3D Mesh adaptation by metric control for CFD

Bijan Mohammadi* — Paul-Louis George**
Frédéric Hecht* — Eric Saltel****

* *Université Montpellier II*
Math. CC51
34090 Montpellier
mohamadi@math.univ-montp2.fr

** *INRIA*
B.P.105
78153 Le Chesnay, France
{ Paul-Louis.George, Eric.Saltel }@inria.fr

*** *Université Pierre et Marie Curie*
4 place Jussieu
75005 Paris, France
Frederic.Hecht@inria.fr

ABSTRACT. *We describe our approach for mesh adaptation for CFD by metric control for 3D configurations including several tools for surface and volume h-adaptation, metric definition and flow solver.*

RÉSUMÉ. *Nous décrivons une méthode d'adaptation de maillage 3D pour la mécanique des fluides. Il s'agit d'un h-méthode qui concerne le maillage des surfaces et des volumes. Cette adaptation se fait dans un champ de métriques qui permet de spécifier les tailles et directions souhaitables pour les éléments du maillage adapté à construire.*

KEYWORDS: *CFD, Mesh generation, Mesh adaptation, anisotropic mesh adaptation, metric.*

MOTS-CLÉS : *CFD, Génération de maillage, adaptation de maillage, maillage anisotrope, métrique.*

1. Motivation

Mesh adaptation by metric control is an elegant way for mesh generation and optimization and also to get solver independent solutions [GEO 98, HEC 97, BOR 96, CAS 00, BOR 96a, HAB 00].

In the past, we have successfully used these techniques for inviscid and viscous laminar and turbulent configurations for problems in two dimensions. This paper presents our first results with this approach for configurations in three dimensions.

2. Flow solver

Our model problem corresponds to the compressible Euler equations:

$$\frac{\partial W}{\partial t} + \nabla . F(W) = 0 \text{ on } \Omega, \qquad [1]$$

with suitable boundary conditions. Here, the unknown W correspond to the conservation variables:

$$W = \begin{bmatrix} \rho \\ \rho \vec{u} \\ \rho(e + \frac{|\vec{u}|^2}{2}) \end{bmatrix} = \text{ conservation variables}, \qquad [2]$$

where $\vec{u} \in \mathbf{R^3}$. The flux vector is given by:

$$F = \begin{bmatrix} \rho \vec{u} \\ \rho \vec{u} \otimes \vec{u} + pId \\ (\rho(e + \frac{|\vec{u}|^2}{2}) + p)\vec{u} \end{bmatrix}. \qquad [3]$$

These equations are discretized by a cell centered Finite Volume scheme. The domain Ω, i.e. the flow, is discretized using node centered cells C_i for the convective part defined from our unstructured grid.

Let $\Omega_h = \cup_j T_j$ be a discretization by tetrahedra of the computational domain Ω and let $\Omega_h = \cup_i C_i$ be its partition by means of cells (see the figure below for this representation in two and three dimensions).

Thus, we can associate with each $w_h \in V_h$, where V_h is the set of the continuous affine functions on our triangulation, a w'_h piecewise constant function on the cells:

$$w'_h|C_i = \frac{1}{|C_i|} \int_{C_i} w_h .$$

Conversely, knowing w'_h piecewise constant, w_h is obtained as $w_h(S_i) = w'_h|C_i$.

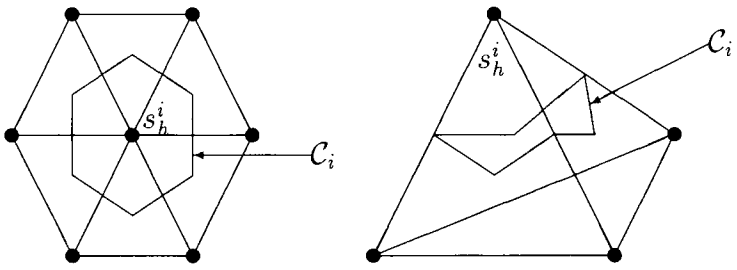

Figure 1. *Representation of a cell*

After discretization, the equations at time step n and node i, are given by :

$$\int_{C_i} \frac{W_i^{n+1} - W_i^n}{\Delta t} + \nabla F(W^n)_i = 0 \qquad [4]$$

$$\nabla F(W^n)_i = \sum_{j \in V(i)} \int_{\partial C_i \cap \partial C_j} F(W^n)_{ij} \cdot n_i + \int_{\partial C_i \cap \partial \Omega} F(W_i^n) \cdot n_i. \qquad [5]$$

