scalar problem, which is quite natural by considering a Smith factorization of this system, see [32] or [18]. In Section 3 we define an optimal algorithm for the third order scalar equation. It is inspired from the idea of the Robin-Robin algorithm [2] applied to a convection-diffusion problem. We also prove by using a Fourier analysis that this algorithm converges in two iterations. Afterwards in Section 4 we back-transform it and define the corresponding algorithm applied to the Euler system at the continuous level and for a general decomposition. In Section 5, we consider the discrete formulation of the algorithm. In the Section 6, numerical results confirm the very good stability of the algorithm with respect to the various parameters of the problem (mesh size, Mach number, ...).

## 2. A THIRD ORDER SCALAR PROBLEM

In this section we will show the equivalence between the linearized and time discretized Euler system and a third order scalar equation. The motivation for this transformation is that a new algorithm is easier to design for a scalar equation than for a system of partial differential equations.

### 2.1. The compressible 2D Euler equations

In the following we will focus ourselves on the conservative Euler equations in two-dimensions:

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathbf{F}(W) = 0 \quad , \quad W = (\rho, \rho \mathbf{V}, E)^t. \quad (2.1)$$

In the above expressions,  $\rho$  is the density,  $\mathbf{V} = (u, v)^t$  is the velocity vector,  $E$  is the total energy per unit of volume and  $p$  is the pressure. In equation (2.1),  $W = W(\mathbf{x}, t)$  is the vector of conservative variables,  $\mathbf{x}$  and  $t$  respectively denote the space and time variables and  $\mathbf{F}(W) = (F_1(W), F_2(W))^T$  is the conservative flux vector whose components are given by

$$F_1(W) = (\rho u, \rho u^2 + p, \rho uv, u(E + p))^t, \quad F_2(W) = (\rho v, \rho uv, \rho v^2 + p, v(E + p))^t.$$

The pressure is deduced from the other variables using the state equation for a perfect gas  $p = (\gamma_s - 1)(E - \frac{1}{2}\rho \|\mathbf{V}\|^2)$  where  $\gamma_s$  is the ratio of the specific heats ( $\gamma_s = 1.4$  for the air).

### 2.2. Equivalence of the Euler system to a scalar equation

The starting point of our analysis is given by the linearized form of the Euler equations (2.1) written in primitive variables  $(p, u, v, S)$ . In the following we suppose that the flow is isentropic, which allows us to drop the equation of the entropy (which is totally decoupled from the others). We denote by  $W = (P, U, V)^T$  the vector of unknowns and by  $A$  and  $B$  the Jacobian matrices of the fluxes  $F_i(w)$  to whom we already applied the variable change from conservative to primitive variables. In the following, we shall denote by  $\bar{c}$  the speed of the sound and we consider the linearized form (we will mark by the bar symbol, the state around which we linearize) of the Euler equations:

$$\mathcal{P}W \equiv \frac{W}{\Delta t} + A\partial_x W + B\partial_y W = f \quad (2.2)$$

characterized by the following Jacobian matrices:

$$A = \begin{pmatrix} \bar{u} & \bar{\rho}\bar{c}^2 & 0 \\ 1/\bar{\rho} & \bar{u} & 0 \\ 0 & 0 & \bar{u} \end{pmatrix} \quad B = \begin{pmatrix} \bar{v} & 0 & \bar{\rho}\bar{c}^2 \\ 0 & \bar{v} & 0 \\ 1/\bar{\rho} & 0 & \bar{v} \end{pmatrix}. \tag{2.3}$$

We can re-write the system (2.2) by denoting  $\beta = \frac{1}{\Delta t} > 0$  under the form

$$\mathcal{P}W \equiv (\beta I + A\partial_x + B\partial_y)W = f. \tag{2.4}$$

In Computational Fluid Dynamics, problems of the form (2.4) have to be solved repeatedly. We shall design a new domain decomposition method for this purpose. We build and analyze our method for the constant coefficient case ( $\bar{c}$ ,  $\bar{u}$ ,  $\bar{v}$  and  $\bar{\rho}$  are constants) and for only two subdomains. But the resulting algorithm can be applied to the general case of variable flows and arbitrary number of subdomains, see Section 4 for the formulation of the algorithm and Section 6 for numerical results.

2.2.1. *Smith factorization*

We first recall the definition of the Smith factorization of a matrix with polynomial entries and apply it to systems of PDEs, see [16, 17, 19] or [32] and references therein.

**Theorem 1.** *Let  $n$  be an integer and  $C$  an invertible  $n \times n$  matrix with polynomial entries in the variable  $\lambda$ :  $C = (c_{ij}(\lambda))_{1 \leq i, j \leq n}$ .*

*Then, there exists three matrices with polynomial entries  $E$ ,  $D$  and  $F$  with the following properties:*

- *$\det(E)$  and  $\det(F)$  are real numbers;*
- *$D$  is a diagonal matrix;*
- *$C = EDF$ .*

*Moreover,  $D$  is uniquely defined up to a reordering and multiplication of each entry by a constant by a formula defined as follows. Let  $1 \leq k \leq n$ ,*

- *$S_k$  is the set of all the submatrices of order  $k \times k$  extracted from  $C$ ;*
- *$Det_k = \{Det(B_k) \mid B_k \in S_k\}$ ;*
- *$LD_k$  is the greatest common divisor of the set of polynomials  $Det_k$ .*

*Then,*

$$D_{kk}(\lambda) = \frac{LD_k(\lambda)}{LD_{k-1}(\lambda)}, \quad 1 \leq k \leq n \tag{2.5}$$

*(by convention,  $LD_0 = 1$ ).*

**Application to the Euler system.** We first take formally the Fourier transform of the system (2.4) with respect to  $y$  (the dual variable is  $\xi$ ). We keep the partial derivatives in  $x$  since in the sequel we shall consider a domain decomposition with an interface whose normal is in the  $x$  direction. We note

$$\hat{\mathcal{P}} = \begin{pmatrix} \beta + \bar{u}\partial_x + i\xi\bar{v} & \bar{\rho}\bar{c}^2\partial_x & i\bar{\rho}\bar{c}^2\xi \\ \frac{1}{\bar{\rho}}\partial_x & \beta + \bar{u}\partial_x + i\xi\bar{v} & 0 \\ \frac{i\xi}{\bar{\rho}} & 0 & \beta + \bar{u}\partial_x + i\bar{v}\xi \end{pmatrix}. \tag{2.6}$$

