

Efficiency of two-level Schwarz Algorithm for incompressible and compressible flows

H. Alcin¹, O. Allain², B. Koobus³, A. Dervieux¹

¹ INRIA Sophia Antipolis, 2004 route de Lucioles, 06902 Sophia-Antipolis, France

² LEMMA - 2000 route des Lucioles - Les Algorithmes - Bt. Le Thales A, Biot, 06410, France

³ Institut de Mathématiques et de Modélisation de Montpellier, Université Montpellier 2, Case Courrier 051, Place Eugène Bataillon, 34095 Montpellier (France)

SUMMARY

The use of volume-agglomeration for introducing one or several levels of coarse grids in an Additive Schwarz (AS) multi-domain algorithm is revisited. The purpose is to build an algorithm applicable to elliptic and convective models. The sub-domain solver is ILU. We rely on algebraic coupling between the coarse grid and the Schwarz preconditioner. The Deflation Method (DM) and the Balancing Domain Decomposition (BDD) Method are experimented for a coarse grid as well as domain-by-domain coarse gridding. Standard coarse grids are built with the characteristic functions of the sub-domains. We also consider the building of a set of smooth basis functions (analog to smoothed-aggregation methods). The test problem is the Poisson problem with a discontinuous coefficient. The two options are compared for the standpoint of coarse-grid consistency and for the gain in scalability of the global Schwarz iteration. Applications to parallel calculation of incompressible and compressible flows are then presented. Copyright © 2011 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: domain decomposition, Schwarz, coarse grid, deflation, balancing, incompressible, compressible

1. INTRODUCTION

The parallel resolution of main CFD models, and in particular of compressible ones such as the Navier-Stokes equations for compressible gas remain an important issue in efficient modelling and design. While Multigrid methods appeared, at least for a while, as the best CFD solution algorithm, Domain Decomposition methods (DDM) emerged as a new star in Computational Structural Mechanics due to matrix stiffness issues in CSM. A DDM relies on a partition of the computational domain into sub-domains and assumes that representative sub-problems on sub-domains can be rather easily computed and can help convergence towards global problem's solution. An ideal DDM should be strongly scalable, that is, when solving a particular problem on a given mesh, it takes some time on p processors, and half of this time on $2p$ processors.

Most software using unstructured meshes apply a domain partition and subdomain local preconditioners. Two families of methods allow the design and analysis of the resulting algorithms. Shur methods the set of local problems allow to reduce the unknowns to interface degrees of freedom. Shur methods apply very efficiently to elliptic systems such as those arising in structural Mechanics and therefore are also applied to incompressible Navier-Stokes.

[†]E-mail: Alain.Dervieux@inria.fr

*Correspondence to: INRIA Sophia Antipolis, 2004 route de Lucioles, 06902 Sophia-Antipolis, France

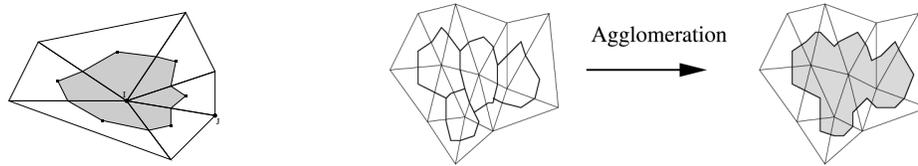


Figure 1. *Left: finite-Volume partition built as dual of a triangulation. Right: Greedy Algorithm for finite-volume cell agglomeration: four fine cells (left) are grouped into a coarse cell*

In Schwarz DDM, the set of local problems preconditions a global loop updating all degrees of freedom. Boundary conditions for each sub-domain problem are fetched in neighboring domains. The resulting iterative solver generally involves a Krylov iteration and is often referred as Newton-Krylov-Schwarz [6]. In contrast to the incompressible case, compressible flows are quasi exclusively addressed with additive Schwarz methods. More precisely, the Restrictive Additive Schwarz algorithm (RAS), which was initially introduced for elliptic models [7], was early extended with success to compressible flow models, [5, 26]. Schwarz and RAS algorithms combine well with various local preconditioners. In [26] the ILU local preconditioner is used. In [12], colored Gauss-Seidel is used.

However, Schwarz methods are subject to a -yet moderate- lack of scalability, in the sense that increasing the number of domains results in degrading the convergence rate. It has been shown by S. Brenner [4] that the resulting algorithm is not scalable, unless an extension called coarse grid is added. In [4], the coarse grid correction is computed on a particular coarser mesh, embedded into the main mesh. A similar approach was applied in [1] to compressible CFD. The advantage of this approach is to produce a convergent coarse mesh solution. However the coarse mesh option is not practical in many cases, in particular for arbitrary unstructured meshes. As a result, it was tried later to build a coarse basis using other principles. An option is to look for a few global eigenvectors of the operator, see for example [25]. For CPU cost reasons, these eigenvectors should not be exactly computed but only approximated. In a recent study [22],[23], it is proposed to compute eigenvectors of the local Dirichlet-to-Neumann operator, which can be computed in parallel on each sub-domain. The evaluation of eigenvectors is difficult when the matrix has a dominant Jordan behavior, which happens for convection dominant models, the privileged domain of finite-volume methods. In general, the eigenvector-based coarse grid corrections are introduced in an algebraic way, either by deflation or by balancing. Deflation and Balancing methods were respectively introduced by Nicolaides [24] and Mandel [19] and an interesting comparison can be found in [29]. Very simplified coarse basis can be used, like characteristic functions of subdomains. In particular, Deflation and Balancing allow for coarse basis without considering the accuracy of the coarse grid system. These methods are progressively compared with multi-grid like ones, and are sometimes found more efficient, see [28].

This paper addresses the combination of Schwarz algorithms with coarse grid methods based on Volume Agglomeration. The idea of Volume Agglomeration is directly inspired by the multi-grid idea, but inside the context of Finite-Volume Method. In this paper we consider meshes made of triangles or tetrahedra. On the mesh we consider a vertex centered approximation, similar to the P^1 finite element. A finite-volume partition is built from the dual cells of triangles, Figure 1, left. In order to build a coarser grid, it is possible to build coarse cells by sticking together neighboring cells for example with a greedy algorithm, Figure 1, right. The coarser grid is *a priori* unstructured as is the fine one. By the magic of FVM, a consistent coarse discretisation of a divergence-based first-order PDE is directly available. Indeed, we can consider that the new unknown is constant over the coarse cell and it remains to apply a Godunov quadrature of the fluxes between any couple of two coarse cells. Elliptic PDE can also be addressed in similar although more complicated way, [16]. As a result, consistent linear and non-linear coarse grid approximations are built using the agglomeration principle. Linear and nonlinear MG have been derived, in contrast with AMG algorithms. This method extends to Discontinuous Galerkin approximations [21]. More elaborated versions and their analysis have been under the aggregated methods such as [14]. The extension

of Agglomeration MG to multi-processor parallel computing, however, are less easily achieved, as compared to Domain Decomposition Methods, see however [9]. The many works on multi-level methods *à la* Bramble-Pasciak-Xu [3] has drawn attention to the question of basis smoothness. Indeed, the underlying basis function in volume-agglomeration is a characteristic function equal to zero or one. In [20], the agglomeration basis is extended to H^1 consistent ones in an analog way to smoothed-aggregation. In [10], a Bramble-Pasciak-Xu algorithm is built on these bases for an optimal design application.

