Ecinads-D3.3: Efficiency of three-level Schwarz Algorithm

H. Alcin, O. Allain, B. Koobus, A. Dervieux

Abstract The use of volume-agglomeration for introducing one or several levels of coarse grids in an Additive Schwarz 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 coefficicent. The two options are compared for the standpoint of coarse-grid consistency and for the gain in scability of the global Schwarz iteration.

Key words: domain decomposition, coarse grid
MSC2010: 65F04, 65F05

H. Alcin
INRIA, B.P. 93, 06902 Sophia-Antipolis, France, e-mail: Hubert.Alcin@inria.fr

O. Allain
LEMMA, Les Algorithmes (Le Thales A), 2000 route des Lucioles,06410 BIOT, France, e-mail: olivier.allain@lemma-ing.com

B. Koobus
University of Montpellier 2, CC51,Montpellier, France,e-mail: koobus@math.univ-montp2.fr

A. Dervieux
INRIA, B.P. 93, 06902 Sophia-Antipolis, France, e-mail: Alain.Dervieux@inria.fr

1 Volume agglomeration in MG and DDM

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, right. In order to build a coarser grid, it is possible to

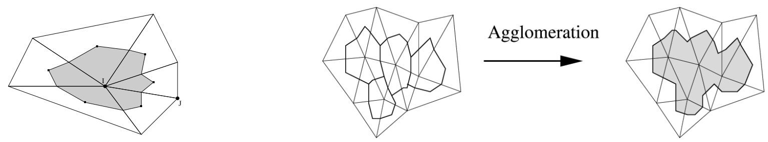

Fig. 1 *Left: finite-Volume partition built as dual of a triangulation. Right: Greedy Algorithm for finite-volume cell agglomeration: four fine cells (left) are grouped into a coarse cell*

build coarse cells by sticking together neighboring cells for example with a greedy algorithm, Figure 1, left. 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.

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 [19]. The extension of Agglomeration MG to multi-processor parallel computing, however, are less easily achieved, as compared to Domain Decomposition Methods.

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 [16], the agglomeration basis is extended to H^1 consistent ones in an analog way to smoothed-aggregation. In [7], a Bramble-Pasciak-Xu algorithm is built on these bases for an optimal design application.

While MG appeared, at least for a while, as the best CFD solution algorithm, Domain Decomposition methods (DDM) were seen as a new star for computational Structural Dynamics due to matrix stiffness issues. Domain decomposition methods assume the partition of the computational domain into sub-domains and assume that representative sub-problems on sub-domains can be rather easily computed and help convergence towards global problem's solution. An ideal DDM should be weakly scalable, that is, when it produces in some time with p processors a result on a given mesh, the result on a two times larger mesh should be produced in the same time with $2p$ processors. In Schwarz DDM, The set of local problems preconditions the global loop. Boundary conditions for each sub-domain problem are fetched in neighboring domains. The resulting iterative solver generally involves a Krylov iteration and is often refered as Newton-Krylov-Schwarz. It has been shown by S. Brenner [4] that the resulting algorithm is not scalable, unless a extension called coarse grid is added. In [4], the coarse grid correction is computed on a particular coarser mesh, embedded into the main mesh. 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 [21]. For CPU cost reasons, these eigenvectors should not be exactly computed but only approximated. In a recent study [17],[18], 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 dominent Jordan behaviour (as for convection dominent models, the privilegiated domain of finite-volume methods). 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. The first test problem we concentrate on is inspired by a pressure-correction phase in compressible Navier-Stokes calculations (see for example [12]), and expresses as a Neumann problem with strongly discontinuous coefficient and writes:

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

in which the well-posedness is fixed with a Dirichlet condition on one cell.
The second test problem on which we concentrate is the linearised compressible Navier-Stokes system to be solved at every iteration of a time-implicit unsteady LES simulation. This system is expressed in terms of the Jacobian of Navier-Stokes fluxes. The Jacobian is discretised with spatially first-order linearised Riemann solvers.