Above, the convective flux $F(W^n)_{ij} \cdot n_i$ on each interface is computed by the Roe [ROE 81] flux difference splitting scheme with second order MUSCL approximation of interface variables. The boundary and initial conditions are classical. In particular, a Stegger-Warming [STE 83] flux splitting scheme is used for in and outflow boundaries.

The resulting algebraic system can be solved by an expiicit Runge-Kutta scheme, as follows:

Loop on n until steady state, then loop on Runge-Kutta sub-steps p:

$$\int_{C_i} \frac{W_i^{n+1,p} - W_i^n}{\alpha_p \Delta t_i} + \nabla F(W^{n,p-1})_i = 0. \qquad [6]$$

This algorithm is implemented as a succession of different loops, which are called at each time step n:

1. Loop on Elements:

– gather nodal values of W,

– compute the gradients ∇W (for MUSCL reconstruction),

– scatter (send and add) results to nodal residuals and gradients.

2. Loop on Edges:

– gather nodal values of W and ∇W,

– compute convective fluxes $\int_{\partial C_i \cap \partial C_j} F(W^n)_{ij} \cdot n_i$,

– scatter results to nodal residuals.

3. Loop on Nodes: update nodal values as predicted by algebraic solver

$$W_i^{n+1,p} = W_i^n - \alpha_p \Delta t_i \nabla F(W^{n,p-1})_i. \qquad [7]$$

3. Metric definition

The key idea is to modify the scalar product used in the mesh generator for distance and volume evaluations. Therefore, the aim, using an automatic mesh generation method (in our case, a Delaunay based approach), is to construct equilateral triangles (in two dimensions), resp. regular tets (in three dimensions) according to a new adequate metric and not as in a classical case in the Euclidean one. The scalar product is then based on the evaluation of the Hessian of the variables of the problem. Indeed, for a P^1 Lagrange discretization of a variable u, the interpolation error is bounded by:

$$\mathcal{E} = |u - \Pi_h u|_0 \leq c h^2 |D^2 u|_0, \qquad [8]$$

where $\Pi_h u$ is the P^1 interpolate of u, h is the element size, $D^2 u$ is the Hessian matrix:

$$D^2 u = \begin{pmatrix} \partial^2 u/\partial x^2 & \partial^2 u/\partial y \partial x & \partial^2 u/\partial z \partial x \\ \partial^2 u/\partial x \partial y & \partial^2 u/\partial y^2 & \partial^2 u/\partial z \partial y \\ \partial^2 u/\partial x \partial z & \partial^2 u/\partial y \partial z & \partial^2 u/\partial z^2 \end{pmatrix}$$

$$= \mathcal{R} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \mathcal{R}^{-1},$$

where R is the eigenvectors matrix of $D^2 u$ and λ_i its eigenvalues which are always real values. Using this information, we introduce the following metric tensor $\mathcal{M}$:

$$\mathcal{M} = \mathcal{R} \begin{pmatrix} \tilde{\lambda}_1 & 0 & 0 \\ 0 & \tilde{\lambda}_2 & 0 \\ 0 & 0 & \tilde{\lambda}_3 \end{pmatrix} \mathcal{R}^{-1}, \qquad [9]$$

where:

$$\tilde{\lambda}_i = min(max(|\lambda_i|, \frac{c\mathcal{E}}{h_{max}^2}), \frac{c\mathcal{E}}{h_{min}^2}),$$

with h_{min} and h_{max} being the minimal and maximal edge lengths allowed in the mesh.