In the proposed study, we try to build a convergent coarse mesh basis for an arbitrary unstructured fine mesh. It has been observed that coarser meshes for unstructured meshes are elegantly build with volume-agglomeration. In this study, we follow this track, define a convergent basis and examine how it behaves as a coarse grid preconditioner. Different ways in partitioning are also recalled. These methods are presented in next Section 2. In Section 3, the first test problem we concentrate on is inspired by a pressure-correction phase in compressible Navier-Stokes calculations (see for example [17]), and expresses as a Neumann problem with strongly discontinuous coefficient. In which the well-posedness is fixed with a Dirichlet condition on one cell. Section 4 is devoted to an example of complete incompressible flow resolution. In Section 5 we concentrate on the linearised compressible Navier-Stokes system to be solved at every iteration of a time-implicit unsteady LES simulation.

2. THE ALGORITHMS

2.1. Basic Additive exact and ILU Schwarz algorithm

Our discrete model has its unknowns attached to vertices of the triangulation. Let us assume that the set of unknown, Ω is split into two sub-sets, Ω_1 and Ω_2 Local systems on Ω_1 and Ω_2 are defined through the operators:

$$R_i = \text{Diag}(a_k), \text{ where } a_k = 1 \text{ if } k \in \Omega_i, 0 \text{ otherwise}$$

$$A_i = R_i A R_i.$$

It results that:

$$(A_i)_{kl} = A_{kl} \text{ if } k \in \Omega_i \text{ and } l \in \Omega_i, 0 \text{ else.}$$

Which turns into Dirichlet-type interface conditions at the first node outside from Ω_i . The *Additive Schwarz* algorithm is written in terms of preconditioning, as

$$M_{AS}^{-1} = \sum_{i=1}^2 A_i^{-1} \quad (1)$$

where we have written in short:

$$(A_i^{-1} f)_j = A_i^{-1}(f|_{\Omega_i}) \text{ if } j \in \Omega_i, 0 \text{ else.}$$

The preconditioner M^{-1} can be used inside a quasi-Newton iteration. In this paper, in order to keep some generality in our algorithms, we use GMRES, also used in [25], or BiCGStab [11]. In the *Additive Schwarz-ILU* version, the exact solution of the Dirichlet on each sub-domain is replaced by the less costly Incomplete Lower Upper (ILU) approximate solution.

$$M_{ASILU}^{-1} = \sum_{i=1}^2 ILU(A_i)^{-1} \quad (2)$$

with analog notation convention. Under some conditions concerning the overlapping of the local systems, both AS methods are convergent, but not completely satisfactory:

Definition: Let us call the *scalability factor* of a DDM method the ratio between n_1 the number

of iterations for converging to zero machine for N subdomains and n_2 the number of iterations for converging to zero machine for $2N$ subdomains. \square

This factor is measured for a given PDE, with a given mesh (strong scalability) or with a mesh two times larger for the run on $2N$ domains (weak scalability).

Definition: A DDM method is scalable if its scalability factor is 1 or smaller. \square

Here we do not discuss in terms of number of operations, but in terms of convergence rate, *i.e.* of number of main iterations. It is known, see for example [4], that:

Lemma: A Schwarz method as defined above is not scalable. \square

2.2. Algebraic Coarse grid

As shown by S. Brenner [4], the combination $M^{-1} = A_0^{-1} + \sum_{i=1}^N A_{|\Omega_i}^{-1}$ of the Additive Schwarz method with a coarse grid A_0^{-1} reduces the convergence rate of the previous iteration to an essentially scalable one. In [4], a coarse grid system is introduced in a geometrical manner, using two embedded meshes. Looking for an *algebraic* way to introduce a coarse grid, we observe that two methods have been proposed in the literature. Both rely on the following ingredients:

- $A_h u = f_h$ is the linear system to solve in V , fine-grid approximation space.
- $V_0 \subset V$ coarse approximation space. $V_0 = [\Phi_1 \cdots \Phi_N]$.
- Z an extension operator from V_0 in V and Z^T a restriction operator from V in V_0 .
- $Z^T A_h Z u_H = Z^T f_h$ is the coarse system.

The Deflation Method (DM) has been introduced by Nicolaidis [24] and is used by many authors. Saad *et al.* [25] encapsulate it into a Conjugate Gradient. Aubry *et al.* [2, 18] apply it to a pressure Poisson equation. In DM, the projection operator is defined as:

$$P_D = I_n - A_h Z (Z^T A_h Z)^{-1} Z^T \text{ with } A_h \in R^{n \times n} \text{ and } Z \in R^{n \times N}$$

The DM algorithm consists in solving first the coarse system $Z^T A_h Z u_H = Z^T f_h$, then the projected system $P_D A_h \check{u} = P_D f_h$ in order to get finally $u = (I_n - P_D^T) u + P_D^T u = Z (Z^T A_h Z)^{-1} Z^T f_h + P_D^T \check{u}$.

The Balancing Domain Decomposition (BD) has been introduced by J. Mandel [19] and applied to a complex system in [27]. In [29] a formulation close to DM is proposed. It consists in replacing the original preconditioner M^{-1} (ex.: global ILU, Schwarz, or Schwarz-ILU) by:

$$P_B = P_D^T M^{-1} P_D + Z (Z^T A_h Z)^{-1} Z^T.$$

The two above algorithms are close to each other. Vuik and Nabben [29] show in a particular context that their convergence rate should be the same. With DM preconditioning, some eigenvalues are replaced by zero. With BDD, they are replaced by one. A consequence is that DM involves the solution by the fixed point iteration of an ill-posed problem, and this may induce difficulties in obtaining an iterative convergence reaching machine zero and staying there. The BDD has not this problem, but involves a larger number of operations. It can be about two times more expensive. We turn now to the way in choosing our coarse basis.

2.3. Smooth and non-smooth coarse grid

The coarse grid is then defined by set of basis functions. A central question is the smoothness of these functions. According to Galerkin-MG, smooth enough functions provide consistent coarse-grid solutions. Conversely, DDM methods preferably use the characteristic functions of the sub-domains, $\Phi_i(x_j) = 1$ si $x_j \in \Omega_i$. In the case of P^1 finite-elements, for example, the typical basis function corresponds to setting to 1 all degrees of freedom in sub-domain. According to [20], the



Figure 2. Left: characteristic coarse grid basis function. Right: smooth coarse grid basis function

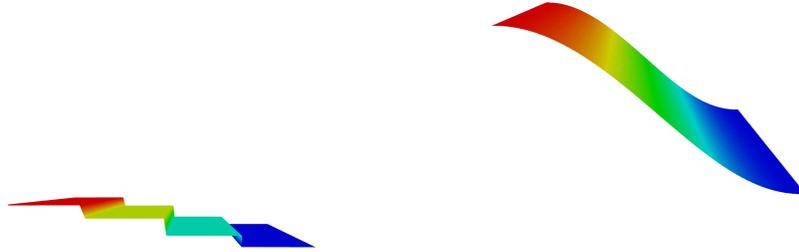


Figure 3. Accuracy of the coarse grid approximation (four basis functions) for a Poisson problem with a *sin* function (of amplitude 2.) as exact solution. Left: coarse grid solution with the characteristic basis (amplitude is 0.06). Right: coarse grid solution with a smooth basis (amplitude is 1.8).

