A New Domain Decomposition Method for the Compressible Euler Equations Using Smith Factorization

Victorita Dolean[1] and Frédéric Nataf[2]

[1] Univ. de Nice Sophia-Antipolis, Laboratoire J.-A. Dieudonné, Nice, France.
dolean@math.unice.fr
[2] Université Paris VI, Laboratoire Jacques-Louis Lions, Paris, France,
nataf@ann.jussieu.fr

Summary. 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 [1]. 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, ...).

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. The preconditioning methods have also known a wide development in the last decade. The Neumann-Neumann algorithms for symmetric second order problems have been the subject of numerous works. An extension of these algorithms to non-symmetric scalar problems (the so called Robin-Robin algorithms) has been done in [1] 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 [2] and [7] for more details). In the subsonic case and in two or three dimensions, we

can find a formulation with classical (natural) transmission conditions in [7, 8] or with more general interface conditions in [3] and optimized transmission conditions in [5]. The analysis of such algorithms applied to systems proved to be very different from the scalar case, see [4]. 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.

In this work, 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 [9]. 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 [1] applied to a convection-diffusion problem. 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. In the Section 5, 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. The starting point of our analysis is given by the linearized form of the Euler equations 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:

$$\mathcal{P}W \equiv (\beta I + A\partial_x + B\partial_y)\, W = f \tag{1}$$

where $\beta = \frac{1}{\Delta t} > 0$, characterized by the following jacobian matrices:

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

In Computational Fluid Dynamics, problems of the form (1) 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.

We first recall the Smith factorization of a matrix with polynomial entries ([9], Theorem 1.4):

Theorem 1. *Let n be an positive integer and A a $n \times n$ matrix with polynomial entries with respect to the variable λ: $A = (a_{ij}(\lambda))_{1 \le i,j \le n}$. Then, there exist matrices E, D and F with polynomial entries satisfying the following properties:*

- *$det(E), det(F)$ are constants,*
- *D is a diagonal matrix uniquely determined up to a multiplicative constant,*
- *$A = EDF$.*

We first take formally the Fourier transform of the system (1) 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

$$\hat{\mathcal{P}} = \begin{pmatrix} \beta + \bar{u}\partial_x + i\xi\mathbf{b} & \bar{\rho}\bar{c}^2\partial_x & i\bar{\rho}\bar{c}^2\xi \\ \frac{1}{\bar{\rho}}\partial_x & \beta + \bar{u}\partial_x + i\xi\mathbf{b} & 0 \\ \frac{i\xi}{\bar{\rho}} & 0 & \beta + \bar{u}\partial_x + i\mathbf{b}\xi \end{pmatrix} \tag{3}$$

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

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \hat{\mathcal{L}}\hat{\mathcal{G}} \end{pmatrix}, \; E = \begin{pmatrix} i\bar{\rho}\bar{c}^2\xi & 0 & 0 \\ 0 & \bar{u} & 0 \\ \beta + \bar{u}\partial_x + i\mathbf{b}\xi & E_2 & \dfrac{\bar{c}^2 - \bar{u}^2}{i\xi\bar{\rho}\bar{c}^2} \end{pmatrix} \tag{4}$$

and

$$F = \begin{pmatrix} \dfrac{\beta + \bar{u}\partial_x + i\xi\mathbf{b}}{i\xi\bar{\rho}\bar{c}^2} & \dfrac{\partial_x}{i\xi} & 1 \\ \dfrac{\partial_x}{\bar{\rho}\bar{u}} & \dfrac{\beta + \bar{u}\partial_x + i\xi\mathbf{b}}{\bar{u}} & 0 \\ \dfrac{1}{(\beta + i\xi\mathbf{b})(\bar{u}^2 - \bar{c}^2)} & \dfrac{\bar{\rho}\bar{u}}{(\beta + i\xi\mathbf{b})(\bar{u}^2 - \bar{c}^2)} & 0 \end{pmatrix} \tag{5}$$

where

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

and

$$\begin{aligned} \hat{\mathcal{L}} &= \beta^2 + 2i\xi\bar{u}\mathbf{b}\partial_x + 2\beta(\bar{u}\partial_x + i\xi\mathbf{b}) + (\bar{c}^2 - \mathbf{b}^2)\xi^2 - (\bar{c}^2 - \bar{u}^2)\partial_{xx} \\ \hat{\mathcal{G}} &= \beta + \bar{u}\partial_x + i\xi\mathbf{b} \end{aligned} \tag{6}$$

