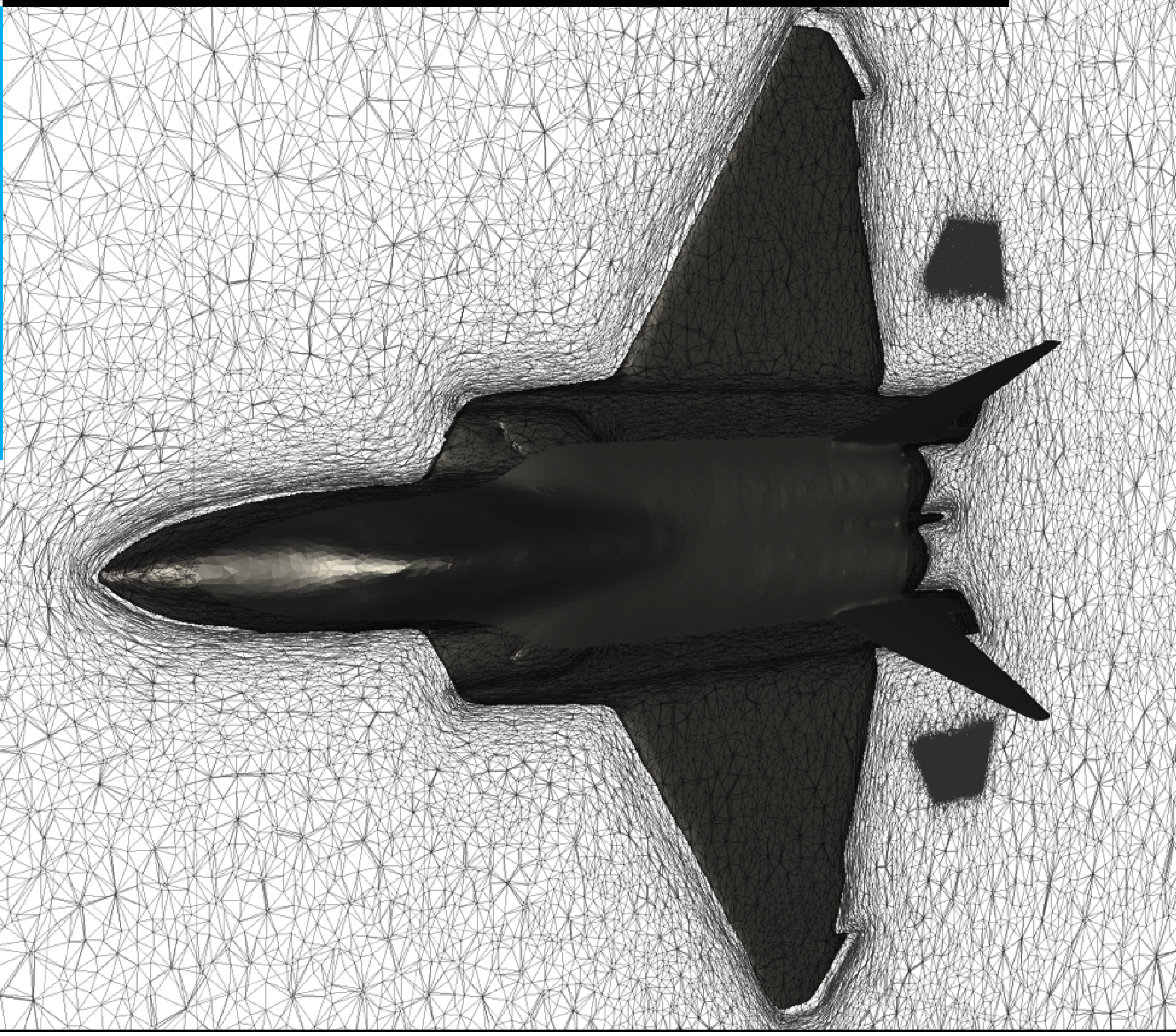
Embedded Boundary Method

For Aerospace Problems

Michele Pisaroni

Master of Science Thesis



Delft University of Technology

Faculty of Aerospace Engineering

Embedded Boundary Method

For Aerospace Problems

Master of Science Thesis

For the degree of Master of Science in Aerospace Engineering at Delft University of Technology

Michele Pisaroni

Thesis supervisors: Prof.dr.ir. H. Bijl, Dr.ir. A.H. van Zuijlen, Dr. D.J.P. Lahaye, Dr.ir. Roeland De Breuker

August 8, 2013

Faculty of Faculty of Aerospace Engineering · Delft University of Technology

Abstract

The application of Computational Fluid Dynamics to model and simulate flows around flexible and moving objects has grown in the last decades fueled by new technological challenges in particular in the aerospace engineering field because of the introduction of highly deformable materials and complex moving systems.

Existing body-fitted mesh methods such as the Arbitrary Lagrangian Eulerian (ALE) approach have been proposed to simulate the flow around moving and deforming structures but their applicability is limited because of the issues arising in deforming the computational grid constrained to the moving/deforming structure.

This thesis focuses on the verification, and validation of an Embedded Boundary method (developed at Stanford University by Prof. Farhat research group) for the solution of fluid-structure interaction problems involving large and complex structural motions and deformations. The Embedded Boundary method works on non-body fitted grids, by using a tracking algorithm is able to impose the effects of an 'immersed' moving and deforming surface mesh on the fixed Eulerian fluid mesh. For this reason this method is gaining popularity because it simplify a number of issues ranging from codes coupling (fluid-structure solvers) to formulating and implementing algorithms for applications that involve very large and complex motions-deformations and for which ALE algorithms are unfeasible.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to thank my supervisors, Professor Hester Bijl, Professor Alexander van Zuijlen, Professor Domenico Lahaye and Professor Roeland de Breuker for their support and suggestions to my research and my dissertation.

I offer my deepest gratitude to Professor Charbel Farhat that gave me the golden opportunity to visit for seven months his research group (FRG) at Stanford University for my thesis project. I would also like to thank all my colleagues at Farhat Research Group and in particular my office mates: Maciej Balajewicz, Vinod Lakshminarayan and Elie Hachem for their support and suggestions.

I am also indebted to Prof. Domenico Lahaye and Ing. Rudy Sady for the opportunity to work for the last two years on a research project at DIAM at TU Delft and at Alamtis B.V. in Rotterdam.

Especially, I would like to warmly thank my parents Lucia and Fabrizio for their patience and unconditional support.

Last but not least, I wish to thank my friends Antonino Parrinello and Rukshan Matteo Perera for their support and the time spent together in the last years in different parts of the world.

Delft, University of Technology — Michele Pisaroni
August 8, 2013

“The purpose of computation is insight, not numbers.”

— *Richard Hamming, Numerical Methods for Scientists and Engineers (1962)*

Chapter 1

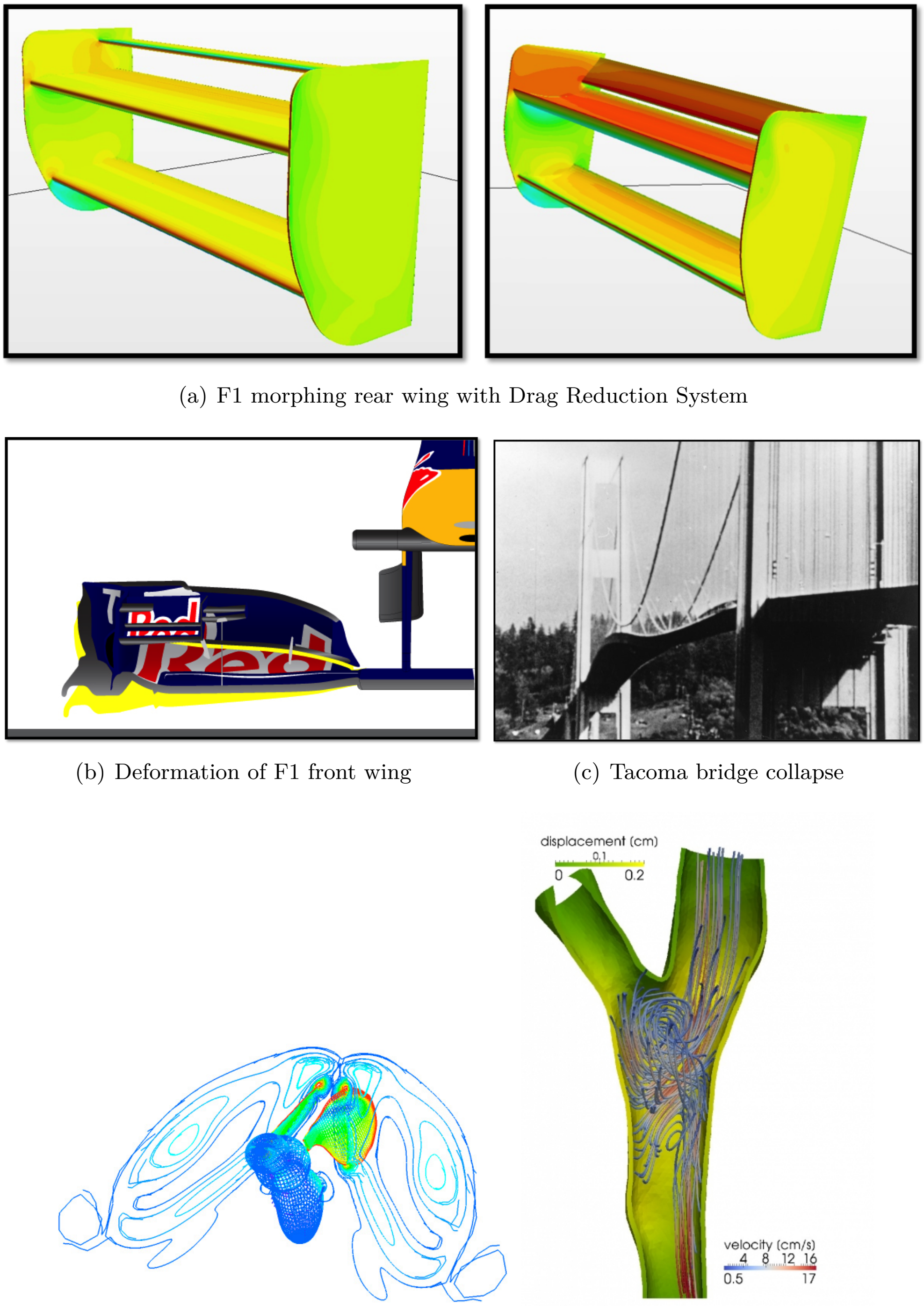
Introduction

The application of Computational Fluid Dynamics (CFD) to simulate flows around rigid obstacles or predict the Fluid-Structure Interaction (FSI) of flexible bodies with the surrounding flow has grown tremendously in the last decades fueled by new technological challenges and applications in a wide range of engineering fields.

In FSI the evolution of each entity (fluid and flexible structure) depends on the evolution of the other and a coupling phenomenon may appears. The aerodynamic forces influence the movement and deformation of the structure and on the other hand the movement and deformation of the structure affect the fluid flow and thus the aerodynamic loads. During the 20th century, with the advent of digital computer, CFD and finite element methods (FEM), FSI has been initially investigated in the field of aeroelasticity to predict potentially catastrophic phenomena such wing flutter and limit-cycle oscillation (LCO). At a later stage FSI gained a prominent role also in other engineering fields. Examples include parachute dynamics, turbomachinery, vibrations of bridges and buildings induced by the wind, deformation of morphing race-car wings to enhance the downforce and reduce the drag, deformation of the heart and the relative blood flow and the flapping flight of insects and birds (Fig.1-1).

During the last decade, the continuous emphasis on innovative materials and parallely on the design of lighter and morphing airplanes and thinner, highly deformable wings, has raised the level of interest in accurate, reliable and robust CFD-based FSI methods to predict and simulate the flow-structure coupling effects but also optimize existing aerospace systems. Inspired by flying birds and insects, flapping wings can be recommendable for small vehicles operating at low speed, generally designed for surveillance purposes, because of the high maneuverability and small dimensions. Higly deformable and flexible wings are currently employed for high altitude long endurance (HALE) flights. To achieve high efficiency at high altitude, their wings are characterized by large aspect ratio and low weight. As a result, they exhibit high flexibility in flight. One example of such aircrafts is the Helios prototype vehicle (Fig.1-2), a solar and fuel-cell powered unmanned aerial vehicles (UAV).

During a test flight in preparation for an endurance test the Helios Prototype broke up and fell into the Pacific Ocean. According to the investigation report: *"the aircraft encountered*



(a) F1 morphing rear wing with Drag Reduction System

(b) Deformation of F1 front wing

(c) Tacoma bridge collapse

(d) Flapping fruit fly

(e) Blood flow in an elastic artery

Figure 1-1: FSI domains of application

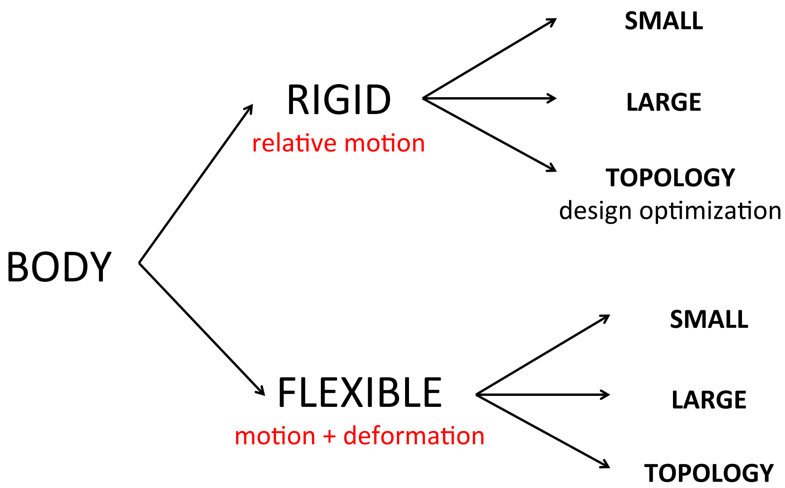
Figure 1-2: NASA Helios Prototype

turbulence and morphed into an unexpected, persistent, high dihedral configuration. As a result of the persistent high dihedral, the aircraft became unstable in a very divergent pitch mode in which the airspeed excursions from the nominal flight speed about doubled every cycle of the oscillation. The aircraft's design airspeed was subsequently exceeded and the resulting high dynamic pressures caused the wing leading edge secondary structure on the outer wing panels to fail and the solar cells and skin on the upper surface of the wing to rip off. The aircraft impacted the ocean within the confines of the PMRF test range and was destroyed." [17]

The investigation report identified a two-part root cause of the accident:

- **Lack of adequate analysis methods** led to an inaccurate risk assessment of the effects of configuration changes leading to an inappropriate decision to fly an aircraft configuration highly sensitive to disturbances.
- Configuration changes to the aircraft, driven by programmatic and technological constraints, altered the aircraft from a spanloader to a highly point-loaded mass distribution on the same structure significantly reducing design robustness and margins of safety.

An extensive study of such problems remains an open challenge due to their strong non-linearity and multidisciplinary nature. Analytical solutions are often impossible to obtain and experiments are usually limited in scope, complex and very expensive, thus numerical simulations are generally needed to investigate the fundamental physics and predict the mutual interaction between rigid or flexible moving objects sometimes subjected to topological changes in a surrounding (and/or internal) flow. Up to these days, FSI problems have been studied for many engineering applications, however in spite of the actual maturity of CFD and the computational power available, several key issues such as the lack of robustness,



accouracy, flexibility and computational requirments for coupled problems still prevent the analysis and the simulation of certain aerospace systems.

The Arbitrary Lagrangian Eulerian (ALE) approach gained a lot of interests because it combines the advantages of Lagrangian representation and those of the Eulerian formulation and allow solving the Navier-Stokes (or Euler) equations in a moving and deformable domain. Unfortunately ALE methods works well for FSI problems as long as the deformations or translation of the solid remains within certain limits. When these limits are exceeded the mesh elements become ill shaped and ALE fails (Fig.1-3).

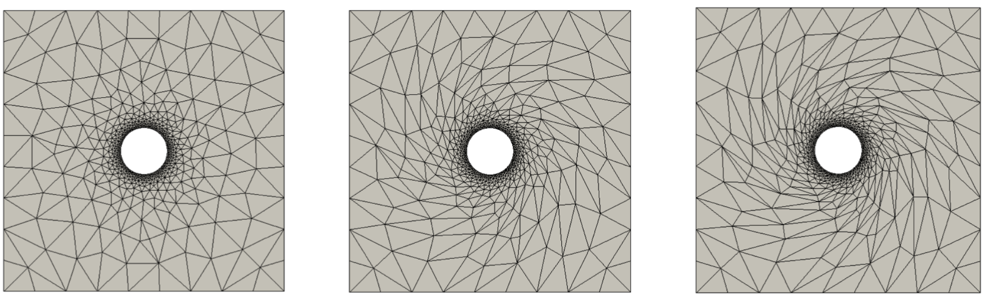


Figure 1-3: Mesh deformation for a rotating cylinder

In the literature a variety of non conform mesh methods, such as the Embedded Boundary Methods (EBM), are still developed to overcome some of the above-mentioned limitations and issues of "body fitted mesh methods" by using a mathematical formulation that allow to impose the effects of an "immersed" moving and deforming body mesh on a fixed fluid mesh. For this reason these methods are very attractive because they simplify a number of issues ranging from codes coupling to formulating and implementing algorithms for applications that involve very large motions and deformations and topological changes, and for which alternative ALE algorithms are unfeasible. However in case of large motion-deformation the mesh requirements (in particular for viscous flows) limit the applicability of such methods for real full scale engineering problems.

The present work highlights these recent developments, shows the flexibility and the robustness of these class of methods for problems involving large motion-deformations and topological changes and proposes new directions to overcome the above mentioned limitations of the EBM.

Chapter 2

State-of-the-art

Numerical techniques designed for handling the interaction of a fluid and moving (deforming) body(es) can be divided into two main classes depending on whether the CFD grid moves (deforms) according to the motion (deformation) of the immersed object. The first class of methods, that we denote as *Body-Fitted* (BF) mesh methods, include the dynamic mesh method [18] [28] [4], the co-rotational approach [8] [24] and the ALE method [40] . The second class of methods, that we denote as *Non-body-fitted* (NBF) mesh methods, includes the immersed boundary, embedded boundary, immersed interface, fictitious domain, and Cartesian methods.

2-1 Body-fitted mesh methods

The main feature of the BF mesh methods is that they operate on dynamic, body-conforming CFD grids and a particular algorithm take care of the mesh motion (deformation) to accomodate the body motion (deformation) and maintain a conformal CFD dynamic wet surface of the solid body.

2-1-1 Dynamic mesh method

The *dynamic mesh method* is based on a Lagrangian flow formulation, and dynamic finite element meshes. The dynamic wet interfaces are treated with a material flow description to avoid the tracking of the moving boundaries. However the dynamic meshes that propagate with the flow become distorted and should be regenerated at each time step. This approach was successfully appied to large-scale simulations of flows with numerous cellular bodies [18], inviscid aerodynamic analysis [4], sloshing problems [28] and in the field of hemodynamics but it is impractical for high Reynolds number viscous engineering problems that require large (and fine) meshes because of the computational expensive remeshing procedure needed at each time step.

2-1-2 Co-rotational approach

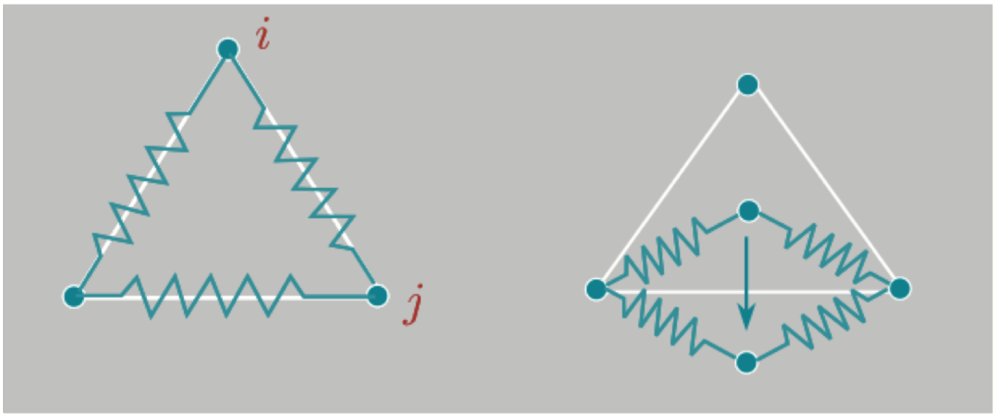
In the *co-rotational* approach the equations for the fluid are represented in multiple moving (translating and rotating) reference frames attached to the nodes of the moving solid body. Such approach allow the simulation of maneuvering wings or wing-body configurations undergoing rigid motions [8]. If the solid moving-deformimg body is represented by a FE model and the reference frames are connected to the discretized structure, a rigid aeroelastic simulation can be performed without explicitly moving the grid nodes. However if the elastic deformation of the structural body is included, an additional (expensive for large deformations) algorithm is needed to update the position of the nodes [24].

2-1-3 The Arbitrary Lagrangian Eulerian (ALE) method

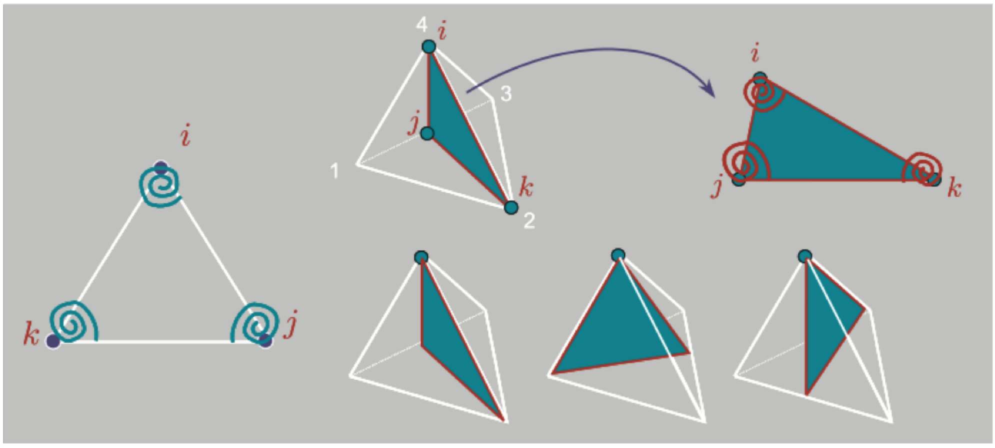
The *ALE* method combines the advantages of the Lagrangian representation (well defined interface and facility in imposing the boundary conditions) with those of the Eulerian formulation (possibility of handling deformation) [40]. In this apporach the solid body moves relative to the mesh as in a Eulerian framework, but the external (wet) shape of the grid volumes is controlled by the boundary conditions of the problem as in a Lagrangian formulation. A direct consequence of this mixed formulation is that the Navier-Stokes (Euler) equations are solved in a moving and deforming domain and hence an additional convection term is needed to take into account the velocity of the mesh (not known a priori). A formal theory for the time-accuracy for ALE schemes was derived by Farhat et al. [19]; the so called Geometric Conservation Law (GCL)/Discrete Geometric Conservation Law (DGCL), based on the fact that the scheme should be able to exactly integrate a uniform flow for any mesh motion, must be satisfied to ensure numerical stability.

