

RESIDUAL SCHEMES FOR PENALIZED NAVIER-STOKES EQUATIONS ON ADAPTED GRIDS

L. Nouveau³, R. Abgrall¹, H. Alcin³, H. Beaugendre², C. Dobrzynski²

¹ Universität Zürich, Institut für Mathematik, CH-8057 Zürich, Switzerland

² Univ. Bordeaux, IMB, UMR 5251, Inria Bordeaux Sud-Ouest, 33405 Talence, France

³ Inria Bordeaux Sud-Ouest, 33405 Talence, France

email^{1,2,3} : name.surname@inria.fr

Key words: Residual Schemes, Immersed Boundary Method, Penalization, Unstructured grids, Mesh Adaptation

Abstract. Immersed boundary methods such as penalization present advantages in computational fluid dynamics. They simplify mesh generation and are widely used for moving bodies. However, an issue remains : the treatment of wall boundary conditions. Therefore mesh adaptation can be performed to improve the accuracy of wall treatments. In this work, we propose to solve the penalized Navier-Stokes equations with a residual distribution scheme combined with mesh adaptation. In addition, different ways of tracking an interface are presented to investigate unsteady problems dealing with moving bodies.

1 INTRODUCTION

When dealing with Computational Fluid Dynamics simulations, two kinds of grids are usually used : body-fitted grids and embedded ones. For body fitted grids, solid wall boundaries are meshed and boundary conditions (BCs) are applied on the wall according to additional equations (Neumann or Dirichlet BC). On the contrary, immersed boundary methods (IBM) are characterised by a mesh covering the entire domain, and BCs are taken into account in a different way. The penalization technique [1] is an IBM in which the BCs are handled inside the equations by adding source terms. In this method, the solid body is located on the mesh by the 0 isovalue of the signed distance function. It is then necessary to define as good as possible this isovalue. Mesh adaptation is then introduced [4, 5] to catch precisely this isovalue and to accurately compute the penalized source terms. We propose here to solve

the penalized compressible Navier-Stokes equations on adapted grids using a residual distribution scheme. Those schemes [6, 7, 8, 9] are chosen for their easy order increase and parallelization.

Penalization combined with mesh adaptation provides a good method to deal with moving bodies. Indeed, for a given (or computed) movement of the solid body, if the new position and a mesh can be provided, the solution at the new time step can be computed. As the position is defined by the level set, it is necessary to follow its displacement. In [5], it is proposed a way of following the level-set by advection and to get dynamic mesh adaptation. We propose in this work to follow the process proposed in [5] and to track surface interface with mesh adaptation.

2 PENALIZATION

2.1 New Equations

Penalization is an IBM in which the solids are covered with the mesh and the Navier-Stokes (NS) equations are modified so as to take into account the boundary conditions with a penalized term acting like a source term [1]:

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \frac{1}{\eta} \sum_{i=1}^{N_s} \chi_{S^i} (\rho \mathbf{u} - \rho \mathbf{u}_{S^i}) = -\nabla p + \nabla \underline{\boldsymbol{\pi}} \\ \frac{\partial (\rho e)}{\partial t} + \nabla \cdot ((\rho e + p) \mathbf{u}) + \frac{1}{\eta} \sum_{i=1}^{N_s} \theta_{S^i} \chi_{S^i} (\rho e - \rho e_{S^i}) \\ \quad + \frac{1}{\eta} \sum_{i=1}^{N_s} \chi_{S^i} (\rho \mathbf{u} - \rho \mathbf{u}_{S^i}) \cdot \mathbf{u} = \nabla \cdot (\underline{\boldsymbol{\pi}} \mathbf{u} + \mathbf{q}) \end{array} \right. \quad (1)$$

where ρ is the density, \mathbf{u} is the velocity, p the pressure, $\underline{\boldsymbol{\pi}}$ is the stress tensor and \mathbf{q} the heat flux. N_s corresponds to the number of solids considered inside the domain, χ_{S^i} is the characteristic function of the solid S^i . θ_{S^i} lets the possibility to penalize the energy (Dirichlet BC) or not (Neumann BC), and η is a parameter chosen small enough ($\frac{1}{\eta} \gg 1$).

