A Mesh-Adaptive Metric-Based Full-Multigrid for the Poisson problem

G. Brèthes[1], O. Allain[2], A. Dervieux[1]

[1] *INRIA Sophia Antipolis, 2004 route de Lucioles, 06902 Sophia-Antipolis, France*
[2] *LEMMA - 2000 route des Lucioles - Les Algorithmes - Bt. Le Thales A, Biot, 06410, France*

Abstract

This paper studies the combination of the Full-Multi-Grid (FMG) algorithm with an anisotropic metric-based mesh adaption algorithm. For the sake of simplicity, the case of an elliptic two-dimentional Partial Differential Equation (PDE) is studied. Meshes are unstructured and non-embedded, defined through the metric-based parametrisation. A rather classical MG preconditionner is applied, in combination with a quasi-Newton fixed point. An anisotropic metric-based mesh adaptation loop is introduced inside the FMG algorithm. FMG convergence stopping test is re-visited. Applications to a few 2D continuous and discontinuous-coefficient elliptic model problems show the efficiency of this combination.

Keywords: anisotropic mesh adaption, full multi-grid, finite element, stopping criterion, Poisson problem

1. INTRODUCTION

Multi-grid methods (MG) can produce fast and robust solution algorithms. They apply to a large variety of models and approximations in Computational Mechanics.

MG uses intensively approximation properties. A simple local iteration is applied on the given grid. The iterative convergence is then accelerated by means of a set of coarser-grid corrections, ranging 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, when expressed in terms of (logarithm of) residual norm decreasing with iterations, is more or less of constant slope. Further this slope does not depend on mesh size. In particular, an accurate enough discrete solution with N degrees of freedom is obtained with a number of operations bounded by $K.N.Log(N)$, a complexity nearly optimal. Lastly, MG can be combined with a nested iteration producing the Full-Multi-grid (FMG) algorithm: FMG involves n phases, working from the 1-st coarsest mesh to the n-th finest mesh. The j-th phase of FMG solves the appproximate PDE on the j-th mesh. This j-th phase starts from an interpolation of the result of $j-1$-th mesh and applies a certain number k_j of MG cycles with the available j coarser meshes, from 1-st to j-th. In an ideal case, the number k_j of cycles in each phase is the same. Then FMG has then an optimal complexity of $K.N$, predicted by theory and observed on many practical examples, see the reference book [23].

But this rosy picture needs some rectifications.

First, in many cases, the single local iteration of MG is frequently 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][22] or by anisotropic mesh coarsening [19], [17], [7], and/or more adapted inter-grid transfers. Indeed, in some case, the directly-coarser grid correction is not able to complement the fine grid iteration, or may even work in a defavourable way. As a consequence, the best set of grids to apply for MG acceleration is not necessarily the best set of grid to apply for the FMG process. In this paper, we apply an isotropic refinement for the FMG nested iteration and an anisotropic mesh coarsening for MG acceleration.

Second, for a lot of complex applications, it has been remarked that FMG does not work, in the sense that the usual stopping criterion produces a discrete final solution with an accuracy deteriorated by an insufficient iterative resolution. See for example [6]. Let us examine a possible FMG failure scenario. The two assumtions in FMG theory are (1) a MG convergence which does not depend of level fineness, (2) the asymptotic high-order convergence of the discrete solution to continuous one on the different meshes of FMG, including coarse ones. Assuming the MG cycling convergence is good, FMG failure can then be explained by the lack of asymptotic convergence to continuous, either because meshes are still too coarse, or because solution involves small details or singularities. As a consequence, when the basic FMG algorithm involving a fixed number of cycles per phase is applied, the solution produced at end of FMG may be inaccurate. Adjusting the number of cycles to the necessary convergence, if not done accurately, may result in a computational cost much larger than the one which the theory of FMG would let expect and may result in loosing the $K.N$ complexity. Stopping criteria for iterative solvers have been the topic of many published works. In the case of quasi-Newton iteration, a typical work is [16]. It is commonly admitted that the best criterion is to stop the iteration when the iteration error between iterated approximated solution and converged approximated solution is smaller than the approximation error between converged approximated solution and exact solution [23],[16]. But computing the approximation error is computationally costly. In [16] its evaluation is replaced by a assumption concerning $O(h^2)$ mesh convergence. In the present paper, we propose to use the a posteriori residual as reference for stopping MG cycles.

MG and FMG have been combined very frequently with mesh adaption. Let us cite a pioneering work of R. Bank, [2], and a few more recent ones, such as [21],[4],[20],[18]. Adaptive works are most frequently based on mesh refinement by local division, producing embedded meshes. More generally, unstructured non-embedded MG and FMG have been penalized during years by the difficulty in building and managing multiple coarse and fine unstructured meshes in particular for industrial applications. This difficulty is more easy to address today, with the recent progress of mesh generation and adaptation, see e.g. [10]. Due to this progress, novel anisotropic strongly mesh-adaptive algorithms are now available. By strongly mesh-adaptive we mean that an anisotropic mesh adaption is strongly coupled with the solver thanks to a nonlinear fixed point iteration. Anisotropic mesh adaptors have been observed as carrying two important advantages. First not only many computations are performed in much better conditions than with traditional methods, but also they allow computations which were simply *not feasible* without anisotropic adaptation, like the propagation of a sonic boom from aircraft to ground [15]. Second, anisotropic mesh adaptors provide mesh convergence at high-order for singular problems [12]. For non-singular problems but rather heterogeneous problems, non-adaptive methods will produce higher order convergence only with very fine meshes. Anisotropic adaption will give a high order numerical mesh convergence with a much smaller number of nodes.

The plan is as follows. The next section introduces Riemannian metrics for defining what we call a continuous mesh model and the fixed-point mesh adaptation algorithm. Section 3 combines MG and mesh adaption. Section 4 presents FMG and proposes a stopping criterion for it. Section 5 defines the complet proposed algorithm, combining FMG and anisotropic mesh adaption. The paper is completed by several test cases and a discussion.

2. Mesh parametrization and mesh adaptation loop

2.1. Continuous mesh model

We recall shortly the continuous mesh framework, introduced in [13, 14]. This framework lies in the class of metric-based methods. A continuous mesh $\mathcal{M}$ of the computational domain Ω is identified to a Riemannian metric field [3] $\mathcal{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric 2×2 matrix. Its diagonalisation writes:

$$\mathcal{M}(\mathbf{x}) = d(\mathbf{x})\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-1}(\mathbf{x}) & \\ & r_2^{-1}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x}), \tag{1}$$

The *total number of vertices* $\mathcal{C}$ is defined as:

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

A discrete mesh $\mathcal{H}$ of the same domain Ω is a *unit mesh with respect to* $\mathcal{M}$, if, to simplify, each edge $\mathbf{e} = \mathbf{ab}$ verifies $\int_0^1 \sqrt{{}^t\mathbf{ab}\,\mathcal{M}(\mathbf{a}+t\,\mathbf{ab})\,\mathbf{ab}}\,\mathrm{d}t \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right]$.

Given a smooth function u, to each unit mesh $\mathcal{H}$ with respect to $\mathcal{M}$ corresponds a local interpolation error $|u - \Pi_{\mathcal{H}} u|$. In [13, 14], it is shown that this interpolation error is well represented by the so-called continuous interpolation error related to $\mathcal{M}$, which is locally expressed in terms of the Hessian H_u of u as follows:

$$|u - \pi_{\mathcal{M}} u|(\mathbf{x}) \;=\; \frac{1}{10}\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x})|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})) \tag{2}$$

where $|H_u|$ 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 $\mathcal{C}$ equal to a parameter N. After solving analytically this optimization problem, this defines the unique optimal $(\mathcal{M}_{opt}^{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ as:

$$\mathcal{M}_{opt}^{\mathbf{L}^p} = D_{\mathbf{L}^p}\,(\det|H_u|)^{\frac{-1}{2p+2}}\,|H_u| \;\text{ with }\; D_{\mathbf{L}^p} = N^{\frac{2}{2}} \left(\int_\Omega (\det|H_u|)^{\frac{p}{2p+2}}\right)^{-\frac{2}{2}}, \tag{3}$$

where $D_{\mathbf{L}^p}$ is a global normalization term set to obtain a continuous mesh with complexity N and $(\det|H_u|)^{\frac{-1}{2p+2}}$ is a local normalization term accounting for the sensitivity of the $\mathbf{L}^p$ norm. In the sequel we choose $p = 2$.