In addition to the solution of the fluid and structure equations, a mesh motion algorithm is needed to deform the fluid domain grid. Different mesh motion algorithms designed for structured and unstructured grids were proposed, the two main classes are the *grid connectivity schemes* that exploit the connectivity of internal grid points and the *point by point schemes* that move each grid point based on its position in space using a radial basis function interpolation [2] . One of the most widely used grid connectivity scheme for unstructured meshes is the *structure analogy*; the grid is considered as a truss structure and the edges or the nodes of the finite volumes are modeled as network of fictitious linear, torsional [11] or ball vertex [36] springs (Fig.2-1), a linear system is solved at each iteration and the position of the grid nodes is updated based on the solution of the pseudo-structural problem.

The main disadvantage of this approach is that for large deformation and topological changes some pseudo-structural edges can penetrate their neighboring triangles and produce negative volume elements, in these occasions the simulation fails and an expensive and time consuming remeshing step is usually needed. The three-field (fluid, structure and fluid grid) formulation [25] was successfully used to simulate high performance military aircraft transonic flutter [6]. In that regime, the popular k and $p - k$ procedure, based on linear flow theory, for subsonic regime and the piston theory for supersonic regime are not reliable. The three-field formulation where the flow is modeled by the ALE fomulation of the fluid equations and the structure by a nonlinear FE model hence become the only available choice and gradually substitued the expensive and time consuming experimental wind tunnel flutter testing procedure in the design of these fighters: "The results of a finite number of CFD



(a) Pseudo-structure with lineal springs

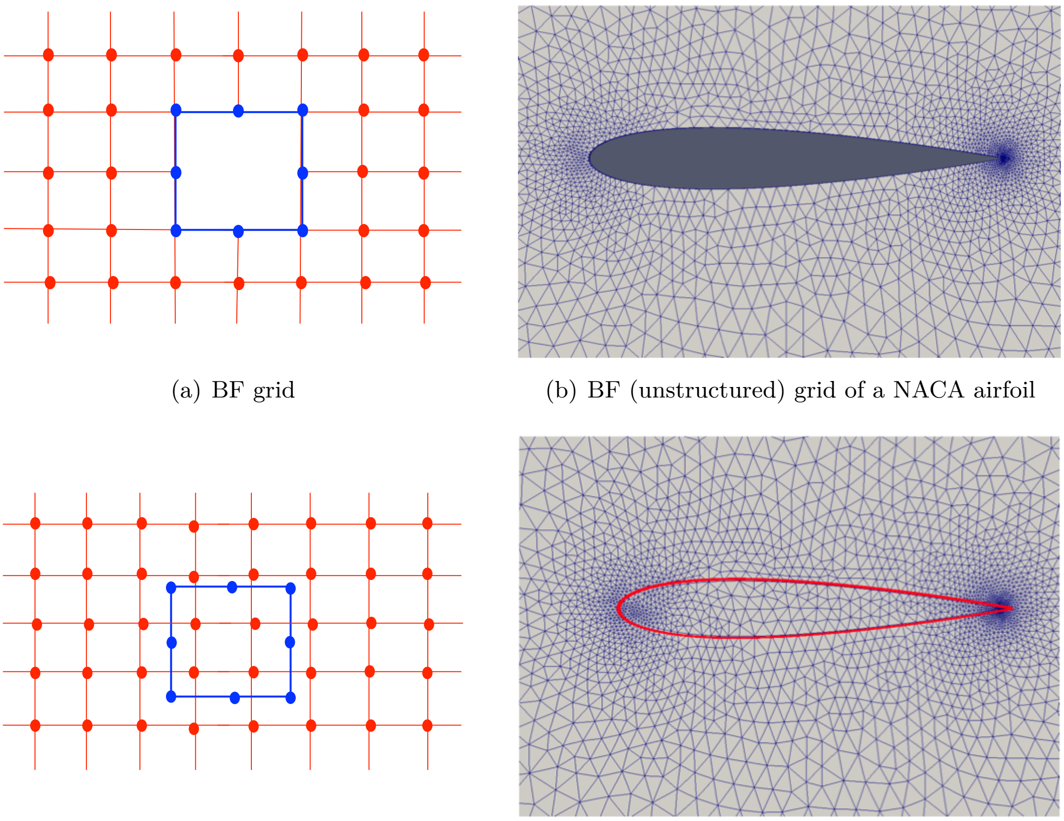


(b) Pseudo-structure with torsional springs

Figure 2-1: Structure analogy mesh motion-deformation approaches solutions could be used as a replacement for wind tunnel testing, assuming a validated code was available."[7]. The ALE methods can naturally accommodate spatially high-order schemes at the fluid-structure interface, and allow a high mesh resolution for boundary layers in viscous flows, however *"most successes have been encountered so far either for complex geometries but inviscid grids, or viscous grids but simple geometries"* [10]. Achieving efficiency, robustness, and high mesh quality requires extreme care when the motion and deformation of the solid body is large and undergoes some kind of topological change.

2-2 Non-body fitted mesh methods

In contrast with BF methods, the NBF methods operate on fixed, non body-conforming CFD grids (Fig:2-2).



(a) BF grid

(b) BF (unstructured) grid of a NACA airfoil

(c) NBF grid

(d) NBF (unstructured) grid of a NACA airfoil

Figure 2-2: Comparison of BF and NBF CFD grids

2-2-1 Immersed Boundary (IB)

The methodology and the term *Immersed Boundary* (IB) was introduced by Peskin in 1972 [32]. He developed mixed Eulerian-Lagrangian approach, based on fixed Cartesian mesh for the Eulerian fluid variables and moving curvilinear mesh for the Lagrangian variables linked by interaction equations, to simulate the FSI of the blood flows through the elastic heart valves. The Cartesian grid, generated with no regard to the solid body surface, is cut by the 'immersed' surface grid and a modification of the governing equations is needed in the vicinity of the boundary to incorporate the appropriate boundary conditions. Different approaches were proposed to enforce the boundary conditions in a non-body conforming framework depending on the problem under consideration.

Continuous forcing approach

In the *continuous forcing approach*, particulary suitable for elastic boundaries, the force applied by the IB to the fluid is distributed over a set of cells around each Lagrangian grid point and than used in the momentum equations of the neighbouring nodes [33].

Virtual boundary method approach

The *virtual boundary method approach* [38], for rigid boundaries, treats the body surface as a virtual boundary that applies force on the fluid to accomodate the no-slip condition at the interface (flow at rest at the surface). The *distributed lagrangian multiplier method* [31] is based on the same idea but introduce Lagrange multipliers (the body force) on the IB to satisfy the no-slip condition.

Direct forcing approach

The *direct forcing approach*, based on the idea of mirroring (the grid nodes inside the IB have a opposite flow field to the external grid nodes) build implicitly the forcing term by calculating the difference between the interpolated velocities at the boundary nodes and the required boundary velocity [27]. The above mentioned approach were successfully tested for low and moderate Reynolds number problems involving complex flow past 2D and 3D bluff bodies [20] [22].

Ghost cell approach

For high Reynolds number the accouracy required in the boundary layer region around the IB assume a great imporance; to avoid the spreading of the effect of the IB (the previous approches emply a smooth forcing term) and increase the local accuracy a particular treatment of the cells near the IB is needed. The *ghost cell* approach [16] [15] implicitly incorporates the boundary condition on the IB by using an interpolation (linear, quadratic or linear in the tangential direction and quadratic in the normal direction) between the 'ghost cells', cells that are inside the solid region and have at least one neighbour in the fluid domain, and the fluid cells. Such approach is particulary suitable for high Reynolds number problems because it can generate a sharp representation of the IB and and hence limits the spreading of the effect of the IB in the boundary layer.

Cut-cell methodology

All of the above mentioned approaches derived form the original Peskin's IB method are not designed to guarantee the conservation laws for the cells in the vicinity of th IB. The only way to re-enstablish the conservation of the mass and momentum is to built a variant of the standard finite-volume apprach based on the *cut-cell* methodology [23]. In this approach, the cells that are cut by the IB are reshaped by removing the portions that lie inside the solid domain if their centers lie inside the fluid domain or are reshaped by removing their solid portions and merged with their neighbouring fluid cells if their centers lie in the solid domain.

After ensuring the accuracy of the above mentioned non-body conforming methods in solving problems that involve flow around fixed [37] [30], moving [23] [39] and deformable [26] [12] objects, the present research trend is to develop hybrid methodologies that gather togheter some of the capabilities of the above mentioned approach and employ them to solve FSI problems with large structural motions, deformations and topological changes [5], for which, as explained in the first section, the ALE apprach fails.

2-2-2 Embedded Boundary Method (EBM)

A particular interesting approach, called *Embedded Boundary Method* (EBM), designed for computations on unstructured grids, was recently developed by Prof. Charbel Farhat research group at Stanford University.

The lack of a representation of the structural wet surface in the unstructured CFD grid require a particular treatment of the boundary conditions and the fluid-structure transmission conditions that are required in FSI simulations. The continuity condition, similar to the velocity boundary condition for rigid obstacles can be enforced by using one of the above mentioned approaches (interpolation, mirroring, ...). However to enforce the equilibrium condition, balance between the fluid and structural surface tractions, it is required to compute the flow-induced load on the wet surface of the structure that is not represented in the CFD grid. Instead of relying for this purpose on interpolation or extrapolation as the existing methods [29] [15], the EBM enforces the appropriate value of the fluid velocity at the wall and recovers the value of the fluid pressure via the exact solution of local, one-dimensional Riemann problems between the ghost and fluid nodes [35] [12] (velocity and pressure conditions on the 'embedded' interfaces are treated simultaneously).

Two consistent and conservative methodologies, the *reconstructed surface* approach and the *control volume* approach are employed to evaluate flow-induced forces and moments on rigid and flexible embedded interfaces. The first one is based on local reconstruction of the embedded discrete interfaces an the other is based on the level set concept [12].

The tracking of the position of the embedded interface with respect to the non body-conforming CFD grid is a key part of immersed/embedded approaches. Most computational methods have been mainly focused on closed embedded interfaces. However, this assumption is limiting as many FSI problems, like flapping wings or membranes, involve open thin shell surfaces. The EBM is equipped with two interface tracking algorithms. The *FRG* algorithm is based on a projection approach, it is very fast but it is restricted to closed interfaces and resolved enclosed volumes. The *PhysBAM* algorithm is based on a collision approach derived from computational graphics [14] [1] is slower than the previous approach but can handle open shell surfaces and underresolved enclosed volumes [13].

The mesh generation procedure is undoubtedly simplified in non-body fitted approaches because there are no particular constraints determined from the geometry of the solid boundary and the grid quality is not affected by the complexity of the geometry as in BF grid. However, in a viscous simulation, as the Reynolds number increases, the size of the grid increases faster than a corresponding BF grid because the informations of the solid surface grid, that are generally used to control the mesh resolution in the vicinity of the body, are not available in the CFD grid [21]. In some cases this increase in grid size does not directly imply a corresponding increase in computational cost because a large part of the grid points can be inside the solid body where the flow equations are not solved. However, for large 3D problems at high Reynolds number (with very fine mesh in the boudary layer) the generation of a grid designed without any information of the immersed geometry is computationally expensive and can become impractical.

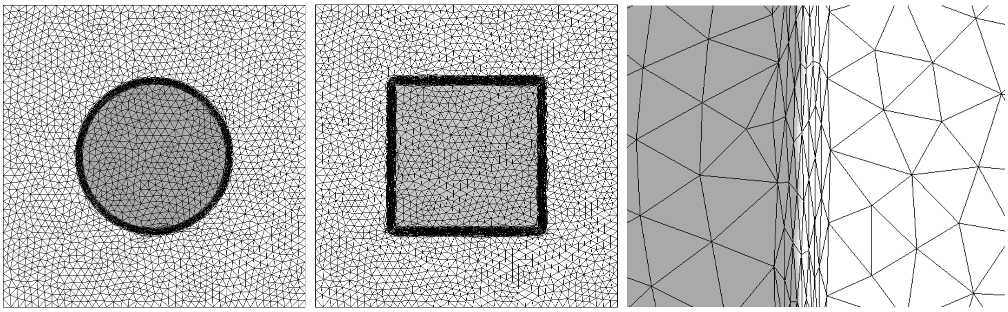
2-2-3 Immersed volume method

To overcome the above mentioned issue, some methods were designed to operate on unstructured grids and in some case also equipped with mesh adaptaton algorithms. An example is the *immersed volume method* [9] that is able to immerse and represent complex moving-deforming geometries inside a unique CFD mesh.

Starting form the initial geometry, a signed distance function (levelset) is computed for each node of the mesh, then the latter is refined anisotropically at the interface between the fluid domain and the structural domain using the gradient of the distance function. The level set function assume a positive value in the fluid domain, a negative value inside the structural domain and zero value at the interface. If multiple structural domains are present in the same fluid, a level set function is defined for each domain. The physical and mechanical properties can then be determined on the whole domain in terms of the level-set function. For the elements crossed by the level-set functions and the their neighbours, a smoothed Heviside function is used to determine the element effective properties. The anisotropic mesh adaptation, by make use of the information of the level set function, improve the accouracy of the solution at the interface by stetching the elments and reducing the dissipation of the effects introduced by the smooth Heaviside function (sharply define the interface) and drastically reduce the size of the mesh required to represent and approximate complex geometries. The mesh adaptation algorithm is very efficient in reducing the mesh size and hence the degrees of freedom of the problem but can slow down the calculation because the level set functions should be recomputed once the solid boundary moves and deform.

2-3 Thesis motivation

By looking at the actual need of accurate and reliable computational methods to simulate, design and optimize aerospace systems with complex geometries, moving parts and highly deformable matherials the immersed/embedded boundary methods are gaining popularity for the flexibility due to the fact that they operate on non-conformal grids. They are a perfect



(a) Immersed disk (b) Immersed square (c) Zoom if the refined interface

Figure 2-3: Anisotropic refined fluid-solid interface of an immersed disk (left), an immersed square (center) and a zoom on the refined interface (right).

candidate to link the gap between the thecnological challenges that are present in today's most demanding aerospace applications and the rigorous and expensive design process needed to guarantee the highest safety standards.

The main intent of this thesis work it to show the the advantages, disadvantages and performances of the EBM developed by Prof. Farhat research group at Stanford University to solve problems involving moving, deforming and flapping structural objects in laminar and turbulent flows and propose new directions to optimize the meshing procedure by employing the informations of the embedded geometry in the design of the fixed CFD grids through level set functions.

In addition an hybrid method based on the co-rotational approach and the mesh deformation algorithm generally used in body-fitted simulations is tested to further reduce the mesh size required in the CFD grid to simulate moving and deformable objects.

Chapter 3

Problem Formulation

This chapter is dedicated to the description of the mathematical models used to describe and analyze the behavior of FSI problems.

Let us consider a fluid domain $\Omega \in \mathbb{R}^3$ with external boundaries $\partial\Omega$. Inside this domain, a structural body B is "immersed"; let ∂B be the boundary of B (Fig.3-1). A mathematical model for this FSI problem involves:

- the governing equations for the fluid domain $\Omega(t) \in \mathbb{R}^3$,
- the governing structural dynamics equations for the solid domain $B(t) \in \mathbb{R}^3$,
- transmission conditions at the fluid-structure interface $\partial B(t)$
- Dirichlet and/or Neumann boundary conditions at the remaining fluid and structural domain boundaries,
- initial conditions (t = 0) for the fluid and structural state vectors.

First we will describe the fluid (3-1) and the structure (3-2) sub problems separately and than we will introduce the modeling of the interaction of those sub problems across the interface (3-3).

3-1 Fluid domain

The domain Ω is filled with a *compressible fluid*[1] governed by the equations of conservation of mass (continuity), conservation of momentum and conservation of energy. Depending on the proprieties of the problem under consideration and the flow regime, we may need to represent the fluid using the viscous Navier-Stokes equations, equipped with a turbulence model in case of high Reynolds number, or with the inviscid Euler equations.

[1] Most aerospace problems are in the

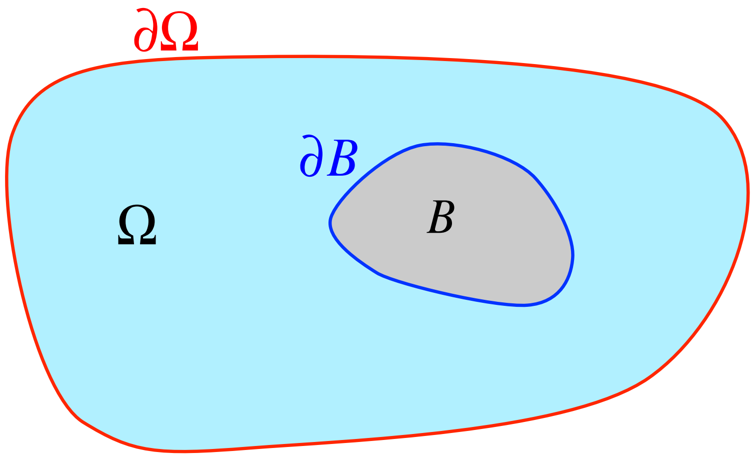


Figure 3-1: The fluid domain Ω, its boundary $\partial\Omega$, the immersed body B and the structure interface ∂B.

3-1-1 Viscous Compressible Fluid

If the domain Ω is filled with a *viscous compressible fluid*, the flow dynamics is governed by the Navier-Stokes equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho v_i) = 0 \\ \frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_j}(\rho v_i v_j + p\delta_{ij} - \tau_{ji}) = 0 \\ \frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}\left[\rho v_j E + v_j p - v_i \tau_{ij} + q_j\right] = 0 \end{aligned} \tag{3-1}$$

where ρ is the mass density, v the velocity, δ the Kronecker delta, p the gas pressure, τ the fluid stress tensor, $E = e + \frac{v_k v_k}{2}$ is the total specific energy of the gas and q is the heat flux. Indices i, j equal $1, 2, 3$ [2].

The above mentioned set of equations is completed by thermodynamic relationships, constitutive relationship, expressions for transport coefficients, initial and boundary conditions.

[2]Repeated indices are summed over.

In this thesis we assume that the gas satisfies the ideal gas equation of state $p = \rho RT$ and behaves as a *perfect gas* [3]:

$$
\begin{aligned}
e = C_v T &= \frac{p}{(\gamma-1)\rho} \\
h = e + \frac{p}{\rho} = C_p T &= \frac{\gamma p}{(\gamma-1)\rho}
\end{aligned}
\tag{3-2}
$$

The constitutive relations provide expressions for the fluid stress and the heat flux in terms of other fluid quantities. For a *Newtonian fluid* [4] the viscous stress are given by:

$$
\tau_{ij} = 2\mu S^*_{ij} \qquad where \qquad S^*_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij}
\tag{3-3}
$$

The heat flux is given by the *Fourier's law*:

$$
q_i = -k\frac{\partial T}{\partial x_i}
\tag{3-4}
$$

where μ is the dynamic viscosity and k is the thermal conductivity coefficient of the gas.

The transport coefficient μ and k are functions of pressure and temperature. In this thesis we used the *Suntherland's Law* to determine the viscosity as function of the temperature:

$$
\mu = \mu_{ref}\left(\frac{T}{T_{ref}}\right)^{\frac{3}{2}}\frac{T_{ref}+S}{T+S}
\tag{3-5}
$$

where T_{ref} is a reference temperature and μ_{ref} is the viscosity at the T_{ref} reference temperature, S is the Sutherland temperature ($S = 110.4[K]$). The Prandtl number, $Pr \approx 0.7$ for many gasses, can than be used to relate μ and k:

$$
Pr = \frac{\mu C_p}{k}
\tag{3-6}
$$

At the solid boundary ∂B the following boundary conditions should be imposed:

$$
\begin{aligned}
\vec{v} &= 0 \qquad (no-slip) \\
T = T_{wall} \quad (fixed\ T) \qquad or \quad \vec{\nabla} T \cdot \vec{n} &= 0 \quad (adiabatic\ wall)
\end{aligned}
\tag{3-7}
$$

The above set of Navier-Stokes equations can be solved directly, *Direct Numerical Simulation (DNS)*, without any turbulence model. In this case the whole range of the turbulent statistical fluctuations at all relevant physical scales (from the smallest dissipative Kolmogorov microscales, up to the integral scale) must be resolved. This approach is very computational demanding because the number of computational grid points needed to satisfy the resolution

[3] Constant specific heats, C_v and C_p and hence specific heat ratio γ
[4] assuming Stokes Law for mono-atomic gasses

requirements is typically very high and grows very fast with the Reynolds number [5]. In case of realistic Reynolds number ($10^5 - 10^7$) the DNS approach becomes prohibitively expensive.

Two options have been proposed and are nowadays employed to compute the turbulent flow around fixed and moving obstacles: the *Large Eddy Simulation (LES)* and the *Favre-averaged Navier-Stokes (FANS)*.

The LES is based on the idea of reduce the range of lenght scales of the turbulence spectrum contained in the solution of the Navier-Stokes equations. This is accomplished by filtering the Navier-Stokes equations to remove the smallest scales of the solution, below that scale (subgrid scale) the turbulence is modeled by semi-empirical laws. By doing so the computational cost and the number of grid points is dramatically reduced. The unresolved small scales (smaller than a cutoff filter width) are modeled [6] instead of directly resolved as in the DNS approach. The lowest identified scales are related to the mesh size. Since the large turbulent fluctuations are directly simulated, the computational requirements of LES are still very high [7].

On the other hand the FANS ignores the turbulent fluctuations and aims at calculating only the turbulent averaged flows. Through the Reynolds decomposition (separation of a variable in mean and fluctuating component) the set of incompressible Navier-Stokes equations is simplified into the RANS equations. Since we are considering the compressible form of the Navier-Stokes equations (3-1), in addition to the classical Reynolds time averaging we need also to use a density weighted time averaging, the *Favre time averaging*, to avoid the introduction of higher-order terms [8] in the averaged Navier-Stokes equations:

$$
\begin{aligned}
(Reynolds\ averaging) \qquad \overline{\Phi} &= \frac{1}{T}\int_T \Phi(t)dt, \qquad \Phi' = \Phi - \overline{\Phi} \\
(Favre\ averaging) \qquad \tilde{\Phi} &= \frac{\overline{\rho\Phi}}{\overline{\rho}}, \qquad \Phi'' = \Phi - \tilde{\Phi}
\end{aligned}
\tag{3-8}
$$

The time-averaged version of the Navier-Stokes equations (3-1), using the standard Reynolds

[5]The size of the smalles turbulent scales related to the turbulent dissipation (Kolmogorov scales) is inversely proportional to $Re^{3/4}$. A resolution of n points per unit length of the smallest eddy determine a total number of mesh points of $n^3 \cdot Re^{9/4}$. The time step is determined by the smalles turbulent time scales which are proportional to $Re^{3/4}$. In conclusion the total computational effort for DNS is proportional to Re^3 for homogeneous turbulence.