We can perform a Smith factorization of  $\hat{\mathcal{P}}$  by considering it as a matrix with polynomials in  $\partial_x$  entries. We have

$$\hat{\mathcal{P}} = EDF \tag{2.7}$$

where

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \hat{\mathcal{L}}\hat{\mathcal{G}} \end{pmatrix} \tag{2.8}$$

and

$$E = \begin{pmatrix} i\bar{\rho}\bar{c}^2\xi & 0 & 0 \\ 0 & \bar{u} & 0 \\ \beta + \bar{u}\partial_x + i\bar{v}\xi & E_2 & \frac{\bar{c}^2 - \bar{u}^2}{i\xi\bar{\rho}\bar{c}^2} \end{pmatrix}$$

and

$$F = \begin{pmatrix} \frac{\beta + \bar{u}\partial_x + i\xi\bar{v}}{i\xi\bar{\rho}\bar{c}^2} & \frac{\partial_x}{i\xi} & 1 \\ \frac{\partial_x}{\bar{\rho}\bar{u}} & \frac{\beta + \bar{u}\partial_x + i\xi\bar{v}}{\bar{u}} & 0 \\ \frac{1}{(\beta + i\xi\bar{v})(\bar{u}^2 - \bar{c}^2)} & \frac{\bar{u}}{\bar{\rho}\bar{u}} & 0 \end{pmatrix} \tag{2.9}$$

where

$$E_2 = \bar{u} \frac{(-\bar{u}\bar{c}^2 + \bar{u}^3)\partial_{xx} + (2\bar{u}^2 - \bar{c}^2)(\beta + i\xi\bar{v})\partial_x + \bar{u}((\beta + i\xi\bar{v})^2 + \xi^2\bar{c}^2)}{\bar{c}^2(i\beta + i\xi\bar{v})},$$

$$\hat{\mathcal{G}} = \beta + \bar{u}\partial_x + i\xi\bar{v} \tag{2.10}$$

and

$$\hat{\mathcal{L}} = \beta^2 + 2i\xi\bar{u}\bar{v}\partial_x + 2\beta(\bar{u}\partial_x + i\xi\bar{v}) + (\bar{c}^2 - \bar{v}^2)\xi^2 - (\bar{c}^2 - \bar{u}^2)\partial_{xx}. \tag{2.11}$$

The operators showing up in the diagonal matrix have a physical meaning:

$$\mathcal{G} = \beta + \bar{u}\partial_x + \bar{v}\partial_y$$

is a first order transport operator where the time derivative is replaced by  $\beta$  and

$$\mathcal{L} = \beta^2 + 2\bar{u}\bar{v}\partial_{xy} + 2\beta(\bar{u}\partial_x + \bar{v}\partial_y) - (\bar{c}^2 - \bar{v}^2)\partial_{yy} - (\bar{c}^2 - \bar{u}^2)\partial_{xx}$$

is the advective wave operator where  $\partial_t^l$  is replaced by  $\beta^l$  for  $l = 1, 2$ . We call  $W_s = FW$  the Smith variables.

Equation (2.8) suggests that the derivation of a domain decomposition method (DDM) for the third order operator  $\mathcal{L}\mathcal{G}$  is a key ingredient for a DDM for the compressible Euler equations.

### 3. A NEW ALGORITHM APPLIED TO A SCALAR THIRD ORDER PROBLEM

In this section we will describe a new algorithm applied to the third order operator found in the Section 2. We want to solve

$$\mathcal{L}\mathcal{G}(Q) = g \tag{3.1}$$

where  $Q$  is scalar unknown function and  $g$  is a given right hand side. The algorithm will be based on the Robin-Robin algorithm [1, 2] for the convection-diffusion problem. Then we will prove its convergence in 2 iterations. We first note that the elliptic operator  $\mathcal{L}$  can also be written as:

$$\mathcal{L} = -\text{div}(A\nabla) + \mathbf{a}\nabla + \beta^2, \quad A = \begin{pmatrix} \bar{c}^2 - \bar{u}^2 & -\bar{u}\bar{v} \\ -\bar{u}\bar{v} & \bar{c}^2 - \bar{v}^2 \end{pmatrix} \quad \text{where } \mathbf{a} = 2\beta(\bar{u}, \bar{v}). \tag{3.2}$$

Without loss of generality we assume in the sequel that the flow is subsonic and that  $\bar{u} > 0$  and thus we have  $0 < \bar{u} < \bar{c}$ .

### 3.1. The algorithm for a two-domain decomposition

We consider now a decomposition of the plane  $\mathbb{R}^2$  into two non-overlapping sub-domains  $\Omega_1 = (-\infty, 0) \times \mathbb{R}$  and  $\Omega_2 = (0, \infty) \times \mathbb{R}$ . The interface is  $\Gamma = \{x = 0\}$ . The outward normal to domain  $\Omega_i$  is denoted  $\mathbf{n}_i$ ,  $i = 1, 2$ . Let  $Q^{i,k}$ ,  $i = 1, 2$  represent the approximation to the solution in subdomain  $i$  at the iteration  $k$  of the algorithm. We define the following algorithm:

**Algorithm 1.** We choose the initial values  $Q^{1,0}$  and  $Q^{2,0}$  such that  $\mathcal{G}Q^{1,0} = \mathcal{G}Q^{2,0}$ . We compute  $(Q^{i,k+1})_{i=1,2}$  from  $(Q^{i,k})_{i=1,2}$  by the following iterative procedure:

**Correction step.** We compute the corrections  $\tilde{Q}^{1,k}$  and  $\tilde{Q}^{2,k}$  as solution of the homogeneous local problems:

$$\begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{1,k} = 0 \text{ in } \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{1,k} \cdot \mathbf{n}_1 = \gamma^k, \text{ on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{2,k} = 0 \text{ in } \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{2,k} \cdot \mathbf{n}_2 = \gamma^k, \text{ on } \Gamma, \\ \tilde{Q}^{2,k} = 0, \text{ on } \Gamma, \end{cases} \quad (3.3)$$

where  $\gamma^k = -\frac{1}{2} [A\nabla\mathcal{G}Q^{1,k} \cdot \mathbf{n}_1 + A\nabla\mathcal{G}Q^{2,k} \cdot \mathbf{n}_2]$ .

**Update step.** We update  $Q^{1,k+1}$  and  $Q^{2,k+1}$  by solving the local problems:

$$\begin{cases} \mathcal{L}\mathcal{G}Q^{1,k+1} = g, \text{ in } \Omega_1, \\ \mathcal{G}Q^{1,k+1} = \mathcal{G}Q^{1,k} + \delta^k, \text{ on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{2,k+1} = g, \text{ in } \Omega_2, \\ \mathcal{G}Q^{2,k+1} = \mathcal{G}Q^{2,k} + \delta^k, \text{ on } \Gamma, \\ Q^{2,k+1} = Q^{1,k} + \tilde{Q}^{1,k}, \text{ on } \Gamma, \end{cases} \quad (3.4)$$

where  $\delta^k = \frac{1}{2} [\mathcal{G}\tilde{Q}^{1,k} + \mathcal{G}\tilde{Q}^{2,k}]$ .

**Proposition 1.** *Algorithm 1 converges in 2 iterations.*

*Proof.* We use the Fourier transform technique. For the sake of the analysis we consider the previous algorithm written in terms of the error vector  $e^{i,k}(x, y) = (Q^{i,k} - Q|_{\Omega_i})(x, y)$ ,  $i = 1, 2$ . The error  $(e^{i,k})_{i=1,2}$  satisfies Algorithm 1 with  $g = 0$ .