Now, if we generate, using a Delaunay procedure as a mesh generation method a mesh with edges close to unit length in the metric $\mathcal{M}/(c\mathcal{E})$, the interpolation error $\mathcal{E}$ is equi-distributed over the edges a_i of the mesh. More precisely, we have

$$\frac{1}{c\mathcal{E}} a_i^T \mathcal{M} a_i = 1. \qquad [10]$$

Nevertheless, the previous definition is not sufficient for a suitable metric in the cases where systems, boundary layers, multi-scales phenomena are to be considered as will be discued below.

3.1. *Systems*

For systems, the previous approach leads to a metric for each variable and we take the intersection of all these metrics. More precisely, for two metrics, we find an approximation of their intersection by the following procedure:

Let λ_i^j and v_i^j, $i, j = 1, 2$ the eigen-values and eigen-vectors of $\mathcal{M}_j$, $j = 1, 2$. The intersection metric $(\hat{\mathcal{M}})$ is defined by

$$\hat{\mathcal{M}} = \frac{\hat{\mathcal{M}}_1 + \hat{\mathcal{M}}_2}{2}. \qquad [11]$$

where $\hat{\mathcal{M}}_1$ (resp. $\hat{\mathcal{M}}_2$) has the same eigen-vectors than $\mathcal{M}_1$, (resp. $\mathcal{M}_2$) but with eigen-values defined by:

$$\tilde{\lambda_i^1} = \max(\lambda_i^1, {v_i^1}^T \mathcal{M}_2 v_i^1), \quad i = 1, 2. \qquad [12]$$

The previous algorithm is easy to extend to the case of several variables. Here, one difficulty comes from the fact that we work with variables with different physical meaning and scale (for instance pressure, density and velocity). We will see that a relative rather than a global error estimation permits to avoid this problem.

3.2. *Boundary layers*

The evaluation of the Hessian is done by the Green formula with Neumann boundary condition. However, this does not lead to a suitable mesh for boundary layers. Indeed, the distance of the first layer of nodes to the wall will be quite irregular. Another important ingredient therefore, is a mixed Dirichlet-Neumann boundary condition for the different components of the metric on wall nodes for viscous computations. More precisely, the eigenvectors for these nodes are the normal and tangent unit vectors and the eigenvalue corresponding to the normal eigenvector is a prescribed value depending on the Reynolds number. The tangential eigenvalue comes from the metric of the solution.

More precisely, along the wall the previous metric $\mathcal{M}(x)$ is replaced by a new metric $\hat{\mathcal{M}}(x)$:

$$\hat{\mathcal{M}}(x) = T \Lambda T^{-1},$$

where

$$\Lambda = diag(\frac{1}{h_n^2}, \lambda_\tau) \quad \text{and} \quad T = (\vec{n}(x), \vec{\tau}(x)).$$

The refinement along the wall can now be monitored through h_n. This allows for instance for shocks and boundary layers to interact. This metric is propagated through the flow by a smoothing operator. This aspect is not used for inviscid solutions.

3.3. *Multi-scale phenomena*

We noticed that when we have several eddies with variable energy, it is difficult to capture the weaker ones, especially if there are shocks involved [HEC 97]. In other words, [8] leads to a global error when we would like to have a relative one. The following estimation takes into account not only the dimension of the variables but also their magnitude:

$$\tilde{\mathcal{E}} = |\frac{u - \Pi_h u}{\max(|\Pi_h u|, \epsilon)}|_0 \leq ch^2 |\frac{D^2 u}{\max(|\Pi_h u|, \epsilon)}|_0, \qquad [13]$$

where we have introduced the local value of the variable in the norm. ϵ is a cut-off to avoid numerical difficulties and also to define the difference between the orders of magnitude of the smallest and largest scales we try to capture. Indeed, when a phenomena falls bellow ϵ, it will not be captured. This is similar to looking for a more precise estimation in regions where the variable is small. Another important consequence of this estimation is that it removes the dimensional problems when intersecting metrics come from different quantities.

3.4. *Implementation of the metric evaluation*

We use a weak formulation for the definition of the Hessian because we use P^1 discretization for the variables. More precisely, we use the following approximation:

$$\frac{\partial^2 \pi_h u}{\partial x_i \partial x_j}(S_k) = \frac{-\int_\Omega \frac{\partial \pi_h u}{\partial x_j} \frac{\partial \varphi}{\partial x_j}}{\int_\Omega \varphi} \qquad [14]$$

with φ the finite element shape function with value 1 at node S_k and 0 everywhere else. We have to specify the physical field u used in such a construction. As specified above, we use all the conservation variables and take the intersection between the corresponding ellipsoids.