[6]The two most common sub-grid scale models are the Smagorinsky model (SGS) and the Germano dynamic model (DSGS).

[7]The total computational effort for LES is proportional to $Re^{9/4}$

[8]Products of fluctuations between density and other variables

averaging for ρ and p and the Favre averaging for u and E are:

$$
\begin{aligned}
\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\widetilde{v}_i\right) = 0 \\
\frac{\partial}{\partial t}\left(\overline{\rho}\widetilde{v}_i\right) + \frac{\partial}{\partial x_j}\left[\overline{\rho}\widetilde{v}_i\widetilde{v}_j + \overline{p}\delta_{ij} + \underbrace{\overline{\rho v_i'' v_j''}}_{(1)} - \widetilde{\tau}_{ji} - \underbrace{\overline{\tau_{ji}''}}_{(2)}\right] = 0 \\
\frac{\partial}{\partial t}\left(\overline{\rho}\widetilde{E}\right) + \frac{\partial}{\partial x_j}\Bigg[\overline{\rho}\widetilde{v}_j\widetilde{E} + \widetilde{v}_j\overline{p} + \underbrace{C_p\overline{\rho v_j'' T}}_{(3)} + \underbrace{\widetilde{v}_i\overline{\rho v_i'' v_j''}}_{(4)} + \underbrace{\frac{\overline{\rho v_j'' v_i'' v_i''}}{2}}_{(5)} \\
-C_p\frac{\mu}{Pr}\frac{\partial \widetilde{T}}{\partial x_j} - \underbrace{C_p\frac{\mu}{Pr}\frac{\partial \overline{T''}}{\partial x_j}}_{(6)} - \widetilde{v}_i\widetilde{\tau}_{ij} - \underbrace{\overline{v_i''\tau_{ij}}}_{(7)} - \underbrace{\widetilde{v}_i\overline{\tau_{ij}''}}_{(8)}\Bigg] = 0
\end{aligned}
\tag{3-9}
$$

This new set of equations contains unknown terms (marked in red) that have to be modeled. The term (2) and (8) can be neglected if $|\widetilde{\tau_{ij}}| >> |\overline{\tau_{ij}''}|$ (valid for all flows). Term (3), corresponding to turbulent transport of heat, can be modeled using the gradient approximation for the turbulent heat-flux:

$$
q_j^{turb} = C_p\overline{\rho v_j'' T} \approx -C_p\frac{\mu_t}{Pr_t}\frac{\partial \tilde{T}}{\partial x_j}, \qquad Pr_t \approx 0.9 \tag{3-10}
$$

Term (5) and (7), corresponding to turbulent transport and molecular diffusion of turbulent energy, can be neglected if, $k_t << \tilde{h} = C - p\widetilde{T}$, turbulent energy is small compared to the enthalpy (valid below hypersonic regime). Term (6), related to heat conduction effects associated with temperature fluctuations, can be neglected if $\left|\frac{\partial^2 \widetilde{T}}{\partial x_j^2}\right| >> \left|\frac{\partial^2 \overline{T''}}{\partial x_j^2}\right|$ (valid for all flows).

Term (1) and (4) contain the Favre-Averaged Reynolds stress tensor $\overline{\rho v_i'' v_j''}$. Using the Bussinesq assumption for the Reynolds stress tensor:

$$
\tau_{ij}^{turb} = -\overline{\rho v_i'' v_j''} \approx 2\mu_t\widetilde{S_{ij}^*} - \frac{2}{3}\overline{\rho}k_t\delta_{ij} \tag{3-11}
$$

where μ_t is the turbulent viscosity. The latter is unkown and should me modeled with a *turbulence model.*

Turbulence Model: Spalart-Allmaras

In the framework of this thesis we decided to use the *Spalart-Allmaras model* [3], a one equation linear eddy viscosity model specifically designed for aerospace applications, which solves a transport equation for a viscosity-like variable $\tilde{\nu}$ (Spalart-Allmaras variable):

$$\begin{aligned}
\nu_t = \tilde{\nu} f_{v1}, \qquad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \qquad \chi = \frac{\tilde{\nu}}{\nu} \\
\frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial}{\partial x_j}(\tilde{\nu} v_j) = C_{b1}\left[1 - f_{t2}\right]\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\left[\nabla \cdot \left[(\nu + \tilde{nu})\nabla \tilde{nu}\right] + C_{b2}|\nabla \nu|^2\right] - \\
\left[C_{w1} f_w - \frac{C_{b1}}{k^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + f_{t1}\Delta U^2 \\
\tilde{S} = S + \frac{\tilde{\nu}}{k^2 d^2} f_{v2}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}
\end{aligned} \tag{3-12}$$

where

$$\begin{aligned}
S = \sqrt{2\Omega_{ij}\Omega_{ij}} \\
\Omega_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i}\right) \\
f_w = g\left[\frac{1 + C_{w3}^6}{g_6 + C_{w3}^6}\right]^{1/6}, \qquad g = r + C_{w2}(r^6 - r), \qquad r = \frac{\tilde{nu}}{\tilde{S} k^2 d^2} \\
f_{t1} = C_{t1}\ exp\left(-C_{t2}\frac{\omega_t^2}{\Delta U^2}[d^2 + g_t^2 d_t^2]\right) \qquad (d \ = \ distance\ to\ the\ closest\ surface) \\
f_{t2} = C_{t3}\ exp(-C_{t4}\chi^2)
\end{aligned} \tag{3-13}$$

The constants are:

$$\begin{aligned}
\sigma &= \frac{2}{3} \\
C_{b1} &= 0.1355 \\
C_{b2} &= 0.622 \\
k &= 0.41 \\
C_{w1} &= \frac{C_{b1}}{k^2} + \frac{(1 + C_{b2})}{\sigma} \\
C_{w2} &= 0.3 \\
C_{w3} &= 2 \\
C_{v1} &= 7.1 \\
C_{t1} &= 1 \\
C_{t2} &= 2 \\
C_{t3} &= 1.1 \\
C_{t4} &= 2
\end{aligned} \tag{3-14}$$

Since we are considering a compressible flow, the turbulent viscosity, needed to close the FANS system of equation, is than obtained from the Spalart-Allmaras variable as:

$$\mu_t = \rho \tilde{\nu} f_{v1} \tag{3-15}$$

where ρ is the local density.

3-1-2 Inviscid Compressible Fluid

If we neglect the viscous effects the Navier-Stokes equations (3-1) are simplified and substituted by the Euler equations:

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho v_j\right) &= 0 \\
\frac{\partial}{\partial t}\left(\rho v_i\right) + \frac{\partial}{\partial x_j}\left(\rho v_i v_j + p\delta_{ij}\right) &= 0 \\
\frac{\partial}{\partial t}\left(\rho E\right) + \frac{\partial}{\partial x_j}\left[v_j\left(\rho E + p\right)\right] &= 0
\end{aligned}
\tag{3-16}
$$

where:

$$
E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho |v|^2 \tag{3-17}
$$

The closure of the above mentioned system of equations is obtained with the equation of state (EOS):

$$
p = \rho R T \qquad (Ideal\ gas) \tag{3-18}
$$

The strong conservative form in Eulerian[9] formulation of the Euler equation (3-16) can be written as:

$$
\frac{\partial \boldsymbol{W}}{\partial t} + \vec{\nabla} \cdot \left(\vec{\boldsymbol{\mathcal{F}}}(\boldsymbol{W})\right) = 0 \tag{3-19}
$$

where

$$
\begin{aligned}
\boldsymbol{W} &= \left(\rho, \rho v_x, \rho v_y, \rho v_z, E\right)^T, \\
\vec{\nabla} &= \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^T, \\
\vec{\mathcal{F}}(\boldsymbol{W}) &= \left(\mathcal{F}_x(W), \mathcal{F}_y(W), \mathcal{F}_z(W)\right)^T,
\end{aligned}
\tag{3-20}
$$

$$
\mathcal{F}_x = \begin{pmatrix} \rho v_x \\ p + \rho v_x^2 \\ \rho v_x v_y \\ \rho v_x v_z \\ v_x(E+p) \end{pmatrix}, \quad
\mathcal{F}_y = \begin{pmatrix} \rho v_y \\ \rho v_x v_y \\ p + \rho v_y^2 \\ \rho v_y v_z \\ v_y(E+p) \end{pmatrix}, \quad
\mathcal{F}_z = \begin{pmatrix} \rho v_z \\ \rho v_x v_z \\ \rho v_y v_z \\ p + \rho v_z^2 \\ v_z(E+p) \end{pmatrix}
\tag{3-21}
$$

[9]Eulerian representation : an observer fixed in an inertial reference

3-2 Structure domain

The linear elastodynamic behaviour of the structural domain B (Fig. 3-2) can be described using the following equilibrium and compatibility conditions:

$$\begin{aligned} \frac{\partial}{\partial x_i}\sigma_{ji} + f_j^{ext} - \rho_0 \ddot{u}_j = 0 \quad in\ B(0) \\ t_j = n_i \sigma_{ij} = \bar{t}_j \quad on\ \partial B_t \subset \partial B, \\ \sigma_{ij} = \sigma_{ji} \\ \varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial_j}\right) \\ u_j = \overline{u}_j \quad on\ \partial B_u \subset \partial B \end{aligned} \tag{3-22}$$

where σ_{ji} is the stress tensor, $x_i = [x_1, x_2, x_3]$ the Cartesian coordinates of a given point of the undeformed body, f_j^{ext} applied body forces, $u_i(x_j, t) = [u_1, u_2, u_3]$ the displacement field observed at any point ($\bar{u}_j$ the displacement imposed on the Dirichlet boundary ∂B_u), ρ_0 the mass density of the body, t_j the surface tractions ($\bar{t}_j$ is the surface traction applied on the Neumann boundary ∂B_t), n_i the direction cosines of the outward normal to the surface and ε_{kl} is the strain tensor.

To close the system, a constitutive law that express the stress tensor in terms of displacement and velocity is required. The structures considered in this thesis can be modeled using linear elasticity, as permanent deformations are usually negligible:

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \tag{3-23}$$

where C_{ijkl} are the elastic coefficients of the material.

We rewrite (3-22) in vector form:

$$\begin{aligned} \rho_s \ddot{\boldsymbol{u}}_{\boldsymbol{s}} - \nabla_x \cdot \sigma_s(\boldsymbol{u}_{\boldsymbol{s}}, \dot{\boldsymbol{u}}_{\boldsymbol{s}}) = \boldsymbol{f}_{\boldsymbol{s}}^{\boldsymbol{ext}}, \quad in\ B(0) \\ \sigma_s \boldsymbol{n}^t = \bar{t} \quad on\ \partial B_t \subset \partial B, \\ \boldsymbol{u} = \bar{\boldsymbol{u}} \quad on\ \partial B_u \subset \partial B \end{aligned} \tag{3-24}$$

3-3 Interface conditions

To describe and model the interaction between the external fluid domain Ω and the immersed solid structual domain B a set of interface condition must be enforced on the boundary ∂B.

If the structure or a subset of it is flexible and is immersed in an inviscid fluid, these interface conditions are applied:

$$\boldsymbol{Continuity\ condition:} \quad \dot{\boldsymbol{u}} \cdot \boldsymbol{n} = \overrightarrow{v} \cdot \boldsymbol{n} \qquad on\ \partial B \tag{3-25}$$

$$\boldsymbol{Equilibrium\ condition:} \quad \sigma_s \cdot \boldsymbol{n} = -p\boldsymbol{n} \qquad on\ \partial B \tag{3-26}$$

In the case of a boundary ∂B fixed in time ($\dot{\boldsymbol{u}} \cdot \boldsymbol{n} = 0$) the Continuity condition reduce to the slip-wall boundary condition for an inviscid flow:

$$\overrightarrow{v} \cdot \boldsymbol{n} = 0 \quad on\ \partial B^0 \tag{3-27}$$

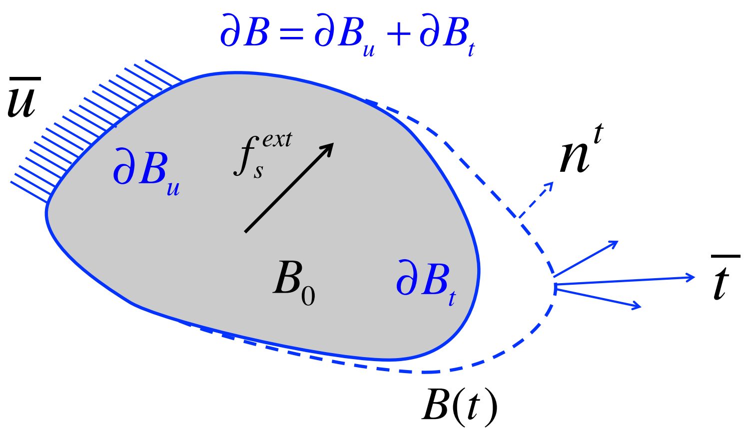


Figure 3-2: The solid structure domain B, its surface boundary ∂B is decomposed in ∂B_u where the displacement $\bar{\boldsymbol{u}}$ is imposed and ∂B_t where the surface traction $\bar{t}$ is applied.

Chapter 4

Embedded Computational Framework

The main components of the Embedded Boundary Methods (EBM) presented in this thesis are: a computational algorithms for tracking the fluid-structure interfaces with respect to the fixed (or moving/deforming) non body-conforming CFD grid, an algorithm based on the exact solution of local 1D fluid-structure Riemann problems to enforce the continuity interface condition at the fluid-structure interface and a consistent and conservative algorithm to enforce the equilibrium transmission condition and compute the loads at the interface.

First we will introduce the Computational Structural Dynamics (CSD) solver (4-1) needed to compute the displacement and velocity of the fluid structure interface, than the CFD solver (3-1), equipped with a tracking algorithm (4-2), the local 1D Riemann solver and a load computation algorithm (4-4), will be presented. At the end of the chapter the complete algorithm will be discussed and explained.

4-1 Structure FE solver

Under the assumption that the displacement field of the structure is assumed to be continuous in space and time the equations for the structural domain B are discretized using finite element method. Considering Eq. (3-24), multiplying by a test function and integrating by part, we obtain a weak formulation for the structural problem:

$$\begin{aligned} &Find\ \boldsymbol{u}(x,t) \in \mathbb{U}\ = \{\boldsymbol{u}(x,t) | \boldsymbol{u}(x,t) \in \mathbb{H}^1(B), \boldsymbol{u} = \bar{\boldsymbol{u}}\ on\ \partial B_u\}\ \ such\ that \\ &\forall \delta\boldsymbol{u}(x) \in \mathbb{U}_0 = \{\boldsymbol{v}(x,t) | \boldsymbol{v}(x,t) \in \mathbb{H}^1(B), \boldsymbol{v} = 0\ on\ \partial B_u\},\ it\ satisfies \\ &\int_B \delta\boldsymbol{u} \cdot \rho_s \ddot{\boldsymbol{u}}_{\boldsymbol{s}}\ dV - \int_B \nabla\delta\boldsymbol{u} \cdot \sigma_s(\boldsymbol{u}_{\boldsymbol{s}}, \dot{\boldsymbol{u}}_{\boldsymbol{s}})\ dV = \int_B \delta\boldsymbol{u} \cdot \boldsymbol{f}_{\boldsymbol{s}}^{\boldsymbol{ext}}\ dV + \int_{\partial B_t} \delta\boldsymbol{u} \cdot \bar{\boldsymbol{t}}\ dS \end{aligned} \tag{4-1}$$

The structural domain B is discretized into non-overlapping elements $e_i,\ i = 1, ..., N_e$.

The approximate solution of (4-1) is found in a finite-dimensional subspace of $\mathbb{U}$:

$$\bar{\mathbb{U}} = \{\boldsymbol{u}(x,t) = \sum_{I=1}^{N_n} N_I(x) u_I(t), \boldsymbol{u}(x,t) = \bar{\boldsymbol{u}}\ on\ \partial B_u\} \tag{4-2}$$

and the test functions in a subspace of $\mathbb{U}_0$:

$$\bar{\mathbb{U}}_0 = \{\boldsymbol{v}(x,t) = \sum_{I=1}^{N_n} N_I(x) v_I(t), \boldsymbol{v}(x,t) = 0 \ on \ \partial B_u\} \tag{4-3}$$

where N_n is the number of nodes and $N_I(x) \in \mathbb{H}^1(B)$ with $I = 1, ..., N_n$ is the set of prescribed shape functions.

By plugging $\boldsymbol{u}(x,t) = \sum_{I=1}^{N_n} N_I(x) u_I(t)$ and $\delta\boldsymbol{u}(x) = N_I(x),\ I = 1, ..., N_n$ in (4-1), it can be shown that the nodal displacement vector $\boldsymbol{u} = [u_1(t), ..., u_{N_n}]^T$ satisfies a nonlinear system that can be written as:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{f^{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f^{ext}} \tag{4-4}$$

where $\boldsymbol{M}$ denotes the symmetric positive definite FE mass matrix, $\boldsymbol{f^{int}}$ the vector of internal forces and $\boldsymbol{f^{ext}}$ the vector of external forces[1].

The nonlinear system (4-4) can be linearized in the well known form:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{4-5}$$

where $\boldsymbol{C}$ and $\boldsymbol{K}$ respectively represent the damping and stiffness matrix.

To advance the semi-discretized system (4-5) in time from t^n to t^{n+1} the family of Newmark schemes is employed:

$$\begin{aligned}(\boldsymbol{M} + \gamma\Delta t^n \boldsymbol{C} + \beta(\Delta t^n)^2 \boldsymbol{K})\ddot{\boldsymbol{u}}^{n+1} &= \boldsymbol{f}^{n+1} - \boldsymbol{C}(\dot{\boldsymbol{u}} + (1-\gamma)\Delta t^n \ddot{\boldsymbol{u}}^n) \\ &\quad -\boldsymbol{K}(\boldsymbol{u}^n + \Delta t^n \dot{\boldsymbol{u}}^n + (0.5-\beta)(\Delta t^n)^2 \ddot{\boldsymbol{u}}^n) \\ \dot{\boldsymbol{u}}^{\boldsymbol{n+1}} &= \dot{\boldsymbol{u}}^n + (1-\gamma)\Delta t^n \ddot{\boldsymbol{u}}^n + \gamma\Delta t^n \ddot{\boldsymbol{u}}^{n+1} \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t^n \dot{\boldsymbol{u}}^n + (\Delta t^n)^2 (0.5-\beta)\ddot{\boldsymbol{u}}^n + (\Delta t^n)^2 \beta \ddot{\boldsymbol{u}}^{n+1}\end{aligned} \tag{4-6}$$

Depending on the choice of two parameters α and β, the integration scheme can be explicit or implicit. In the following table the most common integration schemes are presented (4-1).

The purely explicit scheme is not used in practice, since it always unstable. The cental difference (explicit) scheme is conditionally stable and the time-step required for using this scheme is very small, however it is widely used for problems where a small time step is also required to solve for high frequency phenomenon such as wave propagation and impact response. The Fox and Goodwin algorithm with a periodicity error of the third order is also conditionally stable. The average constant acceleration algorithm is an unconditionally stable scheme (no limitation to the time step) and has the lowest periodicity error. The modified average constant acceleration algorithm is also unconditionally stable and introduce numerical damping to modes of higher frequencies.

[1] We can also write that $\boldsymbol{f^{ext}} = \boldsymbol{f^F}(\boldsymbol{W}) + \boldsymbol{f^{EXT}}$ and define $\boldsymbol{f^F}(\boldsymbol{W})$ as the vector of flow-induced forces and $\boldsymbol{f^{EXT}}$ as the vector of external forces that are not related with the flow interaction.

Algorithm	γ	β	Stability limit ωh	Amplitude error $\rho - 1$	Periodicity error $\frac{\Delta T}{T}$
Purely explicit	0	0	0	$\dfrac{\omega^2 h^2}{4}$	—
Central difference	$\frac{1}{2}$	0	2	0	$\dfrac{-\omega^2 h^2}{24}$
Fox & Goodwin	$\frac{1}{2}$	$\frac{1}{12}$	2.45	0	$O(h^3)$
Linear acceleration	$\frac{1}{2}$	$\frac{1}{6}$	3.46	0	$\dfrac{\omega^2 h^2}{24}$
Average constant acceleration	$\frac{1}{2}$	$\frac{1}{4}$	∞	0	$\dfrac{\omega^2 h^2}{12}$
Average constant acceleration (modified)	$\frac{1}{2} + \alpha$	$\frac{(1+\alpha)^2}{4}$	∞	$-\alpha\dfrac{\omega^2 h^2}{2}$	$\dfrac{\omega^2 h^2}{12}$

Figure 4-1: Integration schemes of the Newmark family. h is the integration step, ω is the highest frequency contained in the model, ρ is the amplitude of the response obtained after one step of the time integrator and $\rho - 1$ is the amplitude error, T is the period of the free oscillation, T_{com} is the period of the computed response, the periodicity error is than $\frac{T_{com}-T}{T} = \frac{\Delta T}{T}$

4-2 Track the embedded surface

Let D_h denote the discretization (*primal mesh*) of the extended fluid domain $\Omega^* = \Omega \cup B$ by an arbitrary non body-fitted unstructured triangular Eulerian grid and D_h^E the discretization of the fluid-structure interface (does not need to coincide with the wet subset of the discretization of ∂B). In an embedded boundary method the Eulerian grid D_h does not contain the representation of the fluid-structure interface (Fig. 4-2). Two algorithms are presented in this section to track the position of the discrete embedded interface D_h^E and collect the following information that are usually directly available in a body-fitted grid:

- status of each grid point in D_h (grid point belongs to the fluid domain in red or structural domain in black in Fig.4-2);
- location of the closest points of D_h^E to a selected set of points in D_h;
- intersection of the edges of D_h with D_h^E (in green in Fig.4-2)

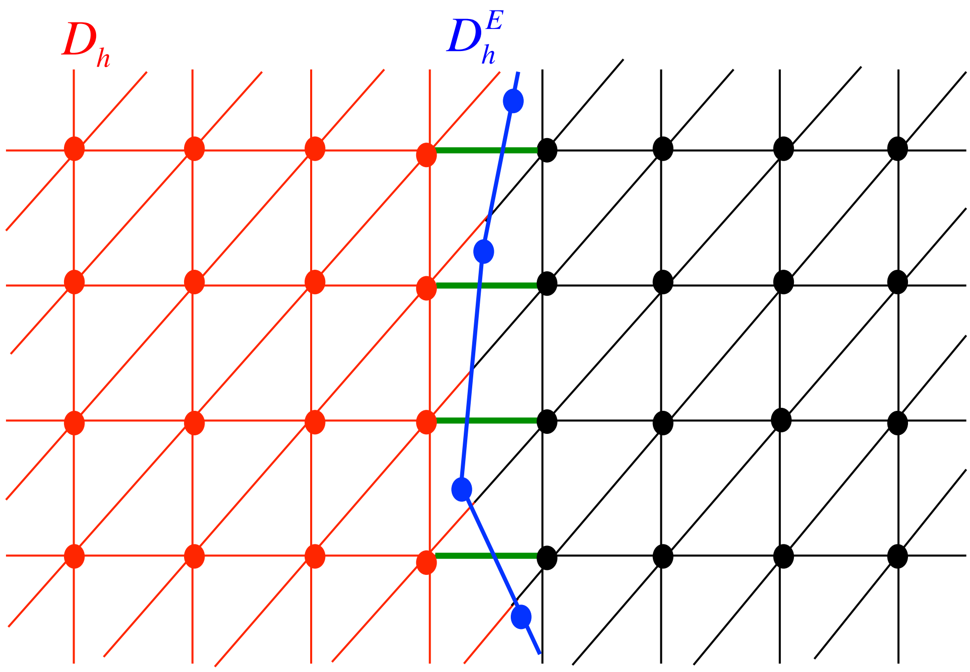


Figure 4-2: Discretization by an arbitrary non body-fitted Eulerian triangular mesh D_h and embedded surface discretization D_h^E.