We will first describe what happens locally inside each subdomain after proceeding to a Fourier transform in the  $y$  direction and then we prove the convergence of Algorithm 1 by computing in Fourier space the effect of the correction and the update steps. We denote by  $\hat{e}(x, \xi)$  the Fourier transform of a function  $e(x, y)$ :

$$\hat{e}(x, \xi) = \int_{\mathbb{R}} e(x, y)e^{-i\xi y} dy.$$

We first study solutions to the homogeneous equation  $\mathcal{L}\mathcal{G}(e_i) = 0$  in domain  $\Omega_i$ ,  $i = 1, 2$ . We take its Fourier transform in the  $y$  direction and get:

$$\hat{\mathcal{L}}\hat{\mathcal{G}}\hat{e}^i = (\beta + \bar{u}\partial_x + i\xi\bar{v})((\bar{u}^2 - \bar{c}^2)\partial_{xx} + 2\bar{u}(\beta + i\xi\bar{v})\partial_x + \xi^2 + (\beta + i\xi\bar{v})^2)\hat{e}^i = 0. \quad (3.5)$$

We seek the solution in the form  $\hat{e}^i(x, \xi) = e^{\lambda(\xi)x}$  and we find three possible values for  $\lambda$ :

$$\lambda_{1,2}(\xi) = \frac{\bar{u}(\beta + i\xi\bar{v}) \pm \bar{c}\sqrt{(\beta + i\xi\bar{v})^2 + \xi^2(\bar{c}^2 - \bar{u}^2)}}{\bar{c}^2 - \bar{u}^2}, \quad \lambda_3(\xi) = -\frac{\beta + i\xi\bar{v}}{\bar{u}} \quad (3.6)$$

therefore the solution writes  $\hat{e}^i(x, \xi) = \alpha_{1i}(\xi)e^{\lambda_1(\xi)x} + \alpha_{2i}(\xi)e^{\lambda_2(\xi)x} + \alpha_{3i}(\xi)e^{\lambda_3(\xi)x}$ . We also impose that  $\hat{e}_1$  (resp.  $\hat{e}_2$ ) is bounded as  $x$  tends to  $-\infty$  (resp.  $\infty$ ). Taking into account the sign of the real parts of  $(\lambda_j)_{j=1,2,3}$

it means that we have:

$$\hat{e}^1(x, \xi) = \alpha_1 e^{\lambda_1(\xi)x} \text{ and } \hat{e}^2(x, \xi) = \alpha_2 e^{\lambda_2(\xi)x} + \alpha_3 e^{\lambda_3(\xi)x}. \tag{3.7}$$

In order to ease the notations we call:

$$a(\xi) = \beta + i\xi\bar{v} \text{ and } R(\xi) = \sqrt{(\beta + i\xi\bar{v})^2 + \xi^2(\bar{c}^2 - \bar{u}^2)}. \tag{3.8}$$

The initial guesses of the algorithm does not satisfy a specific partial differential equation. Therefore, it is possible to use formula (3.7) only for  $k \geq 1$ . With obvious notations, we write:

$$\hat{e}^{1,k}(x, \xi) = \alpha_1^k e^{\lambda_1(\xi)x} \text{ and } \hat{e}^{2,k}(x, \xi) = \alpha_2^k e^{\lambda_2(\xi)x} + \alpha_3^k e^{\lambda_3(\xi)x} \tag{3.9}$$

and

$$\hat{\tilde{e}}^{1,k}(x, \xi) = \tilde{\alpha}_1^k e^{\lambda_1(\xi)x} \text{ and } \hat{\tilde{e}}^{2,k}(x, \xi) = \tilde{\alpha}_2^k e^{\lambda_2(\xi)x} + \tilde{\alpha}_3^k e^{\lambda_3(\xi)x}. \tag{3.10}$$

Using,  $\hat{\mathcal{G}}\hat{e}^{1,1} = \hat{\mathcal{G}}\hat{e}^{2,1}$  we get:

$$\alpha_\Gamma := \alpha_1^1(a(\xi)\bar{c} + \bar{u}R(\xi)) = \alpha_2^1(a(\xi)\bar{c} - \bar{u}R(\xi)). \tag{3.11}$$

Now we estimate the Fourier transform of the *correction*  $\tilde{e}$ . By using (3.10) in the interface conditions of the *correction step* (3.3), we first get  $\hat{\gamma}^1 = i\xi\alpha_\Gamma R(\xi)$  and then:

$$\tilde{\alpha}_1^1 = -\frac{\alpha_\Gamma}{a(\xi) + \bar{u}R(\xi)}, \quad \tilde{\alpha}_2^1 = -\frac{\alpha_\Gamma}{a(\xi)\bar{c} - \bar{u}R(\xi)}, \quad \tilde{\alpha}_3^1 = \frac{i\xi(\bar{c}^2 - \bar{u}^2)}{a(\xi)} \cdot \frac{\alpha_\Gamma}{a(\xi)\bar{c} - \bar{u}R(\xi)}. \tag{3.12}$$

Now we estimate the Fourier transform of the *update*  $e$ . By using (3.9) and (3.10) in the interface conditions of the *update step* (3.4), we first get that  $\hat{\mathcal{G}}\hat{e}^{1,1} + \hat{\delta}^1 = \hat{\mathcal{G}}\hat{e}^{2,1} + \hat{\delta}^1 = 0$  and  $\hat{e}^{1,1} + \hat{\tilde{e}}^{1,1} = 0$  on  $\Gamma$ . This means that the coefficients  $(\alpha_i^2)_{1 \leq i \leq 3}$  satisfy homogeneous systems of linearly independent equations. Therefore  $\alpha_i^2 = 0$  for  $i = 1, 2, 3$  and then  $\hat{e}^{1,2} = \hat{\tilde{e}}^{2,2} = 0$ . Therefore, the convergence is achieved in two steps.  $\square$

#### 4. A NEW ALGORITHM APPLIED TO THE EULER SYSTEM

After having found an optimal algorithm which converges in two iterations for the third order model problem, we focus on the Euler system by translating this algorithm into an algorithm for the Euler system. It suffices to replace the operator  $\mathcal{L}\mathcal{G}$  by the Euler system and  $Q$  by the last component  $F(W)_3$  of  $F(W)$  in the boundary conditions. The algorithm reads:

**Algorithm 2.** We choose the initial values  $W^{1,0}$  and  $W^{2,0}$  such that  $\mathcal{G}F(W^{1,0})_3 = \mathcal{G}F(W^{2,0})_3$  and we compute  $(W^{i,k+1})_{i=1,2}$  from  $(W^{i,k})_{i=1,2}$  by the following iterative procedure:

**Correction step.** We compute the corrections  $\tilde{W}^{1,k}$  and  $\tilde{W}^{2,k}$  as solution of the homogeneous local problems:

$$\left\{ \begin{array}{l} \mathcal{P}\tilde{W}^{1,k} = 0 \text{ in } \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W}^{1,k})_3 \cdot \mathbf{n}_1 = \gamma^k, \text{ on } \Gamma, \end{array} \right. \quad \left\{ \begin{array}{l} \mathcal{P}\tilde{W}^{2,k} = 0 \text{ in } \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W}^{2,k})_3 \cdot \mathbf{n}_2 = \gamma^k, \text{ on } \Gamma, \\ \tilde{F}(W^{2,k})_3 = 0, \text{ on } \Gamma, \end{array} \right. \tag{4.1}$$

where  $\gamma^k = -\frac{1}{2} [A\nabla\mathcal{G}F(W^{1,k})_3 \cdot \mathbf{n}_1 + A\nabla\mathcal{G}F(W^{2,k})_3 \cdot \mathbf{n}_2]$ .