In these modified equations, the red terms are the terms which include directly the BCs in the equations. Indeed, outside the solids, the χ_{S^i} functions are equal to 0 and so the penalization term vanishes and the usual Navier-stokes equations are found back. On the opposite, inside a solid, the characteristic function is equal to one, and as $\frac{1}{\eta}$ is chosen big enough, the red terms dominate and the boundary values are

imposed. The accuracy of the method depends on the value of η , in all simulations, we use an implicit scheme with $\eta = 1 \times 10^{-12}$.

This system can be formulated in a matrix way :

$$\begin{aligned} \partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) + \mathbf{S} &= \nabla \cdot \mathbf{G} \\ \mathbf{U} &= \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho e \end{pmatrix} \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{Id} \\ (\rho e + p) \mathbf{u} \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} 0 \\ \boldsymbol{\pi} \\ \boldsymbol{\pi} \mathbf{u} + \mathbf{q} \end{pmatrix} \\ \mathbf{S} &= \frac{1}{\eta} \sum_{i=1}^{N_s} \chi_{S^i} \begin{pmatrix} 0 \\ \rho(\mathbf{u} - \mathbf{u}_{S^i}) \\ \theta_{S^i} \rho(e - e_{S^i}) + \rho(\mathbf{u} - \mathbf{u}_{S^i}) \mathbf{u} \end{pmatrix} \end{aligned} \quad (2)$$

where \mathbf{U} are the conservative variables, \mathbf{F} and \mathbf{G} are the advection and viscous flux and \mathbf{S} the source term. In addition of those equations, the variables are linked by state law which is in our case the perfect gaz law.

2.1.1 Signed Distance Function

To discretize the system of equations (1), χ_{S^i} the characteristic function of the solid S^i is required. From a practical point of view, the *signed distance function* (SDF) is used. It is a way to locate interfaces or surfaces on a mesh. The SDF is equal to the distance of the considered point to the solid surface with a sign in order to determine if the point is inside or outside the surface. For an inside point, the sign is negative and for an outside one, the sign is positive.

A way of computing the SDF (which is fully studied by Frey and Dapogny in [2]) is to solve the *unsteady Eikonal equation*. The inside of the surface to model is defined by Ω :

$$\Omega = \{x \in \mathbb{R}^d; u_0(x) < 0\} \text{ and } \partial\Omega = \{x \in \mathbb{R}^d; u_0(x) = 0\} \quad (3)$$

with d the dimension of space and u_0 a continuous function.

Then, the SDF can be considered as the solution of the following *unsteady Eikonal equation* :

$$\begin{cases} \partial_t u + \text{sgn}(u_0) (|\nabla u| - 1) = 0 \quad \forall t > 0, x \in \mathbb{R}^d \\ u(t = 0, x) = u_0(x), \quad \forall x \in \mathbb{R}^d \end{cases} \quad (4)$$

With the characteristic method, the following approached solution is obtained :

$$u \approx \begin{cases} u_0 \left(x - dt \frac{\nabla u_0}{\|\nabla u_0\|} \right) + dt, \text{ for } x \in \overline{c\Omega} \\ u_0 \left(x + dt \frac{\nabla u_0}{\|\nabla u_0\|} \right) - dt, \text{ for } x \in \Omega \end{cases} \quad (5)$$