1.1 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:

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

$$A_i = R_i A R_i.$$

The *Additive Schwarz* algorithm is written in terms of preconditioning, as

$$M^{-1} = \sum_{i=1}^{2} A_i^{-1}.$$

The preconditioner M^{-1} can be used in a Krylov subspace method. In this paper, in order to keep some generality in our algorithms, we use GMRES, also used in [21]. In the *Additive Schwarz-ILU* version, the exact solution of the Dirichlet on each subdomain is replaced by the less costly Incomplete Lower Upper (ILU) approximate solution. 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 .□

This factor is measured for a given PDE, with a given mesh (strong scalability) or 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.□

Lemma:[?] A Schwarz method as defined above is not scalable.□

1.2 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 complexity to an essentially scalable one. Two methods have been proposed in the literature for introducing a coarse grid in an *algebraic* manner. 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 Nicolaides [20] and is used by many authors. Saad *et al.*[21] encapsulate it into a Conjugate Gradient. Aubry *et al.* [1, 2] apply it to a pressure Poisson equation. In DM, the projection operator is defined as:

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

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 has been introduced by J. Mandel [15] and applied to a complex system in [13]. In [22] a formulation close to DM is proposed. It consists in replacing the preconditioner M^{-1} (ex.: global ILU, Schwarz, or Schwarz-ILU) by:

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

1.3 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 [16], the coarse system

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

produces a solution U^H which does not converge towards the continous 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 which will support the preconditioner. Level j is made of N_j macro-cells C_{jk}, *i.e.* $\mathscr{G}_j = \cup_{k=1}^{N_j} C_{jk}$. Transfer operators are defined between successive levels (from coarse to fine):

$$P_i^j \; : \; \mathscr{G}_i \rightarrow \mathscr{G}_j \qquad P_i^j(u)(C_{k'i}) = u(C_{kj}) \;\; \text{with} \;\; C_{k'i} \subset C_{kj}$$

Following [16] we introduce the smoothing operator:

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

where $\mathscr{N}(i)$ holds for the set of cells which are direct neigbors 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 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.

Fig. 2 *Left: characteristic coarse grid basis function. Right: smooth coarse grid basis function*

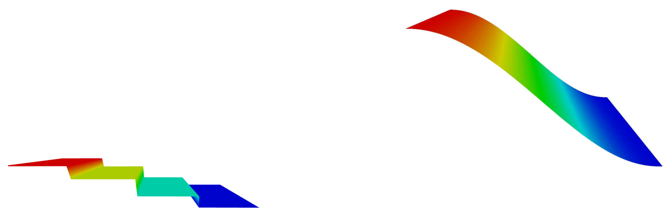

Fig. 3 *Accuracy of the coarse grid approximation for a Poisson problem with a sin function (of amplitude* 2.*) as exact solution. Left: coarse grid solution with the characteristic basis (amplitude is* 0.06*). Right: coarse grid solution with a smooth basis (amplitude is* 1.8*).*

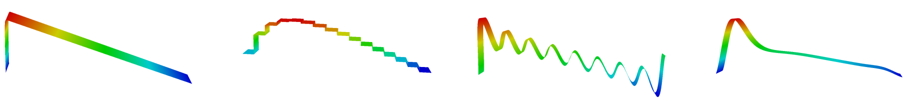

Fig. 4 *Accuracy of the coarse grid approximation for an advection-diffusion problem: (a) fine grid solution, (b) coarse solution with characteristic basis, (c) coarse solution with smooth basis,(d) coarse solution with smooth basis and numerical viscosity.*

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.3a. The characteristic basis produces a not so bad approximation (Fig.3b) 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.3c). 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.3d).

1.4 Two-level Algorithm

We define now how the coarse grid is combined with a Schwarz algorithm.

To fix the ideas, we assume that the decomposition $\Omega_1, ...\Omega_N$ 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 refered below.

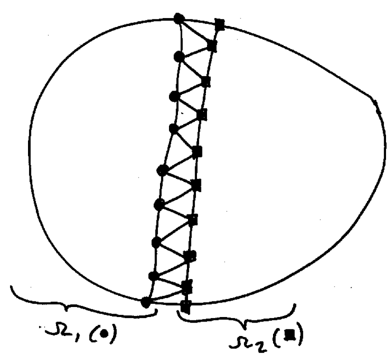

Fig. 5 *Domain decomposition: domain 1 involves "round" vertices, domain 2 involves "square" vertices*

We note in passing that this minimalist option degrades the scalability of the Schwarz algorithm since the overlapping width decrease 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 (**??**), resp. M_{ASILU}, defined according to (**??**), as preconditioner.