coarse system

$$U^H(x) = \sum_i U_i \Phi_i(x) \quad ; \quad \int \nabla U^H \nabla \Phi_i = \int f \Phi_i \quad \forall i$$

produces a solution U^H which does not converge towards the continuous solution U when H tends to 0. In order to build a better basis, we need to introduce a hierarchical coarsening process from the fine grid to a coarse grid \mathcal{G}_j which will support the preconditioner. Level j is made of N_j macro-cells C_{jk} , i.e.:

$$\mathcal{G}_j = \{u, u | C_{jk} = \text{const.}\}.$$

Transfer operators are defined between successive levels (from coarse to fine):

$$P_i^j : \mathcal{G}_i \rightarrow \mathcal{G}_j \quad P_i^j(u)(C_{ik'}) = u(C_{jk}) \quad \text{with } C_{ik'} \subset C_{jk}$$

Following [20] we introduce the smoothing operator:

$$(L_k u)_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \text{meas}(j) u_j / \left\{ \sum_{j \in \mathcal{N}(i) \cup \{i\}} \text{meas}(j) \right\}$$

where $\mathcal{N}(i)$ holds for the set of cells which are direct neighbors of cell i . The smoothing is applied at each level between the coarse level k defining the characteristic basis and the finest level.

$$\Psi_k = (L_1 P_1^2 L_2 \cdots P_{p-2}^{p-1} L_{p-1} P_{p-1}^p) \Phi_k.$$

The resulting smooth basis function is compared with the characteristic one in Figure 2. The inconsistency of the characteristic basis and the convergence of this new smooth basis is illustrated by the solution of a Poisson equation with a *sin* function as exact solution, Figure 3.

Conversely, first-order hyperbolic problems, like advection, allow both types of basis. This is illustrated by the solution of the diffusion convection problem with a Peclet of 100, and an upwind fine approximation. For the fine approximation the mesh numerical Peclet is 1/2 and the approximation solution is free of oscillation, Fig.4a. The characteristic basis produces a not so

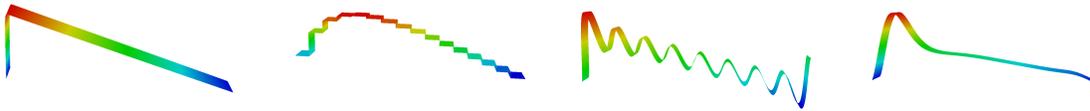


Figure 4. Accuracy of the coarse grid approximation for an advection-diffusion problem: (a) fine grid solution, (b) coarse solution with characteristic basis, (c) coarse solution with smooth basis, (d) coarse solution with smooth basis and numerical viscosity.

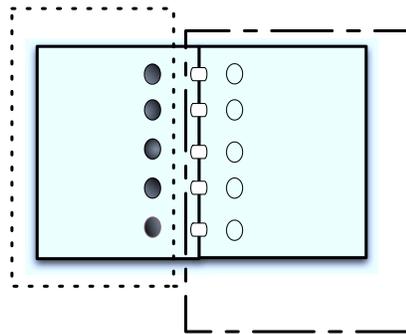


Figure 5. Decomposition 1: Domain decomposition without (vertex) overlap. Domain 1 is inside dots (....), Domain 2 is inside dashes (- - -).

bad approximation (Fig.4b) We force the smooth coarse basis to satisfy the Dirichlet boundary conditions. Since the mesh numerical Peclet is now much larger, the solution oscillates (Fig.4c). We have tried to moderate the oscillation by means of a coarse-grid numerical viscosity, built with the difference between the coarse mass matrix and its lumped version (sum of each line concentrated on the diagonal term)(Fig.4d).

2.4. Schwarz decomposition

The way an Additive Schwarz algorithm will converge strongly depends on the way the domain decomposition is defined. From algorithm complexity standpoint, it is important that overlapping is as small as possible. But for a given problem, overlapping thickness can have an important influence on iterative convergence. Also, in the theoretical analysis -without coarse grid- of e.g. [4], scalability holds if the overlapping thickness does not decrease when the number of nodes is increased. We define now four ways in decomposing the domain and discuss shortly the impact choosing each option on algorithm efficiency.

Decomposition 1, node partition. In Decomposition 1, we assume that the decomposition $\Omega_1, \dots, \Omega_N$ is a nodewise partition in such a way that the range of elements behind two neighboring subdomains is of width 1, Fig.5. Then according to $A_i = R_i A R_i$, each local operator A_i is a discretisation of a Dirichlet problem with zero condition on the vertices which are direct neighbors of vertices of Ω_i , but not belonging to Ω_i . The geometrical overlapping is the range of element of width 1 referred below. We note in passing that this minimalist option degrades the scalability of the Schwarz algorithm since the overlapping width decreases for a finer mesh.

The additive Schwarz (AS) (resp. additive Schwarz ILU, (ASILU)) algorithm is defined as follows:
 - Apply a Conjugate Gradient (CG) with M_{AS} , defined according to (1), resp. M_{ASILU} , defined

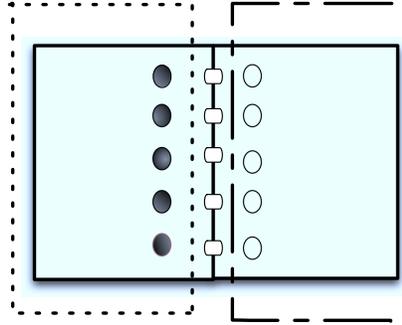


Figure 6. *Decomposition 2: Purely local solver; without overlap, and with a vertex range inside neither domains. (...): Domain 1. (- -): Domain 2.*

according to (2), as preconditioner.

The preconditioner inherits the initial operator symmetry, allowing CG iteration. We also note that the product by preconditioners can be locally computed without an extra communication.

Decomposition 2, Block Jacobi. In Decomposition 2, the $\Omega_1, \dots, \Omega_N$ have empty intersections (in terms of nodes) and their union is not Ω , Fig.6. Further, the nodes of Ω_i are not neighbors of them of Ω_j . The local preconditioner is not of Schwarz type, but some block-Jacobi where the blocks are the Ω_i 's. The preconditioner inherits the initial operator symmetry, allowing CG iteration. We also note that the product of residual by preconditioners can be locally computed without an extra communication.

