International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)
Saratoga Springs, New York, May 3-7, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009)

IMPROVEMENTS OF A FINITE VOLUME BASED MULTIGRID METHOD APPLIED TO ELLIPTIC PROBLEMS

N. Mechitoua*, Y. Fournier, F. Hulsemann
EDF R&D
Department of Fluid Mechanics, Power Generation and Environment
6, Quai Watier, 78401 Chatou Cedex, France

ABSTRACT

The paper focuses on a finite volume aggregation-based multigrid method applied to elliptic dominated problems. We show that for an unstructured finite volume approach on arbitrary shaped cells, it is possible to improve the scaled Galerkin approach based on piecewise constant interpolation for the R_0 restriction and P_0 prolongation operators. The off-diagonal entries of the P0 Galerkin coarse mesh matrix are rescaled by a parameter, which takes into account the mesh spacing ratio between the fine and coarse mesh in the vicinity of the coarse mesh cell boundaries. Some numerical examples are shown to assess the accuracy and the robustness of the proposed approach for elliptic dominated problems. A first level of parallelization has been performed and authorizes equivalent numerical convergence speed to the single-processor version, but for a moderate number of processors. This work is still in progress and new developments are planned for improving the speed up in the case of massively parallel computing.

Key Words: finite volume, multigrid, elliptic problems

1. INTRODUCTION

Multigrid methods represent an efficient alternative to other solution strategies for solving large linear systems, which arise from the discretization of Partial Differential Equations. For multigrid approaches, the computational complexity scales quasi-linearly with the problem size, which is not the case for Gaussian elimination or Krylov subspace based methods alone (even with polynomial or incomplete factorization preconditioners). The ongoing increase in computing power renders the treatment of ever larger linear systems possible, so that the use of algorithms with optimal complexity becomes inevitable.

The key ingredient of multigrid technique relies on the use of a hierarchy of grids (built from fine to coarse grids) for solving a linear set of equations. The principal idea is that, on each grid level, the relaxation scheme converges quickly for certain components of the error and slowly for the others. Put differently, a necessary condition for the fast convergence of a multigrid scheme is that for each component of the error, there exists a grid level on which the error component in question is eliminated efficiently.

In the case of geometric multigrid methods, the hierarchy of grids can be determined *a priori* by geometric considerations, irrespective of the discretized operator. However, the geometric multigrid approach is limited to a narrow range of applications (structured meshes for instance). Thanks to the rapid progress in the field, the algebraic multigrid has become a viable alternative.

* Corresponding author; e-mail address: namane.mechitoua@edf.fr

The hierarchy of grids is automatically created, taking into account the matrix entries of the discretized operator. The algebraic multigrid allows the effective solution of large linear systems arising from highly non-homogeneous Partial Differential Equations discretized with unstructured meshes, stretched or not.

We can consider two classes of multigrid methods. In the "Coarse/Fine" multigrid approach (named C/F multigrid here), the coarse grid is built as a subset of the fine grid. Provided the coarse grid points surrounding the fine grid points, "linear" prolongation operators between the fine and the coarse grid can be built in a quite natural way, avoiding extrapolation. This kind of multigrid was historically developed to effectively solve elliptic dominated systems [BRA 79]. The second one, named aggregation multigrid, is a particularly simple limiting case of the C/F multigrid, briefly presented above. It consists of agglomerating the fine mesh points for creating coarse mesh points.

The agglomeration approach, associated with piecewise constant interpolation for prolongation and restriction operators, is widely used in finite volume based CFD solvers [LON 93], either for elliptic or hyperbolic problems. However, a necessary condition for ensuring optimal multigrid efficiency is given by [HEM 90]:

$$d_P + d_R > d \qquad [1]$$

where d_P and d_R are the highest degree +one of the polynomials that are interpolated exactly by prolongation P and restriction R operators respectively, and d is the order of the partial differential equation to be solved. In this case, as piecewise constant interpolation (also named injection) is used either for restriction and prolongation, standard agglomeration multigrid fails to respect rule [1] for elliptic problems. A simple analysis of a Laplace equation shows that the Galerkin based coarse matrix $\boldsymbol{A_H=R.A_h.P}$ under-evaluates the corrections compared to that obtained by direct re-discretization on the coarse mesh. The simplest remedy [BRS 95] consists of rescaling (divide) the coarse level Galerkin operator $\boldsymbol{R.A_h.P}$ by a number, characteristic of the mesh spacing ratio between the coarse and fine mesh. But this very simple and easy-to-program technique is limited for problems that can be solved effectively with an agglomeration geometric multigrid method constructing almost *homogeneous* aggregates [MAV 95].

