ESAIM: PROCEEDINGS AND SURVEYS, September 2014, Vol. 45, p. 410-419
J.-S. Dhersin, Editor

ADAPTIVE METRIC-BASED MULTIGRID FOR A POISSON PROBLEM WITH DISCONTINUOUS COEFFICIENTS [*, **]

GAUTIER BRÈTHES [1]

Abstract. In order to solve the linear partial differential equation $Au = f$, we combine two methods: Full-Multigrid method and Hessian-based mesh adaptation. First, we define independently an Hessian-based mesh adaptation loop and a FMG algorithm where, at each phase, the equation is solved by a preconditioned GMRES with multigrid as preconditioner. Then we insert the adaptive loop between the FMG phases. We use this new algorithm and we compare its results with those obtained with non-adaptive FMG.

Résumé. On se propose de résoudre une équation différentielle linéaire aux dérivées partielles $Au = f$ en combinant deux méthodes : la méthode FMG et la méthode d'adaptation basée hessien. Tout d'abord, on définit, indépendamment l'un de l'autre, une boucle d'adaptation de maillage basée sur des hessiens et un algorithme FMG où dans chaque phase, l'équation est résolue par un GMRES préconditionné par les multigrilles. Puis, on insère la boucle adaptative entre les phases FMG. On utilise le nouvel algorithme ainsi obtenu et on compare ses résultats avec ceux obtenus avec le FMG non adaptatif.

INTRODUCTION

Multigrid methods (MG) are solution algorithms in which the iterative convergence to discrete solution is accelerated by means of a set of coarser approximations typically from a just twice coarser level to a coarsest level with just a few dozens degrees of freedom. The approach generally produces an iterative convergence which does not depend on mesh size, and therefore, an accurate enough discrete solution with N degrees of freedom obtained with a number of operations bounded by $const.N.Log(N)$, a complexity nearly optimal. Further, MG can be combined with a nested iteration, to build the Full-Multi-grid (FMG) algorithm. It consists in applying, in a first phase, a few one-grid cycles on the coarsest mesh, then transfer the solution to the next finer grid and in a second phase apply the same number of 2-G cycles to compute a solution on this finer grid, then transfer to the new finer grid, etc. FMG has an optimal complexity of $const.N$, predicted by theory and observed on many practical examples, see the reference book [16].

But this rosy picture needs some comments.

First, in many cases, the single local iteration is not sufficient to deal with singular or stiff configurations like discontinuities, or boundary layers. It becomes necessary to use more sophisticated less local iterations, and/or more sophisticated coarse mesh definition, as proposed by Algebraic MG [5] or by anisotropic mesh coarsening [7, 14, 15], and/or more adapted inter-grid transfers. Indeed, in some case, the directly coarse grid correction is not able to complement the fine grid iteration, or may even work in an unfavorable way. As a consequence, the set of grids to apply for MG acceleration is not necessarily the set of grid to apply for the FMG process.

[*] *This work has been supported by French National Research Agency (ANR) through COSINUS program (project ECINADS n° ANR-09-COSI-003)*

[**] *This work has been funded by PACA region and LEMMA company*

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

Article published online by EDP Sciences and available at http://www.esaim-proc.org or http://dx.doi.org/10.1051/proc/201445042

Second, for a lot of complex applications, it has been remarked that classical FMG, with a same number of cycles on each phase, does not work (see for example [6]). A central theoretical condition for FMG is the high-order convergence of the discrete solution to continuous on the different meshes of FMG, including the coarsest ones (cf. [8]). Therefore the most probable explanation for FMG failure is that high-order convergence to continuous does not hold, either because meshes are still too coarse, or because solution involves singularities. As a result, the solution produced at end of FMG is inaccurate when the basic FMG algorithm is applied, or will be obtained with a computational cost much larger than the one the theory of FMG would let expect.

Lastly, MG and FMG have been penalized for years by the difficulty in building and managing multiple coarse and fine meshes for industrial applications, a difficulty which is more easy to address today, with the recent progress of mesh generation and adaptation.

Now, the application of MG to novel strongly mesh-adaptive algorithms needs be clarified. By strongly mesh-adaptive we mean that an anisotropic mesh adaptation is strongly coupled with the solver by a fixed point iteration. These algorithms take benefit from the recently developed fast and adapted unstructured mesh generators.

Anisotropic mesh adaptors have recently proved two important properties. First they result in more efficient computations than with traditional methods, and they allow computations which were not attainable without anisotropic adaptation, like the propagation of a sonic boom from aircraft to ground [13]. Second, anisotropic mesh adaptors provide mesh convergence at high-order for singular problems while non-adaptive methods do not, and for non-singular problems, with a much more modest number of nodes than non-adaptive methods. The latter property exactly compensates the deficiency identified in the classical FMG algorithm.

