

ECOLE POLYTECHNIQUE

CENTRE DE MATHÉMATIQUES APPLIQUÉES
UMR CNRS 7641

91128 PALAISEAU CEDEX (FRANCE). Tél: 01 69 33 41 50. Fax: 01 69 33 30 11

<http://www.cmap.polytechnique.fr/>

**A New Domain Decomposition Method
for the Compressible Euler Equations**

Victorița Dolean and Frédéric Nataf

The report can also be downloaded from

<http://hal.ccsd.cnrs.fr/ccsd-00004319>

arXiv, HAL

R.I. N⁰ 567

February 2005

A New Domain Decomposition Method for the Compressible Euler Equations

V. Dolean*, F. Nataf†

22nd February 2005

Abstract

In this work we design a new domain decomposition method for the Euler equations in 2 dimensions. The basis is the equivalence via the Smith factorization with a third order scalar equation to whom we can apply an algorithm inspired from the Robin-Robin preconditioner for the convection-diffusion equation. 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 preserved 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, ...).

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 2 |
| 2 | A third order scalar problem | 2 |
| 2.1 | The compressible 2D Euler equations | 2 |
| 2.2 | Equivalence of the Euler system to a scalar equation | 3 |
| 2.2.1 | Smith factorization | 3 |
| 3 | A new algorithm applied to a scalar third order problem | 5 |
| 3.1 | The algorithm for a two-domain decomposition | 5 |
| 4 | A new algorithm applied to the Euler system | 7 |
| 5 | Discretization | 9 |
| 5.1 | A finite volume discretization | 9 |
| 5.2 | Discretizations of the interface conditions | 10 |
| 6 | Numerical results | 11 |
| 7 | Conclusion | 14 |
| | References | 14 |

*Univ. d'Évry and CMAP, Ecole Polytechnique, 91128 Palaiseau, France, dolean@cmap.polytechnique.fr

†CMAP, CNRS UMR 7641, Ecole Polytechnique, 91128 Palaiseau Cedex, France, nataf@cmap.polytechnique.fr

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 [Lio90, EZ98], for the Helmholtz equation in [BD97, CN98, GMN01, LVL05] convection-diffusion problems in [JNR01]. For time dependent problems and local time steps, see for instance [GHN01, GH03]. The preconditioning methods have also known a wide development in the last decade. The Neumann-Neumann algorithms for symmetric second order problems [BGLTV89, Man92, RT91] has been the subject of numerous works, see [TW04] and references therein. An extension of these algorithms to non-symmetric scalar problems (the so called Robin-Robin algorithms) has been done in [ATNV00, GGTN04] for advection-diffusion problems.

The generalization of these kinds of methods to the system of the Euler equations is difficult in the subsonic case in dimensions equal or higher to two. As far as Schwarz algorithms 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, where the optimality is expressed by the fact that the number of iterations is equal to the number of subdomains (see Bjørhus [Bjø95] and Quarteroni [Qua90] for more details). In the subsonic case and in two or three dimensions, we can find a formulation with classical (natural) transmission conditions in [Qua90, CFS98, QS96] or with more general interface conditions in [Cle98] and optimized transmission conditions in [DN04]. The analysis of such algorithms applied to systems proved to be very different from the scalar case, see [DLN02, DLN04]. As far as preconditioning methods are concerned, to our knowledge, no extension 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 [WRL95] or [Gan66c]. 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 [ATNV00] 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. All the previous results have been obtained at the continuous level and for a decomposition into 2 unbounded subdomains. After a discretization in a bounded domain we cannot expect that these properties to be preserved exactly. Still we can show in Section 5 by a discrete convergence analysis that the expected results should be very good. In 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 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 concentrate ourselves on the conservative Euler equations in two-dimensions:

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

In the above expressions, ρ 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 γ_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 with respect to 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 constant state around which we linearize) of the Euler equations:

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

characterized by the following jacobian matrices:

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

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

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

We will study this system with the help of the Smith factorization.

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 [Gan66b, Gan66a, Gan98] or [WRL95] 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 exist three matrices with polynomial entries E , D and F with the following properties:

- $\det(E) = \det(F) = 1$
- 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 largest common divisor of the set of polynomials Det_k .