Another approach for accelerating aggregation-based algebraic multigrid consists of improving the piecewise constant interpolation by some smoothing process before the Galerkin operator is calculated [VAN 96]. The original piecewise constant prolongation operator P_0 is then replaced by a smoothed operator P_1, defined as:

$$P_1 = P_0 \, [\; Id - \omega \, diag(A_0)^{-1} \, A_0 \;] \qquad [2]$$

where w is a relaxation factor and A_0 is a simplified matrix of the original matrix A obtained by suppressing the small negative off-diagonal entries and adding their absolute values to the diagonal. This process "mimics" a linear interpolation, and enables us to respect rule [1], but it can dramatically increase the number of off-diagonal entries of the coarse mesh matrix and it can destroy the simplicity of the original approach. Thanks to an unstructured finite volume approach, it is possible to improve the scaled Galerkin based approach and make it tractable with algebraic multigrid for the optimal solving of elliptic dominated problems. The aim of this paper is to describe such a solution, with more details than in the previous paper [MEC 08].

2. THE FINITE VOLUME FLOW SOLVERS NEPTUNE_CFD AND CODE_SATURNE

The general flow solvers *Code_Saturne* [ARC 04], [COS 08] and NEPTUNE_CFD [MEC 03] are mainly developed at EDF R&D and at their partners (CEA). NEPTUNE_CFD is dedicated to the simulation of multi-component/multiphase flows. The multi-fluid set of equations is an extension of the “two fluid-one pressure” model to the case of m phases. Each fluid (fluid component and/or phase) is modeled through at least 3 conservation equations representing mass, momentum and total enthalpy. *Code_Saturne* is dedicated to multi-component flows, with only one momentum equation, representing the momentum of the mixture of gas, liquid and particles. NEPTUNE_CFD is mainly used for nuclear engineering, whereas *Code_Saturne* is used either for nuclear and fossil energy engineering, and for environment (geophysical flows).

The spatial discretisation is based upon a three dimensional full unstructured finite volume approach. A face based data structure allows the use of cells of arbitrary shape. The discrete face separates two cells, as it is illustrated in figure 1. The face to cell connectivity is, for example, considering that surfaces are outing from cell I to the neighbours cells J1, J2 and J3 :

J1 = IFACEL(1, N1); J2 = IFACEL(1,N2) ; J3 = IFACEL(1, N3)
I = IFACEL(2,N1) = IFACEL(2,N2) = IFACEL(2,N3)

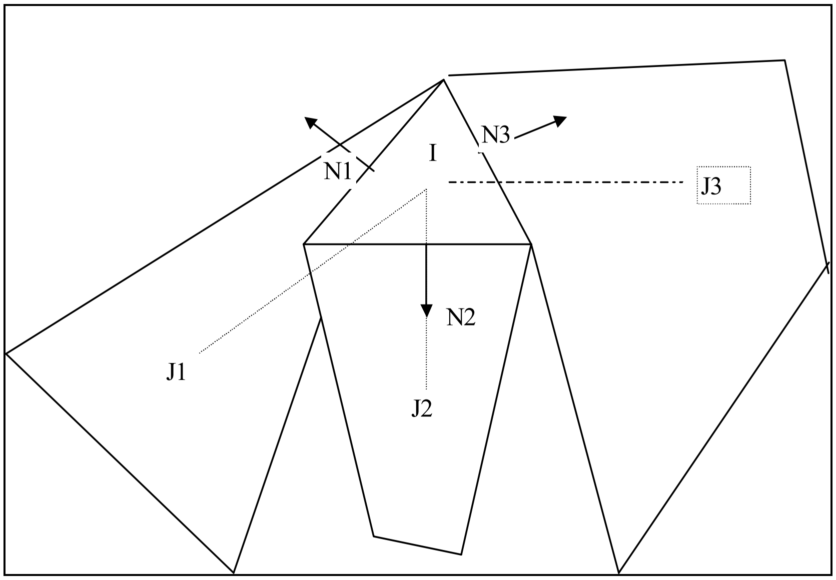


Figure 1 : view of cell’s shape, and data structure

The data structure, used for CFD computations, contains, at least, the face to cell connectivity, the vectorial surfaces, the coordinates of points “associated” to cells and coordinates of gravity centers of faces, the attributes of the boundary faces. Non plane surfaces are split into plane surfaces in order to have a finite volume scheme, at least, at first order in space.
All variables are collocated at the cell “centers”. The cells can be built, from the primitive mesh or around the vertices of the primitive mesh. The main advantage of the primitive mesh is that it

authorizes in a simple manner, non matching meshes, with hanging nodes. The main advantage of the mesh built around the cell vertices is its superior "numerical" quality, if the primitive mesh is highly irregular and non orthogonal. The optimal location of the cell "center", from a numerical point of view, is not necessary at the geometric gravity center of the cell. It depends on the numerical scheme and the physical phenomena. The ortho-center is preferred to geometric gravity center, if it is located inside the cell.
Gradients are calculated at second order for regular cells and, at least, at first order for highly irregular cells. An iterative finite volume procedure is preferred to the least square method because it produces smoother gradients on irregular cells [COS 08].
A careful treatment of gradient terms of momentum equation is needed, in order to avoid spurious oscillations of pressure, velocity components and volume fractions, during the coupling of mass and momentum equations. This behavior comes from the collocated arrangements of all variables for low speed flows and centered interpolations of variables at face centers. Variants of the Rhie & Chow Interpolation [RHI 83] are used for the pressure gradient calculations, for the coupling between the mass equation with a reduced form of the momentum equations [CAR 83].
A large amount of linear systems have to be solved during the velocities prediction step and the mass/ momentum/ energy coupling. These systems, generally diagonal dominant, are solved by simple iterative methods :

- relaxation methods as Jacobi or Gauss Seidel for convective dominant systems,
- gradient methods as Preconditoned Conjugate Gradient, Stabilized Biconjugate Gradient and Conjugate Gradient Square for elliptic dominant systems.