2.2. Fixed-point mesh adaptation

In the case where the function u is the solution of a Partial Differential Equation, the Hessian-based method extends heuristically as follows. Given a discrete solution u_h to the PDE, a Hessian $H(u_h)$ is defined from it. The so-called optimal mesh $\mathscr{M}_{pde}$ is defined by:

$$\mathscr{M}_{pde} = \mathscr{M}_{opt}(H(u_h(\mathscr{M}_{pde})))$$

Where for any metric $\mathscr{M}$, $u_h(\mathscr{M})$ is the discrete PDE solution computed on a unit mesh for $\mathscr{M}$. We solve the non-linear problem giving the optimal mesh $\mathscr{M}_{pde}$ by applying the following loop:

Fixed point for adaptive PDE approximation
1- compute the PDE approximate solution u_h on current mesh $\mathscr{M}$
2- compute an approximate Hessian $H(u_h)$
3- adapt with N nodes according to this Hessian, obtain $\mathscr{M} = \mathscr{M}_{opt}(H(u_h))$
4- go to 1.

For remeshing phases, we used indifferently `MeshGems-Adapt` of Distene and in-house versions of `Yams` ([8, 9]). Due to the discrete and noisy character of remeshing, a strictly-fixed point cannot be obtained, but instead the iteration needs be stopped when further work would be useless. In contrast to the adaptation to an analytic function, the deviation to a target is not available. However, it is possible to rely on the approximation of the interpolation error given by integral (2). In [1], it is proposed to stop the iteration when the difference between to approximate solution field u_h is smaller than a positive quantity to choose cleverly since this difference does not converge to zero. The simple option of a fixed number of adaption iterations is also a rather secure one.

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:

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

As noted in previous section this is the solution of a non-linear coupling system, but the dependancy of $\mathscr{M}_h$ with respect to u_h is explicit, in the sense that the cost

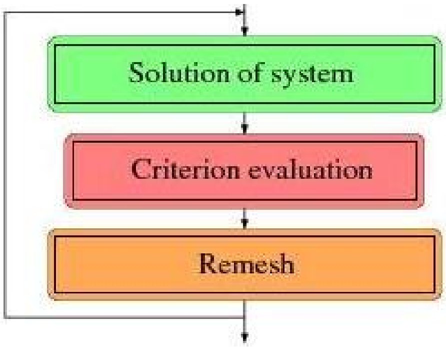

Figure 1: Mesh adaption loop

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. The resulting algorithm is depicted in Figure 2.

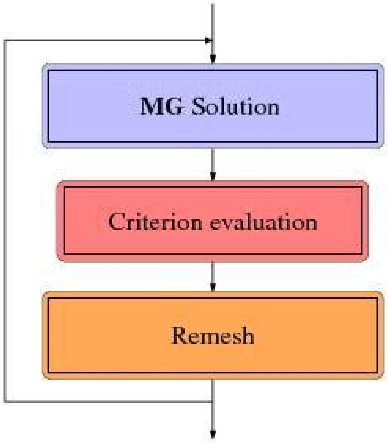

Figure 2: Mesh adaptation loop with Multigrid

The external loop of mesh adaption is iterated five times for convergence of the coupling bertween PDE solution and adapted mesh. We now define in more details the ingredients of this loop.

3.1. Anisotropic coarsening

Let us examine how to build coarser meshes in order to apply MG. The adopted stanpoint is to use the metric based mesh parametrization. Firstly, we specify the number of nodes N_k of the adapted fine mesh of the current phase, phase number k. We have to choose an initial metric $\mathscr{M}$:

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

with:
$\Delta\xi(x,y) =$ 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 specification of the number of nodes of this fine mesh writes:

$$\int (\Delta\xi \ \Delta\eta)^{-1} \, dxdy \ = \ N_k \tag{4}$$

where the integral is taken over the computational domain. 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. In particular the first coarser mesh has about $N_k/4$ nodes, the next coarser has about $N_k/16$ nodes, etc.

Anisotropic coarsening can also be applied. For example, assuming that the ordering of eigenvalues satisfies $\Delta\xi \leq \Delta\eta$, we can coarsen in an equivalent way to [7]:

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

This option has been tested but did not improve the results for the test cases we present, which involve meshes which are not enough stretched.

3.2. MG

The above metric coarsening produces a sequence coarse meshes $\mathscr{H}_1, ...\mathscr{H}_{\ell_{max}}$ to be used together with the fine initial mesh as *levels* $\ell+1, ..., \ell_{max}$ for a MG cycle. Those are kept during the MG cycles and regenerated during the adaptation phase when the fine mesh is adapted. For applying the MG cycle, transfers are defined as follows: correction transfers from coarse to fine are P_1 interpolated in triangles, and residual transfers, from fine to coarse, are accumulated on coarse nodes with barycentric weighting. A saw-tooth V-cycle with 10 Jacobi sweeps as pre-smoothing and without post-smoothing is applied.

3.3. Global fixed point

In the mesh-adapted MG, the adapted solution with a prescribed number of nodes N_k is obtained by encapsulating the MG cycle into the adaption loop. For the adaptation convergence, we have chosen to uniformly apply 5 adaptations.

4. The FMG algorithm

The adaptive FMG is the succession of adaptive MG phases with transfer of the solution between each phase. At phase k_ϕ, the number of nodes is prescribed to be equal to N_{k_ϕ}. We have chosen the usual option of a new mesh size two times smaller in next FMG phase:

$$N_{k_\phi+1} = 4\ N_{k_\phi}. \tag{5}$$

The resulting approximation error will be presumably 4 times smaller. With $N_{k_\phi+1} = 2\ N_{k_\phi}$ the error would be 2 times smaller, and so for the necessary cycles but two phases would be necessary for the same final accuracy, with a similar global cpu effort. Therefore the choice of $N_{k_\phi+1}/N_{k_\phi}$ is not a sensible one.

4.1. Global FMG under O2 convergence assumption

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 solution is transfered to the new mesh, typically by interpolation. The two meshes are available for playing the role of two levels in order to solve fastly the problem with a two-grid iteration on the new mesh, starting with a good initial condition. This process is reiterated with a 3-grid solution on next mesh etc. We call *FMG k_ϕ-th phase* the k_ϕ-th nested iteration phase, using with *k_ϕ-grid cycling*. Due to the initialization by the previous phase, a sufficient convergence at each phase k_ϕ can be obtained by a small number k_c of k_ϕ-grid cycles. Due to the ability of MG in exhibiting a convergence rate quasi-independant of number of level and grid size, an important gain is reachable. The prescription in stopping iterative solutions is generally based on the following criterion:

Stopping criterion: *Iteration error should be smaller than the approximation error.*

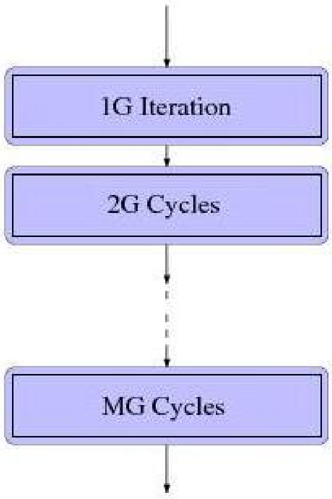

Figure 3: Each phase k_ϕ of the Full Multi-Grid algorithm is made of several k_ϕ-grid cycles

In the FMG theory presented in [11], it is assumed that there exists a constant K such that for any phase k_ϕ, we have

$$||u - u_{k_\phi}|| \leq K h_{k_\phi}^2. \tag{6}$$

The error for u_{k_ϕ} is four times larger than for $u_{k_\phi - 1}$, which suggests a residual reduction by 4 at each phase. A more accurate analysis ([11]) shows that it is enough to converge each FMG cycling phase by dividing the residual by 10, for example:

$$||\mathbf{A}_h u_h^{k_c} - f_h|| \leq \frac{1}{10} ||\mathbf{A}_h u_h^0 - f_h||. \tag{7}$$

The weakness of this theory lies in the central assumption (6). An evident first remark is that (6) is a mesh convergence assumption, which can be established only for a fine enough mesh, that is for *grids/levels* ℓ with ℓ large enough. For ℓ close to unity, the number of nodes is small and the above estimate is clearly *not true*. But the main disaster arises when (6) does not holds for the last fine grids. In that latter case, the dilemma of FMG is:

Either the convergence test (7) based on a fixed decrease of residual at each phase, being not sufficient, will produce an inaccurate solution on the finer mesh, or, assuming we have found a sufficiently severe way to stop convergence in each phase, we get an accurate solution, but we have increased the cost in a rather unpredictible way and have presumably lost the optimal $K \cdot N$ complexity of FMG.