Decomposition 3, Minimal Decomposition. Decomposition 3 is referred as the minimal decomposition, in [5, 7]. The different subdomains overlap on a node row, Fig.7. On each node of $\Omega_i \cap \Omega_j$, a corrector is produced by both local preconditioners. In the Restrictive Additive Schwarz of [7, 26], the nodes shared by two different Ω_i 's are partitioned:

$$\cup_{1 \leq i < j \leq N} (\Omega_i \cap \Omega_j) = \cup_{1 \leq i \leq N} \hat{\Omega}_i$$

in which $\hat{\Omega}_i$'s are disjoint, and $\hat{\Omega}_i \subset \Omega_i$. Only the A_i^{-1} local preconditioner of subdomain Ω_i will update any node of $\hat{\Omega}_i$. In other words, let us define:

$$\begin{aligned} \Omega_{i,0} &= \{j \in \Omega_i, \forall k \neq i, j \notin \Omega_k\} \cup \hat{\Omega}_i \\ (R_i^0 u)_j &= u \text{ if } j \in \Omega_i \text{ and } l \in \Omega_i, \text{ 0 else.} \end{aligned} \quad (3)$$

The RAS preconditioner writes:

$$M_{RAS}^{-1} = \sum_1^N R_i^0 A_i^{-1} R_i.$$

Thanks to the replacement of one of the R_i by R_i^0 , the product of residual by preconditioners can be locally computed without an extra communication. It has been also observed that RAS has generally a better conditioning and better convergence, see also an analysis in [13]. In contrast to the previous Decompositions 1 and 2, due to the choice of $R_{i,0}$, the preconditioner becomes a non-symmetric one and CG needs to be replaced by GMRES or BiCGStab. The inconvenient of GMRES is the storage of Krylov basis, which need to be limited to a "restart" dimension. The inconvenient of BiCGStab is a computational cost about two times higher than the other iterations, while, in general, BiCGStab's convergence is not two times faster.

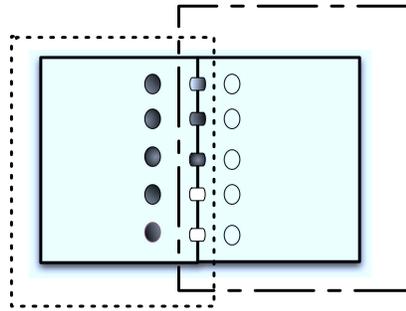


Figure 7. *Decomposition 3: Minimal domain decomposition of Cai et al.[7]: the matrix local solvers apply on sets of node inside bubbles, which have an overlap of one vertex row. In the case of Restricted Additive Schwarz version, updating in Domain 1 is restricted to black vertices, updating in Domain 2 is restricted to white vertices.(...): Domain 1. (- -): Domain 2.*

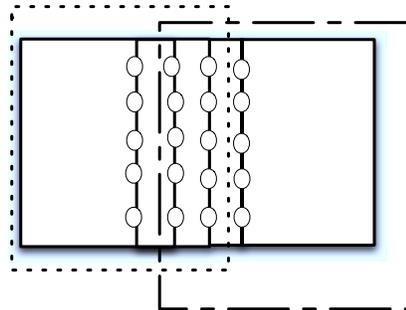


Figure 8. *Decomposition 4: Domain decomposition with two rows of vertex overlap. In the case of Restricted Additive Schwarz version, updating in Domain 1 (resp. 2) is restricted to vertices having no neighbors not belonging to Domain 1, (resp. 2)*

Decomposition 4, Two-row overlap. Decomposition 4 is a second version of the Restricted Additive Schwarz, but with an overlapping between subdomain that is thick of two node rows, Fig.8. The RAS preconditioner writes:

$$M_{RAS}^{-1} = \sum_1^N R_i^0 A_i^{-1} R_i^\delta.$$

Here, R_i^δ is the usual restriction corresponding to the overlapping set of subsets of Ω . Operator R_i^0 corresponds to the restriction to the corresponding nonoverlapping decomposition of Ω . The effect of using RAS instead of AS is to impose an iteration for non symmetric system, since, in contrast to M_{AS}^{-1} , M_{RAS}^{-1} is not symmetric. Therefore CG need be replaced by GMRES or BiCGStab. The second communication for preconditioned residual assembly is not necessary. This algorithm also enjoys the better conditioning of RAS.

In Fig.9 and 10, we shortly compare the above methods for a Neumann problem with a mesh of 10,000 nodes (*i.e.* vertices) and a partition into two subdomains. As remarked above, CG can be applied to Decompositions 1 and 2, and we observe that, although without any overlap, Decomposition 2 is converging in a reasonable way, not so slowly in comparison with Case 1. BiCGStab is used for the four Decompositions. Rather surprisingly, this choice is bad for Decomposition 2. Convergence of Decompositions 1 and 3 are good, similar to the CG convergence of Decomposition 1, but since BiCGStab is twice more complex, we would expect a better behavior. Convergence of Decomposition 4 is two times faster than the two previous ones.

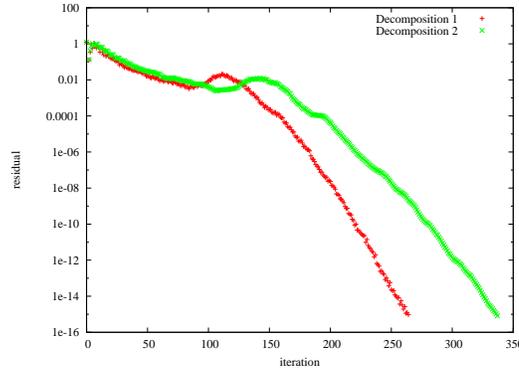


Figure 9. Application of a global conjugate gradient to Decomposition 2 (local preconditioning) and Decomposition 1 (only elementwise overlap)

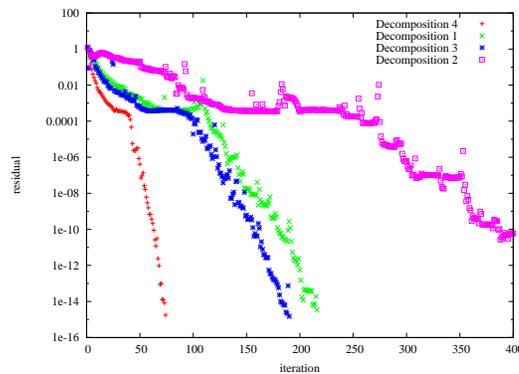


Figure 10. Application of a BiCGStab to (starting from the fastest convergence) Decomposition 4 with RAS, Decomposition 1, Decomposition 3 with RAS, Decomposition 2

2.5. Two-level Algorithm

We define now how the coarse grid is combined with a Schwarz algorithm. The two-level Additive Schwarz algorithm has two versions defined as follows:

Deflation:

- Apply a Conjugate Gradient (CG) with \bar{P}_D as preconditioner, with:

$$\bar{P}_D = M_{AS}^{-1} (I_n - A_h Z (Z^T A_h Z)^{-1} Z^T). \square \quad (4)$$

Balancing:

- Apply a Conjugate Gradient (CG) with P_B as preconditioner, with:

$$P_B = P_D^T M_{AS}^{-1} P_D + Z (Z^T A_h Z)^{-1} Z^T. \quad (5)$$

with M_{AS}^{-1} defined in (1). \square

The two-level Additive Schwarz-ILU algorithm has two versions defined by replacing in (4) or (5) M_{AS}^{-1} by M_{ASILU}^{-1} defined in (2).

In the two-level Restrictive Additive Schwarz (resp. Restrictive Additive Schwarz-ILU), the M_{AS}^{-1} (resp. M_{ASILU}^{-1}) is replaced by M_{RAS}^{-1} (resp. M_{RASILU}^{-1}) and the fixed point iteration is changed into GMRES.