4-2-1 FRG - Projection-based approach

The first interface tracking algorithm presented is based on the following assumptions:

- only the edges of the Eulerian grid D_h connecting a node in the fluid domain Ω and a node in the structural domain B intersect the embedded discrete surface D_h^E,
- D_h^E has an interior and an exterior $\Longrightarrow$ the embedded interface is a closed surface,
- the (interior) volume enclosed by D_h^E is resolved by the Eulerian CFD grid D_h [2].

PROJECTION-BASED ALGORITHM

1. $\forall\ V_i \in D_h$ construct an axis-aligned bounding box b_i defined as the smallest axis-aligned box containing V_i and its adjacent nodes.
2. Construct an axis-aligned bounding box hierarchy B_E which stores the triangles of D_h^E.
3. $\forall\ V_i \in D_h$, determine if it lies close to D_h^E, and if it does, find the location of its closest point on D_h^E and determine the status s_i of V_i.
   (a) status of V_i unknown $\Longrightarrow s_i = -1$
   (b) using B_E find the set of candidate triangles $\mathcal{C}(V_i) \subset D_h^E$ that may contain s the closest point to V_i denoted V_i'
   (c) if $\mathcal{C}(V_i) \neq \emptyset$, find the location of V_i' and compute the signed distance $\phi(V_i', V_i)$ from V_i' to V_i.
      i. If $\phi(V_i', V_i) > 0$, V_i is inside the fluid domain $\Omega \Longrightarrow s_i = 0$
      ii. If $\phi(V_i', V_i) \leq 0$, V_i is inside the structure domain $B \Longrightarrow s_i = 1$
4. Determine the status of all remaining grid points using a "flood-fill" algorithm: $\forall\ V_i \in D_h$, if $s_i \neq -1$, loop through its adjacent nodes $N(V_i)$, $\forall\ V_i \in N(V_i)$, if $s_k = -1$, set $s_k = s_i$.
5. Compute the intersections between the edges of D_h and the elements of D_h^E: for each edge (V_i, V_j), if $s_i \neq s_j$, identify this edge as fluid-structure intersecting edge. Than cast a ray form V_i and V_j and find the intersection point of D_h^E.

The critical components of the Projection-based algorithm presented above are:

- **Search the closest $V_i' \in V_i^E$ to a given grid point V_i**
  The closest point to V_i lying in each triangle $T_k \in \mathcal{C}(V_i)$ denoted $V_i^{(k)}$ is first determined. V_i is projected onto the plane containing T_k and the projection point is uniquely determined by its barycentric coordinates (ξ_A, ξ_B, ξ_C) with respect to T_k (A, B, C denote the vertices of T_k) (Fig.4-3). If all the coordinates are non-negative, the projection point is $V_i^{(k)}$. If one or two coordinates are negative, the projection point lies outside T_k; $V_i^{(k)}$ is located either on the line containing an edge of T_k or at one of its vertices. V_i is reprojected in this case onto each edge of T_k corresponding to a negative coordinate. If the projection point ends up on an edge of T_k, it is $V_i^{(k)}$. Otherwise, the vertex of T_k that has the maximum distance to V_i is $V_i^{(k)}$. Finally, $V_i' \in D_h^E$ is identified as:

$$V_i' = arg \min_{T_k \in\ \mathcal{C}(V_i)} ||V_i - V_i^{(k)}||_2 \tag{4-7}$$

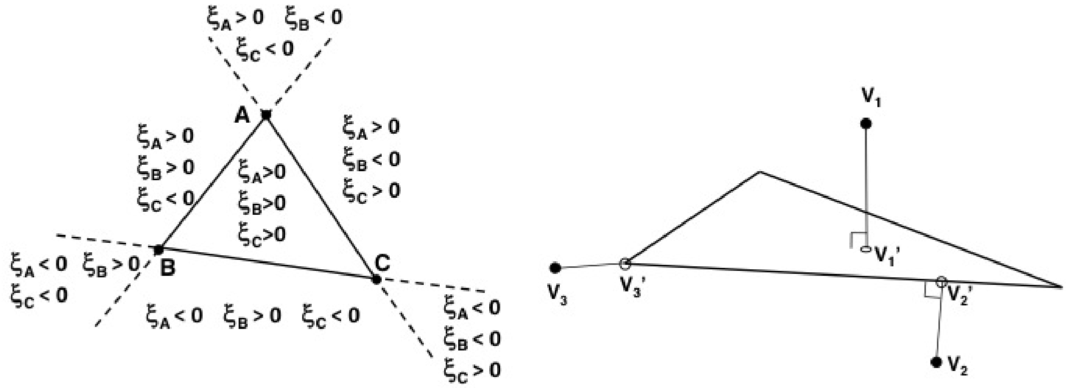


Figure 4-3: Signs of the barycentric coordinates of the projection point in different regions. discretization D_h^E.

- **Calculation of the signed distance $\phi(V_i', V_i)$**
  Once V_i' has been determined, the magnitude of $\phi(V_i', V_i)$ is simply $||V_i - V_i^{(k)}||_2$. To determine the sign of this distance, three different cases must be treated separately:

  - V_i' is inside a triangle $T_k \in \mathcal{C}(V_i) \implies sign(\phi(V_i', V_i))$ is determined as the sign of the dot product $\overrightarrow{n}_k \cdot \overrightarrow{V_i'V_i}$, where $\overrightarrow{n_k}$ is the unit outward normal to T_k and $\overrightarrow{V_i'V_i}$ is the spatial vector connecting V_i' and V_i.
  - V_i' is not inside a triangle $T_k \in \mathcal{C}(V_i)$ but is the projection of V_i onto an edge the triangle $\implies sign(\phi(V_i', V_i))$ is determined using the information from the two triangles of D_h^E sharing this this edge. First V_i is projected onto the plane determined by one of the two triangles that is not coplanar with it (Fig. 4-4). The following quantities are introduced:

$$s_v = \overrightarrow{P_{V_i}V_i} \cdot \overrightarrow{n}_r; \qquad s_c = \overrightarrow{P_C C} \cdot \overrightarrow{n}_r \tag{4-8}$$

    * If $s_v s_c > 0$ (Fig. 4-4 left) $\implies sign(\phi(V_i', V_i))$ is determined as $sign(-s_v)$
    * If $s_v s_c < 0$ (Fig. 4-4 right) $\implies sign(\phi(V_i', V_i))$ is determined as $sign(s_v)$

  - V_i' is a vertex of $\mathcal{C}(V_i)$ (Fig.4-5). The set of triangles adjacent to V_i' is denoted by $\mathcal{N}_i$ and away from V_i', these triangles form and infinite open surface $\tilde{\mathcal{N}_i}$. Considering the plane crossing V_i and orthogonal to $\overrightarrow{V_i'V_i}$, for any point on this plane sufficiently far from V_i, its closest point on $\tilde{\mathcal{N}_i}$ is either on a face or on an edge. Hence this point $\tilde{V_i}$, falls into the above mentioned cases, therefore $sign(\phi(V_i', \tilde{V_i}))$ can be computed as explained above and the $sign(\phi(V_i', V_i))$ is determined as the $sign(\phi(V_i', \tilde{V_i}))$.

[2] This means that a membrane with a thickness that is smaller than the CFD grid element cannot be solved with this approach.

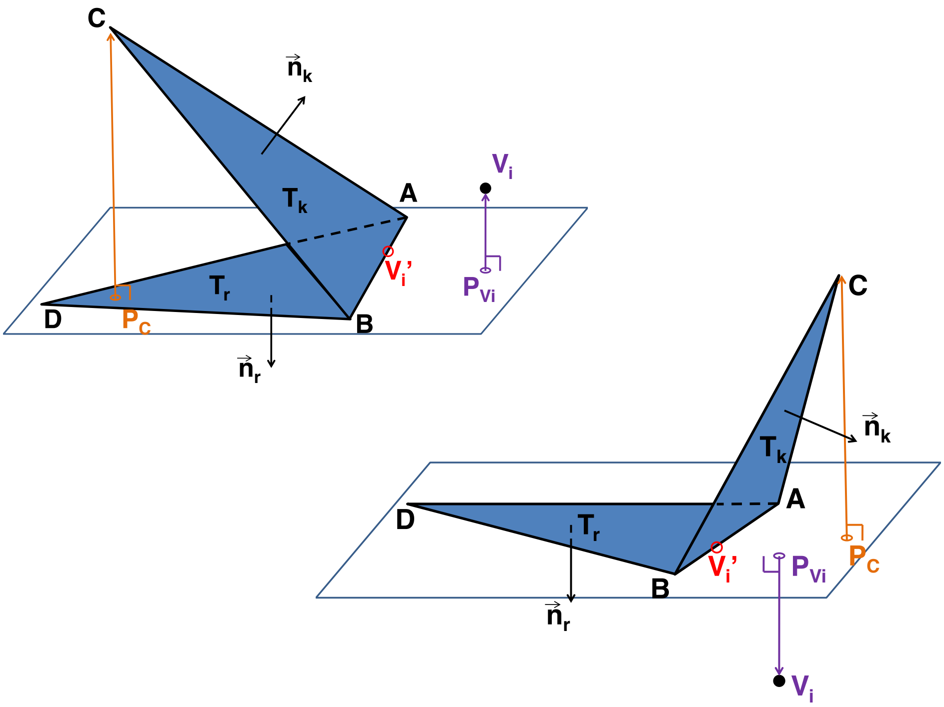


Figure 4-4: Determination of the signed distance $sign(\phi(V_i', V_i))$ when V_i' lies on an edge of a triangle.

4-2-2 PhysBAM - Collision-based approach

When the embedded discrete surface D_h^E is an open surface or a closed surface and the volume it encloses is not resolved (or partially resolved) by a fluid grid, the previous projection-based approach should be substituted by a more sophisticated point-simplex collision based algorithm.

COLLISION-BASED ALGORITHM

1. $\forall\ V_i \in D_h$ construct an axis-aligned bounding box b_i
2. Construct an axis-aligned bounding box hierarchy B_E which stores the triangles of D_h^E.
3. Using B_E find $\forall\ V_i \in D_h$ the set of triangles $\mathcal{C}(V_i) \subset D_h^E$ whose bounding boxes intersect b_i
   (a) Thicken each triangle $T_k \in \mathcal{C}(V_i)$ by a numerical tollernace ε. If V_i lies inside the thickened wedges $\Longrightarrow$ flag it as occluded.
   (b) if $\mathcal{C}(V_i) \neq \emptyset$, find the location of V_i' and compute the signed distance $\phi(V_i', V_i)$ from V_i' to V_i.

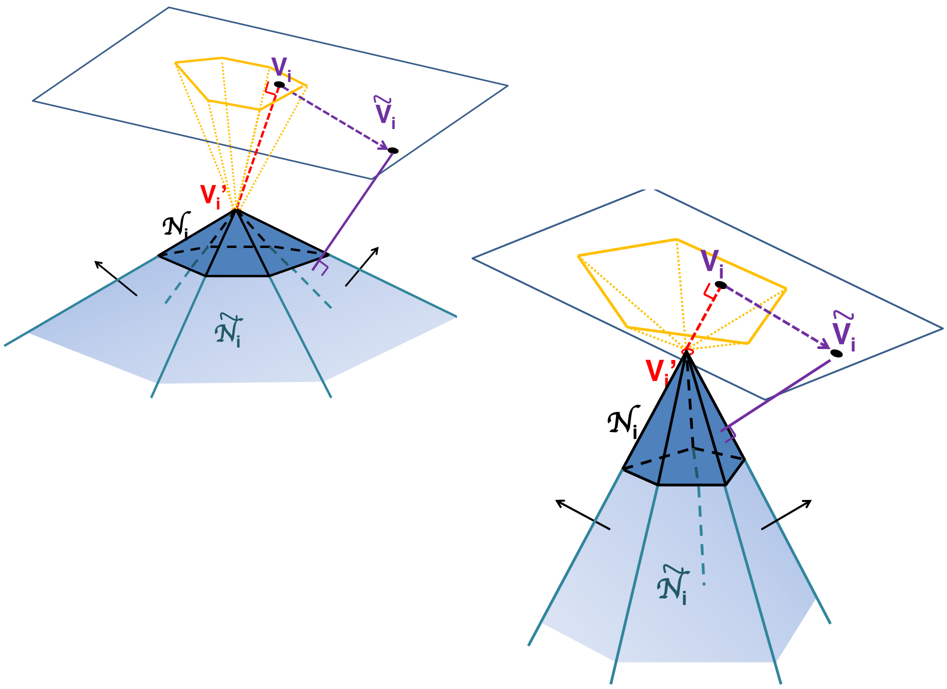


Figure 4-5: Determination of the signed distance $sign(\phi(V_i', V_i))$ when V_i' is the vertex of a triangle.

i. If $\phi(V_i', V_i) > 0$, V_i is inside the fluid domain $\Omega \implies s_i = 0$

ii. If $\phi(V_i', V_i) \leq 0$, V_i is inside the structure domain $B \implies s_i = 1$

4. For every edge (V_i, V_j), cast the rays r_{ij} from V_i to V_j and r_{ji} from V_j to V_i against the triangles in $\mathcal{C}(V_i) \cap \mathcal{C}(V_j)$ using the point-simplex algorithm:

   (a) If both r_{ij} and r_{ji} intersect a triangle in $D_h^E \implies$ classify (V_i, V_j) as a fluid-structure intersecting edge and store the intersection points.

   (b) If either V_i or V_j is occluded $\implies$ classify (V_i, V_j) as a fluid-structure intersecting edge and store the occluded node as the intersection point.

5. Determine the node status n_i using geometric means and its value at the previous time-step. If $\mathcal{C}(V_i) = \emptyset$ then keep n_i unchanged.

   (a) For every triangle $T_k \in \mathcal{C}(V_i)$ use the point-simplex algorithm to determine if T_k crosses over V_i during the given time-step. The point is fixed in space and the simplex travels from its position at t^n to its position at t^{n+1}. If any simplex crosses over $V_i \implies$ set $n_i = -1$ indicating that the status may have changed and therefore must be redetermined.

(b) $\forall\ V_i \in \{V_i \in D_h\ : n_i = -1\}$, search for adjacent nodes V_j with $n_j \neq -1$. If such node is found $\Longrightarrow n_i = n_j$ and repeat this procedure until the status of every node has been determined or no further updates are possible. Flag any remaining node V_i with status $n_i = -1$ as being inside the structural domain B.

The main differences between the two algorithms presented above can be observed by analyzing the three cases illustrated in Fig. 4-6.

- **CASE I**: the discrete embedded surface is a closed surface and the enclosed volume is resolved by the Eulerian grid $\Longrightarrow$ both algorithms give the same results.
- **CASE II**: the discrete embedded surface is a closed surface but the enclosed volume is not fully resolved by the Eulerian grid (similar to the trailing edge of an airfoil) $\Longrightarrow$ projection-based algorithm misses two edges that intersect the embedded interface twice; collision-based algorithm is able to detect the double intersection.
- **CASE III**: the discrete embedded surface is an open surface $\Longrightarrow$ projection-based algorithm fails to detect the intersections; collision-based algorithm is able to detect the intersections.

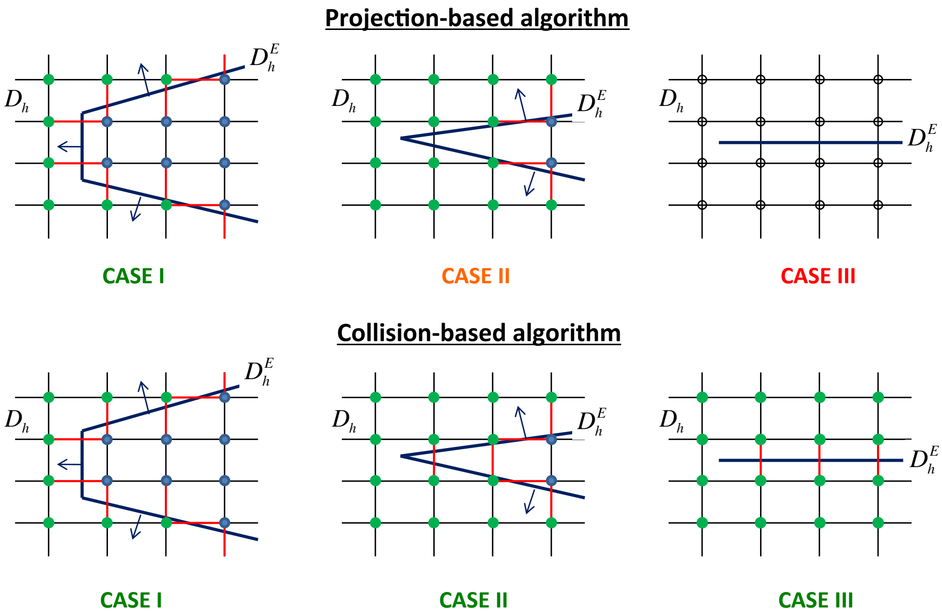


Figure 4-6: Illustration of the results obtained with the projection-based algorithm and the collision-based algorithm for three different cases.

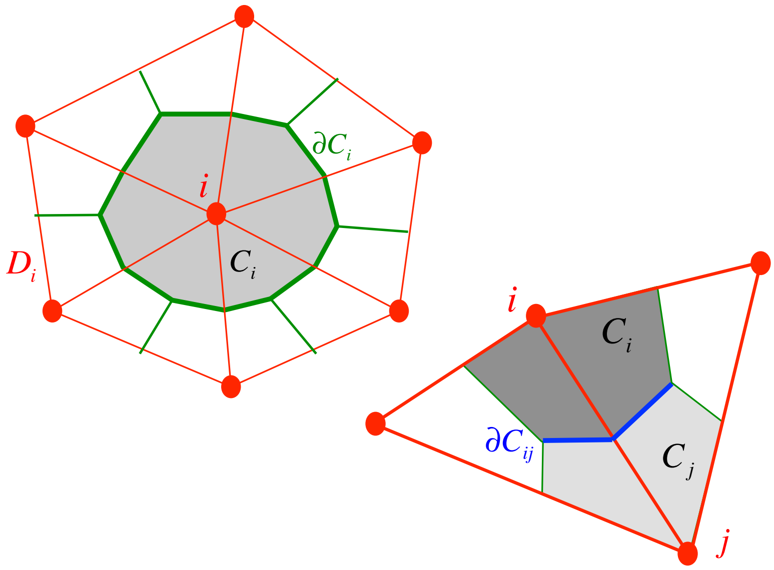


Figure 4-7: Primal mesh D_i in red and the dual cell C_i in grey with boundary surface ∂C_i. On the right the details of the interface between two neighboring dual cells.

4-3 Finite Volume Solver

Depending on the information obtained from the tracking algorithms presented above we need to solve a fluid-fluid problem if both the nodes belong to the fluid domain Ω or a fluid-structure problem if the edge $(V_i, V_j) \in D_h$ intersects the embedded interface D_h^E.

In order to apply Finite Volume variational-like formulation, a cell-centered mesh D_h^* (*dual mesh*) is built over the primal mesh D_h. For a node i, we denote by D_i the set of triangles (or tetrahedra in 3D) that contains node i in the primal mesh D_h, and by C_i the cell associated to i with boundary surface ∂C_i (Fig. 4-7).

Using the standard characteristic function associated with the control volume C_i, a standard variational approach and integration by parts we can rewrite the Euler equation (3-16) as:

$$\begin{aligned} &\int_{C_i} \frac{\partial \boldsymbol{W_h}}{\partial t}\, dV \\ &\qquad + \sum_{j \in K(i)} \int_{\partial C_{ij}} \vec{\boldsymbol{\mathcal{F}}}(\boldsymbol{W_h}) \cdot \overrightarrow{\boldsymbol{n_{ij}}}\, dS \qquad < 1 > \\ &\qquad + \int_{\partial C_i \cap D_h^E} \vec{\boldsymbol{\mathcal{F}}}(\boldsymbol{W_h}) \cdot \overrightarrow{\boldsymbol{n^E}}\, dS \qquad < 2 > \\ &\qquad = 0 \end{aligned} \tag{4-9}$$

$\boldsymbol{W_h}$ represents the discrete approximation of the state vector $\boldsymbol{W}$, $K(i)$ is the set of adjacent vertices of V_i, ∂C_{ij} is the segment of ∂C_i (defined as $\partial C_{ij} = \partial C_i \cap \partial C_j$ (Fig. 4-7)), $\overrightarrow{\boldsymbol{n_{ij}}}$ is the unit outward normal to ∂C_{ij} and $\overrightarrow{\boldsymbol{n^E}}$ is the unit outward normal to D_h^E.

The computation of the previous equation can be performed in one-dimensional fashion by evaluating the fluxes along normal directions to the boundaries of the control volumes.