We are also motivated by a second issue: if a mesh adaption loop is applied, the MG cycling between mesh updates will be initialised by a candidate solution obtained with a just slightly less adapted mesh of same fineness. This candidate solution can be already very close to the converged solution. Then, in order to avoid a rather large amount of unnecessary computing time, we need to recognize it and stop early the MG cycling.

Next two paragraphs deal with introducing a measure of approximation error and a control of iteration error in FMG, in order to improve the cycling stopping test for FMG, consolidating the accuracy of FMG. The second issue, efficiency, will be addressed by introducing mesh adaption, in Section 7.

4.2. Approximation error estimate

Let

$$u_h^k(x) = \sum \mathbf{u}_{h,i}^k N_i(x)$$

be the approximate function at iteration k of a given iterative solver (N_i holds for the finite-element basis function related to node i). For a second-order accurate approximation, Arioli and co-workers propose in [16] a stopping criterion based on:

$$||\mathbf{A}_h u_h^k - f_h||_{H^{-1}} \ \leq \ h^2 \ ||f_h||_{H^{-1}}$$

where H^{-1} holds for the dual of the Sobolev space H_0^1. Using the h^2 factor assumes that the numerical mesh-convergence is close to scheme asymptotic convergence.

We propose here a method which does not rely on mesh convergence. Let us introduce some notations: Let $V = H_0^1(\Omega)$, Ω being the computational domain. The continuous PDE system is written in short:

$$Au = f \ \ or \ \ u \in V \ \ \forall\, \phi \in V \ \ a(u,\phi) = (f,\phi)$$

Let V_h be the usual P_1-continuous finite-element approximation space. The finite-element discretisation is written:

$$u_h \in V_h \ \ \forall \ \phi_h \in V_h \ \ a(u_h, \phi_h) = (f, \phi_h)$$

We introduce the components of u_h for the finite-element basis (N_i) as follows, with $T : \mathbb{R}^n \to V$:

$$u_h = \sum \mathbf{u}_{h,i} N_i \ \ \Leftrightarrow \ \ u_h = T\mathbf{u}_h$$

Let us denote T^* the adjoint of T:

$$T^* : V' \to \mathbb{R}^n \ \ \ [T^* f]_i = (f, N_i).$$

The *variational discretization*:

$$a(\sum \mathbf{u}_{h,j} N_j, N_i) = (f, N_i) \ \ \forall i \ \ \Leftrightarrow \ \ \sum a(N_j, N_i)\mathbf{u}_{h,j} = (f, N_i) \ \ \forall i.$$

transforms into an algebraic one:

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{f}_h, \ \text{ where } \ [\mathbf{A}_h]_{ij} = a(N_j, N_i) \ \text{ and } \ \mathbf{f}_h = T^* f. \tag{8}$$

The exact *a posteriori* estimate:

$$u - u_h = A^{-1}(f - Au_h)$$

can approximated as:

$$u - u_h \approx T\mathbf{A}_h^{-1} T^* \ (f - \hat{A}u_h)$$

where $\hat{A}u_h$ denotes a smoothed approximation of Au_h

$$(\hat{A}u_h, \phi) = \sum_{ij} \frac{|D_{ij}|}{|ij|} \int_{D_{ij}} \phi [\nabla u_h]_{ij} \cdot \mathbf{n}_{ij} \mathrm{d}v$$

where :
- ϕ is an arbitrary function of V,
- the sum $\sum$ is taken for all internal edge ij of the mesh (2D),
- [] holds for the jump of quantity inside bracket across the edge ij,
- the integral $\int_{D_{ij}}$ is taken over surface (2D) of the diamond quadrilateral $D_i j$:

$$D_i j = iG_{ij} jG_{ji}$$

where G_{ij} and G_{ji} are the centroids of the two elements (triangles) having ij as common edge.

4.3. Iterative-Error-controlled FMG

Now the algebraic system (8) is solved by a number k_c of cycles:

$$\mathbf{u}_h^0 = 0 \quad ; \quad \mathbf{u}_h^{k_c+1} = Iterate(\mathbf{u}_h^{k_c}) \quad ; \quad \mathbf{u}_h^\infty = \mathbf{u}_h$$

The *iterative error* can be evaluated by solving the system with a right-hand side equal to the local iteration residual:

$$\mathbf{u}_h - \mathbf{u}_h^{k_c} \; = \; \mathbf{A}_h^{-1}(\mathbf{f}_h - \mathbf{A}_h\mathbf{u}_h^{k_c}).$$

It remains to compare the iterative error with the above approximation error. Remember first that the approximation error above derivation did not use the assumption that u_h is the solution of the discrete system. In particular, the same estimate holds for the result $u_h^{k_c}$ of the incomplete iterative resolution. Let us introduce the element of V_h:

$$u_h^{k_c} = T\mathbf{u}_h^{k_c} = \sum [\mathbf{u}_h^{k_c}]_i N_i,$$

Then, using the *a posteriori* error estimate:

$$u - u_h^{k_c} \approx T\mathbf{A}_h^{-1}T^* \; (f - \hat{A}u_h^{k_c}). \tag{9}$$

It is now useful to transform the algebraic iterative residual $\mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}$ in similar terms. We start from:

$$\mathbf{A}_h(\mathbf{u}_h^{k_c} - \mathbf{u}_h) = \mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h - (\mathbf{A}_h\mathbf{u}_h - \mathbf{f}_h) = \mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h,$$

thus

$$\mathbf{u}_h^{k_c} - \mathbf{u}_h = \mathbf{A}_h^{-1}(\mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h) \quad \Leftrightarrow \quad u_h^{k_c} - u_h = T\mathbf{A}_h^{-1}(\mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h). \tag{10}$$

Heuristics: *Assuming that, in some norm to specify later,*

$$||u - u_h|| \leq \varepsilon, \;\; \varepsilon \textit{ small and positive},$$

and that after k_c solver iterations, we have

$$||u_h - u_h^{k_c}|| \leq \; 0.1 \; ||u - u_h^{k_c}||, \tag{11}$$

then

$$||u - u_h^{k_c}|| \leq \; 1.1 \; ||u - u_h||. \tag{12}$$

As already mentioned, the last statement (12) is for us an acceptable iterative convergence stopping criterion. We observe that stopping criterion (11) is realizable, since as iteration number increases, the norm $||u - u_h^{k_c}||$ is supposed to converge to $||u - u_h||$, assumed to be not zero, while $||u_h - u_h^{k_c}||$ can be driven to machine zero by iterating over k. Unfortunately, evaluating the two terms of (11) involve solving two discrete systems with matrix A, a computation which is more or less computationally as costly as the original system to solve, and is therefore too costly. A possible solution is to solve approximatively the discrete error system with a coarse grid. Here, we propose to decrease the cost of the stopping test, with some risk of decreasing its accuracy, by taking the following l^1 norm of the right-hand side:

$$||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} = \sum_i |[\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}]_i|$$

We rely on the observation that MG-cycles decrease many different norms of the residual with about the same slope. Therefore, we do not claim that the proposed stopping criterion is able to work adequately when associated with another iteration than MG.