Acceleration techniques, as Polynomial Preconditioners and algebraic multigrid solvers, are used.

3. THE MULTIGRID PROCEDURE FOR THE LINEAR SYSTEMS.

We consider in this paragraph the case of a scalar convection/diffusion equation, representative of the transport/diffusion terms of the momentum equations and energy or mass fraction equations. It is equally representative of a type of pressure equation (without the convective term) obtained by the coupling of the mass equation with a reduced form of the momentum equation. The finite volume integration of a convection and diffusion and source term equation *div(* ***Q*** *C-K grad C) = b* over a discrete cell Ω_I is written as:

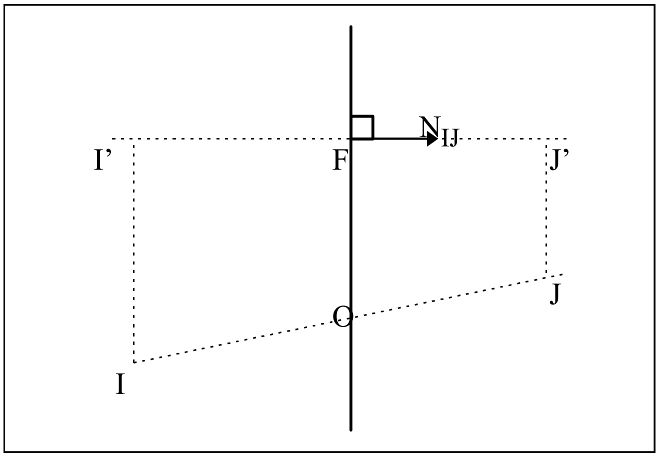
$$\int_{\Omega_I} div(\vec{Q}\,C - K\,\vec{\nabla}C)\,d\Omega = \sum_{J\,\in\,V_I} (\vec{Q}_{IJ}.\vec{N}_{IJ})C_{IJ} \;+\; \sum_{J\,\in\,V_I} -(K\,\vec{\nabla}C)_{IJ}.\vec{N}_{IJ} = b_I\Omega_I \qquad [3]$$

where V_I represents the neighborhood of the cell *I*, i.e the set of cells *J* sharing a non-zero area surface $\vec{N}_{IJ}$ with the cell *I*.
Numerical consistency and precision for diffusive and convective fluxes for non-orthogonal cells are taken into account using a gradient reconstruction technique. This technique is useful for increasing the order of some numerical schemes, when applied to complex situations as unstructured meshes for instance. The use of cells of arbitrary shape does not make easy the construction of shape functions for all type of cells (non matching mesh). It concerns either first order (convection) and second order (diffusion) differential equations, discretized with finite

volume methods. Among the various numerical fluxes, the following one is used for the diffusion:

$$-(K\vec{\nabla}C)_{IJ}.\vec{N}_{IJ} = \frac{K_{IJ}(C_I - C_J) + (\overline{II'} - \overline{JJ'}).(K\vec{\nabla}C)^{c}_{IJ}}{I'J'}\left|\vec{N}_{IJ}\right| \qquad [4]$$

where $(K\vec{\nabla}C)^{c}_{IJ} \approx 0.5.(K_I\,\vec{\nabla}C_I + K_J\,\vec{\nabla}C_J)$ represents the cell gradient projected at the face center F, either evaluated with a finite volume or a least square formulation and K_{IJ} represents the face interpolation of the diffusion coefficient (with arithmetic, harmonic or geometric interpolation).



<u>Figure 2</u>: Diffusion scheme reconstructed at first order at face center F

For the convective part of the fluxes (which is not in the scope of the present paper), the simplest upwind scheme remains consistent for non orthogonal smooth meshes, but the order can be less than one. On unstructured meshes, a reconstruction technique is used to increase the order of convection scheme. Among the various numerical convective fluxes, the following one is used for a centered interpolation of the convected variable C at the face center F (also named “IJ”) of the interface separating two cells I and J (see figure 2):

$$C_{IJ} \equiv C_F .= C_O + \overline{OF}.\vec{\nabla}C^{c}_{IJ}$$

$$C_O = \frac{CJ}{IJ}C_I + \frac{CI}{IJ}C_J \qquad [4]\text{ bis}$$

where $\vec{\nabla}C^{c}_{IJ} \approx 0.5.(\vec{\nabla}C_I + \vec{\nabla}C_J)$ represents the cell gradient projected at the face center F, either evaluated with a finite volume or a least square formulation. In order to avoid instabilities, the convective schemes for all variables, except the pressure, are non linear centered/upwind schemes. The switch between upwinded and centered interpolation is controlled by a slope test which detects the non monotony of the variable in the neighborhood of the interpolation point.