**Update step.** We update  $W^{1,k+1}$  and  $W^{2,k+1}$  by solving the local problems:

$$\begin{cases} \mathcal{P}W^{1,k+1} = f, & \text{in } \Omega_1, \\ \mathcal{G}F(W^{1,k+1})_3 = \mathcal{G}F(W^{1,k})_3 + \delta^k, & \text{on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}\tilde{W}^{2,k+1} = f, & \text{in } \Omega_2, \\ \mathcal{G}F(W^{2,k+1})_3 = \mathcal{G}F(W^{2,k})_3 + \delta^k, & \text{on } \Gamma, \\ F(W^{2,k+1})_3 = F(W^{1,k})_3 + F(\tilde{W}^{1,k})_3, & \text{on } \Gamma, \end{cases} \quad (4.2)$$

where  $\delta^k = \frac{1}{2} [\mathcal{G}F(\tilde{W}^{1,k})_3 + \mathcal{G}F(\tilde{W}^{2,k})_3]$ .

This algorithm is quite complex since it involves second order derivatives of the unknowns in the boundary conditions on  $\mathcal{G}F(W)_3$ . It is possible to simplify it. By using the Euler equations in the subdomain, we have lowered the degree of the derivatives in the boundary conditions. After lengthy computations that we omit here, we find a simpler algorithm. We write it for a decomposition in two subdomains with an outflow velocity at the interface of domain  $\Omega_1$  but with an interface not necessarily rectilinear. In this way, it is possible to figure out how to use for a general domain decomposition.

In the sequel,  $\mathbf{n} = (n_x, n_y)$  denotes the outward normal to domain  $\Omega_1$ ,  $\partial_n = (\partial_x, \partial_y) \cdot \mathbf{n}$  the normal derivative at the interface,  $\partial_\tau = (-\partial_y, \partial_x) \cdot \mathbf{n}$  the tangential derivative,  $U_n = Un_x + Vn_y$  and  $U_\tau = -Un_y + Vn_x$  are respectively the normal and tangential velocity at the interface between the subdomains. Similarly, we denote  $\bar{u}_n$  (resp.  $\bar{u}_\tau$ ) the normal (resp. tangential) component of the velocity around which we have linearized the equations.

**Algorithm 3.** We choose the initial values  $W^{i,0} = (P^{i,0}, U^{i,0}, V^{i,0})$ ,  $i = 1, 2$  such that  $P^{1,0} = P^{2,0}$  and we compute  $W^{i,k+1}$  from  $W^{i,k}$  by the iterative procedure with two steps:

**Correction step.** We compute the corrections  $\tilde{W}^{1,k}$  and  $\tilde{W}^{2,k}$  as solution of the homogeneous local problems:

$$\begin{cases} \mathcal{P}\tilde{W}^{1,k} = 0, & \text{in } \Omega_1, \\ -(\beta + \bar{u}_\tau \partial_\tau) \tilde{U}_n^{1,k} + \bar{u}_n \partial_\tau \tilde{U}_\tau^{1,k} = \gamma^k, & \text{on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}\tilde{W}^{2,k} = 0, & \text{in } \Omega_2, \\ (\beta + \bar{u}_\tau \partial_\tau) \tilde{U}_n^{2,k} - \bar{u}_n \partial_\tau \tilde{U}_\tau^{2,k} = \gamma^k, & \text{on } \Gamma \\ \tilde{P}^{2,k} + \bar{\rho} \bar{u}_n \tilde{U}_n^{2,k} = 0, & \text{on } \Gamma, \end{cases} \quad (4.3)$$

where  $\gamma^k = -\frac{1}{2} [(\beta + \bar{u}_\tau \partial_\tau)(U_n^{2,k} - U_n^{1,k}) + \bar{u}_n \partial_\tau(\tilde{U}_\tau^{1,k} - \tilde{U}_\tau^{2,k})]$ .

**Update step.** We compute the update of the solution  $W^{1,k+1}$  and  $W^{2,k+1}$  as solution of the local problems:

$$\begin{cases} \mathcal{P}W^{1,k+1} = f_1, & \text{in } \Omega_1, \\ P^{1,k+1} = P^{1,k} + \delta^k, & \text{on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}W^{2,k+1} = f_2, & \text{in } \Omega_2, \\ P^{2,k+1} = P^{2,k} + \delta^k, & \text{on } \Gamma, \\ (P + \bar{\rho} \bar{u}_n U_n)^{2,k+1} = (P + \bar{\rho} \bar{u}_n U_n)^{1,k} + (\tilde{P} + \bar{\rho} \bar{u}_n \tilde{U}_n)^{1,k}, & \text{on } \Gamma, \end{cases} \quad (4.4)$$

where  $\delta^k = \frac{1}{2} [\tilde{P}^{1,k} + \tilde{P}^{2,k}]$ .

**Proposition 2.** For a domain  $\Omega = \mathbb{R}^2$  divided into two non overlapping half planes, algorithms 2 and 3 are equivalent and both converge in two iterations.

*Proof.* The Euler equations are invariant with respect to rotations so we can write at the interface the equations (2.4) in the referential given by  $(\mathbf{n}, \tau)$ . We simply have to replace  $\partial_x$  (resp.  $\partial_y$ ) by  $\partial_n$  (resp.  $\partial_\tau$ ),  $U$  (resp.  $V$ ) by  $U_n$  (resp.  $U_\tau$ ) and  $\bar{u}$  (resp.  $\bar{v}$ ) by  $\bar{u}_n$  (resp.  $\bar{u}_\tau$ ). We denote by  $\hat{Q}$  the Fourier transform along the interface of a function  $Q$  and  $\xi$  is the dual variable.

We first consider the boundary conditions (4.2) of Algorithm 2 and prove that they correspond to the ones (4.4) of Algorithm 3. First of all, from (2.9), we have:

$$\widehat{F(W)}_3 = -\frac{1}{\beta + i\xi\bar{u}_\tau}(\bar{u}_n\hat{P} + \bar{\rho}\bar{u}_n^2\hat{U}_n). \tag{4.5}$$

We apply now the operator  $\hat{\mathcal{G}}$  and get:

$$\hat{\mathcal{G}}(\widehat{F(W)}_3) = -\frac{\bar{u}_n}{\beta + i\xi\bar{u}_\tau}(\hat{\mathcal{G}}(\hat{P}) + \bar{\rho}\bar{u}_n\hat{\mathcal{G}}(\hat{U}_n)). \tag{4.6}$$

By using the Euler equations satisfied by  $W$  we can substitute  $\hat{\mathcal{G}}(\hat{U}_n)$  with  $-1/\bar{\rho}\partial_n\hat{P}$  (we can omit the right handside since it will appear on both sides of the boundary condition) and obtain:

$$\hat{\mathcal{G}}(\widehat{F(W)}_3) = -\frac{\bar{u}_n}{\beta + i\xi\bar{u}_\tau}(\hat{\mathcal{G}}(\hat{P}) - \bar{u}_n\partial_n\hat{P}) = -\frac{\bar{u}_n}{\beta + i\xi\bar{u}_\tau}(i\bar{u}_\tau\xi + \beta)(\hat{P}) = -\bar{u}_n\hat{P}. \tag{4.7}$$

Therefore, boundary condition of (4.2) on the boundary of domain  $\Omega_1$  reads:

$$\bar{u}_n(\hat{P}^{1,k+1}) = \bar{u}_n\left(\hat{P}^{1,k} + \frac{1}{2}(\hat{\hat{P}}^{1,k} + \hat{\hat{P}}^{2,k})\right). \tag{4.8}$$

By simplifying with  $\bar{u}_n$ , we get the boundary condition of (4.4) on the pressure on the boundary of domain  $\Omega_1$ .