Hessian definition on the boundaries

The previous formula does not lead to a suitable metric definition along boundaries because the gradients are not correctly evaluated. This is also true in 2D. We use the following smoothing operator to enforce the Hessian along the boundaries:

$$-\Delta \tilde{H} = H \quad \text{on} \quad \Omega, \quad \tilde{H}(\Omega/\partial\Omega) = H. \qquad [15]$$

This means that we solve this elliptic system with Neumann boundary condition everywhere, but keep the internal values unchanged. In this way, the boundary values are adapted to the internal solution to enforce the Neumann boundary condition.

This smoothing is enough for 2D applications, but in 3D, we might have tetrahedra with four nodes on the boundary. In this case, one of the nodes might never see the other nodes of the mesh. This often happens in corners. This of course is to be avoided as far as possible, even if not penalizing with slipping or Neumann boundary condition for the PDE, with Dirichlet boundary conditions this tetrahedra is blind. To correct these values, we tagg such nodes and use another elliptic smoother, this time only on the boundary.

$$-\Delta \overline{H} = \tilde{H} \quad \text{on} \quad \partial\Omega, \quad \overline{H} = \tilde{H} \quad \text{if not tagged.}$$

In this way the blind nodes solution is adapted to the solution around and the other boundary values remain unchanged.

This is particularly important for nodes where the normal prescription rule above is not used. For this former case, this correction is however applied to tangential values.

4. Anisotropic mesh generation

Our aim is to generate a mesh that conforms to a metric coming from the geometry of the domain surface and from information related to the solution of the physical problem. The mesh construction process consists of two separate steps. First of all, a surface mesh is completed based on the above metric specifications. Then, using this surface mesh as an entry, a domain (say a tridimensional) mesh is constructed with the same metric controls.

4.1. *Anisotropic surface meshing*

The geometry of the surface is provided by means of a triangulation enriched with some geometrical informations (ridges, singular points, node normals, ...). This mesh along with these values are used to define the geometry come from the CAD definition of the initial surfaces.

The need for these informations is also to avoid any call to a CAD system during the adaptation process.

Our surface adaptive mesh generation includes the following steps:

– The first step in surface adaptation is to build a geometrical metric integrating the curvatures and local main directions for all the nodes of the triangulation.

– Using this metric, we define a reference geometry called 'the geometry'. As we said, we need this entity as we want to avoid any call-back to CAD.

– Intersect the metric coming from the solution of the physical system with this metric following the procedure described above.

– Generate an equilateral triangulation in this metric with all the edges of length one in this metric. This step is based on four classical ingredients, i.e.:

- Adding nodes over the surface,
- Removing points from the surface,
- Optimization by mesh deformation with respect to the geometry,
- Optimization by edge swaps.

In all these steps, the reference geometry is the one defined in Step 2. To this end, we need the link between the nodes over the new surface and the reference mesh. This implies that we keep this link during mesh adaptation.

In addition, we need an extra link between the new nodes and the triangles of the previous mesh to make the interpolation of the previous solution defined over the former mesh over the new surface mesh possible. In other words, only a link in the volume level is not enough for an accurate interpolation.

4.2. *Volume meshing*

The method used in this work is a Delaunay based mesh generation method. Following [GEO 98, GHS 90, FRE 99], such a method includes several steps. An incremental Delaunay point insertion procedure allows us to:

Insert a point in a given mesh. This procedure is a variation of the classical Delaunay method. Actually, it extended the latter to the case where a metric is specified to which the mesh must conform to. Then, the well-known steps of the mesh generation method concern the boundary enforcement, the way in which the field points are defined (they will be inserted by the above point insertion procedure) and, finally, some extent of optimization.

Specific to our mesh generation context are the way in which the points are created and the optimization procedures used in the final step.