The linear set of equations arising from discrete formulations [3] (without convection, which is not in the scope of the present paper) with the higher order diffusive flux [4], is not solved directly with GMRES or BICGSTAB method, because the resolution can be too expensive or impossible without robust pre-conditioners. The computation of such systems is made through a defect correction technique. The explicit (or initial) diffusive flux (named old) computed with the scheme [4] is put in the right hand side. The iteration matrix for solving the correction φ, making use only of the original finite volume neighborhood, is a positive M matrix, which can be solved by suitable iterative methods, such as conjugate gradient or multigrid solvers.

$$\int_{\Omega_I} div(-K\vec{\nabla}\varphi)\,d\Omega \approx \sum_{J\in V_I} \frac{K_{IJ}\,(\varphi_I-\varphi_J)}{I'J'}\left|\vec{N}_{IJ}\right| = b_I\,\Omega_I - \sum_{J\in V_I} (-K\vec{\nabla}C)^{old}_{IJ}\,.\vec{N}_{IJ} \qquad [5]$$
$$C^{new} = C^{old} + \varphi$$

Defect correction procedure [5] can be repeated (by replacing C^{old} by C^{new}) until the residual tends towards a small user defined value.

Let us now consider a direct discretization of the correction φ on the coarse mesh I_c, arising from the aggregation of the cells I of the fine mesh. Figure 3 shows an example of two aggregates and the boundary between the two aggregates.

$$\int_{\Omega_{I_C}} div(-K\vec{\nabla}\varphi)\,d\Omega = \sum_{I\in I_C} \int_{\Omega_I} div(-K\vec{\nabla}\varphi)\,d\Omega \qquad [6]$$

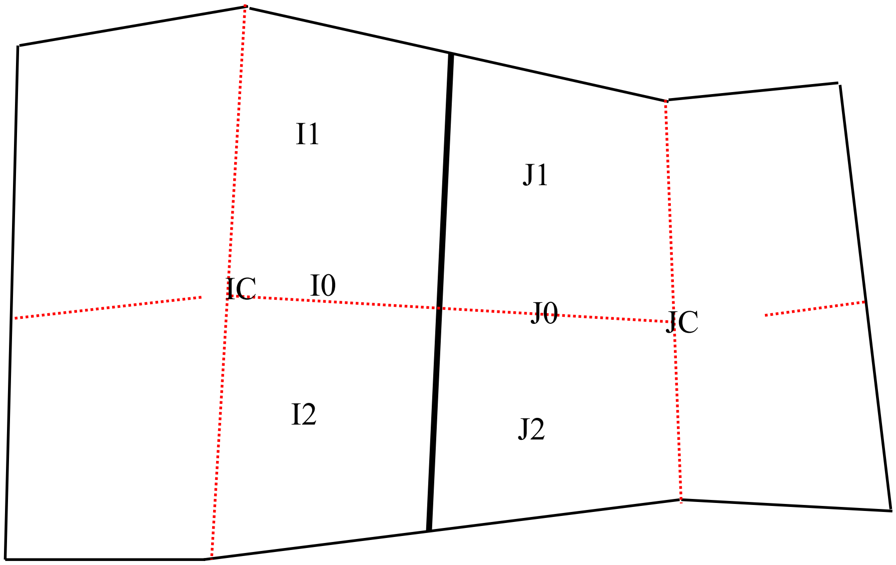


Figure 3: Sketch of an aggregate and coarse mesh boundary (bold)

Thanks to the conservative form of the equations, all fluxes within the aggregate disappear and we can compare the external fluxes discretized with the nearest neighbour stencil [MEC 08] in the following manner:

$$\begin{aligned} &K_{I_C J_C} \frac{\varphi_{I_C} - \varphi_{J_C}}{I_C' J_C'} \left| \vec{N}_{I_C J_C} \right| \equiv \sum_{(I_k, J_k)} K_{I_k J_k} \frac{\varphi_{I_k} - \varphi_{J_k}}{I_k' J_k'} \left| \vec{N}_{I_k J_k} \right| \\ &\vec{N}_{I_C J_C} \equiv \sum_{(I_k, J_k)} \vec{N}_{I_k J_k} \end{aligned} \qquad [7]$$