Then,

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

(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 ξ). 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

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

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

$$(2.7) \quad \hat{\mathcal{P}} = EDF$$

where

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

and

$$E = \frac{1}{(\bar{u}(\bar{c}^2 - \bar{u}^2))^{1/3}} \begin{pmatrix} i\bar{\rho}\bar{c}^2\xi & 0 & 0 \\ 0 & \bar{u} & 0 \\ \beta + \bar{u}\partial_x + i\bar{v}\xi & E_2 & \frac{\bar{c}^2 - \bar{u}^2}{i\xi\bar{\rho}\bar{c}^2} \end{pmatrix}$$

and

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

where

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

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

and

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

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 β 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 ∂_t^l is replaced by β^l for $l = 1, 2$.

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 section 2. We want to solve

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

where Q is scalar unknown function and g is a given right hand side. The algorithm will be based on the Robin-Robin algorithm [AN97, ATNV00] 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:

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

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 $\Omega_2 = (0, \infty, 0) \times \mathbb{R}$. The interface is $\Gamma = \{x = 0\}$. The outward normal to domain Ω_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 $\mathcal{G}Q^{1,0} = \mathcal{G}Q^{2,0}$ and 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:

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

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:

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

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 term of error vector $e^{i,k}(x, y) = (Q^{i,k} - Q|_{\Omega_i})(x, y)$. 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 the 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 Ω_i , $i = 1, 2$. We take its Fourier in the y direction and get:

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

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

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

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. ∞). Taking into account the sign of the real parts of $(\lambda_j)_{j=1,2,3}$ it means that we have:

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

In order to ease the notations we call:

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

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:

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

and we have also

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

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

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

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:

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

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 $\hat{e}^{1,1} + \hat{\hat{e}}^{1,1} = 0$ on Γ . 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 $\hat{e}^{1,2} = \hat{e}^{2,2} = 0$. Therefore, the convergence is achieved in two steps. \blacksquare

4 A new algorithm applied to the Euler system

After having found an optimal algorithm which converges in two steps 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:

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

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:

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

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 Ω_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 Ω_1 , $\partial_n = \nabla \cdot \mathbf{n}$ the normal derivative at the interface, $\partial_\tau = \nabla \cdot \boldsymbol{\tau}$ 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}_τ) 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:

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

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: (4.4)

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

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}, τ) . We simply have to replace ∂_x (resp. ∂_y) by ∂_n (resp. ∂_τ), U (resp. V) by U_n (resp. U_τ) and \bar{u} (resp. \bar{v}) by \bar{u}_n (resp. \bar{u}_τ). We denote by \hat{Q} the Fourier transform along the interface of a function Q and ξ 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:

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

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

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

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:

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

Therefore, boundary condition of (4.2) on the boundary of domain Ω_1 reads:

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

By simplifying with \bar{u}_n , we get the boundary condition of (4.4) on the pressure on the boundary of domain Ω_1 .

We now show how to obtain the second boundary condition in domain Ω_2 both in the update step (4.4) and in the correction step (4.3). From (4.5), we infer:

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

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

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

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

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

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

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

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

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

Using this equation in (4.10),

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

To obtain the first boundary condition of the correction step (4.3), it suffices to multiply (4.1) by $-\bar{u}_n \bar{\rho} \bar{c}^2$. This shows the equivalence of both algorithms. The convergence in two steps comes from the fact that Algorithm 2 was derived directly from Algorithm 1 which converges in two steps. \blacksquare

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 of the method.

5.1 A finite volume discretization

We consider a domain Ω and the boundary value problem associated to (2.4) with classical(natural) boundary conditions (see [DLN04]) 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 the method already described in [DN04]. In order to discretize the BVP we consider a regular quadrilateral grid where a vertex v_{ij} is characterized by

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

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:

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

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 Φ_{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

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

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 Discretizations of the interface conditions

In the following we consider the discretization of the interface conditions. In order to do that we will first write the semi-discrete system (only in the y direction):

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

where

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

where we $M_n = \frac{\bar{u}}{\bar{c}}$ and $M_t = \frac{\bar{v}}{\bar{c}}$ are the normal and the tangential Mach number and D_y^\pm are the usual finite difference operators. When writing the discrete counterpart of the algorithm 3, there is no need for a special treatment in the *update step* since the boundary conditions do not involve any derivatives. But one has to take care of the discretization of the interface conditions in the *correction step*. For this, we proceed as in the proof of proposition 2 but at the semi-discrete level (5.3). First of all, from (5.3) we obtain the discrete counterpart of (4.13)