Stopping test 1: *Assume that after k solver iterations*

$$||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} \leq \varepsilon' ||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^0||_{l^1}, \quad \varepsilon', \text{ small and positive,} \tag{13}$$

and that we have

$$||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} \leq \ 0.1 \ ||T^* \ (f - \hat{A} u_h^{k_c})||_{l^1}, \tag{14}$$

then stop the iteration.

According to (7), ε' is chosen to be $\frac{1}{10}$. In practice, since the computation of the RHS of (14) may need more cpu than a cycle, the test (13) means that several iterations are performed in order to decrease the iterative residual to satisfy (13) before the second test (14) is evaluated. If test (14) is negative, several iterations are again performed before a second test of (14) is again evaluated, etc. This splitting allows for a lower CPU cost. This device is inspired by an analog one proposed by Arioli and co-workers [16]. Our final formulation is as follows:

Stopping test 2:
1. Iterate cycling until $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} \leq \frac{1}{10} ||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^0||_{l^1}$
2. If $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} > \ 0.1 \ ||T^* \ (f - \hat{A} u_h^{k_c})||_{l^1}$ *then* $\mathbf{u}_h^0 = \mathbf{u}_h^{k_c}$ *go to 1.*
3.Stop the iteration.

4.4. Application to the proposed FMG

The MG algorithm which we use is built from:
- a sequence of unstructured meshes which are not necessarily embedded. the basic inter-grid transfers are classically the P^1 interpolation (for a transfer from a coarse mesh to a finer mesh) and an accumulation weighted with barycentric ceofficients (for a tranfer from a fine mesh to a coarse mesh).
- a saw-tooth V-cycle with 10 damped-Jacobi relaxation a a smoother.
- an encapsulation of this MG cycle as a preconditioner of a GMRES loop. When we shall talk about "a cycle", we shall mean the combination of the MG V-cycle with the GMRES updating.

The rather high number of sweeps, together with the use of GMRES is the price we pay in order to get a robust convergence for high-density ratio case.

Three contexts are now examined for showing how works the combination of FMG with the stopping criterion.

- Fig.4: the function to compute is not important, but to fix the ideas, it is the circular test case described in the sequel. The initial solution is uniform. We apply 90 GMRES-MG cycles. The preconditioned (by MG) residual l^1 norm (started at iteration 1) indeed decreases momotonely and fastly from 1 to 10^{-8} (multiply-shaped marked curve). Marked by plus-covered-by-multiply, the equation residual norm $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1}$ starts from less than 1, shows an increasing phase and then decreases to about 2.5 10^{-3}. Marked by plus-symbols, the approximation residual norm $||T^* \ (f - \hat{A}u_h^{k_c})||_{l^1}$ also first increases in a similar way and then goes down to a non-zero limiting value. The two above above curves intersect at about 70 GMRES-MG cycles. This is probably too many cycles, *i.e.* rather conservative. In the chosen example we know the exact solution and can also depict the approximation error norm $||u - u_h^{k_c}||_{L^1}$, which starts from a number close to 1 and decreases to a limiting level $||u - u_h||_{L^1}$ of about 5 10^{-5}. We get confirmation that with 70 cycles, this level is tightly approached.

- Fig.5: the main change is that the initial solution is provided by interpolating the discrete solution computed on previous coarser mesh. Cycles are stopped at 34. The preconditioned (by MG) residual l^1 norm starts with a small level, and decreases fastly to 5 10^{-8} (multiply-shaped marked curve). Marked by plus-covered-by-multiply, the equation residual norm $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1}$ starts from about 0.5, and monotonely (this time) decreases to about 2 10^{-3}. Marked by plus-symbols, the approximation residual norm $0.1||T^* \ (f - \hat{A}u_h^{k_c})||_{l^1}$ starts from a lower value then previous case and monotonely decreases in a similar way to equation residual, going down to a non-zero limiting value. The two above above curves intersect at about 22 GMRES-MG cycles. In contrast to the previous case, the approximation error norm $||u-u_h^{k_c}||_{L^1}$, starts from a low level of about 10^{-4} and decreases to about 5 10^{-5}, indicating (1) that the numerical convergence is not so good (first-order) between the two meshes and (2) that again the iterative error at iteration 22 is much smaller than the approximation one.

Fig.6: we give an example of FMG sequence for a slightly easier problem (Laplace equation with uniform meshes).The convergence on four sucessive meshes is shown. The approximation error is numerically converging at second order. With mesh 2 and mesh 4, the second test is negative and a second MG convergence is applied. The four phases are complete with a total of 16 cycles, that is a reasonable mean number of 4 cycles per FMG phase.

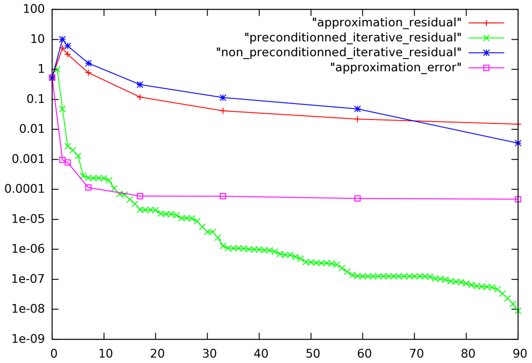

Figure 4: Convergence of the GMRES-MG-iterative l^1 residual norm $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ ($\ast$), the approximation l^1 residual norm $||\hat{A}u_h^k - f||_{l^1}$ ($+$), the preconditioned residual norm ($\times$), the norm $||u-u_h||_{L^1}$ of deviation to exact ($\Box$), for a Poisson problem, starting from a uniform field $u_h^0 = 1$ at iteration 0.

5. The FMG Anisotropicly adaptive algorithm

The synthesis of the above sections is the FMG anisotropicly adaptive algorithm. We insert the adaptation loop as an intermediate loop between FMG phases and MG cycles. Concerning the process of going to a larger number of nodes, we keep the previous meshes and define a finer one by a simple division of each element into four elements of same area. The global algorithm is sketched in Fig. 7. Let us re-visit the ways these loops are stopped. The *external loop*, FMG phase will increase the number of mesh nodes. Theoretically it should stop when some norm of the approximation error $|u-u_h|$ is smaller than a number prescribed by

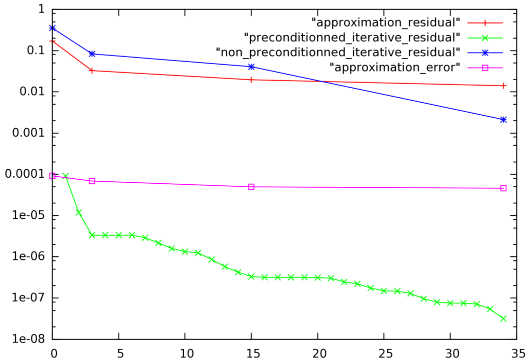

Figure 5: Convergence of the GMRES-MG-iterative l^1 residual norm $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ ($\ast$), the approximation l^1 residual norm $||\hat{A}u_h^k - f||_{l^1}$ ($+$), the preconditioned residual norm ($\times$), the norm $||u - u_h||_{L^1}$ of deviation to exact ($\square$), for a Poisson problem, starting from a coarser-grid interpolated solution at iteration 0.

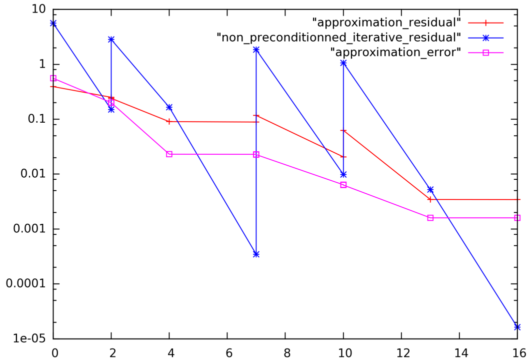

Figure 6: Convergence of the FMG-iterative l^1 residual norm $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ ($\ast$), the approximation l^1 residual norm $||\hat{A}u_h^k - f||_{l^1}$ ($+$), the norm $||u - u_h||_{L^1}$ of deviation to exact ($\square$), for a Poisson problem, starting from the coarsest-grid (grid 1) solution, and performing four FMG phases from grid 2 to grid 5.

user. This option is yet rather far from practice and is not studied in this work. The *intermediate loop*, mesh adaption is stopped after 5 iterations. As concerns *inner loop*, the cycling loop, it is controlled by the stopping criterion defined in Section 6. The stopping criterion is used when (1) changing from a first mesh of N nodes to a finer mesh of $4N$ nodes between two FMG phases, as well as (2) changing from a first mesh of N nodes to a second mesh of same number N of nodes, but more adapted.