At the time the above surface mesh is ready, the point insertion method enables us to complete a mesh of the domain including the vertices of this surface mesh. Then, a boundary enforcement method completes a mesh of the domain including the triangles of this mesh as element faces.

Now, a mesh of the domain is available whose sole vertices are the surface vertices. The point is to create an appropriate set of field points. The key-idea is as before to obtain, in some, a mesh with unit length edges. To this end, the edges of the current mesh serve as support for the point creation. Based on the length of a given edge, zero, one or several points are constructed in such a way as the distance between two neighboring points is close to the unit value. Once all the edges have been examined,

we have a set of points. The latter are inserted and the process is iterated as long as some edges of the current mesh are greater than one.

The field points having been constructed and inserted, an optimization phase enables us to obtain the final mesh. The tools used in this task are basically those of any classical optimization procedure. In this respect, node relocations and edge swaps are mostly used to optimize a quality function based on both the aspect ratio of the elements and their sizes.

At present, while following the above description, we have restricted ourselves to an isotropic context (since the corresponding anisotropic mesh generation method is not fully available at this time). To make sense, the anisotropic metric provided initially has been converted into an isotropic metric in a very simple way. Despite this simplification, the resulting meshes, used in different numerical simulations, proved to be reasonable. Obviously, at the time the anisotropic method will be available, it will be of interest to examine the benefits of such an approach as we would like to do in the near future.

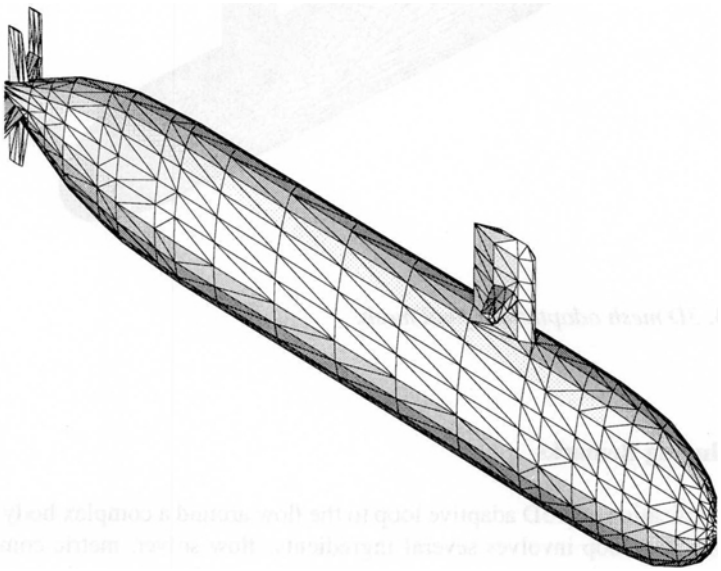

Figure 2. *3D mesh adaptation: initial and background geometry mesh.*

5. Numerical experiences

We show results for the applications of our 3D adaptive loop to the computation of the flow around a complex geometry. The flow is weakly compressible. The flow is considered as being inviscid. The initial mesh has about 12,000 elements and the

final mesh about 640,000 tetrahedra. We can see that the metric intersection between geometry and solution is efficient and that the surface adaptation without CAD interface is effective. One drawback for the moment is the time spent in this former point. Current effort therefore is focused on improveming this point. The next step is to introduce adptivity in the volume level. For the moment only the surface mesh has been adapted.