We can express the first flux $< 1 >$in (4-9) as:

$$\begin{aligned} \sum_{j\in K(i)} \int_{\partial C_{ij}} \overrightarrow{\boldsymbol{\mathcal{F}}}(\boldsymbol{W_h}) \cdot \overrightarrow{\boldsymbol{n_{ij}}}\; dS &= \sum_{j\in K(i)} \Phi_{\mathcal{F}_{ij}}(W_i, W_j, \overrightarrow{\boldsymbol{n_{ij}}}) \\ with \qquad W_i = \frac{1}{|C_i|}\int_{\partial C_i} \boldsymbol{W_h}\; dV, &\qquad W_j = \frac{1}{|C_j|}\int_{\partial C_j} \boldsymbol{W_h}\; dV \end{aligned} \tag{4-10}$$

with $\Phi_{\mathcal{F}_{ij}}$ we indicate the numerical flux evaluated for example with a first-order upwind scheme as Roe's approximate Riemann solver [3]:

$$\Phi_{\mathcal{F}_{ij}}(W_i, W_j, \overrightarrow{\boldsymbol{n_{ij}}}) = \frac{1}{2}\left(\overrightarrow{\boldsymbol{\mathcal{F}}}(W_i) + \overrightarrow{\boldsymbol{\mathcal{F}}}(W_j)\right) - \frac{|\overrightarrow{\boldsymbol{\lambda}}(W_i, W_j) \cdot \overrightarrow{\boldsymbol{n_{ij}}}|}{2}\,(W_j - W_i) \tag{4-11}$$

$\overrightarrow{\boldsymbol{\lambda}}$ represents the vector of Jacobian matrices defined as:

$$\overrightarrow{\boldsymbol{\lambda}} = \left(\frac{\partial \mathcal{F}_x}{\partial \boldsymbol{W}}, \frac{\partial \mathcal{F}_y}{\partial \boldsymbol{W}}, \frac{\partial \mathcal{F}_z}{\partial \boldsymbol{W}}\right)^T \tag{4-12}$$

The computation of the second flux $< 2 >$ in (4-9) involve the continuity transmission condition (3-25) in the case of dynamic or flexible embedded interface or the slip boundary condition (3-27) in the case of static embedded interface.

We indicate here with L and R the grid points, in the Eulerian grid, on the left and on the right of the embedded discrete interface D_h^E. V_i^L and V_i^R respectively denote the the first-layer grid point on the left and on the right of D_h^E connected via V_i^{LR}. M_i^{LR} is the point where the edge V_i^{LR} intersect the control volume boundary $\partial C_i^{LR} = C_i^L \cap C_i^R$ (Fig. 4-8).

For the sake of explanation we assume now that the flow occurs only on one side of D_h^E (left in this case).

<u>ALGORITHM</u>

For each C_i^L and V_i^{LR} intersecting D_h^E

1. If the considered region of D_h^E correspond to a dynamic embedded interface, compute the velocity $\dot{\boldsymbol{u}}_M = \dot{\boldsymbol{u}}(M_i^{LR})$ of D_h^E at M_i^{LR} by interpolation, extrapolation or (combination of both) of the discrete velocity field $\dot{\boldsymbol{u}}$ obtained from the solution of the discrete structural equations (4-5). If D_h^E correspond to a static embedded interface $\dot{\boldsymbol{u}} \cdot \overrightarrow{\boldsymbol{n_M^E}} = 0$ with $\overrightarrow{\boldsymbol{n_M^E}} = \overrightarrow{\boldsymbol{n^E}}(M_i^{LR})$.

[3]For second order MUSCL (Monotonic Upwind Scheme Conservation Law) scheme, the linearly reconstructed states at the mid-edges are used instead of W_i and W_j.

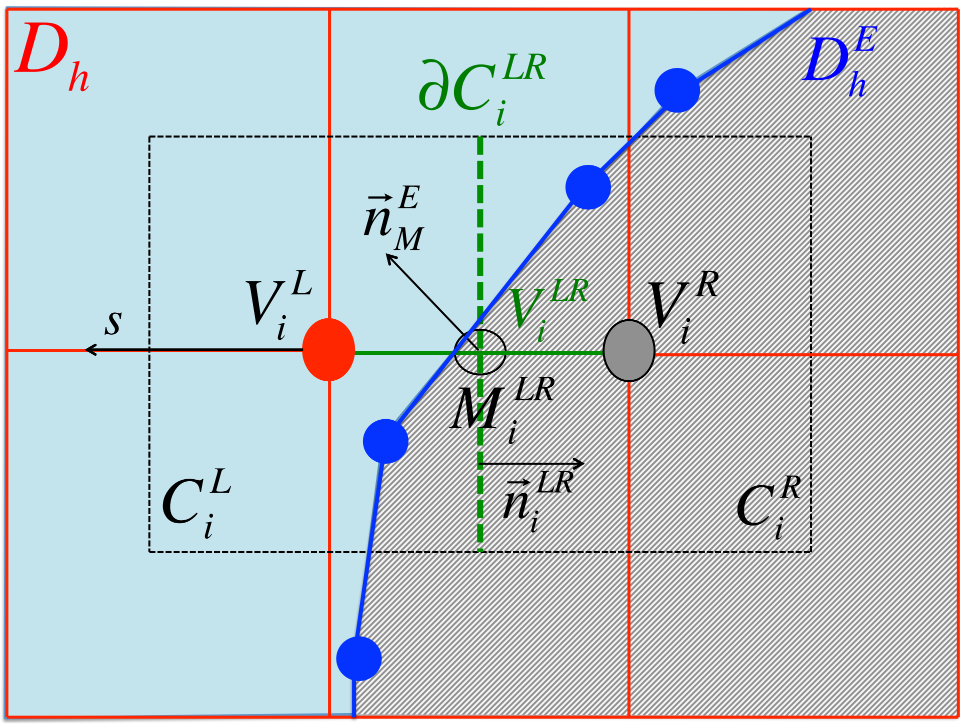


Figure 4-8: Illustration of two control volumes on the left and on the right of the discrete .

2. Assume that at the point M_i^{LR} the boundary facet ∂C_i^{LR} and D_h^E coincide[4] and compute at this point the exact solution $\boldsymbol{W}_M^L$ of the one-sided Riemann problem:

$$\begin{cases} \frac{\partial \boldsymbol{W}}{\partial t} + \frac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}(\boldsymbol{W})}{\partial s} = 0 & \\ \boldsymbol{W}(s,0) = \boldsymbol{W}_i^L & if\ s \geq 0 \\ v(\dot{\boldsymbol{u}}_{\boldsymbol{M}} \cdot \overrightarrow{\boldsymbol{n}_{\boldsymbol{M}}^{\boldsymbol{E}}} t, t) = \dot{\boldsymbol{u}}_{\boldsymbol{M}} \cdot \overrightarrow{\boldsymbol{n}_{\boldsymbol{M}}^{\boldsymbol{E}}} & \forall 0 \leq t \leq \Delta t \end{cases} \tag{4-13}$$

where s is the abscissa along a local axis whose origin is M_i^{LR} and runs in the opposite direction of $\overrightarrow{\boldsymbol{n}_{\boldsymbol{i}}^{\boldsymbol{LR}}}$. We denote with $v(\dot{\boldsymbol{u}}_{\boldsymbol{M}} \cdot \overrightarrow{\boldsymbol{n}_{\boldsymbol{M}}^{\boldsymbol{E}}} t, t)$ the instantaneous velocity of the fluid at $s = 0$. The normal component of the structural velocity at M_i^{LR} denoted as $\dot{\boldsymbol{u}}_{\boldsymbol{M}} \cdot \overrightarrow{\boldsymbol{n}_{\boldsymbol{M}}^{\boldsymbol{E}}}$ is assumed to be constant for $0 \leq t \leq \Delta t$.

3. Evaluate each component of term $< 1 >$ in (4-9) as:

$$\int_{\partial C_i^{LR}} \overrightarrow{\boldsymbol{\mathcal{F}}}(\boldsymbol{W_h}) \cdot \overrightarrow{\boldsymbol{n}_{\boldsymbol{i}}^{\boldsymbol{LR}}}\, dS = \Phi_{\mathcal{F}_i^{LR}}(W_i^L, W_M^L, \overrightarrow{\boldsymbol{n}_{\boldsymbol{i}}^{\boldsymbol{LR}}}) \tag{4-14}$$

If the flow occurs on both sides of the discrete embedded surface D_h^E the above mentioned algorithm is applied twice:

1. for C_i^L and $V_i^{LR} \Longrightarrow \Phi_{\mathcal{F}_i^{LR}}(W_i^L, W_M^L, \overrightarrow{\boldsymbol{n}_{\boldsymbol{i}}^{\boldsymbol{LR}}})$
2. for C_i^R and $V_i^{RL} \Longrightarrow \Phi_{\mathcal{F}_i^{RL}}(W_i^R, W_M^R, \overrightarrow{\boldsymbol{n}_{\boldsymbol{i}}^{\boldsymbol{RL}}})$.

4-3-1 Enforce the continuity condition: Fluid-structure Riemann problem

The system (4-13) is a left fluid-structure Riemann problem (also known as the piston problem) that considers a 1D inviscid compressible flow with a moving wall. In this thesis it is used to enforce the continutiy condition (3-25) and recover the fluid pressure at the fluid structure interface. Given the fluid state $\boldsymbol{W}_L$ and the wall velocity $v_w = \dot{\boldsymbol{u}}_{\boldsymbol{M}} \cdot \overrightarrow{\boldsymbol{n}_{\boldsymbol{M}}^{\boldsymbol{E}}}$, the fluid-structure Riemann problem is defined as follows: considering the 1D time-dependent fluid domain $\Omega(t) = (-\infty, v_w t]$, find the solution $\boldsymbol{W}(x,t)$ to the 1D Euler equations:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \frac{\partial \overrightarrow{\boldsymbol{\mathcal{F}}}(\boldsymbol{W})}{\partial x} = 0 \tag{4-15}$$

$$\boldsymbol{W} = \begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix}, \quad \boldsymbol{\mathcal{F}}(\boldsymbol{W}) = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}$$

with initial condition:

$$\begin{aligned} \boldsymbol{W}(x,0) = \boldsymbol{W}_L, &\quad for \quad x \in \Omega(0) = (-\infty, 0] \\ v(v_w t, t) = v_w, &\quad at\ the\ wall,\ for\ t \geq 0 \end{aligned} \tag{4-16}$$

[4]This assumption introduces a first-order spatial error in the computation of the numerical flux across the embedded surface

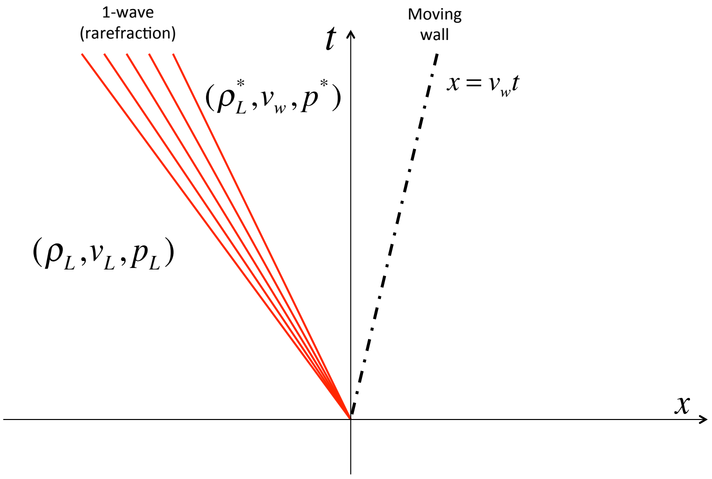


Figure 4-9: A typical solution structure of a fluid-structure Riemann problem with a left rarefraction wave (red).

The solution of the problem consist of a shock wave or rarefaction wave (1-wave) moving from the origin and propagating to the left. At $t \geq 0$ the flow to the left of the 1-wave is unperturbed and the flow between 1-wave and the wall is determined by the intermediate state (ρ_L^*, v_w, p^*). The solution of the problem is obtained by:

- determine if 1-wave is rarefraction or shock
- relate the intermediate state (ρ_L^*, v_w, p^*) to $\boldsymbol{W}_L$ (Fig. 4-9) using the Riemann invariants (if it is a rarefraction) or using the Rankine-Hugoniot jump condition (if it is a shock).
- solve the system obtained at the previous step for ρ_L^* and p^*.
- determine the shock speed using the Rankine-Hugoniot jump conditions or the structure of the solution through the rarefraction waves using the Riemann invariants.

Time integration

We can rewrite in compact form the semi-discretized Euler equation for the discretization D_h of the fluid domain Ω as:

$$\frac{d\boldsymbol{W}}{dt} + \boldsymbol{F}(\boldsymbol{W}) = 0 \tag{4-17}$$

where $\boldsymbol{W}$ and $\boldsymbol{F}$ indicate respectively the cell-averaged state vector and numerical flux vector.

To advance the semi-discretized system (4-17) in time from t^n to t^{n+1}, the *Second-order implicit three point backward difference* scheme is employed:

$$\frac{3\boldsymbol{W}^{n+1} - 4\boldsymbol{W}^n + \boldsymbol{W}^{n-1}}{2} = \Delta t^n \boldsymbol{F}(\boldsymbol{W}^{n+1}) \tag{4-18}$$

4-3-2 Viscous Term

As already mentioned in the previous chapter, to account for the viscous effects, the Euler equations (3-16) are substituted with the Navier-Stokes equations (3-1). The viscous terms, in contrast with the inviscid term [5], is discretized on the primal mesh using piecewise linear continuous representation of the numerical solution. In other words the contribution of the viscous terms to the scheme is given by a Galerkin variational formulation over the elements of the primal mesh as:

$$V_i = -\int_{D_i} \mathbb{K} \cdot \overrightarrow{\nabla \boldsymbol{W}} \cdot \overrightarrow{\nabla \phi_i}\ d\boldsymbol{x}, \tag{4-19}$$

$\mathbb{K}$ is the fourth order diffusive tensor such that

$$\overrightarrow{\boldsymbol{\mathcal{F}}^V}(\boldsymbol{W}, \overrightarrow{\nabla \boldsymbol{W}}) = \mathbb{K} \cdot \overrightarrow{\nabla \boldsymbol{W}} \tag{4-20}$$

and ϕ_i is a Lagrangian basis function at node i.

By adding the viscous contribution (4-19) to (4-9) we can write:

$$\frac{d\boldsymbol{W_h}}{dt} + |C_i| \sum_{j \in K(i)} \Phi_{\mathcal{F}_{ij}}(W_i, W_j, \overrightarrow{\boldsymbol{n_{ij}}}) + \int_{D_i} \mathbb{K} \cdot \overrightarrow{\nabla \boldsymbol{W}} \cdot \overrightarrow{\nabla \phi_i}\ d\boldsymbol{x} = 0 \tag{4-21}$$

4-4 Load computation

After solving the discretized Euler equation, as presented in the previous section, we need to compute the flow-induced loads on the discrete embedded surface D_h^E by enforcing the equilibrium condition (3-26) and send them to the CSD solver. However, as already said, the representation of the discrete embedded surface where the loads need to be computed is not available Eulerian CFD grid D_h. In addition, the structural solver does not have direct access to the pressure field computed in the Eulerian CFD grid D_h. For this reason a consistent and conservative approach is presented here to calculate the localized flow-induced force load in an embedded boundary method.

4-4-1 Local reconstruction of the embedded surface

Let denote τ_q the set of elements of the embedded discrete interface D_h^E.

$$D_h^E = \bigcup_{q=1}^{N_\tau} \tau_q \tag{4-22}$$

Let I_k denote the intersections of D_h^E with D_h computed using the projection-based or the collision-based tracking algorithms presented in Sect.4-2, the we can reconstruct the embedded discrete surface D_h^E as the union $\widetilde{D_h^E}$ of a collection of triangular elements $\tilde{\tau}_q$ obtained by connecting the intersection points I_k:

$$\widetilde{D_h^E} = \bigcup_{q=1}^{N_{\tilde{\tau}}} \tilde{\tau}_q \tag{4-23}$$

[5] Due to the parabolic nature of the viscous terms, there do not exists a truly Finite Volume treatment of the right hand side of (3-1)

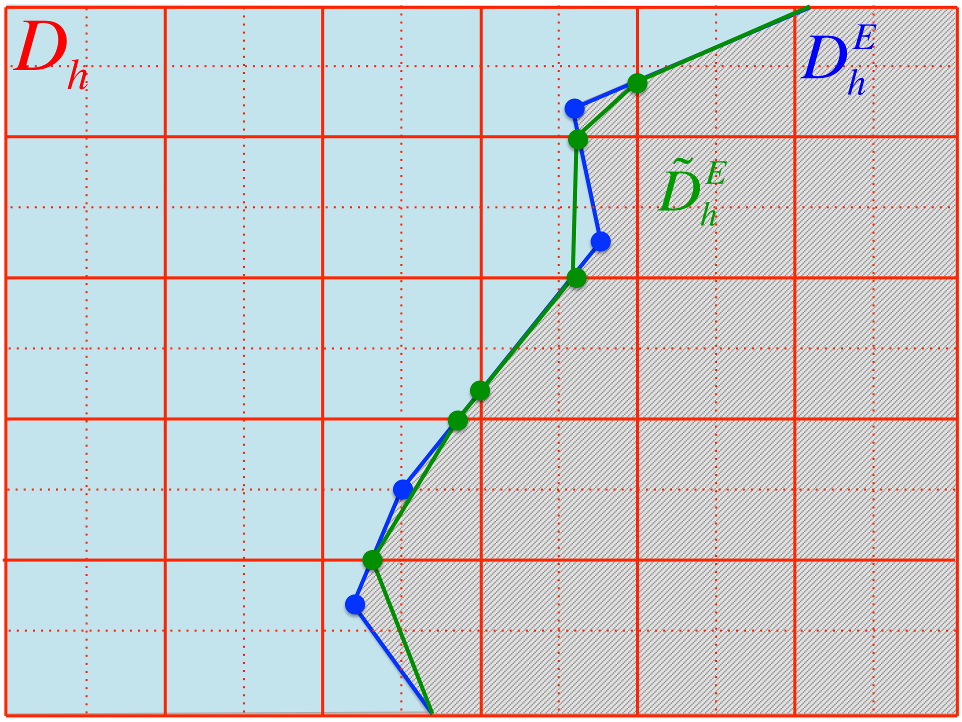


Figure 4-10: Representation of the reconstructed embedded surface.

LOCAL RECONSTRUCTION ALGORITHM

For each element E_h in D_h

- inspect the status of the vertices of the E_h
  - if all status are positive or all the status are negative $\Longrightarrow$ E_h does not cross D_h^E.
  - otherwise $\Longrightarrow$ E_h crosses D_h^E
    1. initialize a list $\mathcal{L}^E$ of intersection points
    2. loop on the edges of the E_h $\Longrightarrow$ for each edge intersecting D_h^E identify I_k and add to $\mathcal{L}^E$
    3. connect all the points of $\mathcal{L}^E$ to form one or more elements $\tilde{\tau}_q$.

We assume now, for simplicity, that D_h^E is the discretization of the wet surface of the structure domain ∂B. Let ξ_k^q denote the natural coordinates of the computed intersection point I_k in τ_q. We can interpolate at each intersection point I_k the displacement and velocity of the reconstructed discrete surface $\widetilde{D_h^E}$:

$$\boldsymbol{u}(I_k) = \sum_j N_j^q(\xi_k^q)\boldsymbol{u}_j, \qquad \dot{\boldsymbol{u}}(I_k) = \sum_j N_j^q(\xi_k^q)\dot{\boldsymbol{u}}_j \tag{4-24}$$

N_j^q is the FE shape function associated with node V_j^q of the element τ_q containing I_k (these shape functions satisfy the partition of unity propriety), $\boldsymbol{u}_j$ and $\dot{\boldsymbol{u}}_j$ are the displacement and velocity of D_h^E at node V_j^q of element τ_q.

Let p_k and $\boldsymbol{v}_k$ denote the computed fluid pressure and velocity vector at I_k (from the solution of the Riemann problem). On the reconstructed interface $\widetilde{D_h^E}$ the fluid pressure and velocity vector are approximated at each $\overrightarrow{x} \in \tilde{\tau}_q$ as:

$$p_h(\overrightarrow{x}) = \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))p_k, \qquad \overrightarrow{v}_h(\overrightarrow{x}) = \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))\boldsymbol{v}_k \tag{4-25}$$

$\tilde{N}_k^q$ is the FE shape function associated with node I_k^q of $\tilde{\tau}_q$, and $\eta(\overrightarrow{x})$ are the natural coordinates of the points $\overrightarrow{x}$ in $\tilde{\tau}_q$.

The discretized version of the continuity transmission condition (3-25) is than:

$$\boldsymbol{v}_k \overrightarrow{\tilde{n}_E^q} = \dot{\boldsymbol{u}}(I_k)\overrightarrow{\tilde{n}_E^q} \tag{4-26}$$

where $\overrightarrow{\tilde{n}_E^q}$ is the normal to an element $\tilde{\tau}_q$ of $\tilde{D_h^E}$ connected to I_k.

Let N_E denote the total number of vertices in D_h^E. Using the same methodology developed for computing the generalized flow-induced load in the the ALE framework, we can compute the load vector $\boldsymbol{f_i^F}$ at the node V_i of the structural model by applying the virtual power principle at the reconstructed fluid-structure interface $\widetilde{D_h^E}$:

$$\begin{aligned}
\sum_{i=1}^{N_E} \boldsymbol{f_i^F} \delta\dot{\boldsymbol{u}}_i &= -\sum_{\tilde{\tau}_q} \int_{\tilde{\tau}_q} (-p_h(\overrightarrow{x}))\tilde{n}_E^q \delta\overrightarrow{v}_h(\overrightarrow{x})\ d\tau \\
&= \sum_{\tilde{\tau}_q} \int_{\tilde{\tau}_q} \left[\sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))p_k \tilde{n}_E^q \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))\delta\boldsymbol{v}_k\right] d\tau \\
&= \sum_{\tilde{\tau}_q} \int_{\tilde{\tau}_q} \left[\sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))p_k \tilde{n}_E^q \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) \left[\sum_j N_j^q(\xi_k^q)\delta\dot{\boldsymbol{u}}_j\right]\right] d\tau
\end{aligned} \tag{4-27}$$

δ denote the virtual quantity. The virtual power of the FE structure loads acting on D_h^E is:

$$\delta P^S = \sum_{i=1}^{N_E} \boldsymbol{f_i^F} \delta\dot{\boldsymbol{u}}_i \tag{4-28}$$

and the virtual power principle at the fluid-structure interface implies that:

$$\delta P^F = \delta P^S \tag{4-29}$$

Using the previous equation and the partition of unity of the shape functions N_i^q and $\tilde{N}_k^q$:

$$\boldsymbol{f_i^F} = \sum_{\tilde{\tau}_q / \exists I_k \in \tilde{\tau}_q,\ I_k \in (\tau_q \ni V_i)} \int_{\tilde{\tau}_q} \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))p_k \tilde{n}_E^q \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))N_i^q(\xi_k^q)\ d\tau \tag{4-30}$$

and

$$
\begin{aligned}
\boldsymbol{f^F} &= \sum_{i=1}^{N_E} \boldsymbol{f_i^F} \\
&= \sum_{i=1}^{N_E} \sum_{\tilde{\tau}_q/\exists I_k \in \tilde{\tau}_q,\ I_k \in (\tau_q \ni V_i)} \int_{\tilde{\tau}_q} \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) p_k \tilde{n}_E^q \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) N_i^q(\xi_k^q)\ d\tau \\
&= \sum_{\tilde{\tau}_q \in \widetilde{D_h^E}} \int_{\tilde{\tau}_q} \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) p_k \tilde{n}_E^q \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) \sum_{i=1}^{N_E} N_i^q(\xi_k^q)\ d\tau \\
&= \sum_{\tilde{\tau}_q \in \widetilde{D_h^E}} \int_{\tilde{\tau}_q} \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) p_k \tilde{n}_E^q \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x}))\ d\tau \\
&= \sum_{\tilde{\tau}_q \in \widetilde{D_h^E}} \int_{\tilde{\tau}_q} \sum_{k=1}^{3} \tilde{N}_k^q(\eta(\overrightarrow{x})) p_k \tilde{n}_E^q\ d\tau
\end{aligned}
\tag{4-31}
$$

If the embedded discrete interface D_h^E does not coincide with the discretization of the wet surface of the structural domain ∂B that we denote as ∂B_h:

1. a projection of D_h^E onto ∂B_h is computed in a pre processing stage,
2. the natural coordiantes of each projected point of D_h^E in the elements of B_h^S are collected,
3. $\boldsymbol{u}$ and $\dot{\boldsymbol{u}}$ in (4-24) are interpreted in D_h^E and are computed by interpolating the nodal displacement and velocities of ∂B_h using the projections and natural natural coordinates collected in the previous steps,
4. the generalized load vector $\boldsymbol{f}_i^F$ in (4-30) is also interpreted on D_h^E and not on ∂B_h,
5. $\boldsymbol{f}_i^F$ is redistributed on the nodes of the FE structural model ∂B_h using again the virtual power principle but with the natural coordinates of the points of the projection of D_h^E onto ∂B_h.