We now show how to obtain the second boundary condition in domain  $\Omega_2$  both in the update step (4.4) and in the correction step (4.3). From (4.5), we infer:

$$\frac{1}{\beta + i\xi\bar{u}_\tau}(\bar{u}_n\hat{P}^{2,k+1} + \bar{\rho}\bar{u}_n^2\hat{U}_n^{2,k+1}) = \frac{1}{\beta + i\xi\bar{u}_\tau}(\bar{u}_n\hat{P}^{1,k} + \bar{\rho}\bar{u}_n^2\hat{U}_n^{1,k} + \bar{u}_n\hat{\hat{P}}^{1,k} + \bar{\rho}\bar{u}_n^2\hat{\hat{U}}_n^{1,k}). \tag{4.9}$$

Multiplying by  $(\beta + i\xi\bar{u}_\tau)/\bar{u}_n$ , we obtain the second boundary condition of (4.3) and of (4.4).

We now derive the first boundary condition of the correction step (4.3) from the corresponding boundary condition of (4.1). We have to consider

$$\mathcal{B}(W) := (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W})_3. \tag{4.10}$$

From (3.2) and (4.7), we have:

$$\widetilde{\mathcal{B}(W)} = -\bar{u}_n((\bar{c}^2 - \bar{u}_n^2)\partial_n - \bar{u}_n(\beta + i\bar{u}_\tau\xi))(\hat{P}). \tag{4.11}$$

In order to replace the normal derivative on  $P$ , we write the Euler system in the form:

$$\partial_x W = -A^{-1}(\beta W + B\partial_y W - f). \tag{4.12}$$

We get (once again omitting the right hand side  $f$  that will appear on both sides of the boundary conditions):

$$\partial_n P = \frac{\bar{u}_n}{\bar{u}_n^2 - \bar{c}^2}[-\bar{u}_n(\beta + \bar{u}_\tau\partial_\tau)(P) + \bar{\rho}\bar{c}^2(\beta + \bar{u}_\tau\partial_\tau)(U_n) - \bar{u}_n\bar{\rho}\bar{c}^2\partial_\tau(U_\tau)]. \tag{4.13}$$

Using this equation in (4.10),

$$\mathcal{B}(W) = -\bar{u}_n\bar{\rho}\bar{c}^2[(\beta + \bar{u}_\tau\partial_\tau)(U_n) - \bar{u}_n\partial_\tau(U_\tau)]. \tag{4.14}$$

To obtain the first boundary condition of the correction step (4.3), it suffices to multiply (4.1) by  $-\bar{u}_n\bar{\rho}\bar{c}^2$ .  $\square$



### 5. DISCRETIZATION

In this section we will first present the discretization method used, a finite volume method on a uniform grid. Then we propose a strategy of discretization of the boundary conditions of the algorithm 3 applied to the Euler system and we present some theoretical discrete estimates of the convergence rate of the method.

#### 5.1. A finite volume discretization

We consider a domain  $\Omega$  and the boundary value problem associated to (2.4) with classical (natural) boundary conditions (see [11]) on  $\partial\Omega$ . This BVP is discretized using a finite volume scheme where the flux at the interface of the finite volume cells is computed using a Roe type solver.

We recall that this method has already been described in [9]. In order to discretize the BVP, we consider a regular quadrilateral grid where a vertex  $v_{ij}$  is characterized by

$$v_{ij} = \left( \left( i - \frac{1}{2} \right) \Delta x, \left( j - \frac{1}{2} \right) \Delta y \right), i, j \in \mathbb{Z}.$$

We associate to each vertex a finite volume cell,  $C_{ij} = [(i - 1)\Delta x, i\Delta x] \times [(j - 1)\Delta x, j\Delta x]$  which is a rectangle having as a center the vertex  $v_{ij}$ . A first order vertex centered finite volume formulation simply writes:

$$\frac{W_{i,j}}{c\Delta t} + \frac{1}{|C_{ij}|} \sum_{e \in \partial C_{ij}} |e| \Phi^e = f, \tag{5.1}$$

where  $|C_{ij}|$  denotes the area of the cell  $C_{ij}$ ,  $|e|$  the length of the edge  $e$  and  $W_{i,j}$  the average value of the unknown on the cell  $C_{ij}$

$$W_{i,j} = \frac{1}{|C_{ij}|} \int_{C_{ij}} W(x, y) dx dy.$$

Here, the elementary flux  $\Phi_{ij}^e$  across edge  $e$  is computed by a Roe type scheme  $\Phi^e = A_{\mathbf{n}}^+ W_{i,j} + A_{\mathbf{n}}^- W_{k,l}$ , where  $\mathbf{n} = (n_x, n_y)$  is the outward normal to the the edge  $e$ ,  $A_{\mathbf{n}} = n_x A_1 + n_y A_2$  and  $C_{kl}$  is the neighboring cell of  $C_{ij}$  sharing the edge  $e$  with it. We can rewrite (5.1) as

$$\frac{W_{ij}}{c\Delta t} + \frac{|A_1|W_{i,j} + A_1^- W_{i+1,j} - A_1^+ W_{i-1,j}}{\Delta x} + \frac{|A_2|W_{i,j} + A_2^- W_{i,j+1} - A_2^+ W_{i,j-1}}{\Delta y} = f. \tag{5.2}$$

We will further denote  $\bar{\Delta}x = \frac{\Delta x}{c\Delta t}$  and  $\bar{\Delta}y = \frac{\Delta y}{c\Delta t}$ , the non dimensioned counterpart of the mesh size in  $x$  and  $y$  directions.