Equation (4) suggests that the derivation of a domain decomposition method (DDM) for the third order operator $\mathcal{LG}$ 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 $\mathcal{LG}(Q) = g$ where Q is scalar unknown function and g is a given right hand side. The algorithm will be based on the Robin-Robin algorithm [1] for the convection-diffusion problem. Then we will prove its convergence in 2 iterations. 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 subdomains $\Omega_1 = (-\infty, 0) \times \mathbb{R}$ and $\Omega_2 = (0, \infty) \times \mathbb{R}$. The interface is $\Gamma = \{x = 0\}$. The outward normal to domain 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 $\mathcal{G}Q^{1,0} = \mathcal{G}Q^{2,0}$. 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:*

$$\begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{1,k} = 0 \text{ in } \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{1,k} \cdot \mathbf{n_1} = \gamma^k, \text{ on } \Gamma, \end{cases} \begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{2,k} = 0 \text{ in } \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{2,k} \cdot \mathbf{n_2} = \gamma^k, \text{ on } \Gamma, \\ \tilde{Q}^{2,k} = 0, \text{ on } \Gamma, \end{cases} \tag{7}$$

where $\gamma^k = -\frac{1}{2}\left[A\nabla\mathcal{G}Q^{1,k} \cdot \mathbf{n_1} + A\nabla\mathcal{G}Q^{2,k} \cdot \mathbf{n_2}\right]$.
Update step. *We update $Q^{1,k+1}$ and $Q^{2,k+1}$ by solving the local problems:*

$$\begin{cases} \mathcal{L}\mathcal{G}Q^{1,k+1} = g, \text{ in } \Omega_1, \\ \mathcal{G}Q^{1,k+1} = \mathcal{G}Q^{1,k} + \delta^k, \text{ on } \Gamma, \end{cases} \begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{2,k+1} = g, \text{ in } \Omega_2, \\ \mathcal{G}Q^{2,k+1} = \mathcal{G}Q^{2,k} + \delta^k, \text{ on } \Gamma, \\ Q^{2,k+1} = Q^{1,k} + \tilde{Q}^{1,k}, \text{ on } \Gamma, \end{cases} \tag{8}$$

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

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

See [6] for the details of the proof.

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:*

$$\begin{cases} \mathcal{P}\tilde{W}^{1,k} = 0 \text{ in } \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W}^{1,k})_3 \cdot \mathbf{n_1} = \gamma^k \text{ on } \Gamma, \end{cases} \begin{cases} \mathcal{P}\tilde{W}^{2,k} = 0 \text{ in } \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W}^{2,k})_3 \cdot \mathbf{n_2} = \gamma^k \text{ on } \Gamma, \\ \tilde{F}(W^{2,k})_3 = 0, \Gamma, \end{cases} \tag{9}$$

where $\gamma^k = -\frac{1}{2}\left[A\nabla\mathcal{G}F(W^{1,k})_3 \cdot \mathbf{n_1} + A\nabla\mathcal{G}F(W^{2,k})_3 \cdot \mathbf{n_2}\right]$.
Update step. *We update* $W^{1,k+1}$ *and* $W^{2,k+1}$ *by solving the local problems:*

$$\begin{cases} \mathcal{P}W^{1,k+1} = f, \text{ in } \Omega_1, \\ \mathcal{G}F(W^{1,k+1})_3 = \mathcal{G}F(W^{1,k})_3 + \delta^k \text{ on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}\tilde{W}^{2,k+1} & = f, \text{ in } \Omega_2, \\ \mathcal{G}F(W^{2,k+1})_3 & = \mathcal{G}F(W^{2,k})_3 + \delta^k \text{ on } \Gamma, \\ F(W^{2,k+1})_3 & = F(W^{1,k})_3 \\ & \quad + F(\tilde{W}^{1,k})_3 \text{ on } \Gamma, \end{cases} \tag{10}$$