The couples (I_k, J_k) are the cells of the fine mesh situated on both sides of the coarse mesh interface. Considering the average values φ_{I0} and φ_{J0} “near” the coarse mesh interface $\vec{N}_{I_C J_C}$ and situated on both sides of this interface, we obtain:

$$\varphi_{M_0} \equiv \frac{\sum\limits_{(I_k, J_k)} \frac{K_{I_k J_k}}{I_k' J_k'} \left| \vec{N}_{I_k J_k} \right| \varphi_{M_k}}{\sum\limits_{(I_k, J_k)} \frac{K_{I_k J_k}}{I_k' J_k'} \left| \vec{N}_{I_k J_k} \right|} \quad \text{with M=I or J} \qquad [8]$$

Hence, equation [7] now reads:

$$XA_{I_C J_C} (\varphi_{I_C} - \varphi_{J_C}) \equiv \left(\sum_{(I_k, J_k)} \frac{K_{I_k J_k}}{I_k' J_k'} \left| \vec{N}_{I_k J_k} \right| \right) (\varphi_{I_0} - \varphi_{J_0}) \qquad [9]$$

where $XA_{I_C J_C}$ represents the off-diagonal entry between the cell I_C and cell J_C of the coarse mesh iteration matrix.

The variables φ_{I_0} and φ_{J_0} can be interpolated at first order according to the coarse mesh variables φ_{I_C} and φ_{J_C} [MEC 08], leading to the above expression:

$$\varphi_{I_0} - \varphi_{J_0} \approx \frac{I_0' J_0'}{I_C' J_C'} (\varphi_{I_C} - \varphi_{J_C}) \qquad [10]$$

where $I_0' J_0'$ represents the projection of the vector $\overrightarrow{I_0 J_0}$ on the normal $\vec{N}_{I_C J_C}$ of the coarse mesh interface. The previous relation [10] combined with [9] leads to:

$$XA_{I_C J_C} = \frac{I_0' J_0'}{I_C' J_C'} XA^0_{I_C J_C} \text{, with } XA^0_{I_C J_C} = \sum_{(I_k, J_k)} \frac{K_{I_k J_k}}{I_k' J_k'} \left| \vec{N}_{I_k J_k} \right| \qquad [11]$$

Equation [11] links the off-diagonal entries of a "linear" Galerkin coarse mesh matrix with the piecewise constant Galerkin one, through a correcting factor $I_0^{'}J_0^{'} / I_C^{'}J_C^{'}$. This correcting factor represents a mesh spacing ratio between the fine and coarse meshes in the neighbourhood of the coarse mesh interface $\vec{N}\ _{I_C J_C}$.

Expression [11] has an important practical interest. We proceed first to a piecewise constant Galerkin coarse mesh computation, which is particularly easy to implement and cheap in computational time. Then, for the elliptic part of the off-diagonal entries, we proceed to a correction with the parameter defined above. The other restriction and prolongation operations are not modified: we continue to proceed to a piecewise constant restriction for the right hand side of the defect correction equation and to a piecewise constant prolongation of the correction.

The simplest cycle in the multigrid resolution, i.e the V cycle, is used, in combination with under relaxed Jacobi or diagonal preconditioned conjugate methods acting as smoothers. The algebraic multigrid procedure is based upon the strength of the matrix connectivity, defined in a symmetric way for an aggregation-based procedure. Two cells numbered *i* and *j* are merged if $a_{ij}^2 /(a_{ii}.a_{jj})$ is greater than a fixed value. Each coarse grid has approximately one third the number of cells of the previous fine grid (coarsening ratio fixed by the user). Hence few smoothing iterations performed at each level are sufficient to efficiently reduce short wavelength errors. Coarsening stops once the number of cells of the coarsest level drops below a prescribed value, given by the user (~10). The coarsest grid linear system is solved "exactly" with a Krylov subspace method or a direct method.

4. ASSESSMENT OF THE PROPOSED NUMERICAL PROCEDURE

4.1 – Square cavity and FEDORA test cases

Figure 4 (left side) shows the number of iterations made by the linear solver with respect to the number of cells when solving the elliptic potential equation in a square cavity. Here, the number of iterations represents all arithmetic and memory access operations made during the resolution, scaled by the number of the same operations of one smoothing iteration on the finest level. The number of iterations of diagonal preconditioned conjugate gradient scales with the square root of the number of cells, which is characteristic of the behaviour of such solvers applied to 2-dimensional problems, when discretized with regular meshes. The number of iterations of the rescaled multigrid linear solver (named here P_1 as linear interpolation) remains quite constant, whatever the number of cells, while the standard aggregation multigrid linear solver (named P_0 as piecewise constant interpolation) used with a V cycle exhibits a mesh dependency, which varies as the number of cells to the power 0.2.