#### 5.2. How to discretize the interface conditions

In the following we will propose a natural way to discretize the interface conditions. In order to do that we will first write the semi-discrete system (only in the  $y$  direction):

$$\begin{cases} \beta P_j + \bar{u} \partial_x P_j + \mathcal{D}_{my} P_j + \bar{\rho} \bar{c}^2 \partial_x U_j + \bar{\rho} \bar{c}^2 \mathcal{D}_{py} V_j = f_1 \\ \beta U_j + \bar{u} \partial_x U_j + \bar{v} D_y^- U_j + \frac{1}{\bar{\rho}} \partial_x P_j = f_2 \\ \beta V_j + \bar{u} \partial_x V_j + \mathcal{D}_{my} V_j + \frac{1}{\bar{\rho}} \mathcal{D}_{py} P_j = f_3 \end{cases} \tag{5.3}$$

where

$$\mathcal{D}_{my} = \left[ \frac{\bar{c} + \bar{v}}{2} D_y^- - \frac{\bar{c} - \bar{v}}{2} D_y^+ \right], \mathcal{D}_{py} = \left[ \frac{\bar{c} + \bar{v}}{2} D_y^- + \frac{\bar{c} - \bar{v}}{2} D_y^+ \right]$$

where  $M_n = \frac{\bar{u}}{c}$  and  $M_t = \frac{\bar{v}}{c}$  are the normal and the tangential Mach numbers and  $D_y^\pm$  are the usual finite difference operators. When writing the discrete counterpart of the algorithm 3 one has to take care of the discretization of the interface in the *correction step*, whereas for the *update step* there is no need for a special treatment.

We follow the same reasoning as in Proposition 2 and we do this using the semi-discrete form (5.3). First of all from (5.3) we obtain the discrete counterpart of (4.13)

$$\partial_n P = \frac{\bar{u}_n}{\bar{u}_n^2 - \bar{c}^2} [-\bar{u}_n(\beta + \bar{u}_\tau \mathcal{D}_{m_y})(P) + \bar{\rho} \bar{c}^2(\beta + \bar{u}_\tau D_y^-)(U_n) - \bar{u}_n \bar{\rho} \bar{c}^2 \mathcal{D}_{p_y}(U_\tau)] \tag{5.4}$$

and then by replacing it into (4.11) we obtain

$$\mathcal{B}_{dis}(W) = \bar{u}_n \bar{\rho} \bar{c}^2 [-(\beta + \bar{u}_\tau D_y^-)(U_n) + \bar{u}_n \mathcal{D}_{p_y}(U_\tau) + \bar{u}_n(\bar{u}_\tau \partial_y - \mathcal{D}_{m_y})\bar{P}] \tag{5.5}$$

which suggests the use of the classical finite difference operator  $D_y^-$  for the normal components of the velocity and  $\mathcal{D}_{p_y}$  for the tangential one. Nevertheless, the question that remains open is which kind of approximation we should use for the pressure term. The most natural one is  $\mathcal{D}_{m_y}$ , which cancels the pressure term, otherwise we will obtain an extra quantity of order  $\Delta y$ . We will see in the discrete analysis that follows that, one should not completely drop this part because it stabilized the boundary term leading to better convergence results.

### 5.3. Theoretical discrete convergence results

We will proceed to a discrete convergence analysis as in [9] in order to decide which discretization of the boundary conditions is better. We will recall briefly the key ingredients of this analysis. We perform a discrete Fourier transform, by looking for the solution under the form:

$$W_{i,j} = \sum_{\xi} \sum_{l=1}^3 \alpha_{\xi l} e^{(i-\frac{1}{2})\lambda_l(\xi)\Delta x} e^{Ij\xi\Delta y} V_l(\xi) \tag{5.6}$$

where  $I^2 = -1$ . By introducing this expression into the discrete equation we get that for each  $\xi$ ,  $\lambda_l(\xi)$  and  $V_l(\xi)$  have to be the solution of

$$\left( Id + \frac{|A_1| + A_1^- e^{\lambda_1(\xi)\Delta x} - A_1^+ e^{-\lambda_1(\xi)\Delta x}}{\Delta x} + \frac{|A_2| + A_2^- e^{I\xi\Delta y} - A_2^+ e^{-I\xi\Delta y}}{\Delta y} \right) V_l(\xi) = 0.$$

We denote by  $L_l(\xi) = \frac{e^{-\lambda_l(\xi)-1}}{\Delta x}$  and by  $e_y(\xi) = \frac{e^{I\xi\Delta y}-1}{\Delta x}$ . We can solve this system numerically for each wave-number  $\xi$ . By introducing expressions (5.6) in the interface conditions of the following algorithm we get the discrete convergence rate:

**Algorithm 4.** We choose the initial values  $W^{i,0} = (P^{i,0}, U^{i,0}, V^{i,0})$ ,  $i = 1, 2$  such that  $P^{1,0} = P^{2,0}$  and we compute  $W^{i,k+1}$  from  $W^{i,k}$  by the iterative procedure with two steps:

**Correction step.** We compute the corrections  $\bar{W}^{1,k}$  and  $\bar{W}^{2,k}$  as solution of the homogeneous local problems:

$$\begin{cases} \mathcal{P}\bar{W}^{1,k} = 0, & \text{in } \Omega_1, \\ -(\beta + \bar{u}_\tau D_y^-)\bar{U}_n^{1,k} + \bar{u}_n \mathcal{D}_{p_y}\bar{U}_\tau^{1,k} + S = \gamma^k, & \text{on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}\bar{W}^{2,k} = 0, & \text{in } \Omega_2, \\ (\beta + \bar{u}_\tau D_y^-)\bar{U}_n^{2,k} - \bar{u}_n \mathcal{D}_{p_y}\bar{U}_\tau^{2,k} - S = \gamma^k, & \text{on } \Gamma, \\ \bar{P}^{2,k} + \bar{\rho}\bar{u}_n\bar{U}_n^{2,k} = 0, & \text{on } \Gamma, \end{cases} \tag{5.7}$$

where  $\gamma^k = -\frac{1}{2} [(\beta + \bar{u}_\tau D_y^-)(U_n^{2,k} - U_n^{1,k}) + \bar{u}_n \mathcal{D}_{p_y}(\bar{U}_\tau^{1,k} - \bar{U}_\tau^{2,k})]$ .

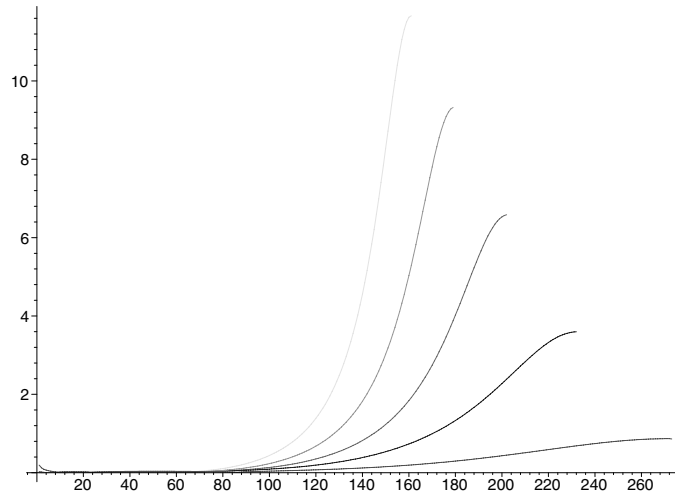


FIGURE 1. Convergence rate *vs.* Fourier number  $\xi$  for different values of the normal Mach number, no stabilization used.