Our two-level algorithms are defined as follows:
- Apply a Conjugate Gradient (CG) with $\bar{P}_D$ or P_B as preconditioner, with:
Deflation: $\bar{P}_D = M_{AS}^{-1}(I_n - A_h Z(Z^T A_h Z)^{-1} Z^T)$.
Balancing: $P_B = P_D^T M_{AS}^{-1} P_D + Z(Z^T A_h Z)^{-1} Z^T$
where again the *ILU* variant will be also examined.

1.5 Three-level Algorithm

Because the local solver is not an exact one but an ILU solver, computing with a larger number of nodes in each sub-domain leads to a degradation of the convergence. It is then interesting to add a coarse grid on each sub-domain. This principle has been investigated in [14], where the authors use a non-smoothed aggregated basis.

Our proposition is to build sub-domain bases which are consistent with the Dirichlet condition of the Schwarz interface condition. To satisfy this, the Dirichlet condition is introduced in each smoothing step of the smooth basis function building process. Fig.6 depicts the resulting local coarse solutions of the Dirichlet problem.

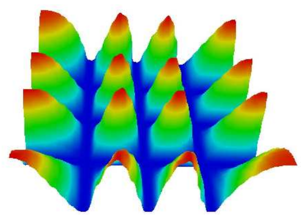

Fig. 6 *A view of the 16 median-coarse solutions*

The *global algorithm* is made of a GMRES iteration preconditioned by the P_B operator combining a global coarse system with sub-domain preconditioners. The latter ones combine the local medium basis and the local ILU solver.

We apply the deflation principle to the subdomain problem:

$$(A_i)^{-1} \approx P_{D_i}^T (ILU_i)^{-1} P_{D_i} + Z_i E_i^{-1} Z_i^T$$

where P_{D_i} is introduced for the projection of problem in V_i into a smaller one. Combinin with the global grid we get:

$$\sum_{i=1}^{N} P_D^T \; P_{D_i}^T (ILU_i)^{-1} P_{D_i} + Z_i E_i^{-1} Z_i^T \; P_D + Z E^{-1} Z^T$$

Another option consists in directly enrich the coarse basis, with the disadvantage of having a much larger coarse system to solve. The two options are compared in a test case depicted in Fig.7, which indicates that they do not much differ but provide an important acceleration with resoect to the two-leve formulation.

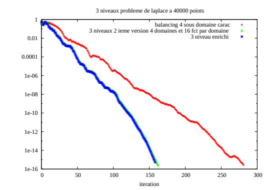

Fig. 7 *A comparison between the 2-level method and the two proposed 3-level methods. The 3-level algorithm using a unique coarse grid basis is only 2% faster than the 3-level algorithm solving in each subdomain the medium grid system. Both are much faster than the 2-level algorithm.*

2 Numerical evaluation

2.1 Elliptic case

We present some performance evaluations for the proposed algorithm. In all cases the conjugate gradient is used as fixed-point. The test case is a Neumann problem with discontinuous coefficient as in Section 2.1. The computational domain is a square. The coefficient 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. In the sequel, convergence is always measured for a division of the residual by 10^{20}. Convergence at this level were problematic with DM and the results are presented for BDD.

We recall first how behaves the *original Schwarz method* with one layer overlapping 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 (Table 1) 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 thinner overlapping.

We continue with the study of the impact of choosing a *smooth basis* for the two-level Additive Schwarz ILU method. We observe that the scalability again does not hold, but it is nearly attained for the smooth basis option. It is rather bad for the characteristic basis. The rest of the paper uses only the smooth basis.

Table 1 *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 2 *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

2.2 Advection dominated case

The difference between characteristic basis and smooth one is not clear for the Peclet-100 advection-diffusion model. One difficulty is the insufficient local resolution by ILU which induces a plateau in the convergence. The introduction of coarse-grid dissipation did not carry any improvement.