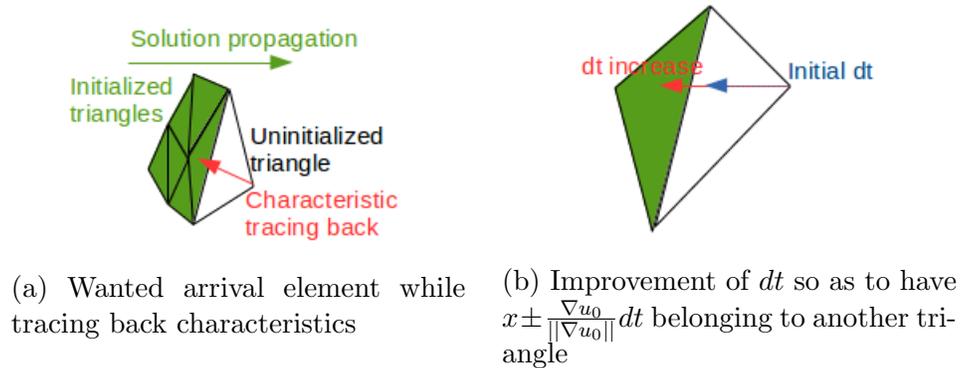


Figure 1: Adaptive time step

In order to be more accurate, the time step dt has been chosen adaptive. Indeed, as the solution is "propagated" over the mesh, the idea is to choose a time step for each triangle which allows to trace back the characteristic "just below" which is already initialized (see figure 1a). The initialization of dt for each triangle is the minimum height. Then, as long as $x \pm \frac{\nabla u_0}{\|\nabla u_0\|} dt$ still belongs to the considered element, dt is increased a little bit (see figure 1b). The algorithm proposed in [2] has been modified as shown in algorithm (1).

3 MESH ADAPTATION

As explained in the previous section, the solid is located on the mesh thanks to the SDF. In order to define as precisely as possible the solid, mesh adaptation with respect to the 0 level set function can be done. Indeed, it allows to refine the mesh close to the solid boundaries. Thus, the penalization is imposed in a very rigorous way. In addition, anisotropic meshes are used. Those meshes are defined by the fact that elements (triangles or tetrahedra) can be stretched a lot which allow to insert less elements in refined areas.

The aim in mesh adaptation is to find a good error estimator of a solution on a mesh. In [4, 5], it is shown that metrics can be used and considered as good error estimator. Those metrics are positive definite symmetric matrices (PDS) defined for each nodes of the mesh. The matrix contains the size and direction of the edges. Indeed, PDS means diagonalizable, and the eigenvalues λ_i are linked to the sizes h_i of the elements in the direction i ($\lambda_i = 1/h_i^2$), those directions being given by the eigenvectors. So as to adapt with respect to the 0 level set, it is shown in [2] that the following metric can control the approximation of the SDF ϕ with an error ϵ :

Algorithm 1 Modified algorithm with adaptative time step

```

for  $n = 1$  untill convergence do
   $u^n(x) = u^{n-1}(x)$  for each node  $x \in \mathcal{T}$ 
  for each simplex  $T$  of  $\mathcal{T}$  do
    for each node  $x \in T$  not initialized do
      if  $x \notin \Omega$  then
         $dt =$  minimum height of  $T$ 
         $drec = \frac{\nabla u^{n-1}}{\|\nabla u^{n-1}\|} dt$ 
         $T_2 = T$ 
        while  $T_2 == T$  do
          if  $x \notin \Omega$  then
             $x_2 = x - drec$ 
          else
             $x_2 = x + drec$ 
          end if
          Find  $T_2$  such that  $x_2 \in T_2$ 
          if  $T_2 == T$  then
             $drec * = 1.1$ 
          end if
        end while
        if  $x \notin \Omega$  then
           $u^n(x) = \min(u^n(x), u^{n-1}(x - drec)) + dt$ 
        else
           $u^n(x) = \min(u^n(x), u^{n-1}(x + drec)) - dt$ 
        end if
      end if
    end for
  end for
end for

```



Figure 2: Area of fine mesh insertion.

$$\mathcal{M} = {}^t\mathcal{R} \begin{pmatrix} \frac{1}{\epsilon^2} & 0 & 0 \\ 0 & \frac{|\lambda_1|}{\epsilon} & 0 \\ 0 & 0 & \frac{|\lambda_2|}{\epsilon} \end{pmatrix} \mathcal{R} \quad (6)$$

with $\mathcal{R} = (\nabla\phi \ v_1 \ v_2)$, where (v_1, v_2) is a base of the tangential plane of the surface defined by the isovalues of ϕ , and the λ_i are the eigen values of the hessian of ϕ . This metric is imposed in a vicinity w of the surface (see figure 2).

Another important and useful way of using mesh adaptation is in areas of large physical variations. Indeed, it allows to refine the mesh in order to catch better some phenomena. Thus, without increasing that much the number of nodes and elements, the solution can be considerably improved.