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

and then by replacing it into (4.11) we obtain

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

which suggests the use of the classical finite difference operator D_y^- for the normal components of the velocity and \mathcal{D}_{py} 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 to cancel the pressure term. We will see in the discrete analysis that follows that it may lead to a diverging algorithm. Therefore, we will not completely drop this part but consider also the discretization of $(\bar{u}_\tau \partial_y - \mathcal{D}_{my}) \bar{P}$ by $\Delta y D_y^+ D_y^- \bar{P}$. As we shall see below, it leads to a converging algorithm. We say then that we have a stabilized algorithm.

Theoretical discrete convergence results We will proceed to a discrete convergence analysis as in [DN04] 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:

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

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

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

If we denote by $L_l(\xi) = \frac{e^{-\lambda_l(\xi)} - 1}{\Delta x}$ and by $e_y(\xi) = \frac{e^{I\xi\Delta y} - 1}{\Delta y}$, we can further solve this systems numerically for each wave-number ξ . By introducing these expressions in the interface conditions, we get the discrete convergence rate.

In figures 1 and 2, we show for a flow normal to the interface the convergence rate as a function of the wavenumber ξ along the interface. Two possibilities are considered: no stabilization and the stabilization mentioned above. These figures show the we need of the stabilization term for the high wavenumbers.

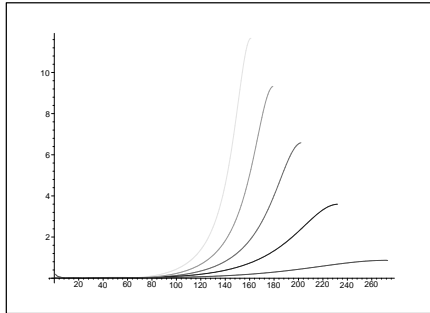


Figure 1: Convergence rate vs. Fourier number ξ for different values of the normal Mach number, no stabilization used

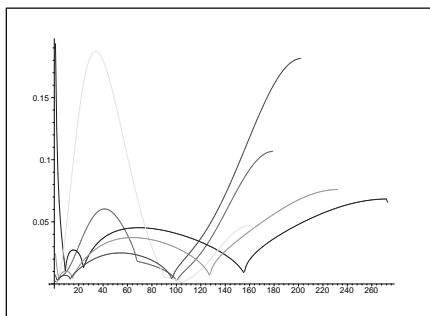


Figure 2: Convergence rate vs. Fourier number ξ for different values of the normal Mach number, with stabilization term

6 Numerical results

We compare the proposed method with the stabilization term and the classical method defined in [CFS98] or [DLN04] involving interface conditions that are derived naturally from a weak formulation of the underlying boundary value problem. We present here a set of results of numerical experiments on a model problem. We consider 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 80×20 points. The numerical investigation is limited to the solving of the linear system resulting from the first implicit time step using a Courant number CFL=100. For all tests, the stopping criterion was a reduction of the maximum norm of the error by a factor 10^{-6} . We consider first a decomposition into 2 subdomains and a linearization around a constant state and we are solving the homogeneous equations satisfied by the error vector at the first time step. In the following, for the new algorithm, each iteration counts for 2 as we need to solve twice as much local problems than with the classical algorithm. For an easier comparison of the algorithms, the figures shown in the tables are the number of subdomains solves. Table 1 summarizes the number of subdomains solves for different values of the reference Mach number for the new and the classical algorithm for a normal flow to the interface $M_t = 0.0$. The same results are presented in figure 3. In Table 2, we consider a linearization around a variable state where the tangential velocity is given by $M_t(y) = 0.1(1 + \cos(\pi y))$ and we vary the normal Mach number. In figure 4, we linearize the equations around a variable state for a normal flow to the interface ($M_t = 0.0$), where the initial normal velocity is gives by $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$. The sensitivity to the mesh size is shown in the Table 3. 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. The next tests concern a decomposition into 3 subdomains. We first consider a linearization around a constant state. Table 5 gives the results for different values of the reference Mach number for the new and the classical algorithm. The same results are presented in figure 5. In figure

| M_n | Classical | New DDM | M_n | Classical | New DDM |
|-------|-----------|---------|-------|-----------|---------|
| 0.001 | 67 | 18 | 0.4 | 25 | 14 |
| 0.01 | 66 | 16 | 0.5 | 20 | 14 |
| 0.1 | 55 | 14 | 0.6 | 16 | 14 |
| 0.2 | 41 | 16 | 0.7 | 13 | 12 |
| 0.3 | 32 | 16 | 0.8 | 15 | 14 |