Figure 4 (right side) shows an example of the numerical behaviour of the multigrid solver applied to the elliptic pressure equation solved in a complex geometry [SIM 88], using an unstructured mesh, as in Figure 5. The grid hierarchy is as follows, where NCEL and NFAC respectively denote the number of cells and half the number of off-diagonal entries.

grid level	0	1	2	3	4	5	6	7	8	9
$NCEL_k$	18876	6361	2100	722	264	85	30	12	6	4
$NFAC_k$	36872	13857	5001	1789	582	160	39	17	6	4

Table 1: Grid hierarchy obtained for the FEDORA test case

The grid complexity, representing the total number of non-zero matrix entries scaled by the same number of the finest level is low (~1.6), indicating that the Galerkin stencils of the lower levels remain almost the same as of the finest level. Hence, it is not useful to proceed to "aggressive" coarsening in order to maintain a low grid complexity, as for the Coarse/Fine splitting multigrid approach [TRO 01]. The P_1 aggregation multigrid is at least 10 times faster than the diagonal conjugate gradient method. The P_0 aggregation multigrid behaves nicely with respect to the diagonal conjugate gradient but it is 3 times slower than the P_1 multigrid approach.

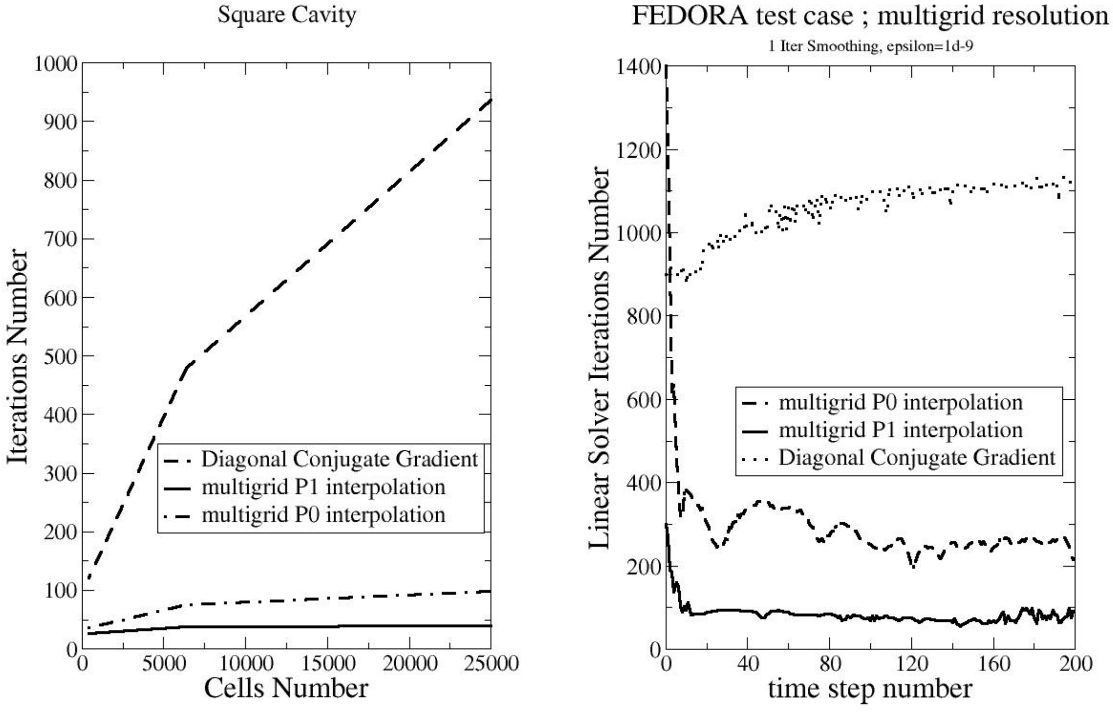


Figure 4: Comparisons of diagonal conjugate gradient and multigrid performances for a Square Cavity and FEDORA tests cases.

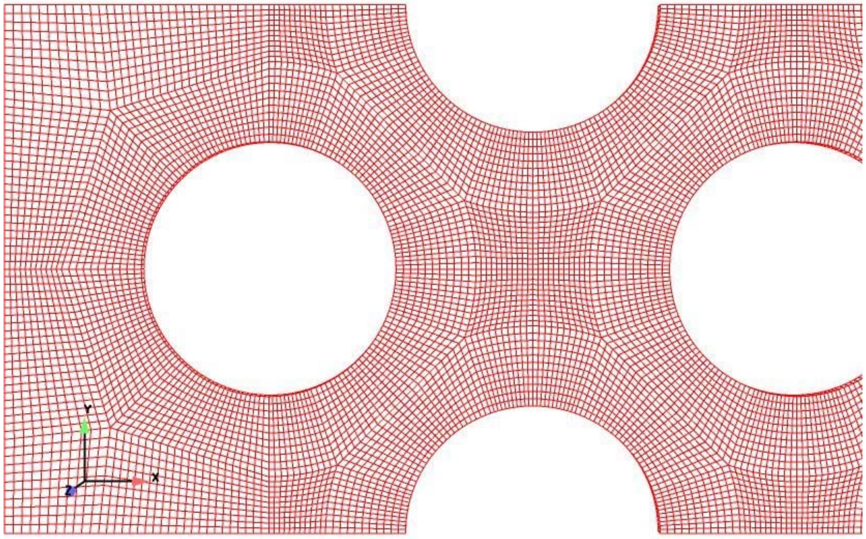


Figure 5 : FEDORA test case mesh near the inlet

4.2 – Flow around a body in a wind tunnel

Table 2 indicates the grid hierarchy performed by the algebraic multigrid solver, applied to the elliptic pressure equation. The number of levels was limited to 6 here.