For this kind of adaptation, the aim is to control the interpolation error between the exact solution u and its interpolant $\Pi_h u$ on the mesh. It has been proved ([4]) that a majoration of this error on an element K is given by :

$$\|u - \Pi_h u\|_{\infty, K} \leq c_d \max_{\mathbf{e} \in K} \langle \mathbf{e}, \mathcal{M}(K) \mathbf{e} \rangle \quad (7)$$

where \mathbf{e} denote the edges of the mesh, c_d is a constant depending on the dimension and with the metric $\mathcal{M}(K)$ computed with the hessian \mathcal{H}_u of u :

$$\mathcal{M} = {}^t \mathcal{R} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \mathcal{R} \quad (8)$$

where \mathcal{R} is the matrix of the eigen vectors of the hessian , and the λ_i are defined by :

$$\lambda_i = \min \left(\max \left(|h_i|, \frac{1}{h_{max}^2} \right), \frac{1}{h_{min}^2} \right)$$

with h_{min} (h_{max}) is the minimum (maximum) size wanted for the mesh and h_i the eigenvalues of \mathcal{H}_u . Thus, in area of large physical variations, fine meshes are inserted and where the solution is quasi constant, large meshes are placed.

4 RESIDUAL DISTRIBUTION SCHEMES

4.1 Residual distribution scheme construction

The scheme which is used to solve the equations introduced in section 2.1 is a residual distribution scheme (RDS). Here is briefly explained how such schemes



(a) Distribution for a P1 triangle (b) Gathering triangle contribution

Figure 3: Principle of residual distribution scheme

are built. More details can be found in [6]. Considering a 2D steady conservation equation :

$$\nabla \cdot \mathcal{F}(u) = 0, \quad \forall (x, y) \in \Omega \quad (9)$$

A tessellation \mathcal{T}_h of Ω is considered. Elements are noted T (boundary ∂T). Let $u^h = \sum_i u_i \varphi_i$, $\varphi_i(x, y)$ being the basis functions chosen on the triangle T . Equation (9) integrated over Ω can be written as :

$$\int_{\Omega} \nabla \cdot \mathcal{F}(u^h) = \sum_{T \in \mathcal{T}_h} \int_T \nabla \cdot \mathcal{F}(u^h) d\mathbf{x} = 0 \quad (10)$$

What is called the *total residual* Φ^T can then be defined for each triangle :

$$\Phi^T = \int_T \nabla \cdot \mathcal{F}(u^h) d\mathbf{x} = \int_{\partial T} \mathcal{F}(u^h) \cdot \mathbf{n}_{ext} \quad (11)$$

where \mathbf{n}_{ext} is the outward normal of the edges. The principle of a RDS is to distribute this residual to each degree of freedom (DoF) of the triangle (see figure 3a). Each DoF receives what is called a *nodal residual* Φ_i^T using a *coefficient distribution* β_i^T :

$$\Phi_i^T = \beta_i^T \Phi^T \quad (12)$$

The distribution coefficients are defined according to the scheme which is used (Lax-Wendroff, Lax-Friedrichs, etc ...). Then, all the contributions of each triangle where i belongs are added (see figure 3b), and the RDS is formulated :

$$\sum_{T \ni i} \Phi_i^T = 0 \quad (13)$$

The solution of (13) can be obtained by solving the pseudo iterative scheme [7] :

$$\begin{cases} \frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{|C_i|} \sum_{T \ni i} \Phi_i^T = 0 \\ u_i^0 \text{ given} \end{cases} \quad (14)$$

or more sophisticated iterative methods. Extensions to the NS equations are defined in [9] and leads to (13), with residual taking into account the viscous terms. Such an example is described in the next paragraph.

4.2 A variant of the SUPG scheme

As said before, the RDS depends on the coefficient distribution chosen. For the computations, we use a variant of the SUPG scheme [9]. The total residual is equally distributed to the DoFs and a stabilization term is added. For P1 triangles, it gives

$$\Phi_i^T = \frac{\Phi^T}{3} + \sum_{T \ni i} \int_T \mathbf{A} \cdot \nabla \varphi_i \tau^T (\mathbf{A} \cdot \nabla u_h - \nabla \cdot (\mathbf{K}(u_h) \nabla u_h)) \quad (15)$$

The stabilization is defined such that both advection and diffusion are considered :

$$\tau^T = \frac{|T|}{3} \left(\sum_{i \in T} (|\mathbf{A} \cdot \nabla \varphi_i| + \mathbf{K} \nabla \varphi) \right)^{-1}$$

Looking at the equation (2), \mathbf{A} is the jacobian matrix of the advection : $\mathbf{A} = \nabla_u \mathbf{F}$ and \mathbf{K} is such that : $\mathbf{G} = \mathbf{K} \nabla U$.