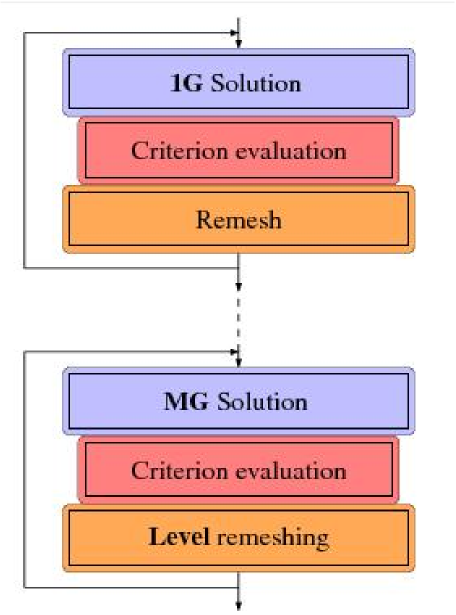

Figure 7: Mesh adaptive Full Multi-Grid

6. Examples

The proposed Adaptive Anisotropic FMG is applied to three test cases and compared with a pure FMG method applied with a sequence of embedded uniform meshes. Due to our restriction to a Poisson-like model, test cases will be toy problems. However, we choose them in order to represent the three following typical difficulties of multiphase incompressible flows:

- boundary layers,
- discontinuous phase changes,
- Dirac layer source terme from capillarity.

Our three simplified representations of these difficulties, in combination with the specification of mesh sizes (number of vertices) constitute a small benchmark for the performance of mesh adaptive methods, which could be used for the evaluation of various sensors.

In cases where the test case has an analytic solution, we shall call *total approximation error* or simply *approximation error* the error between the analytic solution from one side, and, from the other side, the discrete solution produced the algorithm: since the GMRES-MG algorithm is not converged to machine-zero, our approximation error combines numerical scheme approximation error and iterative error.

6.1. A smooth boundary layer test case

For modelling the stiffnes of a boundary layer, we consider a Poisson problem with a smooth solution presenting some anisotropic local variation. Let:

$$rhs(x,y) = \left[\alpha^2(exp(1/\alpha)-1)\right]^{-1} exp(x/\alpha) \text{ with } \alpha = 0.03.$$

We solve $-\Delta u = rhs$ with $\frac{\partial u}{\partial y}(x,0) = \frac{\partial u}{\partial y}(x,1) = 0$ and $u(0,y) = u(1,y) = 0$. Then $u(x,y) = [exp(1/\alpha)-1]^{-1} exp(x/\alpha) + x + [exp(1/\alpha)-1]^{-1}$. An example of approximated solution u_h is shown in Figure 8. That allows us to compute directly

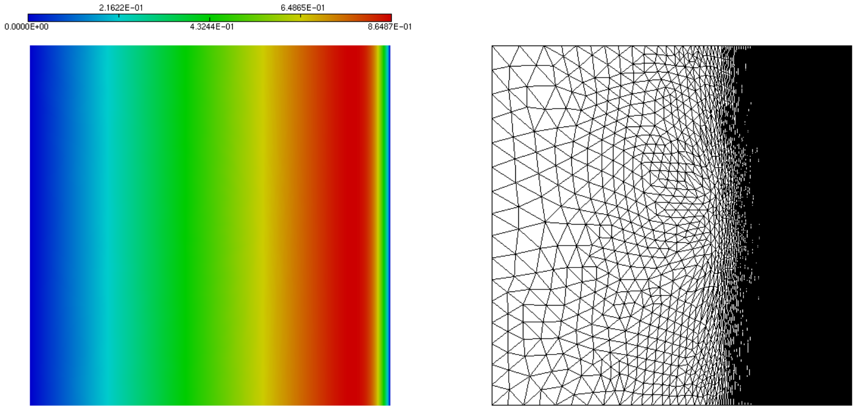

Figure 8: Boundary layer test case solution and adapted mesh

the norm of the approximation error defined by $||u-u_h||_{L^1} = \int_\Omega |u-u_h| dxdy$ which is depicted as a function of the number N of nodes of the mesh. For evaluation of our Hessian-based criterion, we draw the error of interpolation of u on the current meshes as a function of the number N of nodes of the meshes, in Figure 9. We observe a convergence of order two for the non-adaptive case and a similar convergence in the adaptive case. Since the criterion for adaption which we adopted postulates that the interpolation error is a good representation of approximation error, it is interesting to examine the convergence of the interpolation error of the exact solution on the meshes we used, see Figure 10. We observe that indeed both convergence are similar, but that the interpolation error decreases to values which are smaller by a factor larger than two orders of magnitude. This observation will be also done for the other test cases and deliver a strong message saying that the interpolation error and its companion the Hessian criterion, while providing rather good adaptations, are not faithful representations of the actual approximation error. The difference between Figure 9 and Figure 10 measures

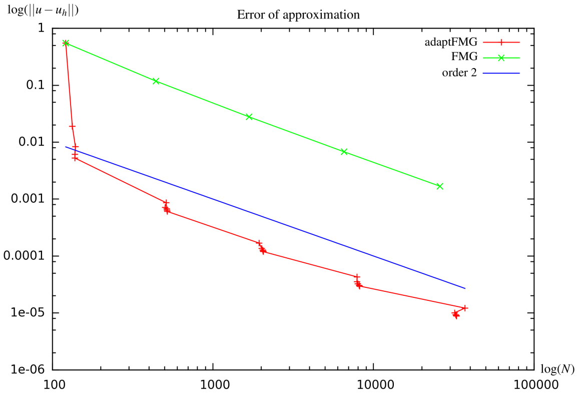

Figure 9: Boundary layer test case. Approximation error $||u-u_h||_{L^1}$ as a function of the number of mesh nodes. (+) non-adaptive FMG, (×) adaptive FMG. The straight line shows the second-order slope.

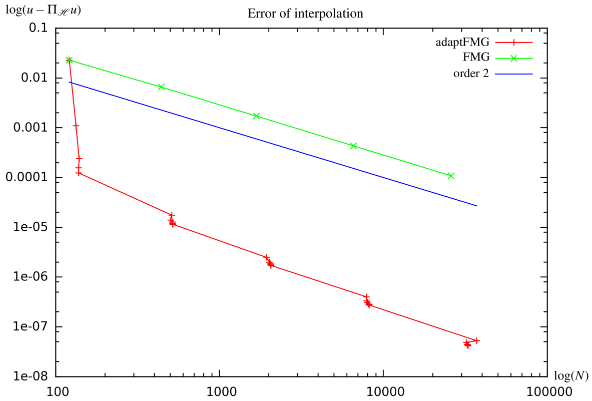

Figure 10: Boundary layer test case. Corresponding behavior of the interpolation error of exact solution $||u-\Pi_h u||_{L^1}$ on the same meshes as in Fig.9. The straight line shows the second-order slope.

the relative inadequacy of the Hessian-based option. Comparing the non-adaptive case and the adaptive one (Figure 9) for the same number of vertices, we observe that the error $||u-u_h||_{L^1}$ is notably smaller in the adaptive case. We also draw this error in function of the CPU time in Figure 11. We distinguish mesh division phase with steep slopes from the mesh adaption ones with less steep slopes.

Thanks to the stopping criterion, the iterations 2 to 5 of adaption consume less cycles and therefore less CPU. For about 1000 seconds of a workstation, the obtained accuracy is $6 10^{-3}$ for the non-adaptive case, and 10^{-5} for the adaptive one. the accuracy fo the non-adaptive calculation with 750 seconds is obtained by adaption with 10 seconds.