A family of anisotropic mesh adaptors rely on the parametrization of a mesh by a Riemannian metric. It is then interesting to examine how Riemannian metrics can address the problem of derivation of the best coarse mesh for MG acceleration of systems non-trivially solved by MG.

1. Continuous mesh model and adaptation

We propose to work in the continuous mesh framework introduced in [10, 11]. The main idea of this framework is to model discrete meshes by Riemannian metric fields. It allows us to define proper differentiable optimization [1, 2], *i.e.,* to use a calculus of variations on continuous metrics which cannot apply on the class of discrete meshes. This framework lies in the class of metric-based methods. A continuous mesh $\mathscr{M}$ of the computational domain Ω is identified to a Riemannian metric field [4] $\mathscr{M} = (\mathscr{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathscr{M}(\mathbf{x})$ is a symmetric 2×2 matrix which we can write:

$$\mathscr{M}(x,y) = \mathscr{R}^t(x,y)\begin{pmatrix} \frac{1}{\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{\Delta\eta^2(x,y)} \end{pmatrix}\mathscr{R}(x,y)$$

where:
$\Delta\xi(x,y)$ = mesh size in the first caracteristic direction
$\Delta\eta(x,y)$ = mesh size in the second caracteristic direction
$\mathscr{R}(x,y)$ = matrix of eigenvectors.

The *vertex density* d is equal to: $d = \sqrt{\det(\mathscr{M})}$. By integrating it, we define the *total number of nodes* $\mathscr{C}$:

$$\mathscr{C}(\mathscr{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathscr{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

Given a continuous mesh $\mathscr{M}$, we shall say, following [10, 11], that a discrete mesh $\mathscr{H}$ of the same domain Ω is a unit mesh with respect to $\mathscr{M}$, if each triangle $K\in\mathscr{H}$, defined by its list of edges $(\mathbf{e}_i)_{i=1\ldots3}$, verifies:

$$\forall i\in[1,6],\quad \ell_{\mathscr{M}}(\mathbf{e}_i)\in\left[\frac{1}{\sqrt{2}},\sqrt{2}\right],$$

in which the length of an edge $\ell_{\mathscr{M}}(\mathbf{e}_i)$ is defined as follows:

$$\ell_{\mathscr{M}}(\mathbf{e}_i) = \int_0^1 \sqrt{{}^t\mathbf{ab}\,\mathscr{M}(\mathbf{a}+t\,\mathbf{ab})\,\mathbf{ab}}\,\mathrm{d}t, \quad \text{with } \mathbf{e}_i = \mathbf{ab},$$

We choose a tolerance α equal to 0.8. We want to emphasize that the set of all the discrete meshes that are unit meshes with respect to a unique $\mathscr{M}$ contains an infinite number of meshes.

Given a smooth function u, to each unit mesh $\mathscr{H}$ with respect to $\mathscr{M}$ corresponds a local interpolation error $|u-\Pi_{\mathscr{H}}u|$. In [10, 11], it is shown that all these interpolation errors are well represented by the so-called continuous interpolation error related to $\mathscr{M}$, which is locally expressed in terms of the Hessian H_u of u as follows:

$$\begin{aligned} |u-\pi_{\mathscr{H}}u|(\mathbf{x},t) &= \frac{1}{10}\mathrm{trace}(\mathscr{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x},t)|\,\mathscr{M}^{-\frac{1}{2}}(\mathbf{x})) \\ &= \frac{1}{10}\sum_{i=1}^{N} h_i(\mathbf{x})^2({}^t\mathbf{v}_i(\mathbf{x}))|H_u(\mathbf{x},t)|\mathbf{v}_i(\mathbf{x}) \end{aligned} \tag{1}$$

where
$h_i(\mathbf{x}) = (\lambda_i(\mathbf{x}))^{-1/2}$ with $(\lambda_i(\mathbf{x}))_{i=1,2}$ the eigenvalues of $H_u(\mathbf{x})$,
$(\mathbf{v}_i(\mathbf{x}))_{i=1,2}$ the eigenvectors of $H_u(\mathbf{x})$,
$|H_u|$ is deduced from H_u by taking the absolute values of its eigenvalues.

We define as optimal metric the one which minimizes the right hand side under the constraint of a total number of vertices equal to a parameter N. After solving analytically this optimization problem, this defines the unique optimal $(\mathscr{M}_{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ as:

$$\mathscr{M}_{\mathbf{L}^p} = D_{\mathbf{L}^p}\,(\det|H_u|)^{\frac{-1}{2p+2}}\,|H_u| \quad \text{with} \quad D_{\mathbf{L}^p} = N^{\frac{2}{2}}\left(\int_\Omega (\det|H_u|)^{\frac{p}{2p+2}}\right)^{-\frac{2}{2}}, \tag{2}$$

where $N = \mathscr{C}(\mathscr{M})$ is the prescribed number of vertices of the mesh.