5 RESULTS

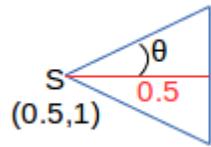
The following result were obtained using the flow solver called **RealFluid**. For the 3D mesh adaptation, the meshes were generated by **MMG3d** [3].

5.1 Supersonic flow around a triangle

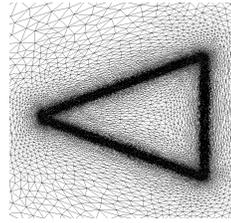
This first test case is the study of a supersonic flow around a triangle. The triangle (heigh $h = 0.5$, half angle $\theta = 20$ deg, see figure 4a) is in a circle of radius 20 . The aim is to capture the shock which is created. The initial mesh is only adapted to the 0 level set (see figure 4b). The penalized variables are the velocity ($\mathbf{u} = 0$) and the temperature ($T = 3$). The Reynolds number is fixed to $Re = 50000$ and the Mach number is chosen equal to $Ma = 2.3664319$ so as to get a shock in contact with the triangle peak. The u-velocity solution on the initial mesh is given in figure (4c). For the adaptations, the following parameters have been chosen :

$$\begin{cases} \epsilon = h_{min} = 5.10^{-4} \\ h_{max} = 2.0 \end{cases}$$

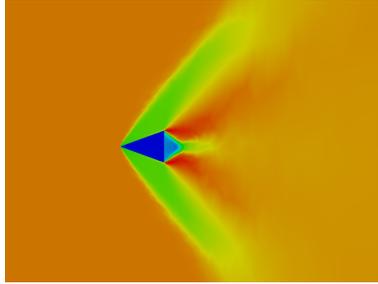
The mesh obtained after 6 cycles of adaptation and the computed u-velocity on this mesh are presented in figure (4d). A way to validate this test case is to compute the angle between the shock and the axis $x = 0$ because this angle can be analytically computed. As in [1], the angle is measured using a point located on the shock near $y = 0$ and we found $\beta = 53.33$ deg for a analytic one of $\beta \approx 53.46$ deg.



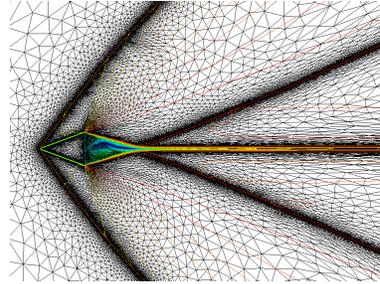
(a) Studied triangle



(b) Initial mesh

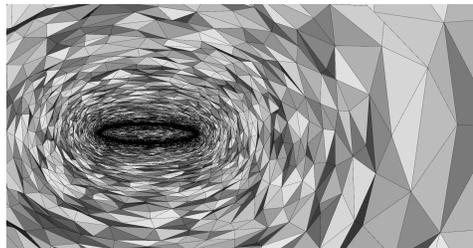


(c) u-velocity computed on the initial mesh

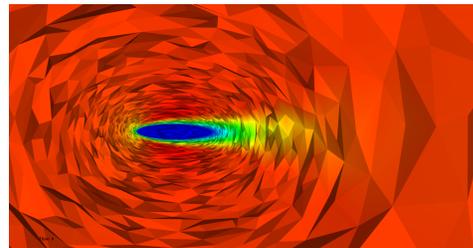


(d) Adapted mesh with u-velocity isoline

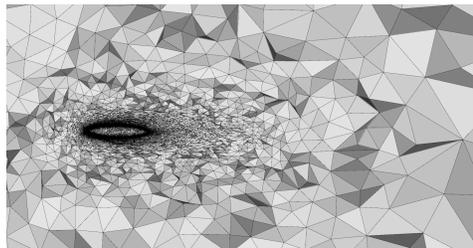
Figure 4: Triangle test case - presentation and results