2.3 Impact of the medium grid

The impact of the *medium grid* is examined in a third series of experiment is performed on a mesh of $40,000$ cells, with 4 sub-domains and a total of 64 medium basis function (8 per sub-domain). In Table 3, we observe that without a coarse grid, the Schwarz-ILU solver is 20% slower than the global (1-sub-domain) ILU solver (in terms of iteration count for 20 decades), the Schwarz-ILU with coarse-grid is slightly faster and the three level is 30% faster.

Table 3 *Convergence of the different preconditioners (*$40,000$ *cells)*

Type of preconditioner M^{-1}	# sub-domains	Iterations
Global ILU	1	348
Schwarz-ILU	4	431
Schwarz-ILU+coarse-grid	4	334
Three-level	4	264
Three-level	16	164

The *speedup* is measured for a given problem, set on a mesh of $40,000$ cells. We compare the iteration count between a 4-sub-domain computation and a 16-sub-domain one. The coarse system solution with 16 unknowns is not parallel, but its cost is very small. Using four times more processors turn into a 6.4 smaller number of iterations before obtaining the solution (Table 3).

For a *scalability* measure, the mesh is taken finer and the number of sub-domain increased accordingly. We compare a $40,000$-cell computation on 4 processors with a $160,000$-cell on with 16 processors. We would like to mention that the Schwarz method with exact sub-domain resolution is far from being scalable: in Table 4, increase in iteration count is 40%. These bad news were announced by Table 1.

We turn the combination of the Schwarz method with our smooth coarse grid. Exact solution is again performed on each sub-domain. Convergence becomes at least twice better. However, passing from $40,000$ cells with 4 sub-domains to $160,000$ cells with 16 sub-domains increases the iteration count by 60%, Table 4. We have checked that results with characteristic coarse grid are worse. In order to perform the analog comparison for the proposed three-level method (smooth coarse grid,

Table 4 *Scalability (10K cells/processors, residual* $*10^{-10}$

Method	# cells	# sub-domains	# medium basis fonct	# it.	Scalability factor
Exact-Schwarz	40,000	4		91	
Exact-Schwarz	160,000	16		265	1.7
Schwarz-ILU	40,000	4		354	
Schwarz-ILU	160,000	16		650	1.35
Schwarz-ILU(1)	40,000	4		252	
Schwarz-ILU(1)	160,000	16		530	1.45
2-lev.Schwarz-ILU	40,000	4		275	
2-lev.Schwarz-ILU	160,000	16		347	1.12
3-lev.Schwarz-ILU	40,000	4	4×16	164	
3-lev.Schwarz-ILU	160,000	16	16×16	176	1.015

smooth medium grid, ILU), we specify a four times higher number of medium-grid basis functions for the computation with four times higher number of cells (and sub-domains). Scalability in iterations is nearly satisfied, with 7% loss, Table 4.

2.4 Advection dominated case

The algorithm also work in case of advection dominated floes. In some case where the partition is stretched in the direction of advection, the baseline Schwarz-ILU iteration is very poorly converging. A Cartesian mesh on a rectangle is considered, with 105 000 cells. The two-level one carries an improvement but is still rather slow. In contrast, the three level converges as fastly as usually. In the case depicted in Fig.9, the three level convergence is 4.5 better than the two-level one.

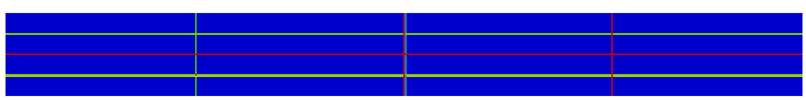

Fig. 8 *Stretched partition of a rectangle*

3 Concluding remarks

The building of a coarse grid for deflated or balanced formulation is presented. We study the effect of coarse-grid consistency. Choosing a consistent coarse grid with