grid level	0	1	2	3	4	5
$NCEL_k$	64148	21475	7248	2353	769	254
$NFAC_k$	124405	64602	32856	13626	4859	1589

Table 2: Grid hierarchy obtained for the VOITURE test case

The grid complexity, which is about 1.94, remains low, although the Galerkin stencil of the lower levels tends to increase, probably because of the full unstructured nature of the finest level mesh, composed of tetrahedra, as shown in figure 6. Figure 7 shows the multigrid solver behavior used either with NEPTUNE_CFD or *Code_Saturne*. The P_1 aggregation multigrid is at least 3 to 20 times faster than the diagonal conjugate gradient method. The noticeable difference in terms of performance for the multigrid solver is the result of a slightly different implementation of the k-ε turbulence model, which generates some local sensitive differences for the turbulent viscosity, near the walls. Then, the right hand sides of the same elliptic pressure equation are locally quite different and the convergence of the multigrid solver is affected. This difference in terms of convergence speed can be corrected by increasing the number of smoothing iterations or by using more robust smoothers, but this behavior would indicate that an algebraic multigrid procedure only based on the matrix entries could not reach the optimal performance, when it is applied to linear systems, with highly disturbed right hand sides.

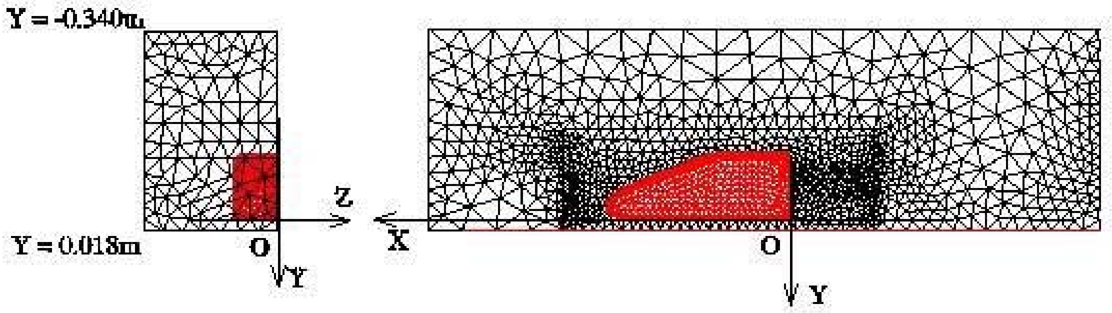


Figure 6 : Mesh around a simplified car in a wind tunnel

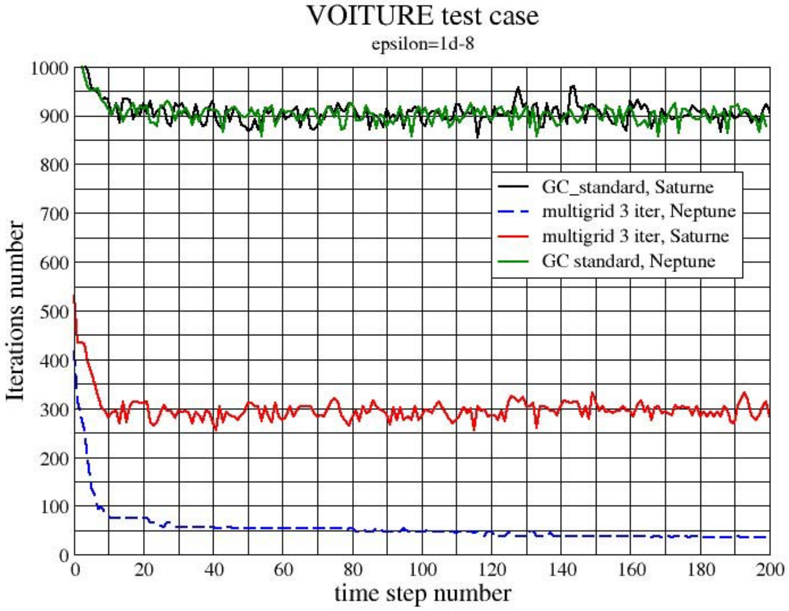


Figure 7 : Comparisons of diagonal conjugate gradient and multigrid performances for VOITURE tests case

4.3 – Work in progress: parallelization of the numerical scheme

The presented numerical procedure has been parallelized with *Code_Saturne* in a quite simple manner. The critical steps of this work concern:

- the creation of the grid hierarchy independently for each process, as in the single-processor case,
- the creation of the connectivity and halos (ghosts cells) between parts of the coarse grids of the same levels, managed by different processors.