Table 1: Subdomain solves counts for different values of M_n

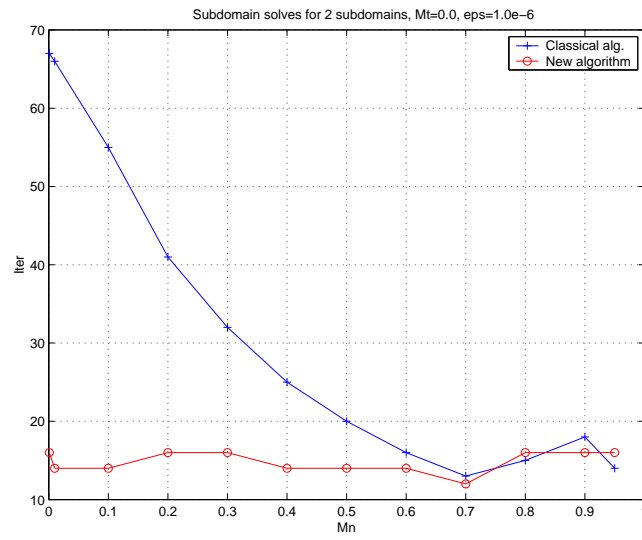


Figure 3: Subdomain solves counts for different values of M_n

| M_n | Classical | New DDM | M_n | Classical | New DDM |
|-------|-----------|---------|-------|-----------|---------|
| 0.001 | 32 | 16 | 0.4 | 18 | 14 |
| 0.01 | 30 | 16 | 0.5 | 16 | 12 |
| 0.1 | 28 | 14 | 0.6 | 15 | 12 |
| 0.2 | 24 | 14 | 0.7 | 14 | 12 |
| 0.3 | 20 | 14 | 0.8 | 14 | 14 |

Table 2: Subdomain solves counts for different values of $M_n, M_t(y)$

| 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: Subdomain solves counts for different mesh size

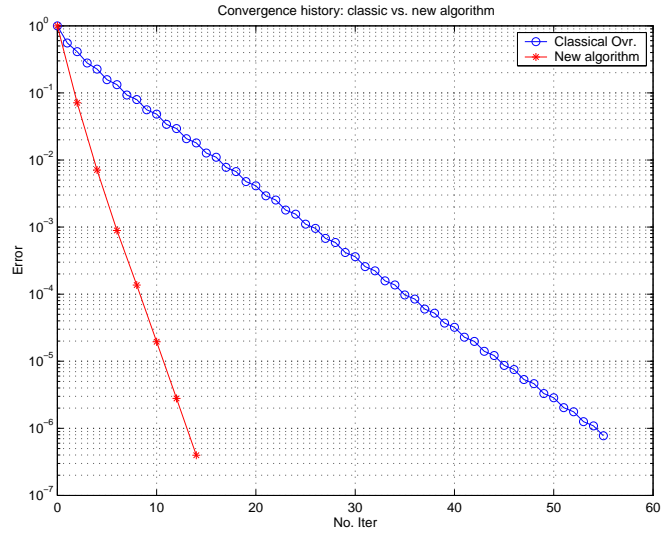


Figure 4: Convergence curves for the classical and the new algorithms

Table 4: Classical vs. the new algorithm

| M_n | Classical | New DDM | M_n | Classical | New DDM |
|-------|-----------|---------|-------|-----------|---------|
| 0.001 | 71 | 18 | 0.4 | 31 | 20 |
| 0.01 | 68 | 16 | 0.5 | 25 | 16 |
| 0.1 | 61 | 14 | 0.6 | 21 | 16 |
| 0.2 | 49 | 16 | 0.7 | 17 | 16 |
| 0.3 | 39 | 18 | 0.8 | 16 | 16 |