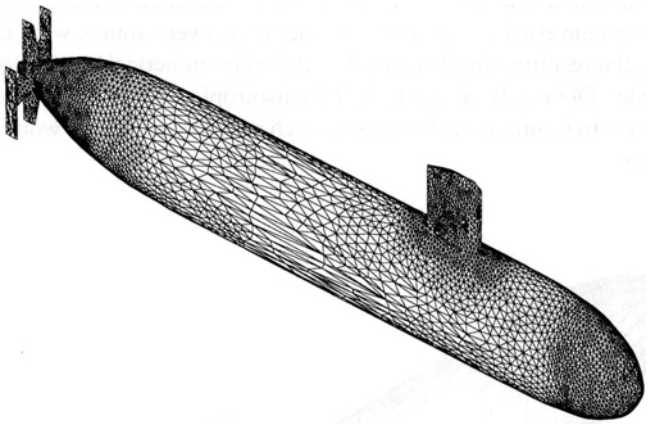

Figure 3. *3D mesh adaptation: final mesh.*

6. Concluding Remarks

The application of a 3D adaptive loop to the flow around a complex body has been presented. This loop involves several ingredients: flow solver, metric computation, metric intersection between geometry and solution, surface adaptation and surface mesher, volume mesher. The volume mesh generation is an isotropic Delaunay mesh generator. Several numerical difficulties still exist for an efficient use of this approach for general geometry. In particular, the surface mesh adaptation is still too slow. Current work is focused on improving these two points.

7. References

[GEO 98] P.L. GEORGE, H. BOROUCHAKI, *Delaunay triangulation and meshing*, Hermès, Paris, 1998.

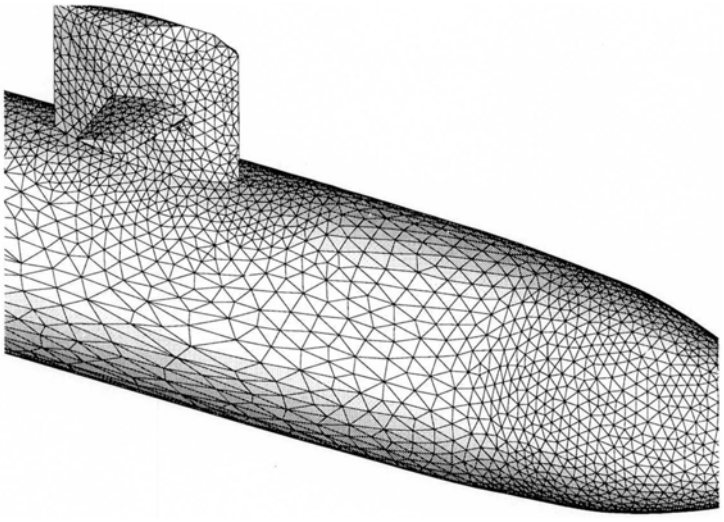

Figure 4. *3D mesh adaptation: zoom of the final mesh.*

[HEC 97] F. HECHT, B. MOHAMMADI, « Mesh Adaption by Metric Control for Multi-scale Phenomena and Turbulence », *AIAA*, paper 97-0859, 1997.

[BOR 96] H. BOROUCHAKI, M.J. CASTRO-DIAZ, P.L. GEORGE, F. HECHT AND B. MOHAMMADI, « Anisotropic adaptive mesh generation in two dimensions for CFD », *5th Inter. Conf. on Numerical Grid Generation in Computational Field Simulations*, Mississipi State Univ., 1996.

[CAS 00] M. CASTRO-DIAZ, F. HECHT, B. MOHAMMADI, « Anisotropic Grid Adaption for Inviscid and Viscous Flows Simulations », submitted to IJNMF.

[BOR 96a] H. BOROUCHAKI, P.L.GEORGE, B. MOHAMMADI, « Delaunay Mesh Generation Governed by Metric Specifications. Part II: Applications », *Finite Element in Analysis and Design*, special issue on mesh adaption, 1996.

[HAB 00] W. HABASHT ET AL., « A Step toward mesh-independent and User-Independent CFD », *Barriers and Challenges in CFD, Kluwer Ac. pub.* pp. 99-117,

[ROE 81] P.L. ROE, « Approximate Riemann Solvers, Parameters Vectors and Difference Schemes », *J.C.P.* Vol.43, 1981.

[STE 83] J. STEGER, R.F. WARMING, « Flux Vector Splitting for the Inviscid gas dynamic with Applications to Finite-Difference Methods », *J.C.P.* 40, pp:263-293, 1983.

[GHS 90] P.L. GEORGE, F. HECHT, E. SALTEL, « Fully automatic mesh generator for 3d domains of any shape », *Impact of Comp. in Sci. and Eng.*, 2, pp.187-218, 1990.

[FRE 99] P.J. FREY, P.L. GEORGE, *Maillages*, Hermès, Paris, 1999.