3. NUMERICAL EVALUATION OF TWO LEVEL FORMULATIONS

3.1. Elliptic case

We present some performance evaluations for the Additive Schwarz and Additive Schwarz-ILU algorithms, for a partition described in “Decomposition 1” (see Section 2.4). The conjugate gradient is used as fixed-point. The test case is a Neumann problem with a discontinuous coefficient ρ :

$$-\nabla^* \frac{1}{\rho} \nabla p = RHS \text{ in } \Omega \quad \frac{\partial p}{\partial n} = 0 \text{ on } \partial\Omega \quad p(0) = 0. \quad (6)$$

The computational domain is a square. The coefficient ρ takes two values with a ratio 100., on two regions separated by the diagonal of the domain. The right-hand side is a *sin* function. DM and BDD produced essentially the same convergence rate. We present convergence statistics for a division of the residual by 10^{20} . Convergence at this level were sometimes problematic with DM, and the results are presented for BDD.

We recall first how behaves the *original Schwarz method* when the number of domains is fixed but the number of nodes increased. We compare in Table 1 a 2D calculation with two domains and 400 nodes with the analog computation with two domains and 10,000 nodes, which correspond to a h ratio of 5. We observe (Tab. I) that the convergence of a Schwarz-ILU is four times slower on the finer mesh. We also observe that the convergence of the Schwarz algorithm with exact sub-domain solution is also degraded by a factor 2.6, a loss which may be explained by the fact that with one layer overlapping, passing to a finer mesh makes a thinner overlapping.

Table I. *Additive Schwarz method*

# sub-domains	# cells	Local solver	# Iterations
2	400	ILU	55
2	400	Direct	28
2	10,000	ILU	221
2	10,000	Direct	74

Table II. *Scalability of the two-level AS-ILU method*

Cells	10K	20K	47K	94K
Domains	12	28	66	142
Cells/domain	833	714	712	661
Char. basis	480	546	750	810
Smooth basis	400	391	444	491

We continue with the study of the impact of choosing a *smooth basis* for the two-level Additive Schwarz ILU method. We observe (Tab. II) that the scalability again does not hold. Scalability is rather bad for the characteristic basis. Conversely, it is nearly attained for the smooth basis option. The rest of the paper uses only the smooth basis for the purely elliptic cases.

3.2. Advection dominated case

We evaluate now the extension to an advection model. This kind of models combines two main difficulties. First, These models correspond to Jordan matrices, and show 1D local behavior along advection trajectories, these are obstacles to the application of coarse grid correctors. Second, we have already noted the difficulty in developing a coarse basis applying to both convective and diffusive effects.

Advection-diffusion model. We consider solving the following partial differential equation :

$$\begin{cases} -\operatorname{div}(c(x)u) + \vec{b} \cdot \nabla u + u = f \text{ dans } \Omega \\ u = 0 \text{ sur } \partial\Omega \end{cases} \quad (7)$$

This model is discretised on a triangulation with nodes at vertices, in order to apply for diffusion terms the same P^1 finite element approximation as for the previous elliptic model. The convection

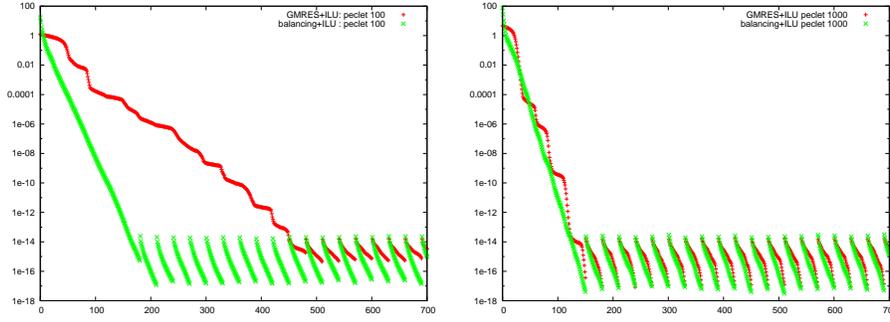


Figure 11. Application of the characteristic coarse basis on an advection diffusion model with Péclet=100 (left) and Péclet=1000 (left).

term $-\text{div}(\mathcal{V}(x)u)$ is approximated by an upwind first-order accurate vertex-centered finite volume formulated in dual cells built from triangle medians. We define the mesh Péclet number as $\text{Min} \frac{\|b\| \Delta x}{\nu}$ where the Min is taken over triangulation edges. We consider only Cartesian meshes ($\Delta x = \Delta y = \text{const.}$).

Regular partition of a square. A first experiment show the difficulty of the problem. We consider a Cartesian mesh of 10 000 nodes regularly partitioned in 16 subdomains. At Péclet 100, both coarse bases give an improvement but the characteristic one is already better. At Péclet 1000, the smooth basis gives a too slow convergence. Figure 11 depicts the characteristic case. The convergence characteristic one is only slightly better than the Schwarz iteration without coarse basis. However, it must be observed that in both cases the coarse grid allows a convergence of 14 decades in less than 150 iterations.

4. APPLICATION TO INCOMPRESSIBLE FLOW

4.1. Numerical scheme

The Navier-Stokes system for incompressible flow writes:

$$\rho \frac{\partial \mathbf{U}}{\partial t} + \rho \nabla \cdot (\mathbf{U} \otimes \mathbf{U}) = \nabla \cdot (v(\rho) \nabla \mathbf{U}) - \nabla p + \rho \mathbf{g} \text{ in } \Omega \quad (8)$$

$$\nabla \cdot \mathbf{U} = 0 \text{ in } \Omega \quad (9)$$

where \mathbf{U} denotes the fluid velocity, p the pressure, ρ the density, and $v(\rho)$ the viscosity. Let $V = \{\psi \in \mathcal{C}^0(\bar{\Omega}) \mid \psi|_K \text{ is affine } \forall K \in \mathcal{H}\}$ which is the usual P_1 Finite Element space. V is spanned by the set of basis functions ψ_i where ψ_i verifies for any vertex \mathbf{x}_i of \mathcal{H} , $\psi_i(\mathbf{x}_i) = 1$ and $\forall j \neq i$, $\psi_i(\mathbf{x}_j) = 0$. Let $\mathbf{V} = V^d$, where d is the space dimension. The discretized multi-fluid variables are:

$$\mathbf{U} = \sum_i \mathbf{U}_i \psi_i, \quad p = \sum_i p_i \psi_i \text{ and } \phi = \sum_i \phi_i \psi_i.$$

A transfer operator into \mathbf{V} is defined as follows: for any $u \in \mathbf{L}^2(\Omega)$, we denote by $\mathcal{P}u : \mathbf{L}^2 \mapsto \mathbf{V}$ the function such that for any vertex \mathbf{x}_i of \mathcal{H} :

$$\mathcal{P}u(\mathbf{x}_i) = \frac{\int_{\Omega} u \psi_i \, d\mathbf{x}}{\int_{\Omega} \psi_i \, d\mathbf{x}}.$$

And, for all $\mathbf{U} = (u, v) \in (\mathbf{L}^2(\Omega))^2$, we denote by $\mathcal{P}\mathbf{U} = (\mathcal{P}u, \mathcal{P}v)$ the transfer into \mathbf{V} . The transfer operator \mathcal{P} will be used for transforming a discrete field that is constant by element into a discrete field that is continuous and piecewise linear.