2. Fixed-point mesh adaptation

2.1. Fixed point for PDE approximate solution

In the case where the function u is the solution of a Partial Differential Equation, it is possible to look for the optimal mesh by applying the following loop:
Fixed point for adaptive PDE approximation
1- compute the PDE approximate solution on current mesh
2- compute an approximate Hessian
3- adapt according to this Hessian
4- go to 1.

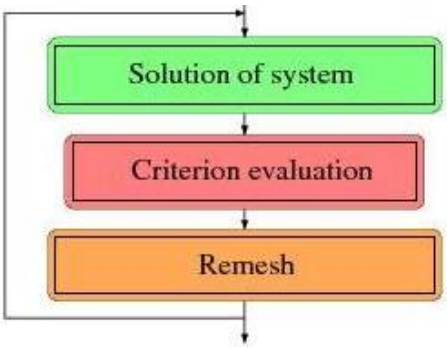

FIGURE 1. Mesh adaptation loop

2.2. **Observed convergence**

It has been observed in several studies that anisotropic mesh adaptation produced high order numerical mesh convergence for contexts for which non-adaptive strategies do not converge at high order. That happens for singular cases, and also for small number of nodes for which the same non-adaptive strategies do not show high order convergence. We refer to [12]. A general analysis of this favorable behavior is not yet available for the multi-dimensional case. Although not mathematically demonstrated, the property of early capturing and early asymptotic convergence has been observed in all numerical experiments, see [9].

3. THE MG ANISOTROPIC FIXED-POINT

Let us assume now that we want to solve our mesh-adaptive discrete PDE by means of a MG algorithm. This means that the PDE to solve, used for finding the approximate solution u_h, is replaced by the problem of finding the couple $(\mathscr{M}_h, u_h)$ such that:

$$\mathscr{M}_h \text{ is adapted to } u_h \text{ and } u_h \text{ is computed on mesh } \mathscr{M}_h.$$

This is the solution of a strongly non-linear compling system, but the dependancy of $\mathscr{M}_h$ with respect to u_h is explicit, in the sense that the cost of systems to solve is neglectible with respect to the cost of computation of the solution u_h on a given mesh. Therefore, we propose to apply the adaptive loop as an external one, the MG resolution by MG being an internal loop.

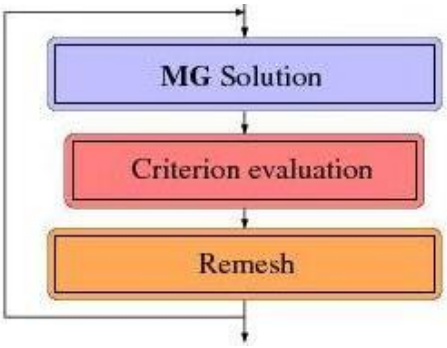

FIGURE 2. Mesh adaptation loop with Multigrid

The use of an anisotropic coarsening has a negligible influence in the considered cases and, so, it is not used. However, the adaptation itself is still anisotropic. The adopted stanpoint is to use the metric based mesh parametrization. Firstly, we specify the number of nodes of the adapted mesh. We have to choose an initial metric $\mathscr{M}$:

$$\mathscr{M}(x,y) = \mathscr{R}^t(x,y) \begin{pmatrix} \frac{1}{\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{\Delta\eta^2(x,y)} \end{pmatrix} \mathscr{R}(x,y)$$

Then coarser metrics are build using the metric-based embedding:

$$\mathscr{M}_{coarser_1}(x,y) = \mathscr{R}^t(x,y) \begin{pmatrix} \frac{1}{4\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{4\Delta\eta^2(x,y)} \end{pmatrix} \mathscr{R}(x,y)$$

and even coarser:

$$\mathscr{M}_{coarser_2}(x,y) = \mathscr{R}^t(x,y) \begin{pmatrix} \frac{1}{16\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{16\Delta\eta^2(x,y)} \end{pmatrix} \mathscr{R}(x,y)$$

etc...

Once these grids are built, we apply the MG method which we recall in short.

The two-grid algorithm uses a fine mesh and a coarse mesh. We partially solve the equation on the fine mesh by using a

smoother which will be, for us, several sweeps of the Jacobi method. Then, we transfer the residual on the coarse mesh to compute a corrector. Finally, we transfer this corrector on the fine mesh and we apply it on the fine grid iterate. That gives us a better approximation of the solution. Extension to Multigrid is done simply via recursion. When we compute the corrector on the coarse grid, we can use an even coarser grid to compute the corrector by using the two-grid method. We can continue on this way with as many grids as necessary.
The loop of Figure 2 needs generally five iterations or less. In the present study, we force five iterations.