Table 5: Subdomain solves counts for different values of M_n

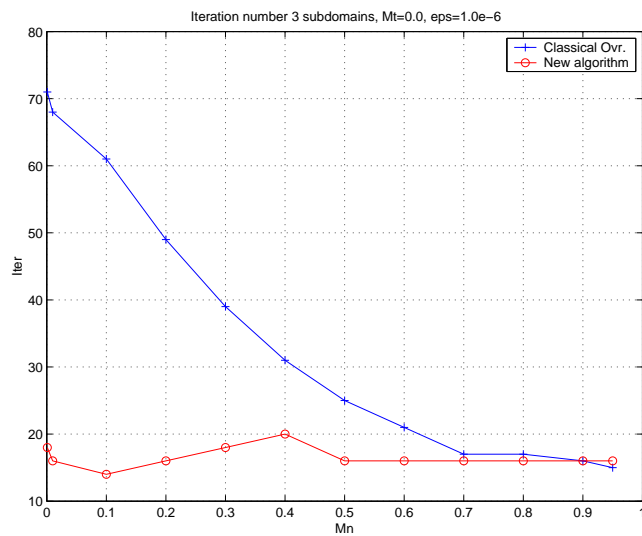


Figure 5: Convergence curves for the classical and the new algorithm

6, we consider for a three subdomains decomposition a linearization around a variable state for a normal flow to the interface. The normal velocity is given by $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$ (the same as for the 2 subdomain case). These tests show that the new algorithm is very stable with respect to various

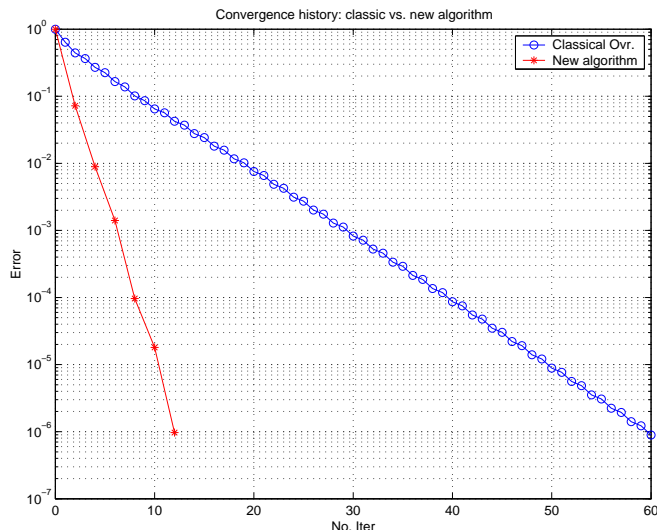


Figure 6: Convergence curves for the classical and the new algorithm

parameters such as the mesh size and the Mach number. We see that the convergence in two iterations of the continuous algorithm is lost at the discrete level although the subdomain solves are very reasonable. Moreover, a stabilization was necessary for the discretization of the interface condition (4.3) in order to keep the algorithm converging. The optimal discretization of this interface condition is not yet quite well known. The comparison with the classical algorithm is favorable for Mach numbers smaller than 0.5 and especially very low Mach numbers by a factor of almost 4.

7 Conclusion

In this paper we designed a new domain decomposition for the Euler equations inspired by the idea of the Robin-Robin preconditioner [AN97] 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 low Mach numbers, where usually the classical algorithm doesn't give very good results. We reduce the number of subdomain solves by almost a factor 4 for linearization around a constant and variable state as well. A general theoretical study and more comprehensive numerical tests have to be done in order to firmly assess the applicability of the proposed algorithm to large scale computations. This work can also be seen as a first step for deriving new domain decomposition methods for the 2D or 3D compressible Navier-Stokes equations. Indeed, the derivation of the algorithm which is based on the Smith factorization is in fact general and can be applied to arbitrary systems of partial differential equations, see [NR] for the Stokes and Oseen equations and [DNR] for a general presentation.

References

- [AN97] Y. Achdou and F. Nataf. A robin-robin preconditioner for an advection-diffusion problem. *C. R. Acad. Sci. Paris*, 325, Série I:1211–1216, 1997.