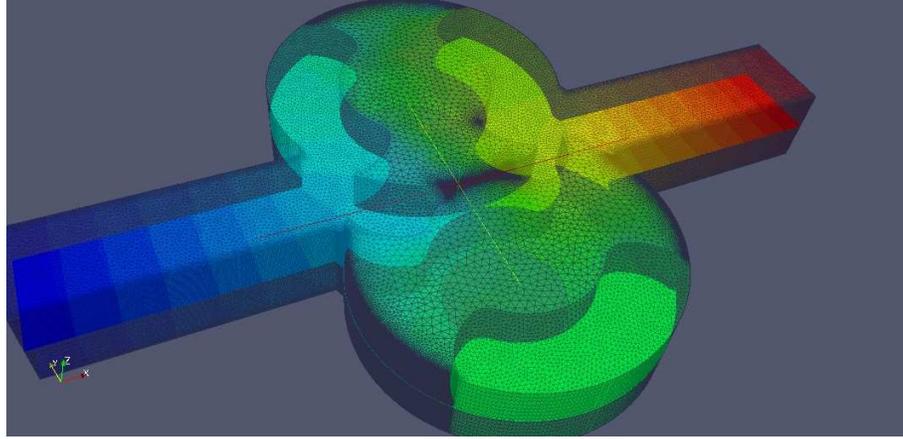


Figure 12. Mesh and pressure in a pump. Courtesy of PCM.

The global algorithm for advancing in time writes:

Stage 1 is an explicit prediction step:

$$\bar{\mathbf{U}}_i = \mathbf{U}_i^n - \frac{\Delta t}{|C_i|} \int_{\Omega} \psi_i (\nabla \cdot (\mathbf{U} \otimes \mathbf{U}) - \nu \nabla \cdot \nabla \mathbf{U} - \mathbf{g}) \, \mathbf{d}\mathbf{x} ,$$

where $|C_i| = \sum_j \int_{\Omega} \psi_i \psi_j \, \mathbf{d}\mathbf{x}$.

Stage 2 is a projection step imposing Relation (9):

$$\int \frac{1}{\rho} \nabla p^{n+1} \cdot \nabla \psi \, \mathbf{d}\mathbf{x} = \frac{1}{\Delta t} \int \nabla \psi \cdot \bar{\mathbf{U}} \, \mathbf{d}\mathbf{x} \quad \forall \psi \in V ,$$

$$\mathbf{U}^{n+1} = \bar{\mathbf{U}} + \Delta t \mathcal{P} \left(\frac{1}{\rho} \nabla p^{n+1} \right) \quad \text{and} \quad \mathbf{U}^{n+1} = 0 \quad \text{on} \quad \partial\Omega .$$

We observe that only the projection step needs the solution of a matrix system and that this matrix system is the same as in our model problem (6).

4.2. Original Solution Algorithm

The linear system arising from the projection step is solved with a RAS algorithm with an overlapping domain decomposition as defined by *Decomposition 4* above. In our algorithm, the GMRES iteration is applied.

4.3. Two level algorithm

A two-level version of the above algorithm is defined by combining the above DM preconditioner into the RAS one.

4.4. Example: Incompressible flow in a pump

We consider the steady flow through a pump. The geometry of the pump is depicted in Fig.12. It involves thin boundary layers. The mesh involves 2M cells and is distributed on 100 processors. We compare in (Tab.III) the efficiency of a single implicit pressure projection step with (1) a pure RAS-ILU preconditioner and (2) the same combined with DM. We observe that with the second option, convergence of the projection linear solver is 12 time faster in terms of iterations. the gain in efficiency is about a factor 9.7.

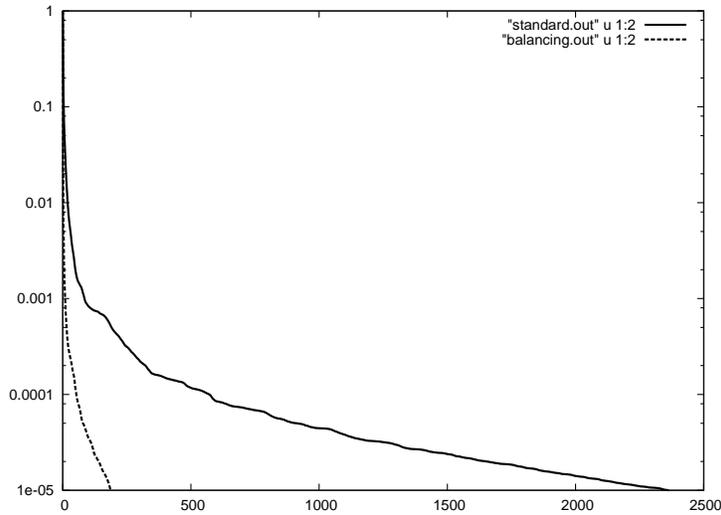


Figure 13. Mesh and pressure in a pump. Courtesy of PCM.

Table III. Flow through a pump, comparison of # of iterations for convergence

Type of preconditioner M^{-1}	# sub-domains	Iterations	CPU(sec.)
RA-Schwarz-ILU	40	2364	291
Balanced-RA-Schwarz-ILU	40	186	30

5. APPLICATION TO COMPRESSIBLE FLOW

5.1. Numerical scheme

The deflation and balancing preconditioners have been adapted to a software computing turbulent compressible flows. In the original numerical scheme, the spatial approximation is a vertex centered mixed-element-volume approximation stabilised by an upwind term introducing a sixth-order dissipation, see [8]. The flow equations are advanced in time with an implicit scheme, based on a second-order time-accurate backward difference scheme.

$$F(W^{n+1}, W^n, W^{n-1}) = 0 \quad (10)$$

Where W is the five-component discretisation of $(\rho, \rho \mathbf{u}, \rho E)$, where ρ is the density, \mathbf{u} the velocity, and ρE the total energy per unit volume. This non-linear system has to be solved at each time step to find W^{n+1} . It is solved by a (Newton-like) defect-correction iteration

$$A(W^{(\alpha+1)} - W^{(\alpha)}) = -F(W^{(\alpha)}, W^n, W^{n-1}) \quad (11)$$

in which a simplified Jacobian A is used. Since Equation (10) has 5 fields as unknown, A is defined as a block 5×5 sparse matrix. The Jacobian is built from the sum of a first-order discretisation of the linearized Euler fluxes and of a linearization of the second-order accurate diffusive fluxes. Typically, 2 defect-correction iterations are performed, each of them requiring two linear solutions. The performances of this algorithm has been studied for example in [15]. The most cpu consuming part of the algorithm is the resolution of the sparse linear system in (11). It is solved by 20 iterations of a Restricted Additive Schwarz (RAS) method, in the formulation proposed in [26], which we know describe. The linear system (11) is first transformed with the block 5 diagonal $D = \text{BlockDiag}(A)$ as follows:

$$D^{-1}A \delta W = D^{-1}f. \quad (12)$$

The domain decomposition is *Decomposition 3*. For the local, *i.e.* subdomain solve, an ILU(0) factorisation is applied to the product $D_i^{-1}A_i$. The preconditioner then writes:

$$M^{-1} = \sum_{i=1}^N R_i^0 ILU^{-1}(D_i^{-1}A_i)R_i^1. \quad (13)$$

This preconditioner is a right one, *i.e.* we solve

$$(D^{-1}A)M^{-1}v = D^{-1}f,$$

and then put $\delta W = M^{-1}v$. This keeps the same residual $D^{-1}A\delta W - D^{-1}f$ as for the unpreconditioned iteration. The RAS formulation (13) needs less communication (thanks to the use of R_i), and has proved to have better convergence properties than the analog AS formulation [7].