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

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. 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, $\partial_n = \nabla \cdot \mathbf{n} = (\partial_x, \partial_y) \cdot \mathbf{n}$ 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:*

$$\begin{cases} \mathcal{P}\tilde{W}^{1,k} = 0, \text{ in } \Omega_1, \\ -(\beta + \bar{u}_\tau\partial_\tau)\tilde{U}_n^{1,n} + \bar{u}_n\partial_\tau\tilde{U}_\tau^{1,k} = \gamma^k \text{ on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}\tilde{W}^{2,k} = 0, \text{ in } \Omega_2, \\ (\beta + \bar{u}_\tau\partial_\tau)\tilde{U}_n^{2,k} - \bar{u}_n\partial_\tau\tilde{U}_\tau^{2,k} = \gamma^k, \Gamma \\ \tilde{P}^{2,k} + \bar{\rho}\bar{u}_n\tilde{U}_n^{2,k} = 0 \text{ on } \Gamma, \end{cases} \tag{11}$$

where $\gamma^k = -\frac{1}{2}\left[(\beta + \bar{u}_\tau\partial_\tau)(U_n^{2,k} - U_n^{1,k}) + \bar{u}_n\partial_\tau(\tilde{U}_\tau^{1,k} - \tilde{U}_\tau^{2,k})\right]$.
Update step. *We compute the update of the solution* $W^{1,k+1}$ *and* $W^{2,k+1}$ *as solution of the local problems:*

$$\begin{cases} \mathcal{P}W^{1,k+1} = f_1, \text{ in } \Omega_1, \\ P^{1,k+1} = P^{1,k} + \delta^k \text{ on } \Gamma, \end{cases} \quad \begin{cases} \mathcal{P}W^{2,k+1} & = f_2, \text{ in } \Omega_2, \\ P^{2,k+1} = P^{2,k} + \delta^k \text{ on } \Gamma, & \\ (P + \bar{\rho}\bar{u}_n U_n)^{2,k+1} & = (P + \bar{\rho}\bar{u}_n U_n)^{1,k} \\ & \quad + (\tilde{P} + \bar{\rho}\bar{u}_n\tilde{U}_n)^{1,k} \text{ on } \Gamma, \end{cases} \tag{12}$$

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

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.*

See [6] for the details of the proof.

5 Numerical Results

We present here a set of results of numerical experiments on a model problem. We compare the method proposed and the classical method defined in [4]. 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 80×20 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 GMRES method is also presented. We are solving the homogeneous equations verified by the error vector at the first time step.

The first 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 1 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

Table 1. 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

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 yield 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 2×2 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 does not converge. Nevertheless, the accelerated algorithm

by a GMRES method converges as showed in Table 2 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 as the classical one for low Mach numbers.

Table 2. 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

The latest results show clearly the need 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.

6 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 does not give very good results. We can reduce the number of iteration by almost a factor 4 both for linearization around a constant and variable state.

References

[1] Y. Achdou, P. Le Tallec, F. Nataf, and M. Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 184:145–170, 2000.

[2] M. Bjørhus. A note on the convergence of discretized dynamic iteration. *BIT*, 35:291–296, 1995.

[3] S. Clerc. Non-overlapping Schwarz method for systems of first order equations. *Cont. Math*, 218:408–416, 1998.

[4] V. Dolean, S. Lanteri, and F. Nataf. Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations. *Appl. Numer. Math.*, 49:153–186, 2004.

[5] V. Dolean and F. Nataf. An optimized Schwarz algorithm for the compressible Euler equations. Technical Report 556, CMAP - Ecole Polytechnique, 2004.

[6] V. Dolean and F. Nataf. A new domain decomposition method for the compressible Euler equations. *ESAIM-M2AN (Modelisation Mathematique et Analyse Numerique)*, 40:689–703, 2006.

[7] A. Quarteroni. Domain decomposition methods for systems of conservation laws: spectral collocation approximation. *SIAM J. Sci. Stat. Comput.*, 11:1029–1052, 1990.

[8] A. Quarteroni and L. Stolcis. Homogeneous and heterogeneous domain decomposition methods for compressible flow at high Reynolds numbers. Technical Report 33, CRS4, 1996.

[9] J.T. Wloka, B. Rowley, and B. Lawruk. *Boundary Value Problems for Elliptic Systems*. Cambridge University Press, Cambridge, 1995.