4-5 Summary

The Embedded Boundary method presented above is based on the following steps:

1. In the Computational Structural Dynamics (CSD) solver (4-1), send the updated displacement and velocity of the fluid structure interface to the CFD solver.
2. In the CFD solver (3-1), update the position of the interface D_h^E and track it with respect to the Eulerian grid D_h (4-2). For each edge (V_i, V_j) in D_h connecting the vertices V_i and V_j compute the fluid-fluid and fluid-structure fluxes as follow:
    - if both V_i and V_j belong to the fluid domain Ω and the edge (V_i, V_j) does not intersect the embedded interface $D_h^E \implies$ compute fluid-fluid flux between V_i and V_j.

- if only one of the two vertices belongs to Ω, declare that (V_i, V_j) intersects $D_h^E \Longrightarrow$ solve a one-dimensional fluid-structure Riemann problem between the active vertex and the interface. Than compute a fluid-structure flux using the fluid interface state obtained from the Riemann solver.
- if both V_i and V_j belong to the fluid domain Ω and the edge (V_i, V_j) intersects D_h^E $\Longrightarrow$ solve one-dimensional fluid-structure Riemann problem between each vertex and the interface, than compute a fluid-structure flux on each side of the interface.
- if neither V_i nor V_j belongs to $\Omega \Longrightarrow$ do not perform any computation.

3. In the CFD solver integrate the semi-discretized fluid equations form t^n to t^{n+1} (4-3-1), compute the fluid-induced load on a surrogate surface (4-4), distribute it on the fluid-structure intersection points and send the load to the CSD solver.
4. in the CSD solver, integrate the semi-discretized structural equations form t^n to t^{n+1} using the loads received from the CFD solver.

Chapter 5

Computational Results

In this chapter, we present the computational results obtained with the Embedded Boundary Method presented in the previous chapters.

5-1 Airfoil with moving flap

One of the main advantage of the non-body fitted mesh methods is that large and complex motions of rigid structures can be accommodated without additional mesh motion algorithms and without the needs of expensive remeshing steps needed if the computational grid volumes becomes highly stretched like in the ALE approach.

In this section, the problem of computing the unsteady viscous low past a rigid airfoil with moving flap (Fig. 5-1) is considered.

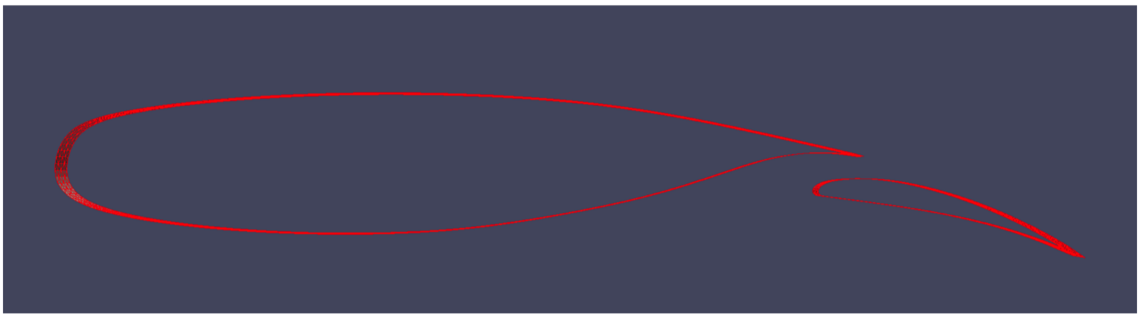


Figure 5-1: Airfoil with flap.

The unstructured Eulerian grid is composed by 3,285,154 nodes (6,569,866 elements) and is designed to ensure that the non-dimensional distance based on local cell fluid velocity $y^+ = \frac{u^* y}{\nu} = 1$ (u^* is the friction velocity at the wall, y is the distance to the wall and ν the kinematic viscosity) in the region where the airfoil and the flap embedded surfaces move[1] (Fig.5-2).

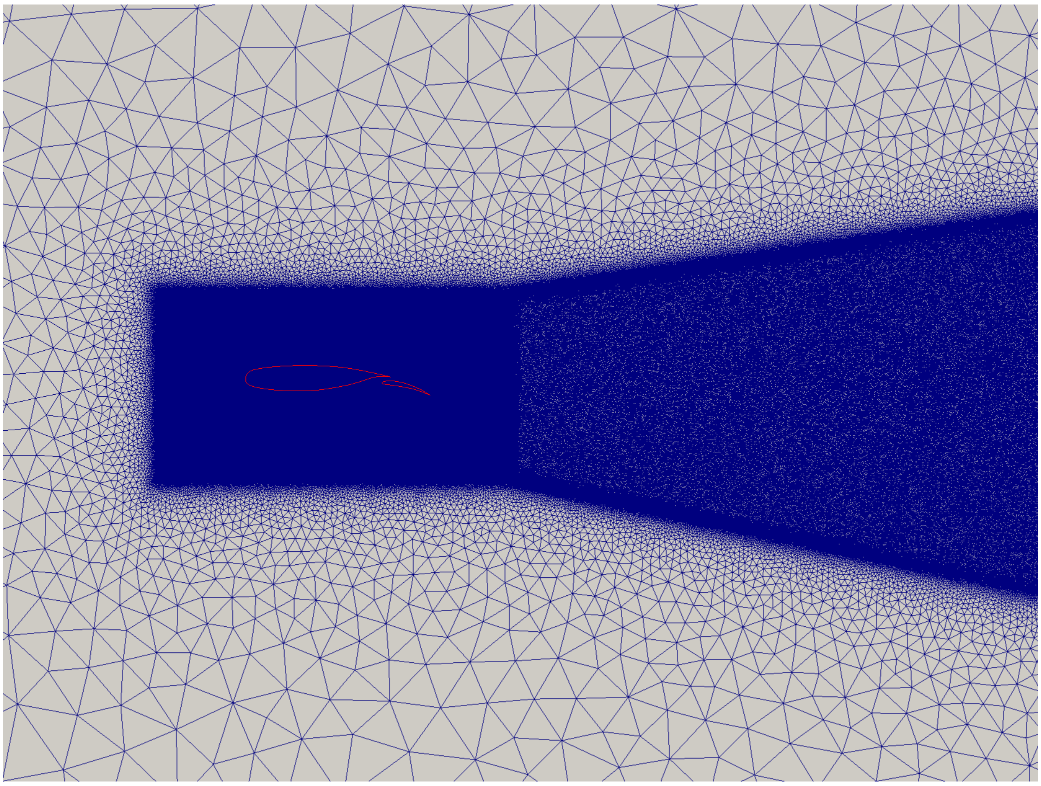


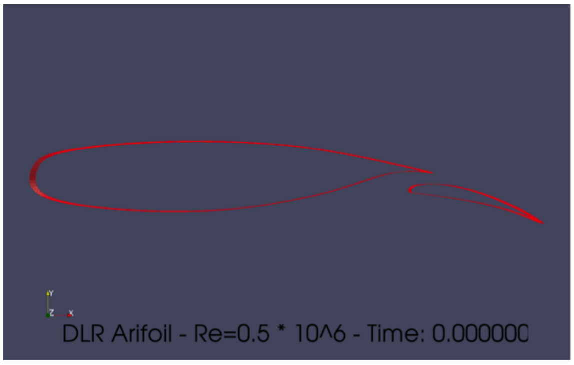
Figure 5-2: Eulerian and embedded grid of the airfoil and flap (in red).

The viscous air flow is modeled using the Navier-Stokes equations with Spalart-Allmaras turbulence model. The free-stream conditions are set to $M = 0.3$, $Re = 500,000$. The sinusoidal prescribed motion of the flap with frequency $f = 0.5$ $[Hz]$ is presented in Fig.5-3.

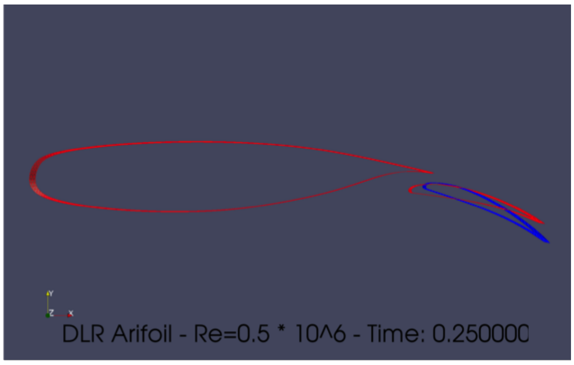
Using the three-point backward difference time-integrator (4-3-1) an implicit numerical simulation based on the FRG - Projection-based approach (4-2-1) to track the embedded moving surfaces and the local reconstruction algorithm (4-4-1) to compute the flow induced loads on the embedded surface produced the following unsteady results for the airfoil with moving flap.

The Embedded Boundary method simulation of the wing with moving flap was performed on 256 processors cluster. The simulation took $38,526$ $[sec]$.

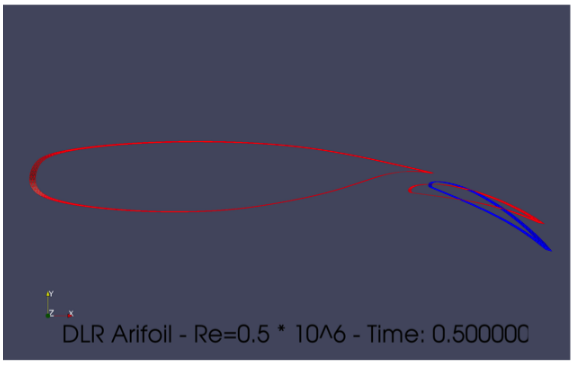
[1]The grid spacing in the rectangular region around the airfoil and flap was derived using the following formula: $\Delta y = Ly^+\sqrt{74}Re_L^{-13/14}$ where L is the flow characteristic length scale and Re_L is the Reynolds Number based on the problem characteristic length scale



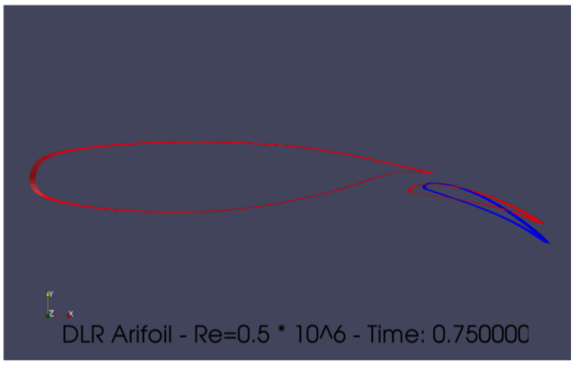
(a) Flap position at t=0



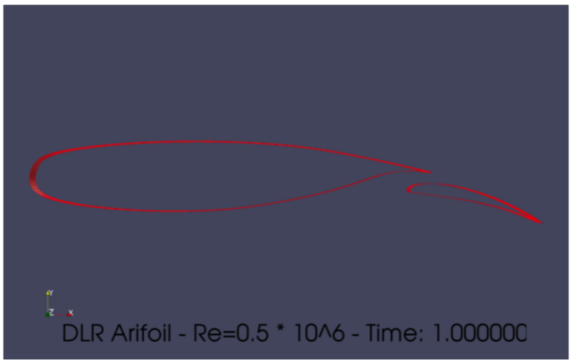
(b) Flap position at t=0.25



(c) Flap position at t=0.5

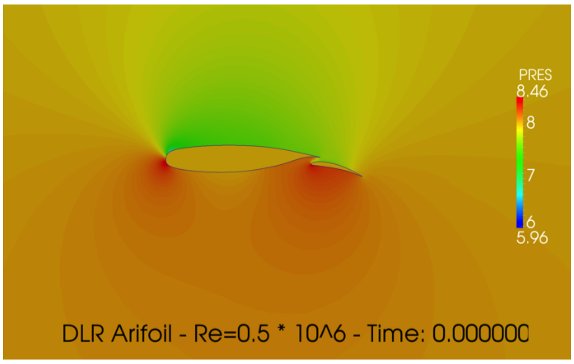


(d) Flap position at t=0.75

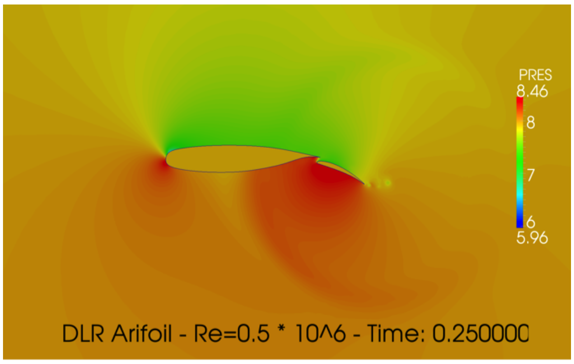


(e) Flap position at t=1

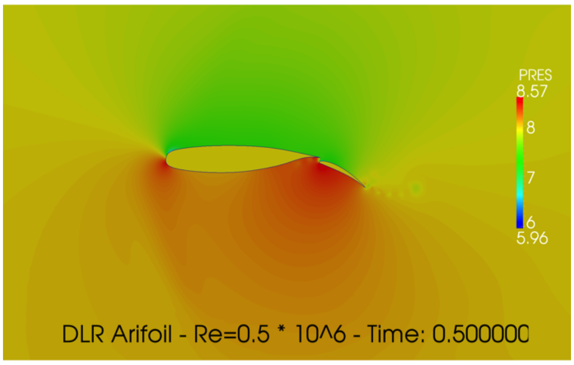
Figure 5-3: Flap position at different time steps.



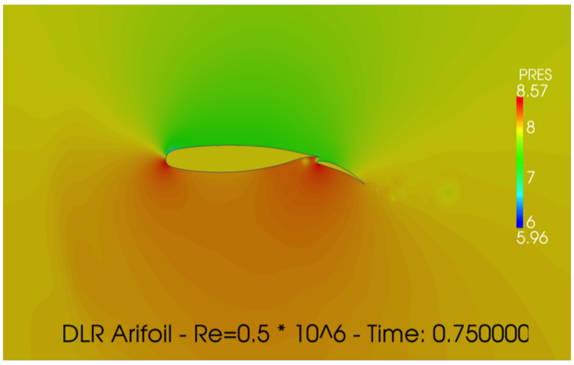
(a) Pressure profile at t=0



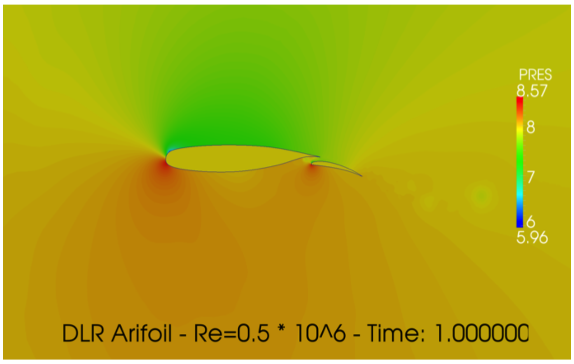
(b) Pressure profile at t=0.25



(c) Pressure profile at t=0.5

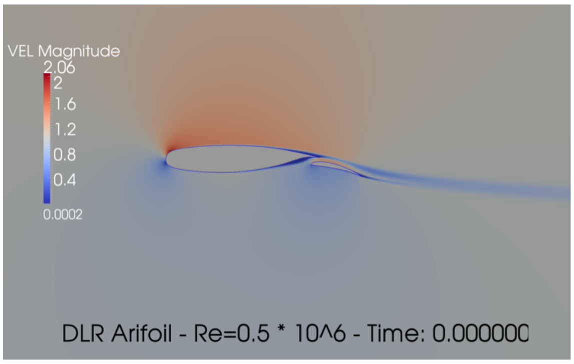


(d) Pressure profile at t=0.75

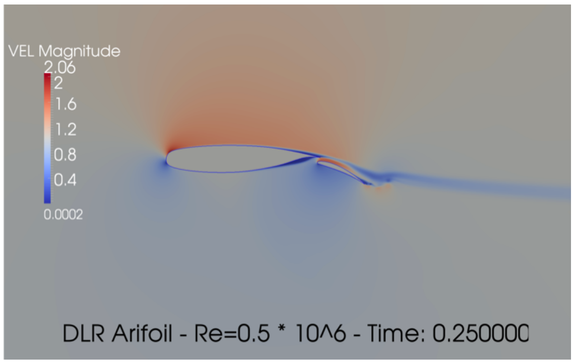


(e) Pressure profile at t=1

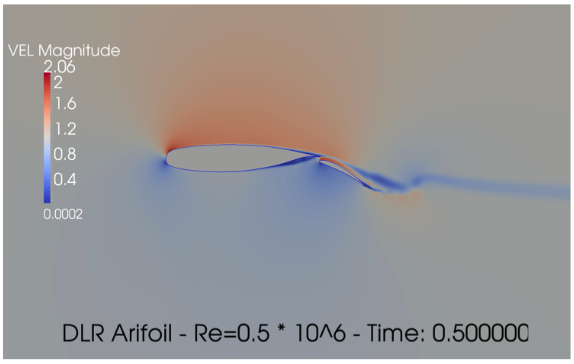
Figure 5-4: Non-dimensional pressure profile at different time steps.



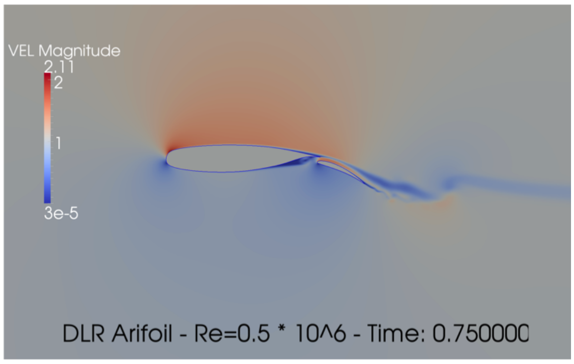
(a) Velocity profile at t=0



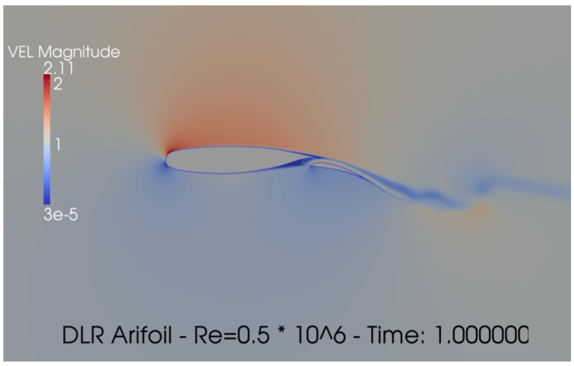
(b) Velocity profile at t=0.25



(c) Velocity profile at t=0.5



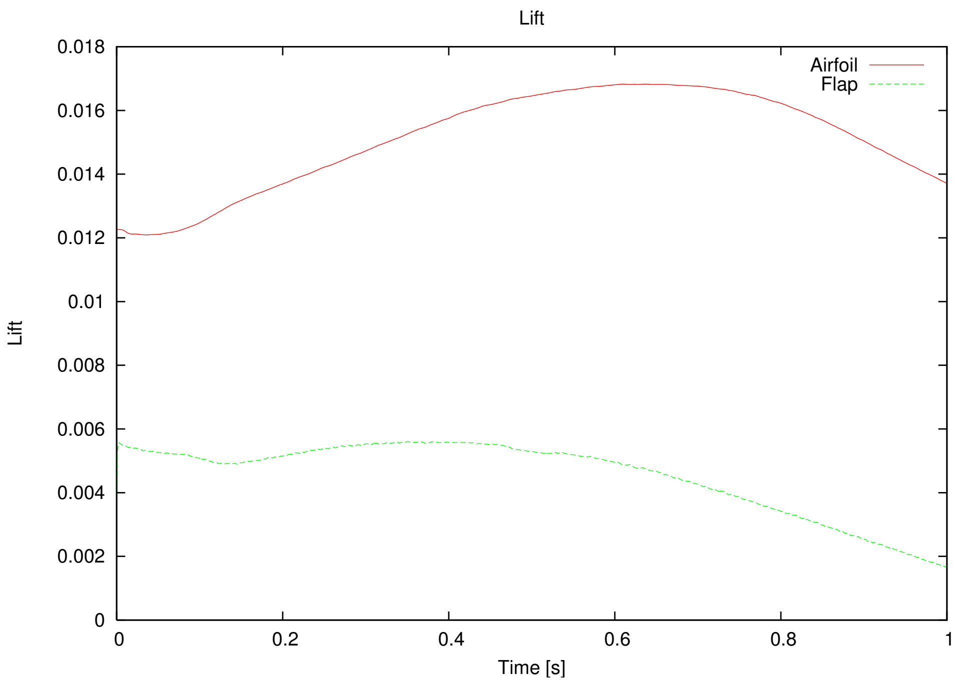
(d) Velocity profile at t=0.75



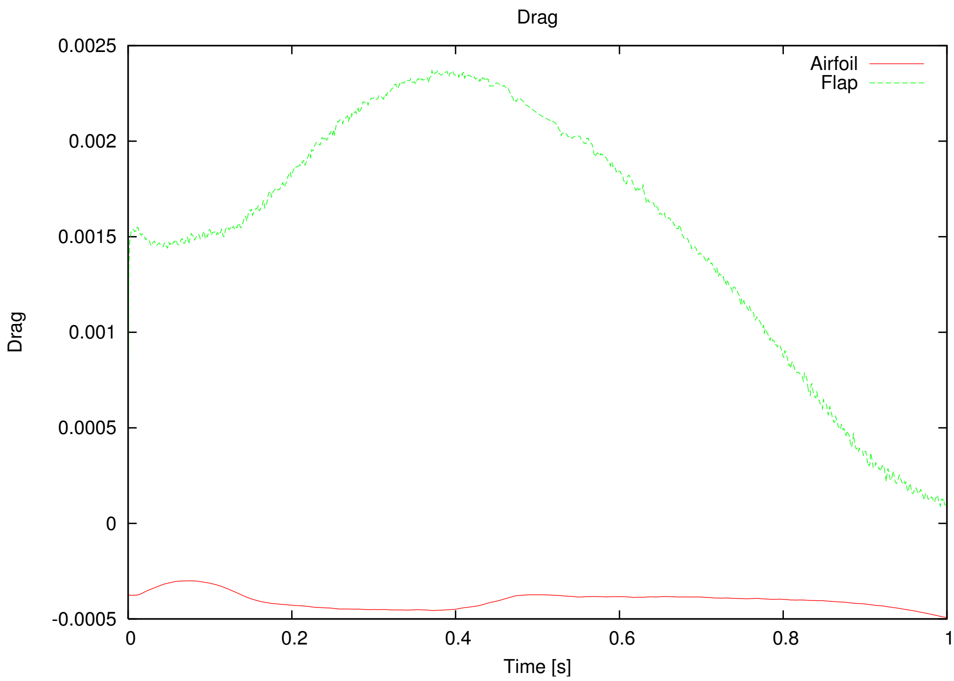
(e) Velocity profile at t=1

Figure 5-5: Non-dimensional velocity profile at different time steps.