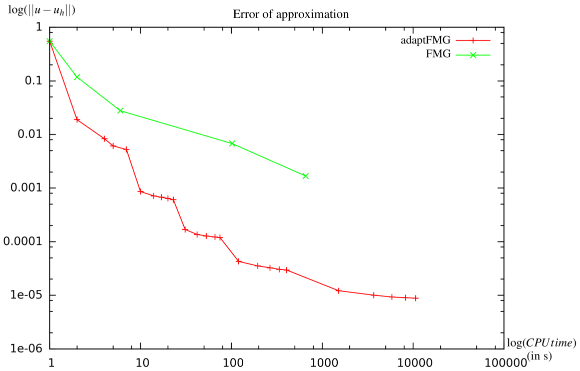

Figure 11: Boundary layer test case. Approximation error $||u-u_h||_{L^1}$ as a function of the CPU time. (+) non-adaptive FMG, (×) adaptive FMG.

6.2. *A non-smooth internal layer case*

The second test case exemplifies the singularity which is met in the simulation of multiphase non-mixed flows with a large deviation between the physical properties of each phase. Let us consider the equation of Poisson $-div(\frac{1}{\rho}\nabla u) = rhs$ with discontinuous coefficient ρ and a right hand side rhs which are strongly discontinuous on the domain, as it is shown in Figure 12. The solution u of the homogeneous Dirichlet prolem has discontinuous gradients along the coefficient discontinuity. A mesh-adaptive approximation u_h is depicted in Fig.13. Since this time an analytic solution is not available, we compute the L^1-norm of the solution u_h: $||u_h||_{L^1} = \int_\Omega |u_h| dxdy$ and compare with an interpolated evaluation on uniform mesh. Figure 14 shows this norm in function of the number of points. It is also shown in Tables 1 and 2. The uniform-mesh approach is definitively penalized by the singularity of the solution. We can expect first-order convergence and indeed the observed numerical convergence order of the non-adaptive L1-norms is 0.96 for the finest computations. Since the L^1 norm is an integral, we can try an extrapolation of it, which gives $||u||_{L^1} \approx 0.82$, rather close to our mesh adaptive results. But the L^1 norm obtained with more than a $100,000$-node uniform mesh and a CPU time of 1811 seconds still show an error of more than 8%. With adaptation,

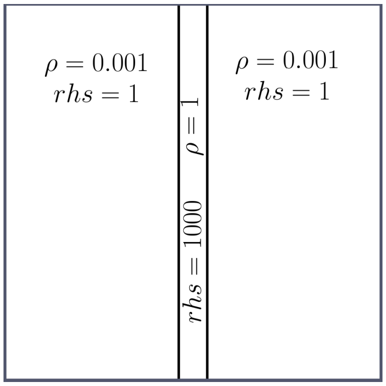

Figure 12: Stiff layer test case domain

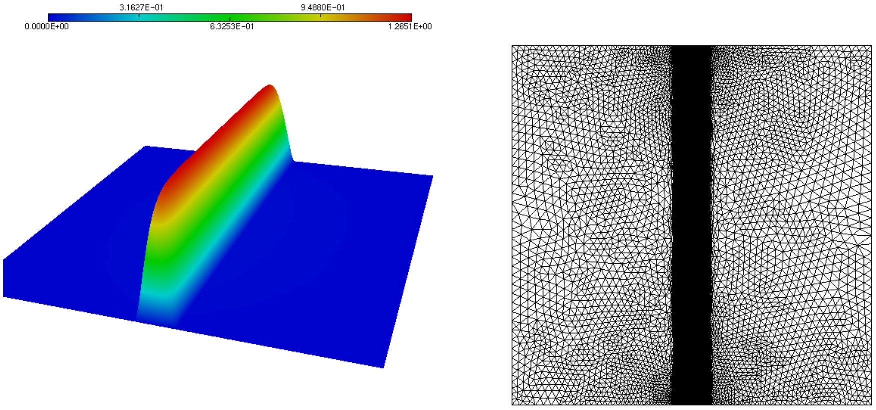

Figure 13: Stiff internal layer test case solution and adapted mesh

numerical convergence is rather noisy. However, a L^1 norm at less than 1% from fine-mesh one is already obtained with 552 nodes and a CPU time of 57 seconds. Figure 15 shows the same norm in function of the CPU-time.

6.3. Circular test case

Capillary models exhibit, along the interfaces, Dirac layer source terms for the pressure equation. These terms imply discontinuous pressures. For example the pressure could be is equal to 1 on a disk at center and equal to 0 in the rest of the domain. Instead of considering a strictly discontinuous solution, we approach it by defining a thickness ε of the layer between the two uniform phases as shown in Figure 16. If (x, y) is located inside the thickness of the layer, $u_c(x, y)$ is

Numb. nodes L^1 norm	121 0.1354	441 0.01929	1681 0.03806	6561 0.05679	25921 0.06869	103041 0.07488

Table 1: Stiff internal layer: convergence of L^1-norm of the approximate solution for a series of embedded uniform meshes.

Numb. nodes L^1 norm	142 0.07512	552 0.08211	2089 0.08292	9243 0.0831	36126 0.08376

Table 2: Stiff internal layer: convergence of L^1-norm of the approximate solution for a series of adapted meshes.

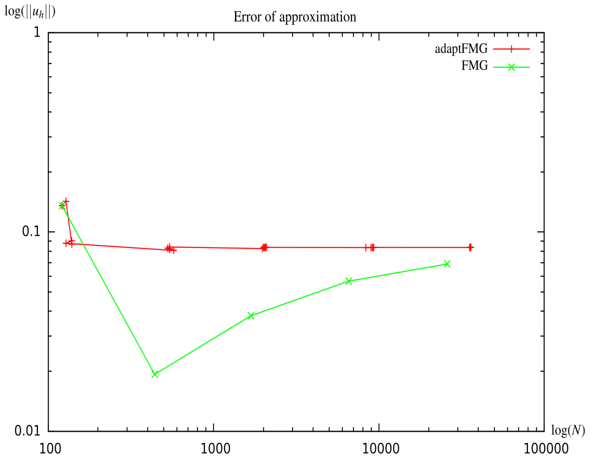

Figure 14: Stiff layer test case results: L1-norm $||u_h||_{L^1}$ of the approximate solution as a function of the number of points.($\times$) non-adaptive FMG, ($+$) adaptive FMG.

given by: $u_c(x,y) = \frac{1}{2}[1+\frac{\psi}{\varepsilon}+\frac{1}{\pi}sin(\frac{\pi\psi}{\varepsilon})]$ with $\psi = R-\sqrt{(x_C-x)^2+(y_C-y)^2}$. The value of ε controls the thickness of the transition between $u_c = 1$ and $u_c = 0$ and is chosen equal to 0.02. Let $rhs = \Delta u_c$. We consider the Dirichlet problem $\Delta u = rhs$ in Ω, $\;u = 0$ on $\partial\Omega$. The right-hand side rhs is close to a Dirac distribution concentrated along the circle limiting the disk. In practical nonlinear situations as capillary models, that kind of feature is not *a priori* known. Then we choose in our variational formulation to integrate the discrete RHS (rhs, N_i) (N_i: finite element basis function) on the given mesh without particular care of

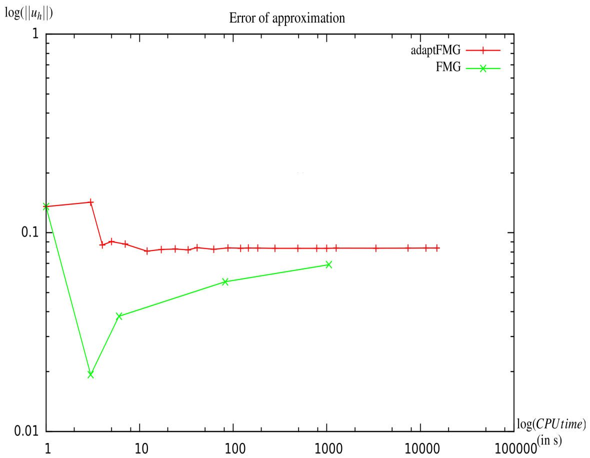

Figure 15: Stiff layer test case results: L1-norm $||u_h||_{L^1}$ of the approximate solution as a function of the CPU time.(×) non-adaptive FMG, (+) adaptive FMG.