This first level of parallelization authorizes equivalent numerical convergence speed to the single-processor version, but for a moderate number of processors (up to 512 processors). The parallel version has already been tested on meshes ranging from 300 000 cells up to 61 millions cells and coming from industrial cases. In order to improve the performances for massively parallel computations, complementary developments are necessary. They consist in grouping together the coarse grids in a reduced number of processors which may reduce the communication costs, but more importantly, they allow to continue the creation of coarser global grids and hence to speed up the convergence of the multigrid solver.

5. CONCLUSION

Thanks to a full finite volume approach, making use of cells of arbitrary shape, the scaled Galerkin approach based on piecewise constant interpolation can be greatly improved in order to make it tractable with algebraic multigrid techniques for optimally solving elliptic problems. The off-diagonal entries of the P_0 Galerkin coarse mesh matrix are rescaled by a numerical value, representative of the mesh spacing ratio between the fine and coarse grid in the neighborhood of the coarse mesh interface.

Three numerical examples are shown firstly to assess the accuracy and the robustness of the proposed approach and secondly to indicate future improvements. The first one shows the linear CPU cost of the rescaled algebraic multigrid with the mesh size; the second one shows the performance of that solver in the case of an close industrial geometry. The third one exhibits the importance of the right hand side (when it is highly disturbed), which is not taken into account in classical algebraic multigrid procedure.

A first level of parallelization has been performed and authorizes equivalent convergence speed to the single-processor version, but for a moderate number of processors (up to 512 processors). This work is still in progress and new developments are planned for improving the performances for massively parallel computations.

ACKNOWLEDGMENTS

Part of the work presented here has been achieved in the framework of the project “Simulations frontières”, dedicated to the use and the deployment of HPC machines at EDF. The financial support comes from EDF. The authors are grateful to the NEPTUNE_CFD and *Code_Saturne* teams for their support with the algorithm of these codes. The authors thank S. Bosse for the implementation of the method in *Code_Saturne* and M. Boucker, who has developed the first version of the method, written with P_0 interpolation and restriction, either for the right hand side, the correction and the coarse mesh matrix.

REFERENCES

[ARC 04] ARCHAMBEAU F., MECHITOUA N., SAKIZ M., "*Code_Saturne*: a finite volume code for the computation of turbulent incompressible flows- Industrial Applications", *Int. J. on Finite Volumes*, http://www.latp.univ-mrs.fr/IJFV/, **Vol.1**, 2004.

[BRA 79] BRANDT A., "Multilevel adaptive computations in fluid dynamics", *AIAA 79-1455*.

[BRS 95] BRAESS D., "Towards algebraic multigrid for elliptic problems of second order". *Computing,* **Vol 55**, pp 379-393 (1995).

[CAR 83] CARVER M.B. "Numerical Computation of Phase Separation in Two Fluid Flows". *Journal of Fluids Engineering*, **Vol. 106**, June 1984

[COS 08] http://www.code-saturne.org

[HEM 90] HEMKER P.W., "On the order of prolongations and restrictions in multigrid procedures". *J. Comp. Applied Math.*, pp. 423-429 (1990).

[LON 93] LONSDALE R.D., "An algebraic multigrid solver for the Navier Stokes equations on unstructured meshes". *Int. J. Num. Meth. Heat and FluidFlow,* **Vol. 3**, pp 3-14 (1993).

[MAV 95] MAVRIPLIS D.J., VENKATAKRISHNAN V., "Agglomeration Multigrid for 2 Dimensional Viscous Flows". *Computers and Fluids*, **Vol. 24**, N°5, pp 553-570 (1995).

[MEC 08] MECHITOUA N., BOSSE S., HULSEMANN F., BOUCKER M., "Some improvements of a FV Aggregation-Based Multigrid Method Applied to Elliptic Problems", *Proceeding of FVCA'5,* Aussois, France, June 8-13, 2008, ISBN 978-1-84821-035-6

[MEC 03] MECHITOUA N., BOUCKER M., LAVIEVILLE J, HERARD J.M., PIGNY S., SERRE G., "An Unstructured Finite Volume Solver for 2-Phase Water/Vapor Flows Modelling Based on an Elliptic Oriented Fractional Step Method", *Proceeding of NURETH 10,* Seoul, South Korea, October 5-9, 2003.

[RHI 83] RHIE C.M, CHOW W.L, "A Numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation". *AIAA Journal*, **Vol. 21**, pp. 1525-1532 (1983)

[SIM 88] SIMONIN O., BARCOUDA M., "Measurements and prediction of turbulent flows entering a staggered tube bundle". *EDF internal report HE-44/88.25*, 1988.

[TRO 01] TROTTENBERG U., OOSTERLEE C., SCHULLER A., *Multigrid,* Elsevier Academic Press*,* 2001. ISBN 0-12-701070-X.

[VAN 96] VANEK P., MANDEL J., BREZINA M., "Algebraic Multigrid by smoothed aggregation for 2nd order and 4th order elliptic problems". *Computing*, **Vol. 56**, pp.179-196 (1996).