**Update step.** We compute the update of the solution  $W^{1,k+1}$  and  $W^{2,k+1}$  as solution of the local problems:

$$\begin{cases} \mathcal{P}W^{1,k+1} = f_1, & \text{in } \Omega_1, \\ P^{1,k+1} = P^{1,k} + \delta^k, & \text{on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}W^{2,k+1} = f_2, & \text{in } \Omega_2, \\ P^{2,k+1} = P^{2,k} + \delta^k, & \text{on } \Gamma, \\ P^{2,k+1} + \bar{\rho}\bar{u}_n U_n^{2,k+1} = (P^{1,k} + \bar{\rho}\bar{u}_n U_n^{1,k}) + (\bar{P}^{1,k} + \bar{\rho}\bar{u}_n \bar{U}_n^{1,k}), & \text{on } \Gamma, \end{cases} \tag{5.8}$$

where  $\delta^k = \frac{1}{2} [\bar{P}^{1,k} + \bar{P}^{2,k}]$ .

where  $S$  designs the discretized pressure term  $\bar{u}_n(\bar{u}_\tau \partial_y - \mathcal{D}_{my})\bar{P}$ .

In Figures 1 and 2 we show for a flow normal to the interface two possible choices of discretization of the interface conditions. The first one gives  $S = 0$  and the second one  $S = M_n \mathcal{O}(\Delta y)$ . These two figures show that we need to keep the stabilization term in order to have a better convergence rate.

### 6. NUMERICAL RESULTS

We compare the method proposed and the classical method defined in [6,28,29] and analyzed in [11] where we formulated a Schwarz algorithm (interface iteration which relies on the successive solving of the local decomposed problems and the transmission of the result at the interface) involving transmission conditions that are derived naturally from a weak formulation of the underlying boundary value problem. In this method, the solutions in the subdomains are matched using interface conditions derived from the Roe decomposition of the fluxes on the interface.

We present here a set of numerical experiments on a model problem. We considered a decomposition into different number of subdomains and for a linearization around a constant or non-constant flow. The computational domain is given by the rectangle  $[0, 4] \times [0, 1]$  with a uniform discretization using  $80 \times 20$  points. The numerical investigation is limited to the resolution of the linear system resulting from the first implicit time step using a Courant number  $CFL = 100$ . In the following, for the new algorithm, each iteration counts for 2 as we need to solve twice as much local problems than the classical one. For an easier comparison of the algorithms, the figures shown in the tables are the number of subdomains solves. We also used substructuring (solving a system with interface variables only) and the iteration number necessary to achieve convergence by means of a

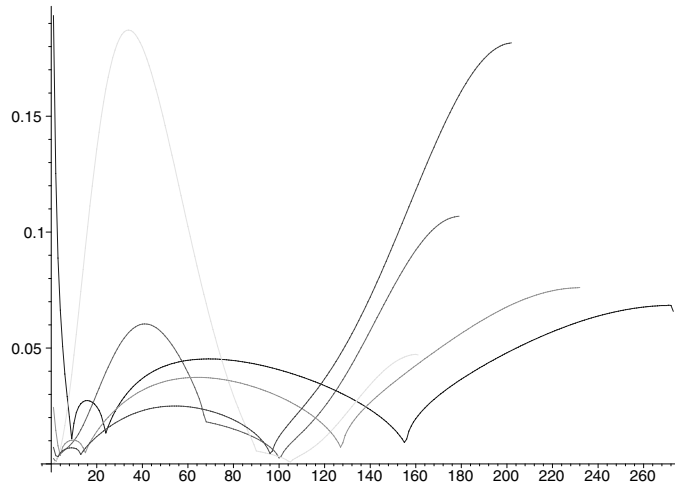


FIGURE 2. Convergence rate *vs.* Fourier number  $\xi$  for different values of the normal Mach number, with stabilization term.

TABLE 1. Iteration count for different values of  $M_n$ ,  $M_t(y)$ .

$M_n$	Classical (iterative)	Classical (GMRES)	New DDM (iter)	New DDM (GMRES)
0.001	32	26	16	16
0.01	30	26	16	16
0.1	28	21	14	14
0.2	24	19	14	14
0.3	20	16	14	14
0.4	18	14	14	14
0.5	16	13	14	14
0.6	15	12	14	14
0.7	14	11	14	14
0.8	14	11	14	14

GMRES method is also presented. We are solving the homogeneous equations verified by the error vector at the first time step.

### 6.1. Two-subdomain case

We consider first a decomposition into 2 subdomains and a linearization around a variable state where the tangential velocity is given by the expression  $M_t(y) = 0.1(1 + \cos(\pi y))$  and the normal Mach number remains constant at the interface. The results for different values of  $M_n$  are presented in Table 1.

We also consider the equations linearized around a variable state for a general flow at the interface where the tangential Mach number is given by  $M_t = 0.1(1 + \cos(\pi y))$ , and the initial normal velocity is given by the expression  $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$ . The iterative version of the new algorithm converges in 18 iterations whereas the classical one need 45 iterations to attain the same tolerance. For the accelerated version the new algorithm needs 14 iterations and the classical one 21, to achieve convergence.

The sensitivity to the mesh size is shown in the Table 2 for the iterative versions of both the classical and the new algorithm. We can see that for the new algorithm the growth in the number of iterations is very weak as the mesh is refined, the same property being already known for the classical one, see [11].

TABLE 2. Iteration count for different mesh size – Iterative versions.

$h$ ( $M_n = 0.001$ )	Classical	New DDM	$h$ ( $M_n = 0.1$ )	Classical	New DDM
1/10	65	18	1/10	56	12
1/20	67	18	1/20	57	14
1/40	70	18	1/40	59	16

TABLE 3. Iteration count for different values of  $M_n$ .

$M_n$	Classical (iterative)	Classical (GMRES)	New DDM (iter)	New DDM (GMRES)
0.001	32	26	20	16
0.01	31	26	20	16
0.1	29	21	18	16
0.2	25	19	18	16
0.3	23	16	18	16
0.4	21	15	16	16
0.5	19	13	16	14
0.6	16	12	16	14
0.7	14	11	16	14
0.8	13	11	16	14

TABLE 4. Iteration count for different values of  $M_n$ .

$M_n$	Classical(iter)	Classical (GMRES)	New DDM (GMRES)
0.001	101	28	28
0.01	86	28	28
0.1	54	26	26
0.2	38	23	30
0.3	35	23	32

### 6.2. General case

The next set of tests concerns a stripwise decomposition into 3 subdomains. The same kind of tests are carried out as in the 2 subdomain case. Table 3 summarizes the number of Schwarz iterations required to reduce the initial linear residual by a factor  $10^{-6}$  for different values of the reference Mach number for the new and the classical algorithm (the tangential velocity is given by the expression  $M_t(y) = 0.1(1 + \cos(\pi y))$ ). For a linearization around a variable state for a general flow at the interface where the tangential Mach number is given by  $M_t = 0.1(1 + \cos(\pi y))$ , and the initial normal velocity is given by the expression  $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$ , the same conclusion yields as in the two-domain case. As of intermediate conclusion we can state that the iteration number is only slightly increasing when going from 2 to 3 subdomains.