5.2. New linear solution algorithm.

Now DM or BDD are applied to the solution of (12). They are used as *right preconditioners*, and the residual is again the same as for unpreconditioner iteration. Using the same notation $E = Z'(D^{-1}A)Z$, we write the Deflation-RAS iteration by defining the following preconditioners:

$$\begin{aligned} P &= I - (D^{-1}A)ZE^{-1}Z' \\ Q &= I - ZE^{-1}Z'(D^{-1}A) \end{aligned}$$

then we solve:

$$(D^{-1}A)QM^{-1}v = PD^{-1}f$$

and finally put:

$$\begin{aligned} \delta\tilde{W} &= (M^{-1})v \\ \delta W &= ZE^{-1}Z'D^{-1}f + Q\delta\tilde{W}. \end{aligned}$$

The Balancing writes:

$$P_B = Z(E^{-1})Z' + QM^{-1}P \quad (14)$$

for solving:

$$(D^{-1}A)(P_B)v = (D^{-1})f$$

and then putting $u = (P_B)v$.

5.3. Example: Compressible flow around a cylinder

Test case. The compressible 3D flow (Mach=.1) around a cylinder with circular section is computed using a Smagorinsky Large Eddy Simulation. The Reynolds number is 20 000. The mesh involves 1.8 Million cells and is stretched near the cylinder wall with a maximum aspect ratio of 500. It is split into 64 to 1024 processors and we examine the convergence of a single implicit phase for a CFL of 100.

The flow is convection dominant and a characteristic coarse basis should be a reasonable choice. In a preliminary study, we try to evaluate and confirm the consistency of the characteristic coarse grid. For this, we introduce a manufactured solution ϕ in the linearised system. The function ϕ is a quadratic one for each component. We define the RHS as the corresponding residual, in such a way that the solution of the linearised system is exactly ϕ . The coarse grid defined from 64 subdomains. In Fig.15 we show the coarse grid solution. It matches quite well with the manufactured quadratic function.



Figure 14. VMS Large Eddy Simulation of a turbulent flow around a cylinder, $Re=20000$, with 1.8 million cells; vorticity

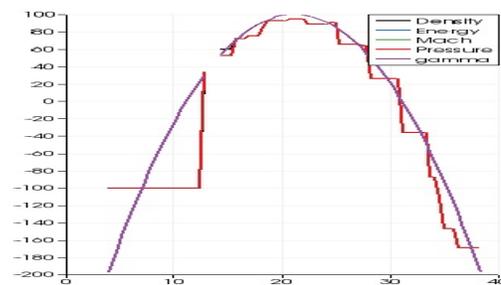


Figure 15. Evaluation of coarse-grid consistency for a characteristic basis: comparison of a quadratic discrete solution with its coarse grid approximation

Performance of the original algorithm. We first study the strong scalability of the original RAS algorithm. In Fig.16, we observe that the convergence degrades, with a number of iteration 29% larger when the number of subdomain is doubled, which expresses a lack of scalability. See also Tabs.IV and V.

Performance of the two-level algorithms. From Fig.16 and Tabs.IV and V, we observe that the scalability is better than 1.

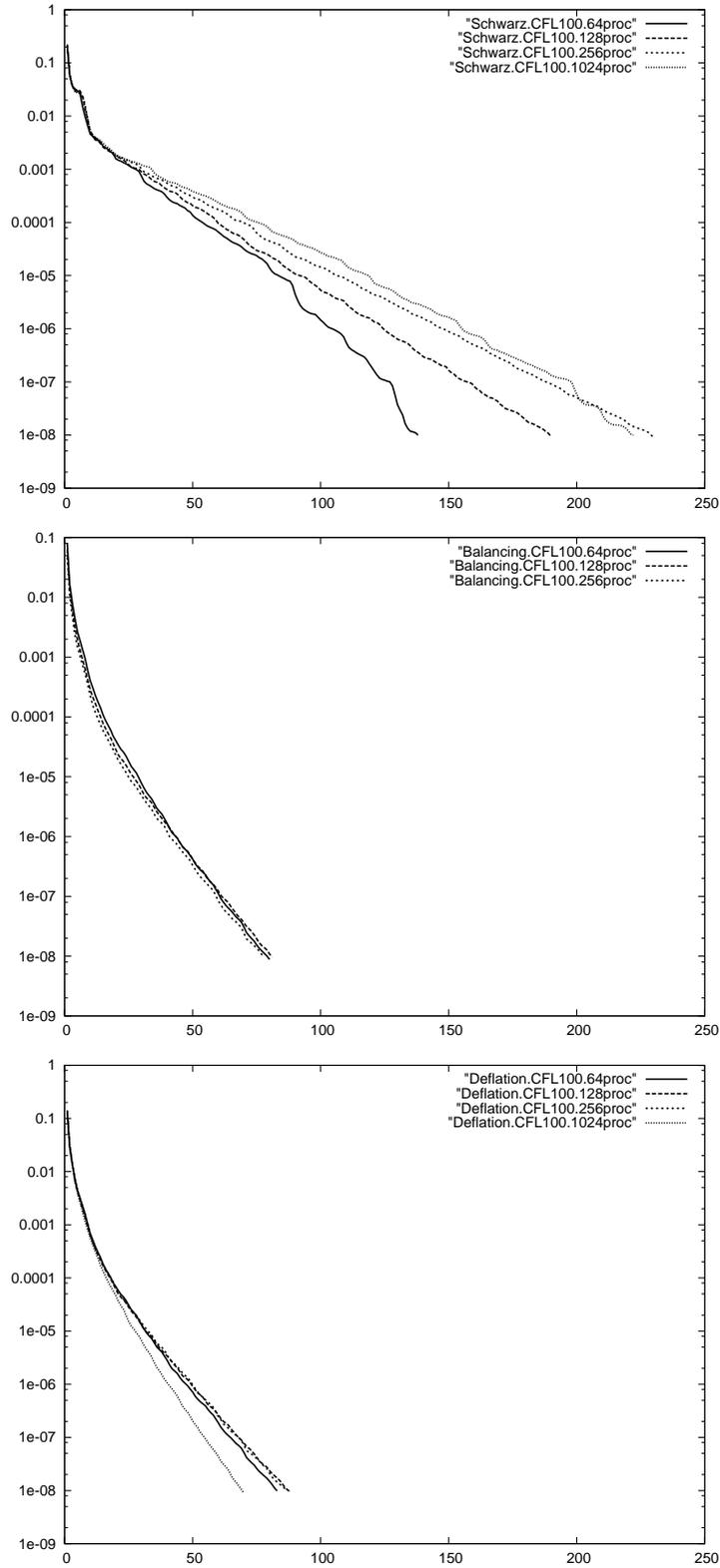


Figure 16. Compressible LES: convergence during one time step ($CFL=100$) of the RAS, RAS with Deflation, RAS with Balancing .