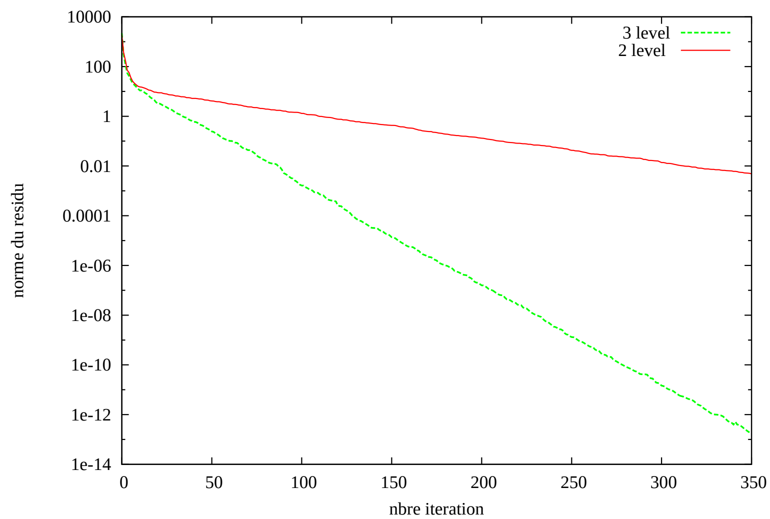

Fig. 9 *Convergence acceleration provided by the median-coarse grid for the stretched partition*

smooth bais functions can help for a better scalability in the case of a diffusion dominated model. For a convection dominated one, from one hand, a non-smooth basis may be consistent, from the other hand, the smooth basis is of more delicate use. In the future, we shall look for the best strategy for phenomena in which part of the domain is convection dominated and part of the domain diffusion dominated. We have examined the introduction of a median-coarse grid on each subdomain. This kind of method may further reinforce the overall scalability when the local preconditioner is of ILU type.

4 Acknowledgements

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. Aubry , F. Mut , R. Lohner , J. R. Cebral, Deflated preconditioned conjugate gradient solvers for the Pressure-Poisson equation, Journal of Computational Physics, 227:24, 10196-10208, 2008.

2. 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 (2010)
3. J. Bramble, J. Pasciak, and J. Xu, Parallel multilevel preconditioners, Math. Comput., 55(191):122, 1990.
4. S. Brenner, Two-level additive schwarz preconditioners for plate elements, Numerische Mathematik, 72:4, 1994
5. X.-C. Cai and M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput., 21 (1999), pp. 792-797.
6. 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, vol. 184, 2000, pp. 391-400.
7. F. Courty, A. Dervieux, Multilevel functional Preconditioning for shape optimisation, IJCFD, 20:7,481-490,2006
8. B. Koobus, S. Camarri, M.V. Salvetti, S. Wornom, A. Dervieux, Parallel simulation of three-dimensional flows: application to turbulent wakes and two-phase compressible flows, Advances in Engineering Software, 38, 328-337, 2007
9. M. Sarkis and B. Koobus. A scaled and minimum overlap restricted additive Schwarz method with application to aerodynamics. *Comput. Methods Appl. Mech. Eng.*, 184:391–400, 2000.
10. 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.
11. 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.
12. 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, 2007.
13. P. Le Tallec, J. Mandel, M. Vidrascu, Balancing Domain Decomposition for Plates, Eigth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Penn State, October 1993, Contemporary Mathematics, 180, AMS, Providence, 1994, 515-524.
14. P.T . Lin, M. Sala, J.N. Shadi, R S. Tuminaro, Performance of Fully-Coupled Algebraic Multilevel Domain Decomposition Preconditioners for Incompressible Flow and Transport
15. J. Mandel, Balancing domain decomposition, Comm. Numer. Methods Engrg., 9, 233-241, 1993.
16. N. Marco, B. Koobus, A. Dervieux, An additive multilevel preconditioning method and its application to unstructured meshes, INRIA research report 2310, 1994 and Journal of Scientific Computing, 12:3, 233-251, 1997
17. F. Nataf, H. Xiang, V. Dolean, A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps, C. R. Mathématiques, 348:21-22, 1163-1167, 2010
18. F. Nataf, H. Xiang, V. Dolean, N. Spillane, A coarse space construction based on local Dirichlet to Neumann maps, to appear in SIAM J. Sci Comput., 2011.
19. C.R. Nastase, D. J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, Journal of Computational Physics 213:1, 330-357, 2006.
20. R. A. Nicolaides, Deflation of conjugate gradients with applications to boundary value problem , SIAM J.Numer.Anal., 24, 355-365, 1987.
21. 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.
22. C. Vuik , R. Nabben A comparison of deflation and the balancing preconditionner, SIAM J. Sci. Comput., 27:5, 1742-1759, 2006.