- [ATNV00] Yves Achdou, Patric Le Tallec, Frédéric Nataf, and Marina Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 184:145–170, 2000.
- [BD97] J. D. Benamou and B. Després. A domain decomposition method for the Helmholtz equation and related optimal control. *J. Comp. Phys.*, 136:68–82, 1997.
- [BGLTV89] Jean-François Bourgat, Roland Glowinski, Patrick Le Tallec, and Marina Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM.
- [Bjø95] Morten Bjørhus. A note on the convergence of discretized dynamic iteration. *BIT*, 35:291–296, 1995.
- [CFS98] X.-C. Cai, C. Farhat, and M. Sarkis. A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations. *Contemporary Mathematics*, 218:479–485, 1998.
- [Cle98] S. Clerc. Non-overlapping Schwarz method for systems of first order equations. *Cont. Math*, 218:408–416, 1998.
- [CN98] Philippe Chevalier and Frédéric Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.
- [DLN02] 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*, 40:1485–1492, 2002.
- [DLN04] 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.*, 49:153–186, 2004.
- [DN04] V. Dolean and F. Nataf. An optimized schwarz algorithm for the compressible euler equations. Technical Report 556, CMAP - Ecole Polytechnique, 2004.
- [DNR] V. Dolean, F. Nataf, and G. Rapin. New constructions of domain decomposition methods for systems of pdes.
- [EZ98] Bjorn Engquist and Hong-Kai Zhao. Absorbing boundary conditions for domain decomposition. *Appl. Numer. Math.*, 27(4):341–365, 1998.
- [Gan66a] F. R. Gantmacher. *Théorie des matrices. Tome 1: Théorie générale*. Traduit du Russe par Ch. Sarthou. Collection Universitaire de Mathématiques, No. 18. Dunod, Paris, 1966.
- [Gan66b] F. R. Gantmacher. *Théorie des matrices. Tome 2: Questions spéciales et applications*. Traduit du Russe par Ch. Sarthou. Collection Universitaire de Mathématiques, No. 19. Dunod, Paris, 1966.
- [Gan66c] Felix R. Gantmacher. *Theorie des matrices*. Dunod, 1966.
- [Gan98] 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.
- [GGTN04] 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.*, 193:745–764, 2004.

- [GH03] Martin J. Gander and Laurence Halpern. Méthodes de relaxation d'ondes pour l'équation de la chaleur en dimension 1. *C.R. Acad. Sci. Paris, Série I*, 336(6):519–524, 2003.
- [GHN01] Martin J. Gander, Laurence Halpern, and Frédéric Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. Technical Report 469, CMAP, Ecole Polytechnique, September 2001.
- [GKM⁺91] R. Glowinski, Y.A. Kuznetsov, G. Meurant, J. Periaux, and O.B. Widlund, editors. *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1991. SIAM.
- [GMN01] Martin J. Gander, Frédéric Magoulès, and Frédéric Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 2001. to appear.
- [JNR01] Caroline Japhet, Frédéric Nataf, and Francois Rogier. The optimized order 2 method. application to convection-diffusion problems. *Future Generation Computer Systems FUTURE*, 18, 2001.
- [Lio90] Pierre-Louis Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.
- [LVL05] Seung-Cheol Lee, Marinos N. Vouvakis, and Jin-Fa Lee. A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays. *Journal of Computational Physics*, 2005.
- [Man92] J. Mandel. Balancing domain decomposition. *Comm. on Applied Numerical Methods*, 9:233–241, 1992.
- [NR] F. Nataf and G. Rapin. A new domain decomposition method for the Stokes and Oseen systems. submitted.
- [QS96] A. Quarteroni and L. Stolicis. Homogeneous and heterogeneous domain decomposition methods for compressible flow at high reynolds numbers. Technical Report 33, CRS4, 1996.
- [Qua90] A. Quarteroni. Domain decomposition methods for systems of conservation laws : spectral collocation approximation. *SIAM J. Sci. Stat. Comput.*, 11:1029–1052, 1990.
- [RT91] Y.H. De Roeck and P. Le Tallec. Analysis and Test of a Local Domain Decomposition Preconditioner. In *R. Glowinski et al. [GKM⁺91]*, 1991.
- [TW04] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer Series in Computational Mathematics. Springer Verlag, 2004.
- [WRL95] J. T. Wloka, B. Rowley, and B. Lawruk. *Boundary value problems for elliptic systems*. Cambridge University Press, Cambridge, 1995.