Table IV. *Compressible Navier-Stokes simulation, residual $\times 10^{-8}$*

Type of preconditioner	CFL	64procs # it.	128procs # it.	256 procs # it.	1024procs # it.
Schwarz-ILU	100	138	190	230	222
Deflated Schwarz-ILU	100	83	88	87	70
Balanced Schwarz-ILU	100	80	81	78	

Table V. *Compressible Navier-Stokes simulation, residual $\times 10^{-8}$*

Type of preconditioner	CFL	128procs # it.	256 procs # it.	Sca. Factor
Schwarz-ILU	100	190	230	1.21
Deflated Schwarz-ILU	100	88	87	.988
Balanced Schwarz-ILU	100	81	78	.963

6. CONCLUDING REMARKS

The building of a coarse grid for deflated or balanced formulations are presented. We study the effect of coarse-grid consistency. Choosing a consistent coarse grid with smooth basis functions can help for a better scalability in the case of a diffusion dominated model. The case of diffusion-convection is better addressed with characteristic bases for Péclet as small as 100. Probably, developing a zonal strategy adapted to phenomena in which part of the domain is convection dominated and part of the domain diffusion dominated might be a source of extra efficiency.

An application of the two-level method is presented for the elliptic part of an incompressible flow. The best option is Deflation. The gain in efficiency is already 5 for a medium number of processors, e.g. 100. An application to a compressible flow is then presented. Although Balancing is converging a little faster, the most efficient option (in term of CPU) is Deflation. The improvement factor in convergence is smaller than for the elliptic case, even for a number of processors as high as 1024, but still interesting (about 3). In particular, weak and strong scalability is observed, which tend to show that a higher factor can be reached for larger computations.

7. ACKNOWLEDGEMENTS

We thank Guillaume Houzeaux and Mariano Vazquez for the fruitful discussion we had with them. This work has been supported by French National Research Agency (ANR) through COSINUS program (project ECINADS n^o ANR-09-COSI-003). HPC resources from GENCI-[CINES] (Grant 2010-x2010026386 and 2010-c2009025067) are also gratefully acknowledged.

REFERENCES

1. R. Aitbayev, X.-C. Cai, and M. Paraschivoiu. Parallel two-level methods for three-dimensional transonic compressible flow simulations on unstructured meshes. In J. Periaux D. E. Keyes, A. Ecer and N. Satofuka, editors, *Proceedings of Parallel CFD'99, Williamsburg, Virginia, May 23-26, 1999*. Elsevier, 2000.
2. R. Aubry, F. Mut, R. Lohner, and J. R. Cebal. Deflated preconditioned conjugate gradient solvers for the pressure-Poisson equation. *Journal of Computational Physics*, 227:24:10196–10208, 2008.
3. J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1–22, 1990.
4. S.C. Brenner. Two-level additive Schwarz preconditioners for plate elements. *Wuhan University Journal of Natural Sciences*, 1:658–667, 1996.

5. X.-C. Cai, C. Farhat, and M. Sarkis. A minimum overlap restricted additive Schwarz preconditioner and applications in 3D flow simulations. In C. Farhat J. Mandel and AMS X.-C. Cai, eds, editors, *The Tenth International Conference on Domain Decomposition Methods for Partial Differential Equations, August 10-14, 1997 Boulder, Colorado*, volume 218. Contemporary Mathematics, 1998.
6. X. C. Cai, D. E. Keyes, and M. D. Tidriri. Newton-Krylov-Schwarz methods in CFD. In *Proceedings of the International Workshop on Numerical Methods for the Navier-Stokes Equations*, pages 17–30, 1994.
7. X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21:792–797, 1999.
8. S. Camarri, B. Koobus, M.-V. Salvetti, and A. Dervieux. A low-diffusion MUSCL scheme for LES on unstructured grids. *Computers and Fluids*, 33:1101–1129, 2004.
9. G. Carre, L. Fournier, and S. Lanteri. Parallel linear multigrid algorithms for the acceleration of compressible flow calculations. *Computer Methods in Applied Mechanics and Engineering*, 184(2:22):427–448, 2000.
10. F. Courty and A. Dervieux. Multilevel functional preconditioning for shape optimisation. *I.J. CFD*, 20:7:481–490, 2006.
11. H. A. Van der Vorst. Bi-CGstab: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 13, pages 631–644, 1992.
12. R. Walters E. Nielsen, W. K. Anderson and D. Keyes. Application of Newton-Krylov methodology to a three dimensional unstructured Euler code. In *AIAA 95-1733-CP, June, 1995*.
13. A. Frommer and D.B. Szyld. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM J. Numer.Anal.*, 39:2:463–479, 2001.
14. H. Guillard, A. Janka, and P. Vanek. Analysis of an algebraic Petrov- Galerkin smoothed aggregation multigrid method. *Applied Numerical Mathematics*, (58):18611874, 2008.
15. B. Koobus, S. Camarri, M.V. Salvetti, S. Wornom, and A. Dervieux. Parallel simulation of three-dimensional complex flows: Application to two-phase compressible flows and turbulent wakes. *Adv. Eng. Software*, 38:328–337, 2007.
16. B. Koobus and M.-H. Lallemand A. Dervieux. Unstructured volume-agglomeration mg: solution of the Poisson equation. *I.J. for Numerical Methods in Fluids*, 18:27–42, 1994.
17. A.-C. Lesage, O. Allain, and A. Dervieux. On level set modelling of bi-fluid capillary flow. *Int.J. Numer. Methods in Fluids*, 53:8:1297–1314.
18. R. Lohner, F. Mut, J.R. Cebral, R. Aubry, and G. Houzeaux. Deflated preconditioned conjugate gradient solvers for the pressure-Poisson equation: Extensions and improvements. *Int. J. Numer. Meth. Engng, Special Issue: In Memory of Professor Olgierd C. Zienkiewicz (19212009)*, 87, Issue 1-5:2–14, 2011.
19. J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9:233–241, 1993.
20. N. Marco, B. Koobus, and A. Dervieux. An additive multilevel preconditioning method and its application to unstructured meshes. *Journal of Scientific Computing*, 12:3:233–251, 1997.
21. C.R. Nastase and D. J. Mavriplis. High-order discontinuous galerkin methods using an hp-multigrid approach. *Journal of Computational Physics*, 213:1:330–357, 2006.
22. F. Nataf, H. Xiang, and V. Dolean. A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps. *C. R. Mathématiques*, 348:21-22:1163–1167, 2010.
23. F. Nataf, H. Xiang, V. Dolean, and N. Spillane. A coarse space construction based on local Dirichlet to Neumann maps. *to appear in SIAM J. Sci. Comput.*, 2011.
24. R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problem. *SIAM J.Numer.Anal.*, 24:355–365, 1987.
25. Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc’H. A deflated version of the conjugate gradient algorithm. *SIAM J. Sci. Comput.*, 21:5:1909–1926, 2000.
26. M. Sarkis and B. Koobus. A scaled and minimum overlap restricted additive Schwarz method with application on aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 184:391–400, 2000.
27. P. Le Tallec, J. Mandel, and M. Vidrascu. Balancing domain decomposition for plates. *Eighth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Penn State, October 1993, Contemporary Mathematics, AMS, Providence*, 180:515–524, 1994.
28. J.M. Tang, S.P. MacLachlan, R. Nabben, and C. Vuik. A comparison of two-level preconditioners based on multigrid and deflation. *SIAM J. Matrix Anal. Appl.*, 31:4:1715–1739, 2010.
29. C. Vuik and R. Nabben. A comparison of deflation and the balancing preconditionner. *SIAM J. Sci. Comput.*, 27:5:1742–1759, 2006.