4. THE FMG ALGORITHM

FMG can be defined as the combination of a MG loop with a nested iteration. A first coarse mesh is used for a first evaluation of the solution. On the coarse mesh, in principle, a coarser level for acceleration is not necessary since the convergence of a standard iterative solution algorithm is rather fast. A finer mesh is built, generally by uniformly refining the first mesh. The previous result is transfered to the new mesh, typically by an interpolation. The two meshes are available for playing the role of two MG levels in order to solve fastly the problem on the new mesh, while enjoying a good initial condition for the two-grid iteration. This process is reiterated with a 3-grid solution on next mesh etc. We call FMG phase each nested iteration phase, i.e. a phase with k-grid loop. Due to the initialization by previous grid, the convergence at each phase can be of a smaller number of k-grid cycles. Due to the property of MG to show a convergence ratio independant of number of level and grid size, an important gain is reachable.

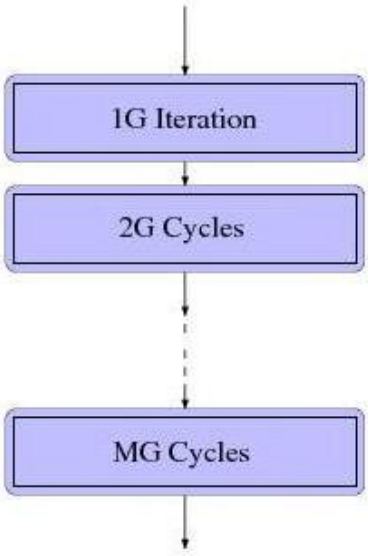

FIGURE 3. Full Multi-Grid algorithm

More precisely, the FMG algorithm is known as producing a second order solution on a mesh of N nodes with a computational effort of $O(N)$. A condition for this property is a second order estimate for approximation of the current level [8]:

$$||u - u_k|| \leq Kh_k^2. \tag{3}$$

However, the proof of this property relies on [3]. Now this property can be established only for a mesh fine enough, that is for levels k with k large enough. In the present study, we stop cycling when the iterative residual has been driven to a ten times smaller value.

5. THE FMG ANISOTROPICALLY ADAPTIVE ALGORITHM

We consider the FMG algorithm and the mesh adaptation loop with Multigrids. We obtain the FMG Anisotropically adaptive algorithm by combining these two methods. Firstly, we consider the FMG algorithm and, in each phase, we replace the Multigrid process by the mesh adaptation loop in which the Multigrid process is used to compute the approximate solution inside the loop. This gives the following algorithm:

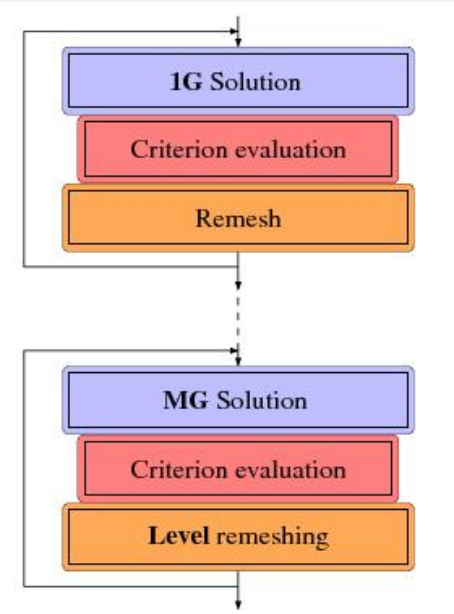

FIGURE 4. Mesh adaptive Full Multi-Grid

In contrast to the non-adaptive case, with the use of anisotropic adaptation, we enjoy the early capturing and convergence property and will have a good mastering of the iteration necessary at each FMG phase, and therefore, a good mastering of the $O(N)$ complexity.

6. EXAMPLES

The proposed Adaptive Anisotropic FMG is applied to two test cases and compared with a pure FMG method.

6.1. **A boundary layer test case**

We consider a test case in which we solve a Poisson problem. We define the right hand side by the following formula: $rhs(x,y) = \frac{1}{\alpha^2(exp(1/\alpha)-1)} exp(x/\alpha)$ with $\alpha = 0.03$.

We consider $-\Delta u = rhs$ with $\frac{\partial u}{\partial y}(x,0) = \frac{\partial u}{\partial y}(x,1) = 0$ and $u(0,y) = u(1,y) = 0$ as boundary conditions.