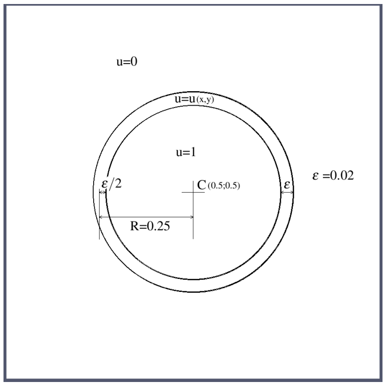

Figure 16: Circular-test-case-domain

the quasi singularity of *rhs*, which means that a coarse mesh may produce a very inaccurate solution. Indeed, in our computations we observe in Fig. 22 that very large errors are produced by coarse uniform meshes. With $10,000$ nodes, a 100 % L^1 error is still produced. Full second-order asymptotic convergence seems be reached only after 30000 nodes are used. This behavior can be a strong handicap for 3D calculations where the number of nodes cannot be much increased. In contrast, the mesh adaptive computation produces much smaller errors with coarse meshes and always perform as well or better. An mesh-adapted approximate solution u_h is shown in Figure 17. A second remark is that most gain of adaption is obtained at about $1000-2000$ nodes while adaption for finer meshes seem to

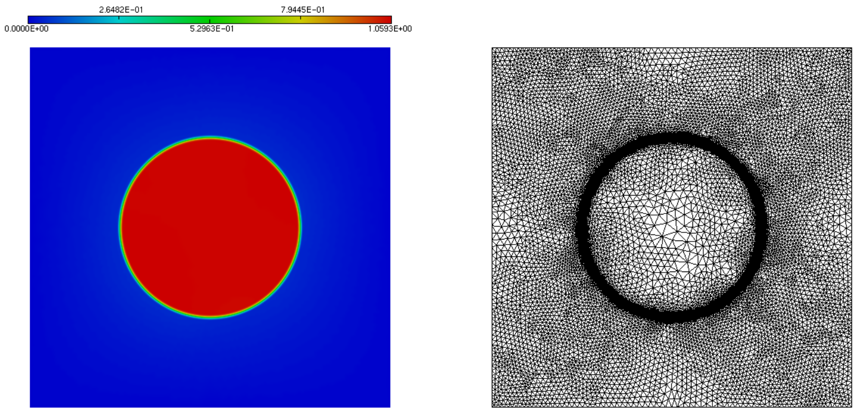

Figure 17: Circular layer test case solution and adapted mesh

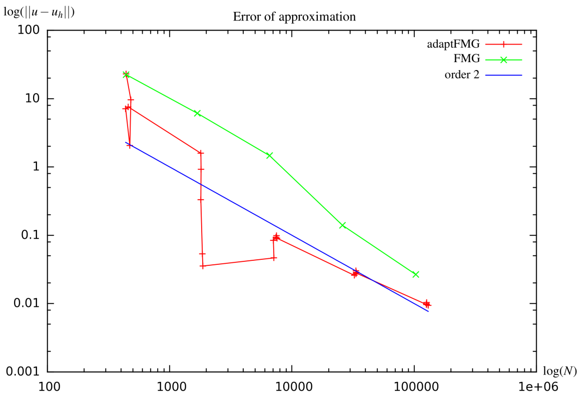

Figure 18: Circular test case: errors as functions of the number of mesh nodes. Approximation error $||u-u_h||_{L^1}$ as a function of the number of mesh nodes. ($\times$) non-adaptive FMG, ($+$) adaptive FMG. The straight line shows the second-order slope.

bring no further acceleration (slope remains second-order). This phenomenon is observed also for the interpolation error of the exact solution onto the different meshes which are used. Our interpretation is that once the layer around the circle is captured, no further adaption in needed, since a fine enough, but uniform mesh is quasi-optimal in the vicinity of the layer described by a *sinus* function. Then the subsequent efforts in adaption are useless, which explains that after an interesting performance for 2000 nodes, the adaption option looses its CPU advantage.

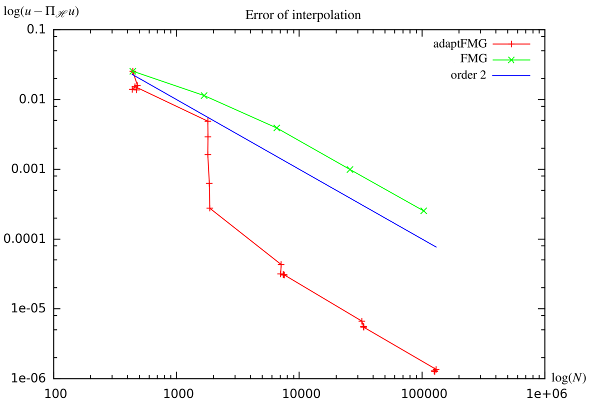

Figure 19: Circular test case. Behavior of the interpolation error of exact solution $||u-\Pi_h u||_{L^1}$ as a function of the number of mesh nodes, on the same meshes. The straight line shows the second-order slope.

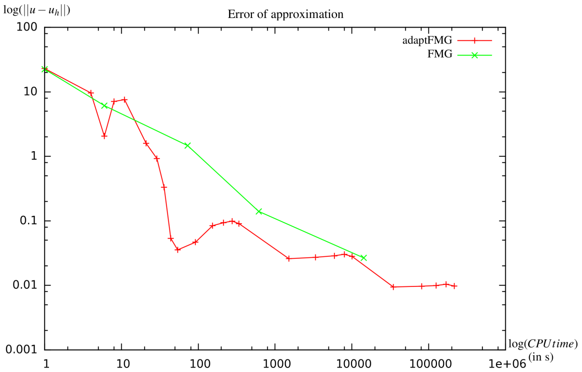

Figure 20: Circular test case: errors as a function of the CPU time. Approximation error $||u-u_h||_{L^1}$ as a function of the CPU time. ($\times$) non-adaptive FMG, ($+$) adaptive FMG. The straight line shows the second-order slope.

6.4. *Thinner circular test case*

Now, the value of ε controlling the thickness of the transition between $u=1$ and $u=0$ and is chosen equal to 0.001. An approximate (adapted) solution u_h is shown in Figure 21 (right). Now, due to the very thin definition of the Dirac-type right-hand-side, the brut force use of an uniform mesh of $100,000$ nodes does not allow the computation of a good solution, see Fig. 21 (left). Probably, a good solution is obtainable when a sufficiently fine uniform mesh is considered in a subsequent nested-iteration phase, but in that latter case, MG convergence on that mesh would need be iterated during many iterations and the FMG ideal

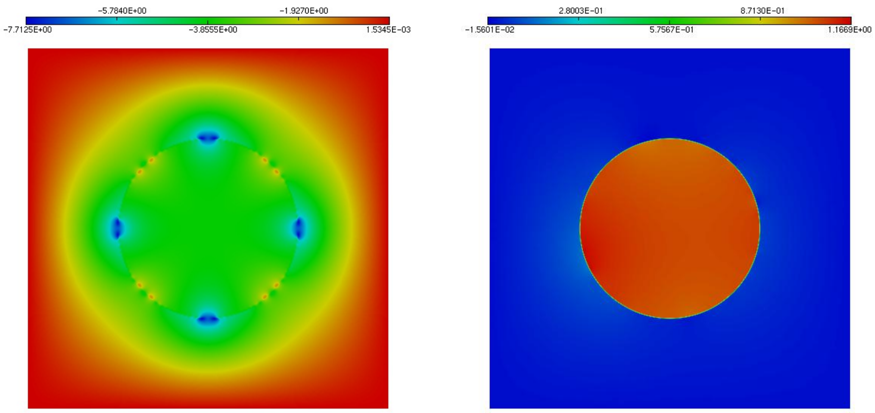

Figure 21: Thin-circular layer test case uniform-mesh solution and mesh-adaptive solution.