The next set of tests concerns a decomposition into 4 subdomains using a  $2 \times 2$  decomposition of a  $40 \times 40 = 1600$  point mesh. No special treatment of the cross points is done or coarse space added. This could be a reason why the iterative version of the algorithm doesn't converge. Nevertheless, the accelerated algorithm by a GMRES method converges as showed in Table 4 which summarizes the number if iterations for different values of the reference Mach number for both algorithms (the tangential velocity is given by the expression  $M_t(y) = 0.1(1 + \cos(\pi y))$  and the normal Mach number is constant at the interface). We can see the the new algorithm behaves similarly to the classical one for low Mach numbers. The latest results show clearly the need for an appropriate treatment of the corners or/and of a coarse space as this is done for the FETI-DP methods, in order to improve the performance of the method which has already shown promising results in the case of the stripwise decompositions.

## 7. CONCLUSION

In this paper we designed a new domain decomposition for the Euler equations inspired by the idea of the Robin-Robin preconditioner applied to the advection-diffusion equation. We used the same principle after reducing the system to scalar equations via a Smith factorization. The resulting algorithm behaves very well for the low Mach numbers, where usually the classical algorithm doesn't give very good results.

## REFERENCES

- [1] Y. Achdou and F. Nataf, A Robin-Robin preconditioner for an advection-diffusion problem. *C. R. Acad. Sci. Paris Sér. I* **325** (1997) 1211–1216.
- [2] Y. Achdou, P. Le Tallec, F. Nataf and M. Vidrascu, A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.* **184** (2000) 145–170.
- [3] J.D. Benamou and B. Després, A domain decomposition method for the Helmholtz equation and related optimal control. *J. Comp. Phys.* **136** (1997) 68–82.
- [4] M. Bjørhus, A note on the convergence of discretized dynamic iteration. *BIT* **35** (1995) 291–296.
- [5] J.-F. Bourgat, R. Glowinski, P. Le Tallec and M. Vidrascu, Variational formulation and algorithm for trace operator in domain decomposition calculations, in *Domain Decomposition Methods*, T. Chan, R. Glowinski, J. Périaux and O. Widlund Eds., Philadelphia, PA, SIAM (1989) 3–16.
- [6] X.-C. Cai, C. Farhat and M. Sarkis, A minimum overlap restricted additive Schwarz preconditioner and application in 3D flow simulations, in *Proceedings of the 10th Domain Decomposition Methods in Sciences and Engineering*, C. Farhat J. Mandel and X.-C. Cai Eds., *Contemporary Mathematics*, AMS **218** (1998) 479–485.
- [7] P. Chevalier and F. Nataf, Symmetrized method with optimized second-order conditions for the Helmholtz equation, in *Domain Decomposition Methods, 10 (Boulder, CO, 1997)*. *Amer. Math. Soc.*, Providence, RI (1998) 400–407.
- [8] S. Clerc, Non-overlapping Schwarz method for systems of first order equations. *Cont. Math.* **218** (1998) 408–416.
- [9] V. Dolean and F. Nataf, *An optimized Schwarz algorithm for the compressible Euler equations*. Technical Report 556, CMAP, École Polytechnique (2004).
- [10] V. Dolean, S. Lanteri and F. Nataf, Construction of interface conditions for solving compressible Euler equations by non-overlapping domain decomposition methods. *Int. J. Numer. Meth. Fluids* **40** (2002) 1485–1492.
- [11] V. Dolean, S. Lanteri and F. Nataf, Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations. *Appl. Num. Math.* **49** (2004) 153–186.
- [12] B. Engquist and H.-K. Zhao, Absorbing boundary conditions for domain decomposition. *Appl. Numer. Math.* **27** (1998) 341–365.
- [13] M.J. Gander and L. Halpern, Méthodes de relaxation d'ondes pour l'équation de la chaleur en dimension 1. *C.R. Acad. Sci. Paris, Sér. I* **336** (2003) 519–524.
- [14] M.J. Gander, L. Halpern and F. Nataf, *Optimal Schwarz waveform relaxation for the one dimensional wave equation*. Technical Report 469, CMAP, École Polytechnique (2001).
- [15] M.J. Gander, F. Magoulès and F. Nataf, Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.* **24** (2002) 38–60.
- [16] F.R. Gantmacher, *Théorie des matrices. Tome 1: Théorie générale*. Traduit du Russe par C. Sarthou. Collection Universitaire de Mathématiques, No. 18. Dunod, Paris (1966).
- [17] F.R. Gantmacher, *Théorie des matrices. Tome 2: Questions spéciales et applications*. Traduit du Russe par C. Sarthou. Collection Universitaire de Mathématiques, No. 19. Dunod, Paris (1966).
- [18] F.R. Gantmacher, *Theorie des matrices*. Dunod (1966).
- [19] F.R. Gantmacher, *The theory of matrices. Vol. 1*. AMS Chelsea Publishing, Providence, RI (1998). Translated from the Russian by K.A. Hirsch, Reprint of the 1959 translation.
- [20] L. Gerardo-Giorda, P. Le Tallec and F. Nataf, A Robin-Robin preconditioner for advection-diffusion equations with discontinuous coefficients. *Comput. Methods Appl. Mech. Engrg.* **193** (2004) 745–764.
- [21] R. Glowinski, Y.A. Kuznetsov, G. Meurant, J. Periaux and O.B. Widlund, Eds. *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, SIAM (1991).
- [22] C. Japhet, F. Nataf and F. Rogier, The optimized order 2 method. Application to convection-diffusion problems. *Future Generation Computer Systems* **FUTURE 18** (2001).
- [23] S.-C. Lee, M.N. Vouvakis and J.-F. Lee, A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays. *J. Comput. Phys.* **203** (2005) 1–21.
- [24] J. Li, A Dual-Primal FETI method for incompressible Stokes equations. *Numer. Math.* **102** (2005) 257–275.
- [25] J. Li and O. Widlund, *BDDC algorithms for incompressible Stokes equations*. Technical report (2006) (submitted).

- [26] P.-L. Lions, On the Schwarz alternating method. III: a variant for nonoverlapping subdomains, in *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, held in Houston, Texas, March 20–22, 1989, T.F. Chan, R. Glowinski, J. Périaux and O. Widlund, Eds., Philadelphia, PA, SIAM (1990).
- [27] J. Mandel, Balancing domain decomposition. *Commun. Appl. Numer. M.* **9** (1992) 233–241.
- [28] A. Quarteroni, Domain decomposition methods for systems of conservation laws: spectral collocation approximation. *SIAM J. Sci. Stat. Comput.* **11** (1990) 1029–1052.
- [29] A. Quarteroni and L. Staldis, *Homogeneous and heterogeneous domain decomposition methods for compressible flow at high reynolds numbers*. Technical Report 33, CRS4 (1996).
- [30] Y.H. De Roeck and P. Le Tallec, Analysis and Test of a Local Domain Decomposition Preconditioner, in R. Glowinski *et al.* [21] (1991).
- [31] A. Toselli and O. Widlund, *Domain Decomposition Methods – Algorithms and Theory*. Springer Series in Computational Mathematics. Springer Verlag (2004).
- [32] J. T. Wloka, B. Rowley and B. Lawruk, *Boundary value problems for elliptic systems*. Cambridge University Press, Cambridge (1995).