Thanks to the equation, the right hand side and the boundary conditions, we can find the expression of the analytical solution u and we obtain:

$$u(x,y) = \frac{-1}{exp(1/\alpha)-1} exp(x/\alpha) + x + \frac{1}{exp(1/\alpha)-1}$$

We also solve $-\Delta u = rhs$ with the boundary conditions thanks to a standard finite elements approximation. It gives us an approximated solution u_h represented in Figure 5.

That allows us to compute directly the norm of the approximation error defined by $||u-u_h|| = \int_\Omega |u-u_h| dxdy$ which is depicted as a function of the number N of nodes of the mesh.

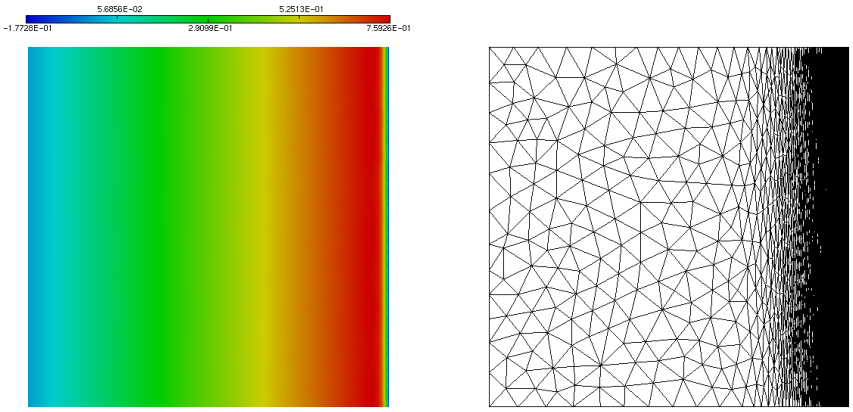

FIGURE 5. Boundary layer test case solution and adapted mesh

We also compute the error of interpolation. We draw it in function of the number N of nodes of the meshes, given in Figure 6. We observe a convergence of order 2 for the non-adaptive case and a convergence even better in the adaptive case. The difference between Figure 6 left and Figure 6 right corresponds to the slight inadequacy of choosing interpolation

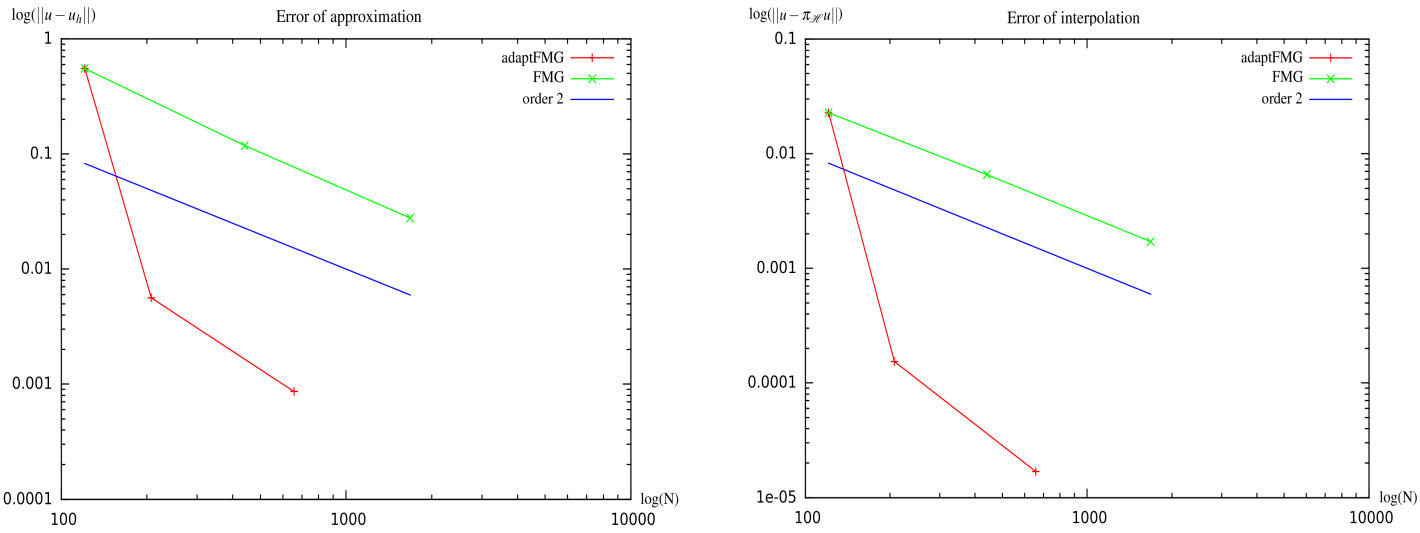

FIGURE 6. Boundary layer test case: errors as a function of the number of points

error as mesh adaptive criterion to minimize. Probably a three times smaller error can be expected with a better criterion. For the same number of points, the errors are notably smaller in the adaptive case. Then, we draw these errors again but in function of the CPU time to obtain Figure 7. We can see, on these pictures, that, for the same CPU time, adaptive FMG gives us a smaller error than non-adaptive FMG: for 10 seconds, the ratio is around 3.5.