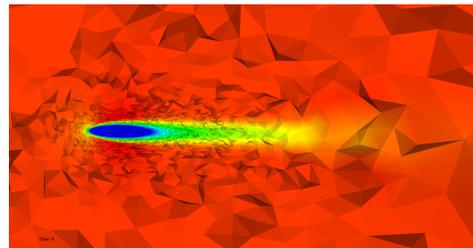
(a) Initial mesh



(b) u-velocity on the initial mesh



(c) Adapted mesh



(d) u-velocity on the adapted mesh

Figure 5: Ellipse test case - meshes and solutions

5.2 Flow around an ellipse

In this 3D test case taken from [1], the flow around an ellipse (sizes : (0.5, 0.1, 0.2)) centered in a sphere (radius $r = 10$) is studied. The parameters are set to $Re = 500$ and $Ma = 0.375$. As for the previous test case, the initial mesh is only adapted to the 0 level set (see figure (5a)). The u-velocity obtained on this mesh is presented in figure (5b). Two cycles of adaptation have been done with the following parameters (for both level set and physical adaptation) :

$$\begin{cases} \epsilon = 0.001 \\ h_{min} = 0.003 \\ h_{max} = 2.0 \end{cases} \quad (16)$$

The adapted mesh is presented in figure (5c) and its solution (u-velocity) in figure (5d).

6 INTERFACE FOLLOW

The aim will be to deal with unsteady problem with moving bodies. As the body is defined by the SDF, the purpose is to track the movement of the 0 level set. With a body moving at speed $\mathbf{V}(t, \mathbf{x})$, the SDF is governed by the advection equation :

$$\begin{cases} \partial_t \phi(t, \mathbf{x}) + \mathbf{V}(t, \mathbf{x}) \cdot \nabla \phi(t, \mathbf{x}) = 0 \\ \phi(t = 0, \mathbf{x}) = \phi_0(\mathbf{x}) \end{cases} \quad (17)$$

where $\phi_0(\mathbf{x})$ corresponds to the initial position of the object.

Here again the goal is to adapt the mesh to the 0 level set, and this time at each time step. As studied in [5], a characteristic method can be used. The nodes at time t^{n+1} are considered and their "original position" at time t^n are searched by tracing back the characteristic. The mesh is then adapted to this new SDF. As we want to deal with physical displacements, time step are very small and so are the displacements. Instead of using an iterative process as presented in [5], due to the small movement, the new 0 level set is defined in an already adapted area. Indeed as explained in section 3, when adaptation to 0 level set is done, the vicinity w of the 0 level set is also refined. Then, just after the advection, adaptation can be done.

A well known problem when dealing with level set advection is the mass loss. Indeed, each advection leads to a small deformation of the surface. Therefore, another way of thinking is, instead of advecting the whole signed distance function, only the surface defining the solid is advected, and the SDF is computed using method presented section 2.1.1. Thus, no mass loss is introduced. The problem is that computation

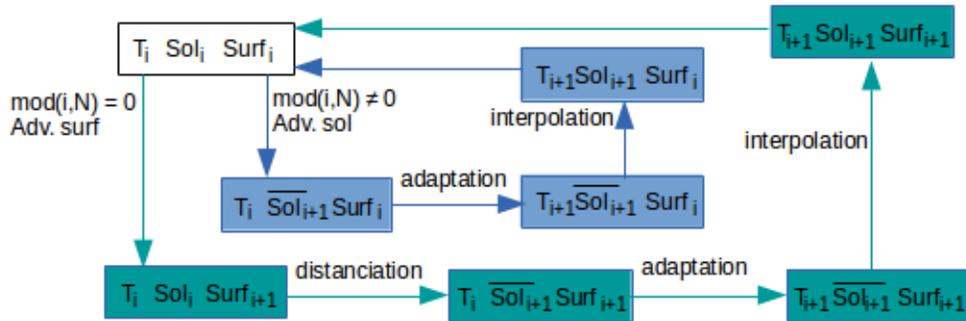
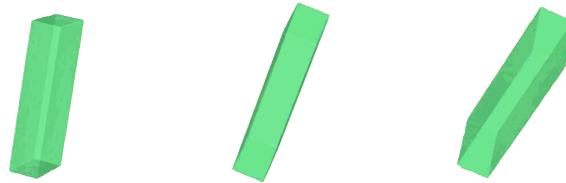
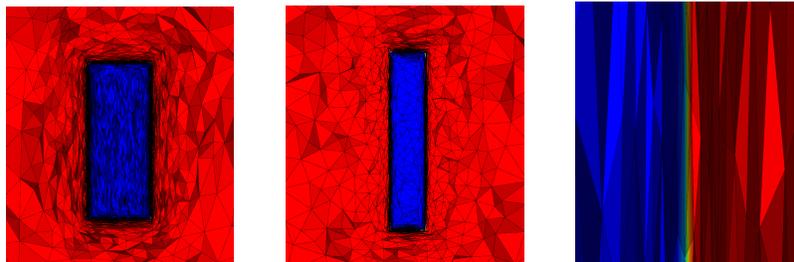