The non-dimensional lift and drag history of the airfoil and flap are presented in the following figures:



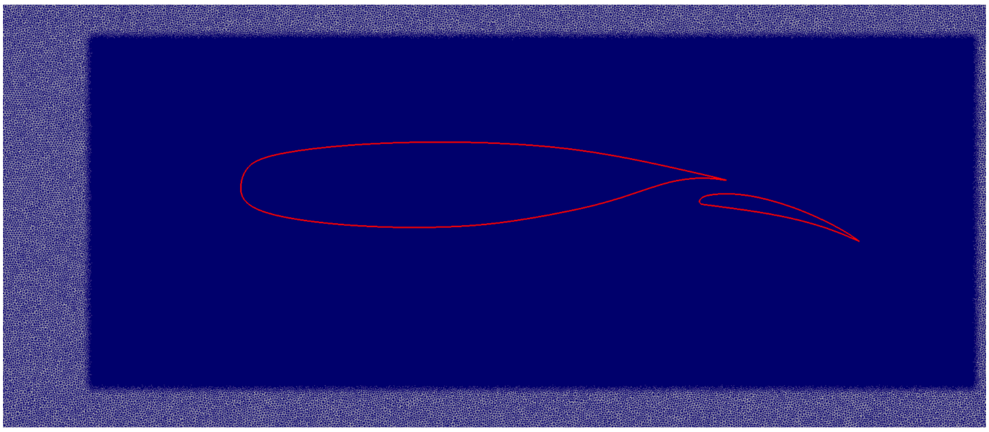
(a) Non-dimensional lift



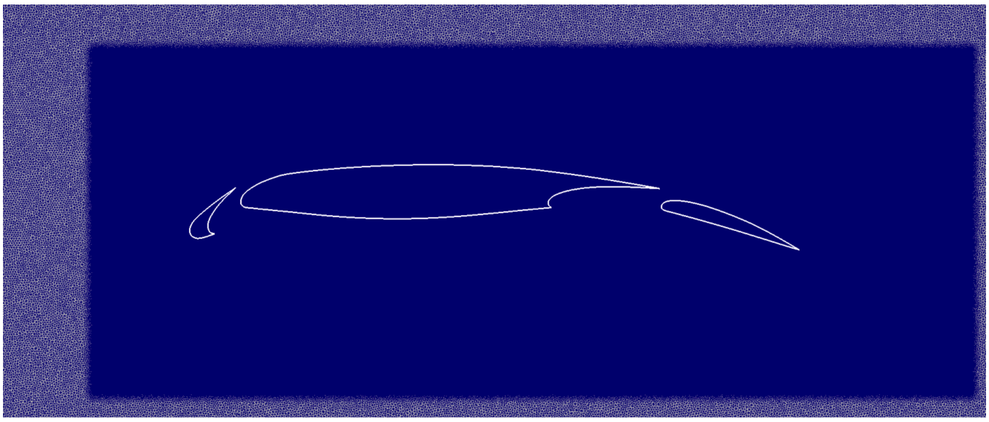
(b) Non-dimensional drag

Figure 5-6: Non-dimensional lift and drag of the airfoil and moving flap.

An interesting feature of the non-body fitted mesh methods is that the Eulerian CFD grid can be re-used for different problems without the need of recomputing and design a new mesh for each problem. For this reason this class of methods are particular attracting for optimization and design purposes since can dramatically reduce the time needed to compute and design an efficient CFD mesh. Once an initial Eulerian CFD mesh is designed based on the flow characteristics and the dimensions of the problems under consideration it can be used as a 'background' mesh for different embedded grids. Indeed, generating a high-quality body-fitted grids around a complex geometry is usually both time-consuming and tedious. For example the above presented Eulerian grid, used to simulate the flow around an airfoil and moving flap, can be used to compute the flow around a completely different airfoil with slat and flap (Fig.5-5).



(a) DLR Airfoil with flap



(b) AGARD airfoil with slat and flap

Figure 5-7: The same Eulerian CFD grid can be used with different embedded surfaces.

However as can be observed from the previous figures a large number of grid volumes that lie inside the embedded surface and in the refinment rectangular box are not needed for the flow computation. In case of 3D problems (in particular for viscous mesh) the computational load due to the large number of grid volumes becomes tremendously high and different strategies, proposed and tested in the following sections, are needed to design the Eulerian grid and compute the flow around moving embedded surfaces.

5-2 Missile Drop

The simulation of fluid flow around moving objects in relative motions is particularly challenging with body-fitted mesh methods because the computational grid volumes becomes highly stretched in the region between the objects since the grid nodes are constrained to the moving surfaces.

In this section, a verification of the proposed embedded boundary method is performed in the context of an unsteady, inviscid flow around a 3D wing with a dropping missile. The complex geometry of this system (Fig.5-8) and the relative motion of the missile with respect to the wing indicates that the embedded boundary method can be a perfect candidate for studying the aerodynamic properties of the wing in presence of the moving missile.

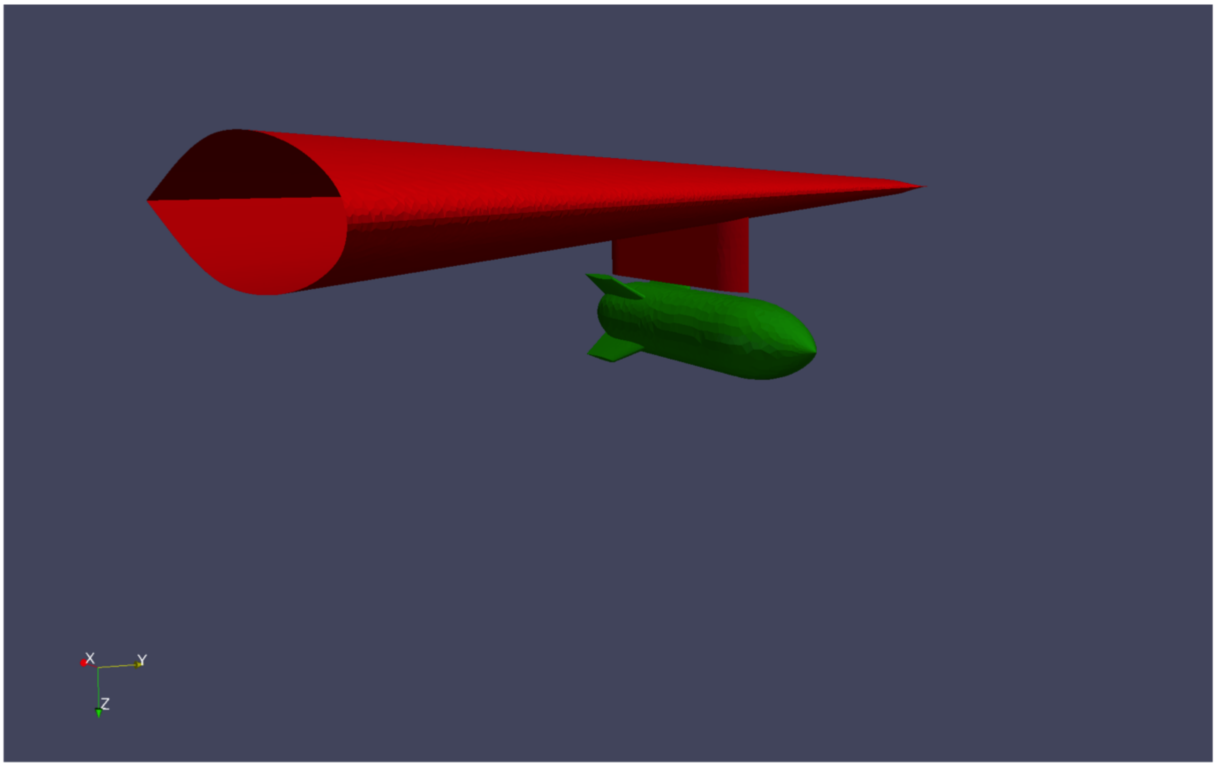
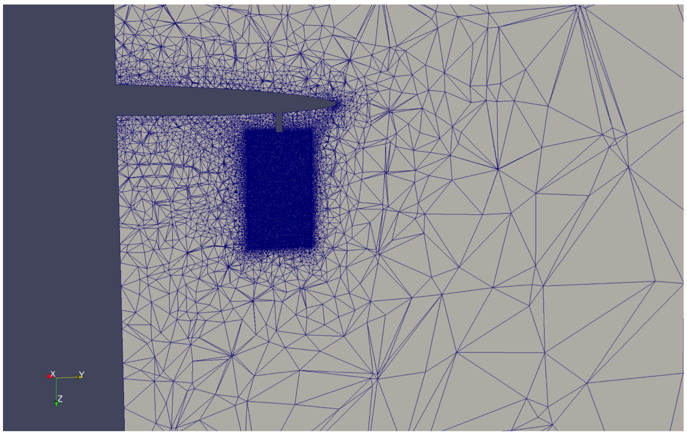


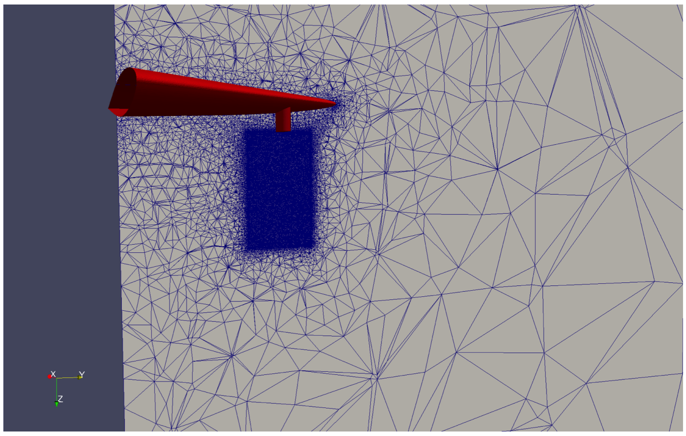
Figure 5-8: Wing with missile (Type A).

To reduce the number of grid volumes required to accurately simulate the flow around the wing and the moving missile we decided to create a body-fitted grid around the fixed wing and 'immerse' in the Eulerian grid only the embedded surface of the moving missile (Fig.5-9).

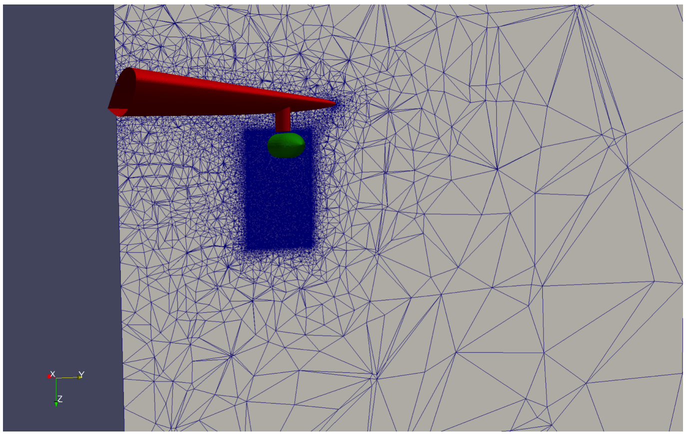
The inviscid flow around the wing and dropping missile is modeled using the Euler equations. The free-stream conditions are set to $M = 0.3$. The missile is set in forced motion in perpendicular direction to the wing. Using the three-point backward difference time-integrator (4-3-1) an implicit numerical simulation based on the PhysBAM - Collision-based approach (4-2-2) to track the embedded moving surfaces and the local reconstruction algorithm (4-4-1)



(a) Body-fitted mesh around the wing

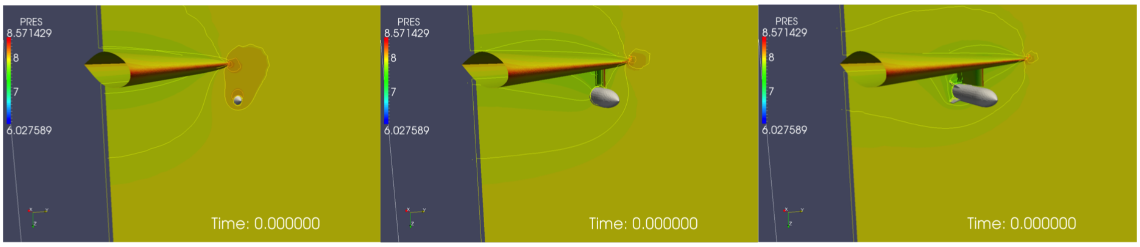


(b) Body-fitted mesh around the wing (red)

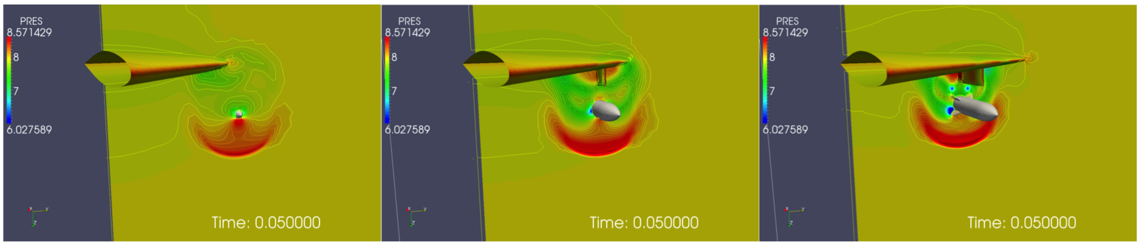


(c) Rectangular refinment region around the embedded missile surface.

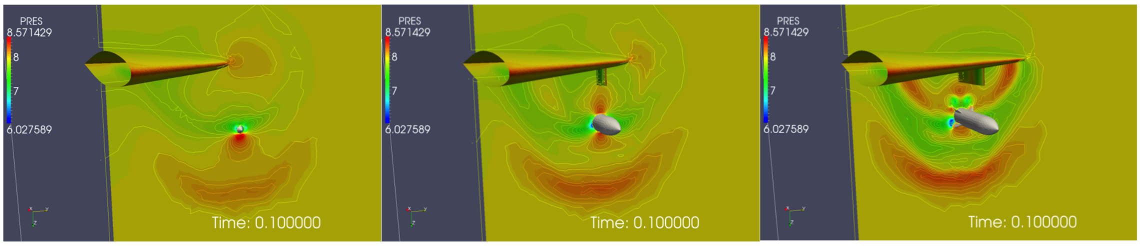
Figure 5-9: Body-fitted mesh for the wing and embedded mesh for the moving missile (section view) to compute the flow induced loads on the embedded surface produced the following unsteady results for the wing with dropping missile.



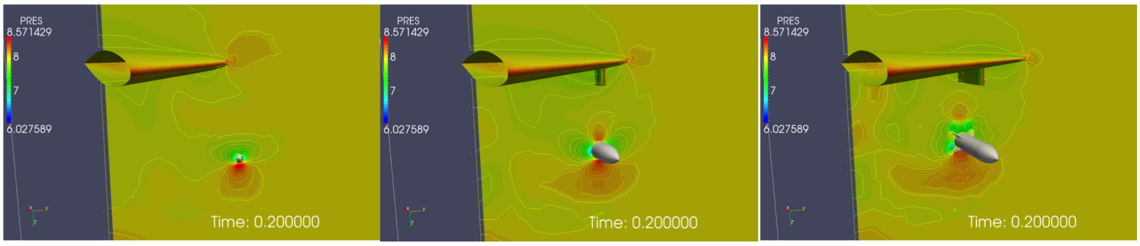
(a) Pressure profile at t=0



(b) Pressure profile at t=0.05



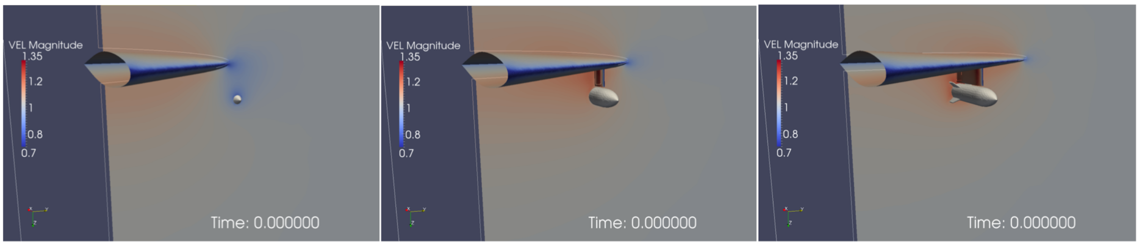
(c) Pressure profile at t=0.1



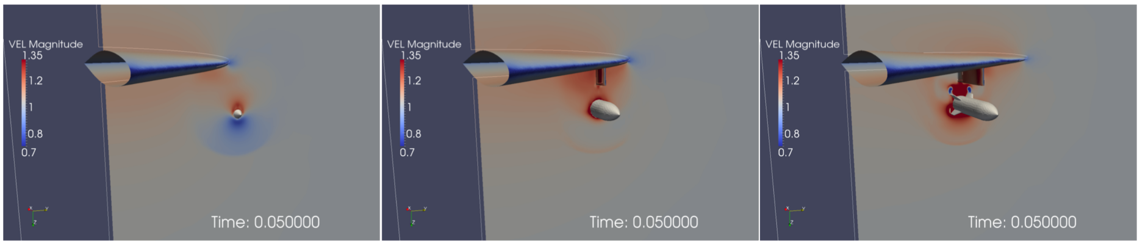
(d) Pressure profile at t=0.2

Figure 5-10: Non-dimensional pressure profile at different time steps and in different sections perpendicular to the wing

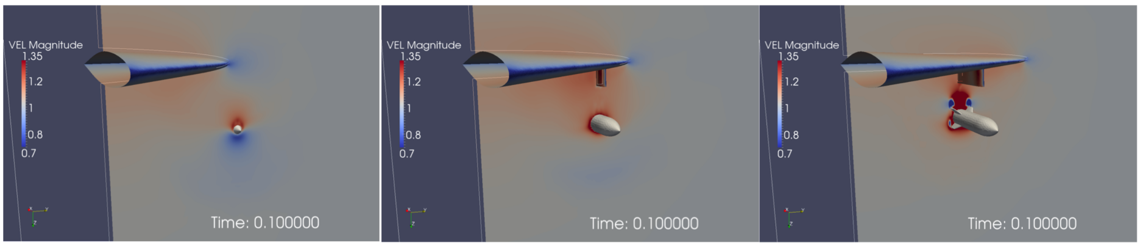
The Embedded Boundary method (computational grid with 2,546,692 nodes) simulation of the wing with dropping missile was performed on 256 processors cluster. The simulations took $15,237.43$ $[sec]$.