6.2. **Anisotropic peak test case**

We solve $-\Delta u = rhs$ with Dirichlet conditions. We choose an analytical solution u and we compute the right hand side. Then, we solve the equation to find an approximation of the solution and we compare it to the analytical one. The

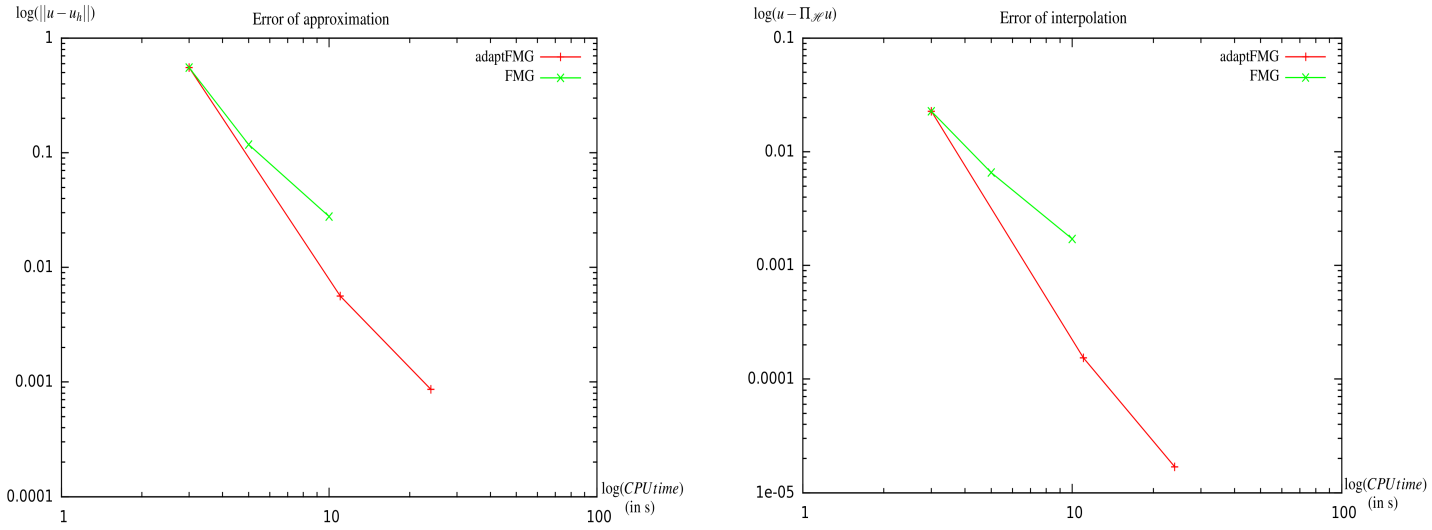

FIGURE 7. Boundary layer test case: errors as a function of the CPU time

analytical solution is given by the following formula:$u = \alpha e^{Kg_0(x,y)} x(x-1)y(y-1)$ where $\alpha = 0.1$ and $K = 50000$ and where $g_0(x,y) = y(y-1)(x-0.45)(x-0.55)$. This function presents a strong variation which looks like a peak and will be close to a Dirac function for too coarse meshes. Refining the mesh in peak location permits to address this problem. Mesh adaptation method *automatically* computes not only a refined mesh but the optimal refined mesh. The Figure 8 shows the final discretized solution and the corresponding adapted mesh:

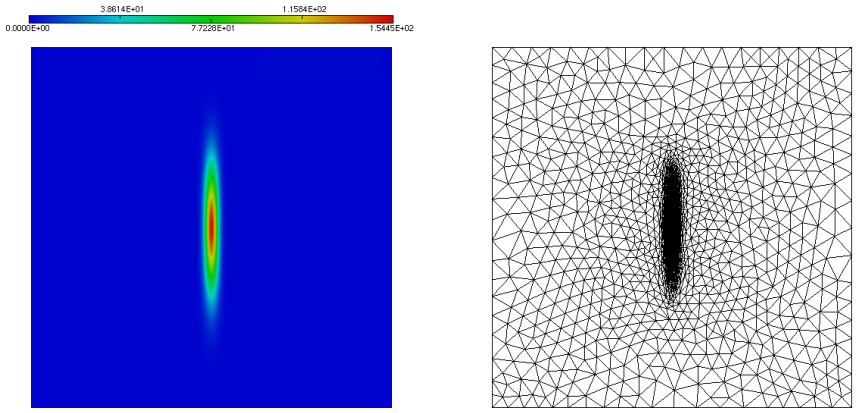

FIGURE 8. Anisotropic peak test case solution ans adapted mesh

Like for the previous test case, we compare the errors of approximation and interpolation, firstly in function of the number of points and, then, in function of the CPU time. It is shown in the Figures 9 and 10.