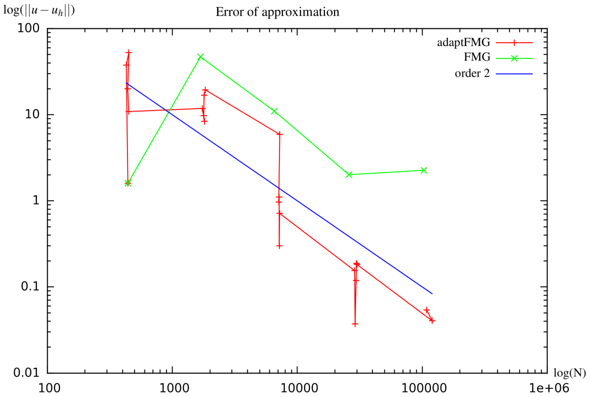

Figure 22: Thin circular test case: errors as functions of the number of mesh nodes. Approximation error $||u-u_h||_{L^1}$ as a function of the number of mesh nodes. ($\times$) non-adaptive FMG, ($+$) adaptive FMG. The straight line shows the second-order slope.

complexity is lost. In the same figure is depicted the approximate mesh-adaptive solution with the same number of nodes. The adaptive L^1 approximation error norm is 0.3 with $10,000$ nodes, and 0.04 with $100,000$ nodes.

7. Concluding remarks

We have proposed a combination of the well-established FMG method with an anisotropic mesh adaption method. The mesh adaption fixed point loop is introduced in the FMG process.

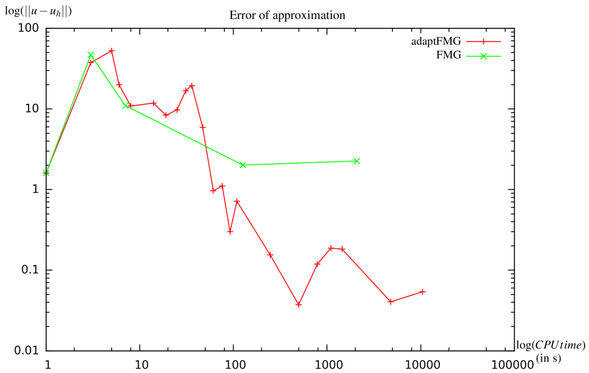

Figure 23: Thin circular test case: errors as a function of the CPU time. Approximation error $||u-u_h||_{L^1}$ as a function of the CPU time. ($\times$) non-adaptive FMG, ($+$) adaptive FMG. The straight line shows the second-order slope.

In order to master the extra computational complexity, an improved stopping criterion for MG cycling inside FMG is proposed. We believe that it is important for robustness and efficiency. We emphasize that this stopping criterion assumes that the iterative MG convergence rate is more or less norm-independant, a property surely not enjoyed by many other (non-MG) solution algorithms, for which our stopping criterion is definitively not recommanded.

The number of adaption iterations is fixed once for all. The overall anisotropic adaptive FMG is of rather high programming/algorithmic complexity, due to the higher number of tests and embedded loops. The central question is therefore: does it enjoy a robust computational efficiency, that is, is it computationally efficient for difficult problems.

The few numerical experiments tend to promote a positive answer. The four test cases are run with a unique set of parameters, *i.e.* without parameter tuning. In contrast to the non-adaptive case, with the use of anisotropic adaptation, we observe the early capturing of many different scales. A significative comparison between AFMG and FMG relies on the total approximation error as a function of CPU time. In most test cases, for a same CPU time, the mesh adaptive computation produces a lower or much lower approximation error. For a same CPU time, the number of nodes is much lower. A second measure concerns the asymptotic behavior in terms of number of unknowns and CPU. When it works, our FMG indeed shows $O(N)$ complexity, namely the considered norm of the total approximation error is proportional to the logarithm of the number of nodes N. This also approximatively holds for the AFMG version. For FMG, with some variations, the norm of total approximation error is also proportional to the logarithm of CPU time. Because of our choice of limiting adaption iteration to 5, we expect that this also holds for AFMG, and we observe it, again approximatively. The discontinuous coefficient case desserves a particular mention since even for the L^1 norm of solution, the convergence on uniform meshes is first-order. The improvement in that case is of two orders of magnitude.

We have not proposed a smart stopping criterion for the adaption loop. We plan to discuss this issue in a forthcoming paper in combination with a different mesh adaption criterion.

Indeed, the present study relies on the Hessian-based mesh adaptive criterion. This simple and robust option has some limits, which we have measured by comparing the convergence of the interpolation error and the convergence of the actual approximation error. The Hessian-based criterion is designed exactly for the interpolation error, and, by the way, the interpolation error converges fastly to small values. This shows that the different approximations of u by u_h and of the Hessian of u_h do not introduce a too important penalty. The -expected- bad news is that the approximation error does not decrease as fastly and as low. We interpret this as an effect of the lack of consistency between the interpolation error and the approximation error. Introducing adjoint-based adaption criteria may improve this issue. We plan to discuss this in a forthcoming paper using the benchmark proposed in the present work.

Of course, we are not sure that the extension to 3D will enjoy the same qualities, but the present results are encouraging. Then, such an evaluation is now in progress. Also, in our opinion, the extension of the proposed methods to other models of Continuum Mechanics can be envisaged as far as the application of MG works satisfactorily.

8. Acknowledgements

We thank Adrien Loseille and Frédéric Alauzet for the fruitful discussions we had together and help in using mesh software. Thanks also for Françoise Loriot from Distene. This work has been supported by French National Research Agency (ANR) through COSINUS program (projects ECINADS n^o ANR-09-COSI-003 and MAIDESC n^o ANR-13-MONU-0010). HPC resources from GENCI-[CINES] (Grant 2010-x2010026386 and 2010-c2009025067) are also gratefully acknowledged.

References

[1] F. Alauzet and A. Loseille. High order sonic boom modeling by adaptive methods. *J. Comp. Phys.*, 229:561–593, 2010.

[2] R.E. Bank. PLTMG: A software package for solving Elliptic Partial Differential Equations. Users' Guide 11.0. Technical report, Department of Mathematics, University of California at San Diego, 2012.

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

[4] M.E. Braaten and S.D. Connell. Three-dimensional unstructured adaptive multigrid scheme for the Navier-Stokes equations. *AIAA Journal*, 34:281–290, 1996.

[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 agglomération directionnelle pour le calcul d'écoulements turbulents*. PhD thesis, Université de Nice - Sophia Antipolis, Nice, France, 1998. (in French).

[8] P.J. Frey. About surface remeshing. In *Proceedings of the 9th International Meshing Roundtable*, pages 123–136, New Orleans, LO, USA, 2000.

[9] P.J. Frey. `Yams`, a fully automatic adaptive isotropic surface remeshing procedure. Inria-rt0252, INRIA, November 2001.

[10] P.J. Frey and P.L. George. *Mesh generation. Application to finite elements*. ISTE Ltd and John Wiley & Sons, 2nd edition, 2008.

[11] W. Hackbusch. *Multigrid methods and applications*. Springer, 1985.

[12] 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).

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

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

[15] 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.

[16] A.J. Wathen M. Arioli, D. Loghin. Stopping criteria for iterations in finite element methods. *Numerische Mathematik*, 99:381–410, 2005.

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

[18] W.F. Mitchel. The hp-multigrid method applied to hp-adaptive refinement of triangular grids. *Numerical Linear Algebra with Applications*, 17:211–228, 2010.

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

[20] C. Nastase and D. J. Mavriplis. A parallel hp-multigrid solver for three-dimensional Discontinuous Galerkin discretizations of the Euler equations. *AIAA Aerospace Sciences Meeting, Reno, NV*, AIAA Paper 2007-0512, 2007.

[21] U. Rüde. Fully adaptive multigrid methods. *SIAM J. Numer. Anal.*, 30:230248, 1993.

[22] K. Stüben. Appendix A: An introduction to Algebraic Multigrid. In *Multigrid by U. Trottenberg, C.W. Osterlee and A. Schüller*, pages 413–532. Academic Press, San Diego, 2001.

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