Volume-Agglomeration Coarse Grid In Schwarz Algorithm

H. Alcin, O. Allain, and 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 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.

Keywords domain decomposition, coarse grid
MSC2010: 65F04, 65F05

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 the finite-volume partition considered is built as the dual of triangles, Fig. 1, right. In order to

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

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

J. Fořt et al. (eds.), *Finite Volumes for Complex Applications VI – Problems & Perspectives*, Springer Proceedings in Mathematics 4,
DOI 10.1007/978-3-642-20671-9_1,

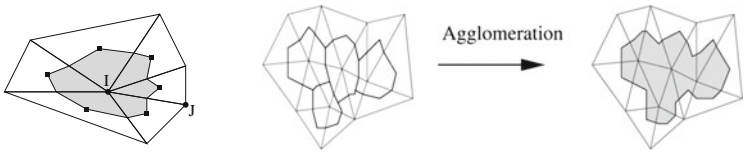

Fig. 1 Left: finite-Volume partition built as dual of a triangulation. Right: Greedy Algorithm for finite-volume cell agglomeration: four fine cells (left) are grouped into a coarse cell

build a coarser grid, it is possible to build coarse cells by sticking together neighboring cells for example with a greedy algorithm, Fig. 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 [13]. 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 [2] 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 [10], the agglomeration basis is extended to H^1 consistent ones in an analog way to smoothed-aggregation. In [4], 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 [3] that the resulting algorithm is not scalable, unless a extension called coarse grid is added. In [3], 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 [15]. For CPU cost reasons, these eigenvectors should not be exactly computed but only approximated. In a recent study [11, 12], 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 test problem we concentrate on is inspired by a pressure-correction phase in Navier-Stokes (see for example [6]), and expresses as a Neumann problem with strongly discontinuous coefficient and writes:

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

in which the well-posedness is fixed with a Dirichlet condition on one cell.

1.1 Basic Additive exact and ILU Schwarz algorithm

Our discrete model relies on a vertex-centered formulation expressed on a triangulation. Let us assume that the computational domain Ω is split into two sub-domains, Ω_1 and Ω_2 , with an intersection $\overline{\Omega_1} \cap \overline{\Omega_2}$ with a thickness of at least one layer of elements. The *Additive Schwarz* algorithm is written in terms of preconditioning, as $M^{-1} = \sum_{i=1}^{2} A_{|\Omega_i}^{-1}$ where $A_{|\Omega_i}^{-1}$ holds for the Dirichlet problem on sub-domain Ω_i. 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 [15]. 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.

1.2 Algebraic Coarse grid

As shown by S. Brenner [3], 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 [14] and is used by many authors. Saad *et al.* [15] encapsulates it into a Conjugate Gradient. Aubry *et al.* [1] 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 [9] and applied to a complex system in [7]. In [16] 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 [10], the coarse system

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

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.* $\mathcal{G}_j = \cup_{k=1}^{N_j} C_{jk}$. Transfer operators are defined between successive levels (from coarse to fine):

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

Following [10] we introduce the smoothing operator:

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

where $\mathcal{N}(i)$ 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 Fig. 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, Fig. 3.

1.4 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 [8], where the authors use a non-smoothed aggregated basis.

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

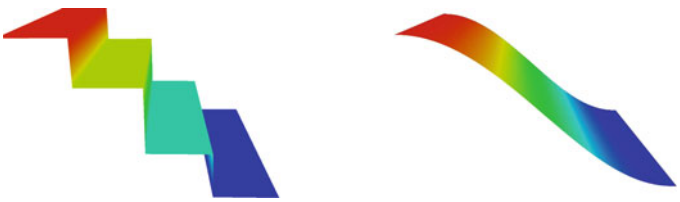

Fig. 3 Accuracy of the coarse grid approximation for a Poisson problem with a sin function (of amplitude 2.) as exact solution. Left: coarse grid solution with the characteristic basis (amplitude is 0.06). Right: coarse grid solution with a smooth basis (amplitude is 1.8)

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.

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.

2 Numerical evaluation

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

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,

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 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.

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

Table 4 Scalability for the Schwarz, two-level Schwarz and three level Schwarz-ILU

Method	# cells	# sub-domains	# medium basis funct	Iterations
Schwarz	40,000	4		320
Schwarz	160,000	16		451
two-level Schwarz	40,000	4		130
two-level Schwarz	160,000	16		212
Three level	40,000	4	64	164
Three level	160,000	16	256	176

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, 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.

3 Concluding remarks

We have proposed a three-level algorithm for solving a linear system with a Schwarz method. The basis functions are independent of the system to solve and building them is not computationally expensive. The coarse grid solution is obtained after one iteration and yields a good initial solution. A few preliminary results show that the proposed method appears to be suitable for a pressure-projection system. The CPU cost (measured on a 2.6GHz workstation) for the heaviest example is of $0.05nS$ for the coarse factorization, $660nS$ ($20nS$ per processor) for the coarse system assembly while the Schwarz preconditioner cost is $124\mu S$. Further measures and applications to convection-diffusion models are in progress, as well as the introduction into a compressible Navier-Stokes model, [5].

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. J. Bramble, J. Pasciak, and J. Xu, Parallel multilevel preconditioners, Math. Comput., 55(191):122, 1990.
3. S. Brenner, Two-level additive schwarz preconditioners for plate elements, Numerische Mathematik, 72:4, 1994.
4. F. Courty, A. Dervieux, Multilevel functional Preconditioning for shape optimisation, IJCFD, 20:7, 481-490, 2006.
5. 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.
6. 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.
7. 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.
8. 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.
9. J. Mandel, Balancing domain decomposition, Comm. Numer. Methods Engrg., 9, 233-241, 1993.
10. 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.
11. 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.
12. 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.
13. 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.
14. R. A. Nicolaides, Deflation of conjugate gradients with applications to boundary value problem, SIAM J.Numer.Anal., 24, 355-365, 1987.

15. 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.
16. C. Vuik , R. Nabben A comparison of deflation and the balancing preconditionner, SIAM J. Sci. Comput., 27:5, 1742-1759, 2006.

The paper is in final form and no similar paper has been or is being submitted elsewhere.