Like in the previous test case, when we use mesh adaptation, the errors decrease quicker, with an apparent order of convergence better than 2. For the same CPU time, the errors are much smaller: for 9 seconds, the ratio is around 20 . A remarkable difference with the boundary layer is that the decreasing accelerates with the second adaptation. This can be explained by the mesh becoming fine enough to consider the strong variations seen as discontinuities by coarse meshes.

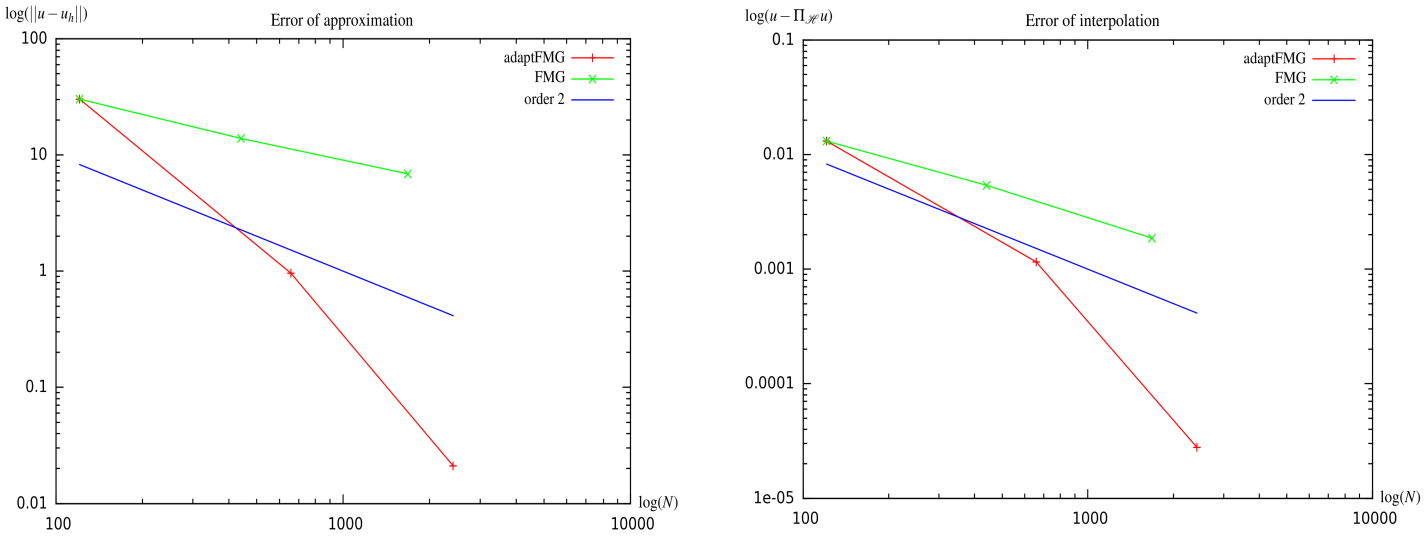

FIGURE 9. Anisotropic peak test case: errors as a function of the number of points