Figure 6: Adaptation process, T_i , Sol_i and $Surf_i$ denote respectively mesh, solution and surface at time i .

time for large meshes are considerably higher than with advection. Consequently, our idea is to find a way of coupling advection and SDF computation so as to limit the mass loss with a reduced computational time. The idea is to advect during $(N - 1)$ time steps the SDF, and every N time step to advect the surface to compute the whole SDF (see figure 6). Some images of different advection time for a box (dimension $[-0.1, 0.1] \times [-0.2, 0.2] \times [-0.05, 0.05]$) submitted to a rotation are presented in figure (7a). The adaptation is the same at each time and is presented for $t = 0$ figure 7b.



(a) 0 isoline, from left to right : $t = 0$, $t = 1.64$, $t = 2.71$



(b) Left to right : cross-section $x = 0$, cross-section $y = 0$, zoom for cross-section $x = 0$

Figure 7: Plane rotation

7 Conclusion

In this work, we combined the simplicity given by the penalization method to compute flow solutions around complex geometries with the power of mesh adaptation to improve accuracy. The accuracy is seek in the localization of the 0 level set function (to be able to impose correctly boundary conditions inside our IBM) and also in the physical solution (to correctly capture shock waves for example). Another advantage of using IBM is the facility of studying moving bodies. To be able to perform unsteady computations we are studying a way of tracking interfaces on a mesh.

Acknowledgement

This work has been funded in part by the European program FP7/2007-2013 called STORM-FP7-605180 (www.fp7-storm.eu).

Experiments presented in this paper were carried out using the PLAFRIM experimental testbed, being developed under the Inria PlaFRIM development action.

REFERENCES

- [1] R. Abgrall, H. Beaugendre and C. Dobrzynski, An immersed boundary method unusing unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *J. Comput. Phys.* 257 (2014) 83-101.
- [2] C. Dapogny and P. Frey, Computation of the signed distance function to a discrete contour on adapted triangulation. *Calcolo* (2012) 49:193-219.
- [3] C. Dobrzynski and P. Frey, Anisotropic delaunay mesh adaptation for unsteady simulations., *Proc.of 17th Int. Meshing Roundtable*, Pittsburgh, USA, 2008.
- [4] P. Frey and F. Alauzet, Anisotropic mesh adaptation for CFD computations, *Comput. Methods Appl. Mesh. Engrg.* 194 (2005) 5068-5082.
- [5] C. Bui, C. Dapogny and P. Frey, An accurate anisotropic adaptation method for solving the level set advection equation, *Int. J. Numer. Meth. Fluid*(2011).
- [6] R. Abgrall, Essentially non oscillatory Residual Distribution schemes for hyperbolic problems, *J. Comput. Phys.* 214,2 (2006) 773-808.
- [7] R. Abgrall and P.L. Roe, High Order Fluctuation Schemes on Triangular Meshes, *J. Sc. Comp.*,Vol.13,Nos1-3,December 2003.
- [8] R. Abgrall and D. De Santis, High order residual distribution scheme for Advection-Diffusion problems,*SIAM J. Sci. Comput*, 2014, accepted.
- [9] D. De Santis, *Development of a high order residual distribution method for Navier Stokes and RANS equation*, PhD. thesis, Univ. of Bordeaux I, December 2013.