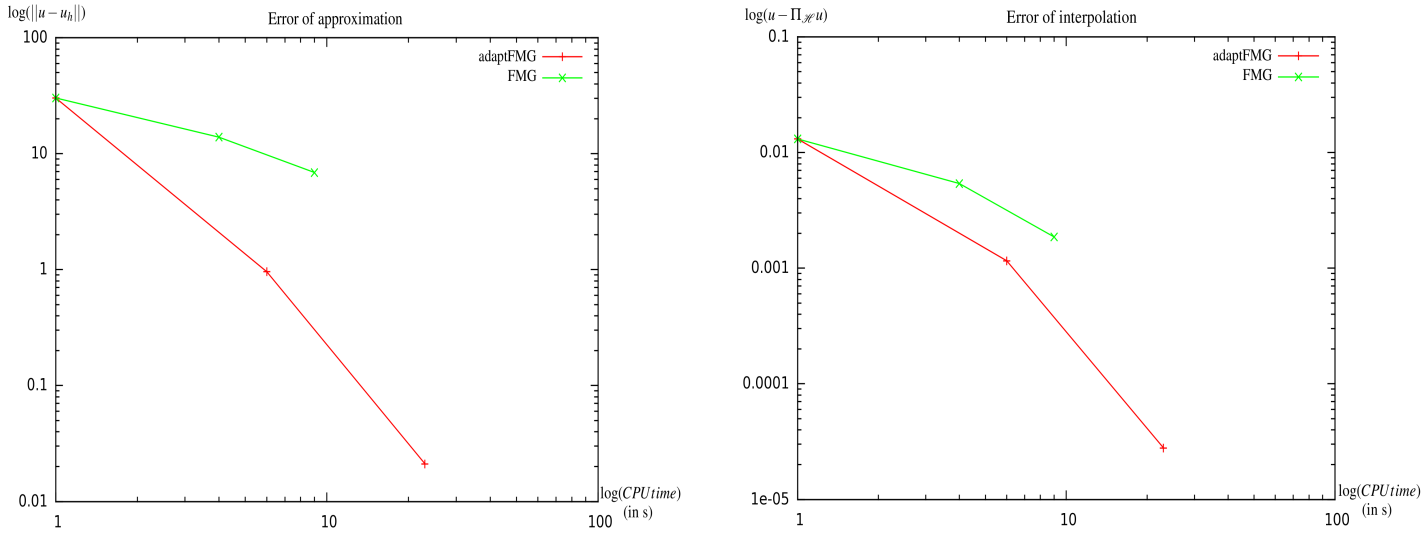

FIGURE 10. Anisotropic peak layer test case: errors as a function of the CPU time

7. CONCLUDING REMARKS

To obtain our new algorithm, we used new mesh adaptation technology: thanks to Hessians, we can compute optimal metrics and use them to generate adapted meshes. We combined the fixed-point mesh adaptation to multigrids and we inserted it in the FMG algorithm to finally obtain the FMG anisotropically adaptive algorithm. We applied it on two test cases and compared the analytical solutions to the computed ones. We have observed that introducing this more complex algorithm brings a higher safety in the accuracy of results and a better control of computational cost: by using adaptive FMG instead of classical FMG, we have obtained better results in less CPU time.

However, it is still possible to optimize our method and two improvements are currently studied. To avoid unnecessary computations, we shall in a forth coming paper stop Multigrid iterations accorded to a comparison of approximation error

and iteration error. For the same reason, we also consider controlling adaptation loop convergence. For both, we plan to use a posteriori error estimators.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magn. Reson. Med.*, 56(2):411–421, 2006.

[3] R.E. Bank. Pltmg: A software package for solving elliptic partial differential equations. In *users' guide 11.0. Technical report, Department of Mathematics, University of California, San Diego*, 2012.

[4] M. Berger. *A panoramic view of Riemannian geometry*. Springer Verlag, Berlin, 2003.

[5] A. Brandt, S. F. McCormick, and J. W. Ruge. Algebraic Multigrid for automatic multigrid solutions with application to geodetic computations, Technical Report, Institute for Computational Studies, Fort Collins, Colorado, october 1982.

[6] G. Carré and A. Dervieux. On the application of FMG to variational approximation of flow problems. *Comp. Fluid. Dyn. J.*, 12:99–117, 1999.

[7] J. Francescatto. *Méthodes multigrilles par agglomration directionnelle pour le calcul d'coulements turbulents*. PhD thesis, Université de Nice - Sophia Antipolis, Nice, France, 1998. (in French).

[8] W. Hackbusch. Multi-grid methods and applications. Springer, 1985.

[9] A. Loseille. *Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique*. PhD thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2008. (in French).

[10] A. Loseille and F. Alauzet. Continuous mesh framework. Part I: well-posed continuous interpolation error. *SIAM NUMA*, 49(1):38–60, 2011.

[11] A. Loseille and F. Alauzet. Continuous mesh framework. Part II: validations and applications. *SIAM NUMA*, 49(1):61–86, 2011.

[12] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *J. Comp. Phys.*, 229:2866–2897, 2010.

[13] A. Loseille and R. Löhner. Adaptive anisotropic simulations in aerodynamics. In *Proceedings of 48th AIAA Aerospace Sciences Meeting, AIAA 2010-169, Orlando, FL, USA, January 2010*, 2010.

[14] D.J. Mavriplis. Adaptive mesh generation for viscous flows using Delaunay triangulation. *J. Comp. Phys.*, 90:271–291, 1990.

[15] W.A. Mulder. A new multigrid approach to convection problems. *J. Comp. Phys.*, 83:303–323, 1989.

[16] U. Trottenberg, C. W. Osterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.