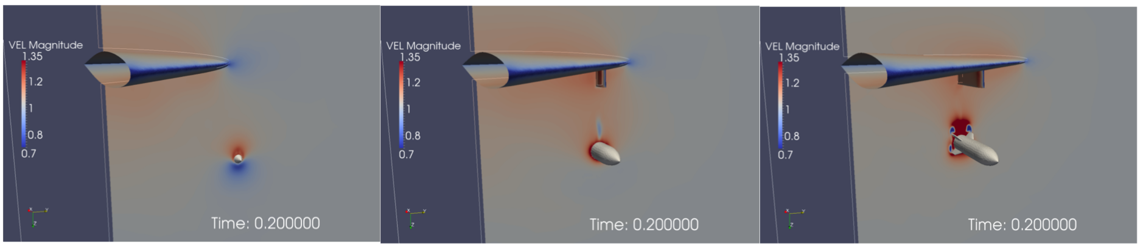
(a) Velocity profile at t=0



(b) Velocity profile at t=0.05

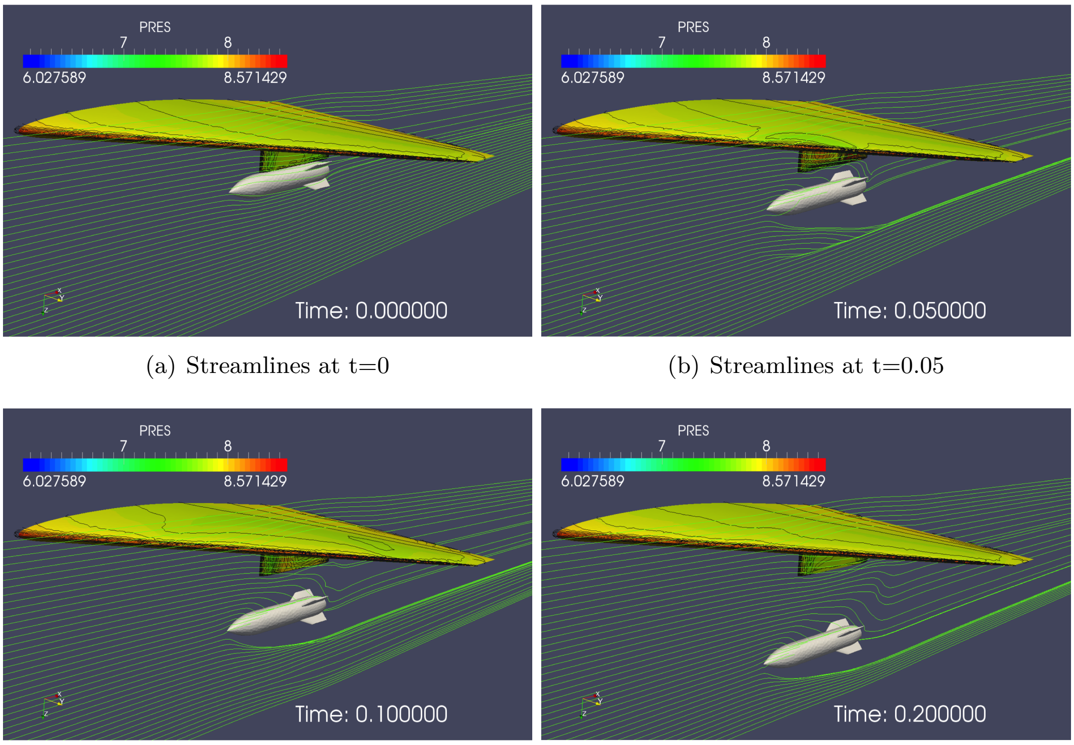


(c) Velocity profile at t=0.1



(d) Velocity profile at t=0.2

Figure 5-11: Non-dimensional velocity profile at different time steps and in different sections perpendicular to the wing



(a) Streamlines at t=0

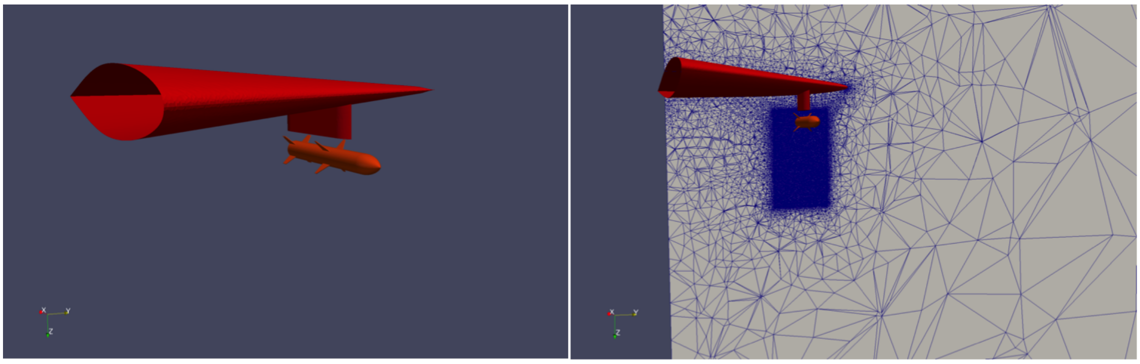
(b) Streamlines at t=0.05

(c) Streamlines at t=0.1

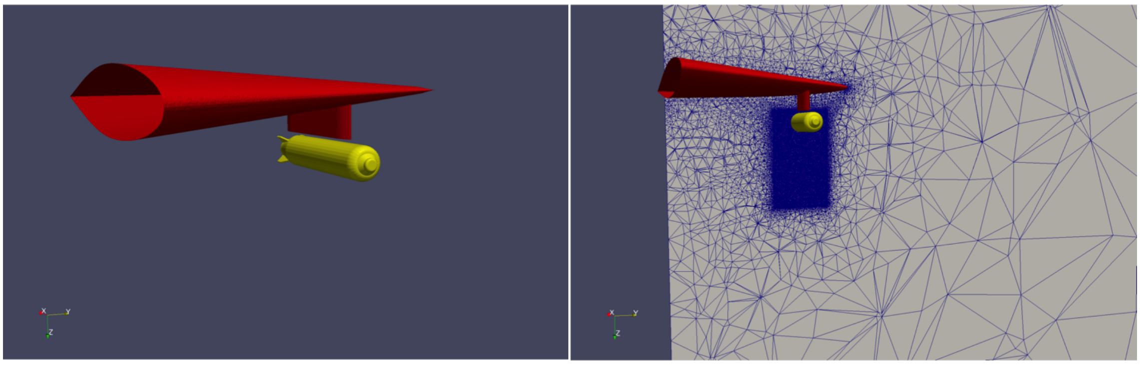
(d) Streamlines at t=0.2

Figure 5-12: Streamlines around the wing and moving missile and pressure contour lines on the wing surface at different time steps.

As already said in the previous case the Eulerian grid designed for a particular case can be used in the design process to test different configurations or in particular for this case different shapes of the missile. This is very useful because reduce the time spent in design the computational grid that has to be recomputed for each shape in case of body-fitted mesh methods.



(a) Missile type B



(b) Missile type C

Figure 5-13: Two different shapes of the missile that can be used in the same Eulerian grid.

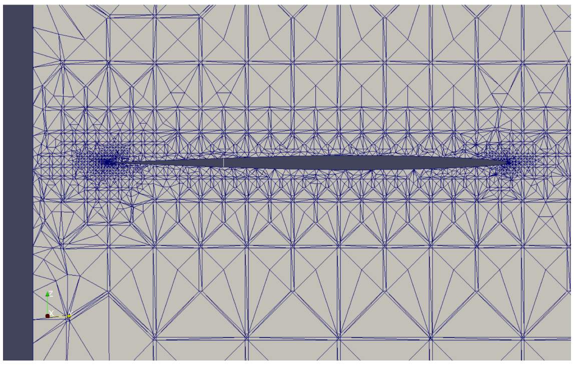
5-3 Heaving AGARD Wing

In this section, the problem of computing the unsteady inviscid flow past a rigid wing in heaving motion is considered. The wing is the AGARD Wing 445.6 (Fig.5-14) (root chord length $L_c = 22.0$ $[in]$, a semi-span $L_s = 30.0$ $[in]$, a tip chord length $Lt = 14.5$ $[in]$, a quarter-chord sweep angle of 45 deg, panel aspect ratio is equal to 1.65 and its taper ratio is equal to 0.66). Its airfoil section is the NACA 65A004 with a maximum thickness of approximately 0.9 $[in]$.

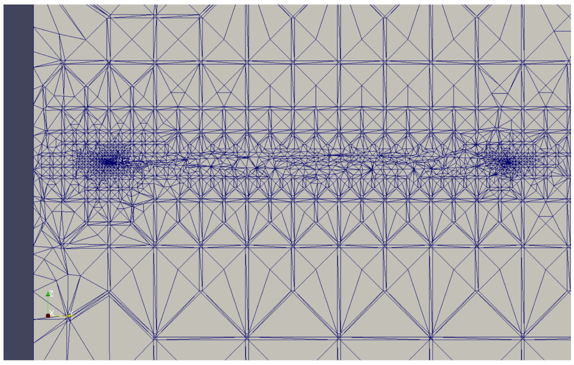
Figure 5-14: Top and side view of the AGARD Wing 445.6 .

The wing is set in a prescribed harmonic heaving motion with an amplitude $A = 3$ $[in]$ and a frequency $h = 100$ $[Hz]$. The inviscid flow around the wing is modeled using the Euler equations. The free-stream conditions are set to $M = 0.3$, $\rho_\infty = 9.357255\ 10^8$ $[(lb/in^4)sec^2]$ and $p = 14.5$ $[psi]$.

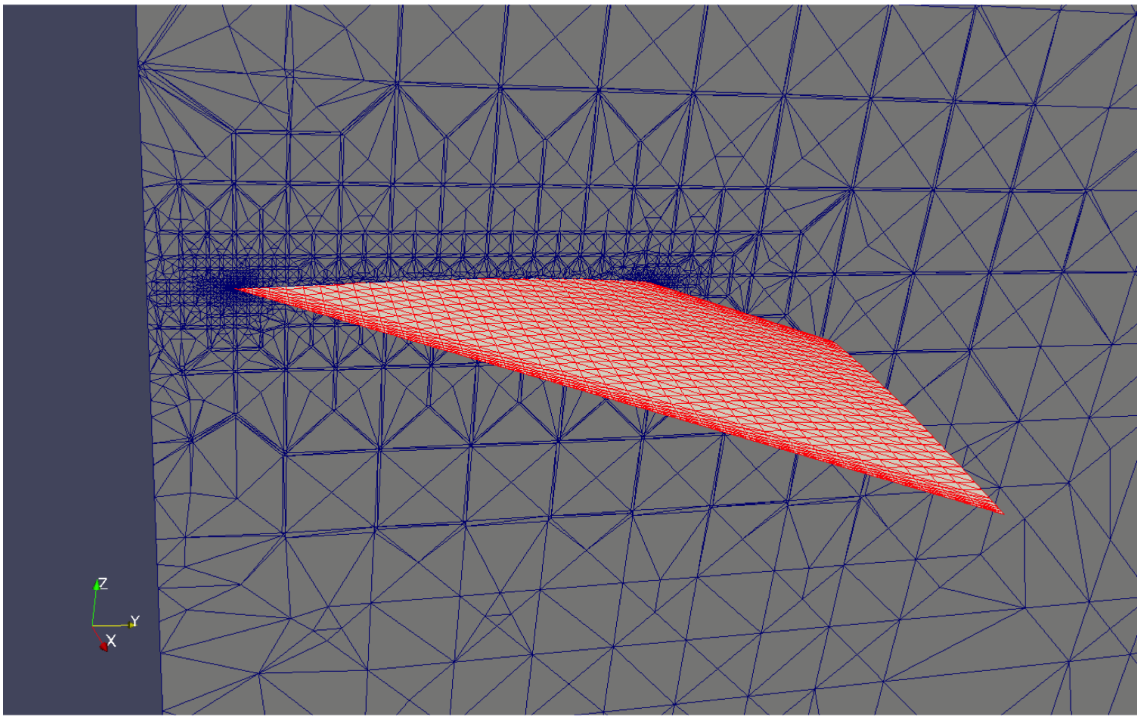
We will compare the computational results obtained with the body-fitted mesh using the ALE approach and with the Eulerian grid using the Embedded Boundary Method. The Eulerian grid used with the Embedded Boundary method is obtained from the body-fitted grid by meshing the interior of the wing; the Embedded surface of the wing is obtained from the surface discretization of the wing in the body-fitted grid (Fig.5-15).



(a) Body-fitted grid



(b) Eulerian gird



(c) Surface grid/Embedded surface grid of the wing (red)

Figure 5-15: Two different shapes of the missile.

Since the wing its quite thin we will use the PhysBAM - Collision-based approach (4-2-2) that is able to accurately track the embedded moving surface even if the volume enclosed by the surface is not fully resolved. All the simulations are initialized with a uniform flow at the free-stream condition specified above.

The lift of the heaving Agard wing obtained with the ALE approach (considered as reference) and the Embedded Boundary methods is presented in the following figure:

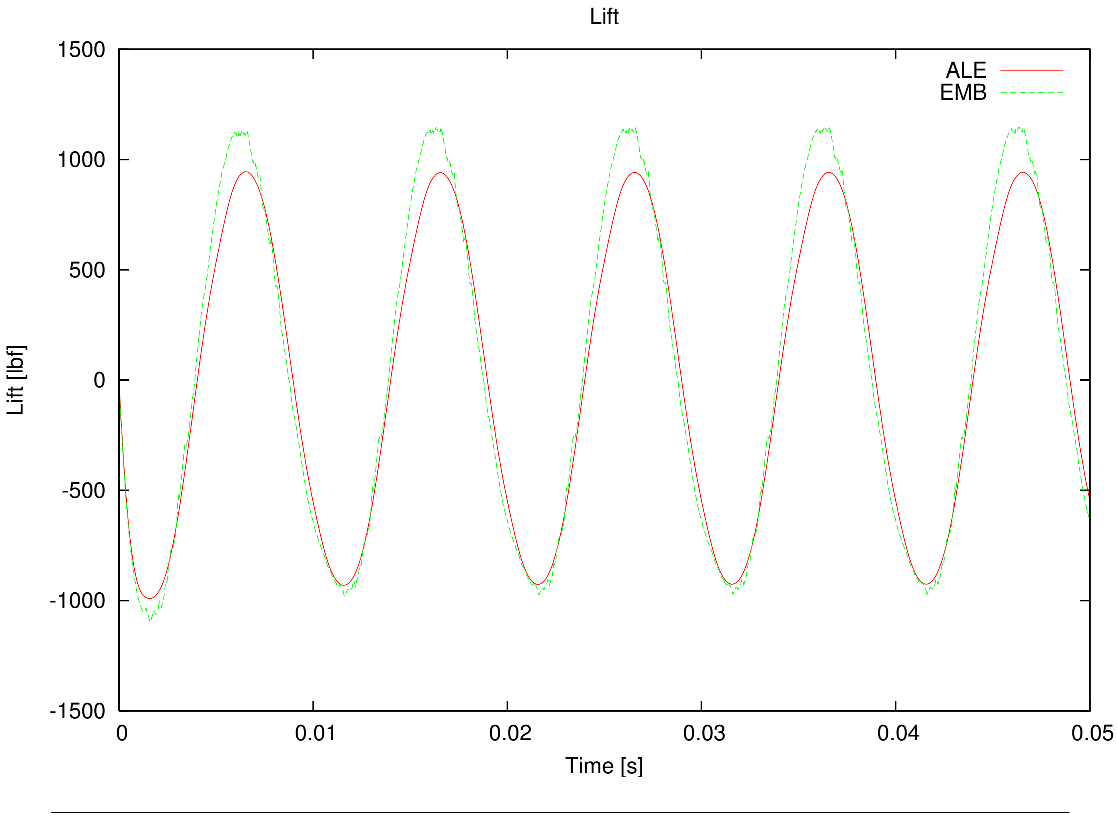


Figure 5-16: Lift time-history of the AGARD Wing 445.6 in heaving motion.

As it is possible to observe from the previous plot the lift computed using the Embedded Boundary method looks quite inaccurate in particular at the peaks of the oscillation. This is because we are no more using an Eulerian grid designed for the Embedded Boundary method with a refinment in the region of the motion of the embedded surface but a computational mesh derived from a typical body fitted grid. The ALE approach thanks to a mesh motion algorithm (corotational approach) is able to move the entire fluid grid and follow the motion of the wing. By doing so the refined region is always in proximity of the wing. In the Embedded Boundary method the Eulerian grid is fixed and the embedded surface moves form a fine to a coarse region in the grid. The tracking algorithm and the load computation algorithm are no more able to accurately reconstruct the surface of the wing and compute the loads on it.

As already said in the previous sections, one of the main problem in the Embedded Boundary method is that the Eulerian grids for 3D problems with moving surfaces require a large number of computational grid volumes. This limit the applicability of the method because of the computational load required to treat large meshes.

By looking at the above mentioned issues we wonder if was possible to introduce a prescribed motion, as in the ALE approach, also to the 'background' grid in the Embedded Boundary method to keep the embedded surface always in the refined region without having to create large refined region in the computational grid. The results of the Embedded boundary method with an a grid that moves with the heaving wing are presented in the following figure.

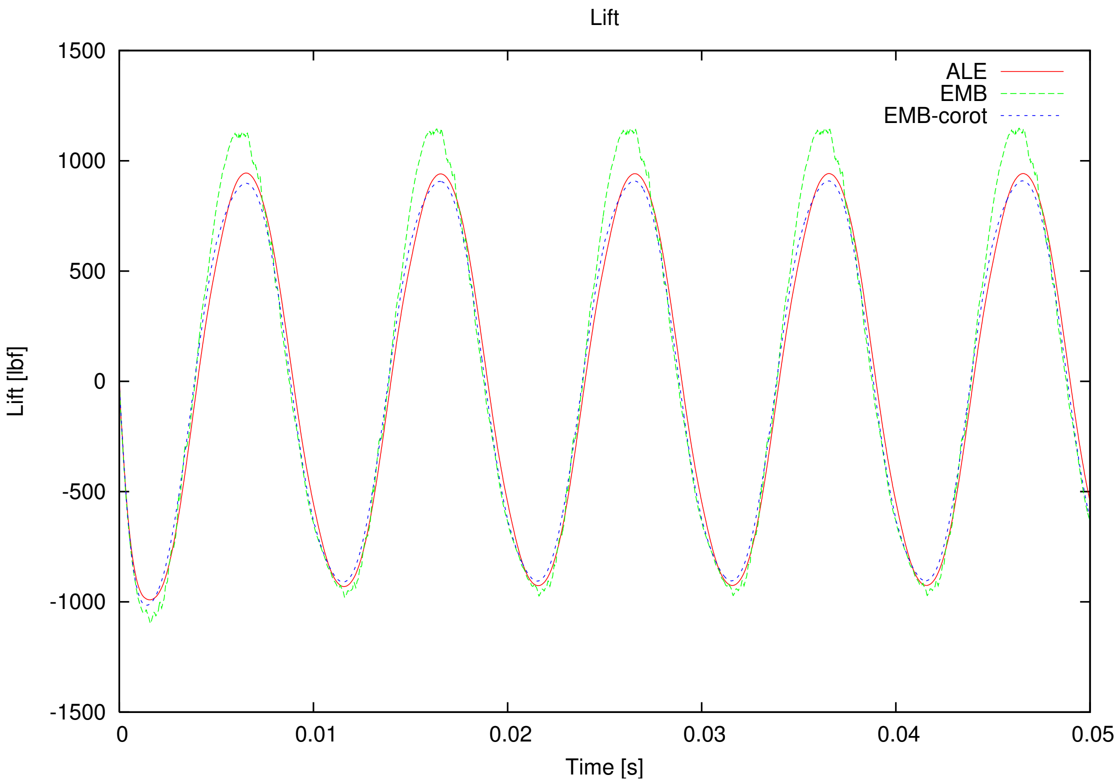


Figure 5-17: Lift time-history of the AGARD Wing 445.6 in heaving motion.

As it is possible to observe the Embedded Boundary method with a moving grid that 'follows' the motion of the heaving wing is much more accurate and match the lift predicted with the ALE approach.

The ALE method (computational grid with 101,623 nodes) and the Embedded Boundary method (computational grid with 105,030 nodes) with and without mesh motion simulations of the AGARD wing in heaving motion were performed on 16 processors cluster. The simulations took $11,322.92$ [sec] with the ALE approach, $13,055.12$ [sec] with the standard Embedded Boundary method and $16,236.33$ with the latter equipped with mesh motion algorithm.

5-4 Mesh deformation and mesh adaptation

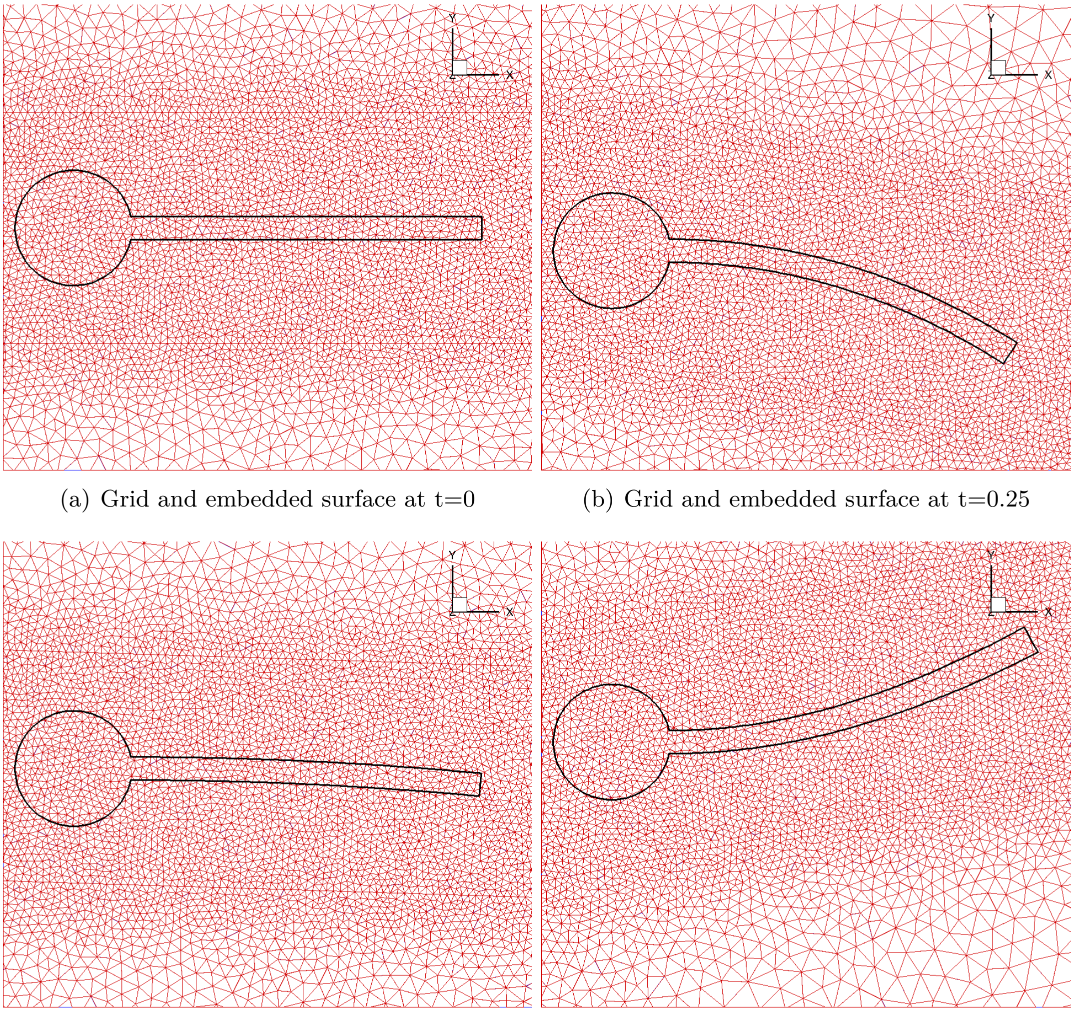
Motivated by the results obtained for the heaving AGARD wing we wonder if was possible to further improve the quality of the mesh needed to accurately compute the flow around a complex moving surface using the Embedded Boundary method and to limit the number of computational grid volumes.

As already said, in the Embedded Boundary method the Eulerian grid does not contain the information of the embedded surface. For this reason it is usually quite difficult to design an efficient grid that does not contain a large number of grid volumes that lie inside the embedded surface and hence are not needed in the computation. In addition, for viscous problems that require a boundary layer mesh one option is to create a very fine mesh, as in the case presented in the previous section for the arifoil with moving flap, around the embedded surface and in the region of the motion of the latter. However for 3D problems this approach may lead to tremendously large mesh that require an huge computational power.

We tested the mesh deformation algorithm based on the torsional spring, widely used in the ALE approach to deform the computational grid around a moving/deforming object, to deform the background grid in the Embedded Boundary method. By doing so, as in the previous case of the AGARD wing, it should be possible to reduce the computational grid size by restrict the refined regions of the mesh only in the vicinity of the embedded grid and deform and move such regions accordingly to the motion and the deformation of the embedded surface. The deformation of the background mesh is driven by the information of the embedded grid however the nodes are not constrained to that surface as in the body-fitted methods. This reduce the stretching of the computational volumes and does not produce degenerated elements in the mesh. Such hybrid Embedded-ALE approach was successfully tested in simulate the unsteady flow around a cylinder with a flapping beam.

The deforming grid around the embedded surface of the cylinder with flapping beam is presented in Fig.5-18. As it is possible to observe the grid volumes in the refined region are not constrained to he embedded surface but they deform and move to follow the deformation of the flapping beam.

In Fig.5-19 the non-dimensional pressure contour around the cylinder with flapping beam, computed using the Embedded Boundary method equipped with the FRG - Projection-based approach (4-2-1) to track the embedded moving surfaces and the local reconstruction algorithm (4-4-1) to compute the flow induced loads on the embedded surface, is presented.



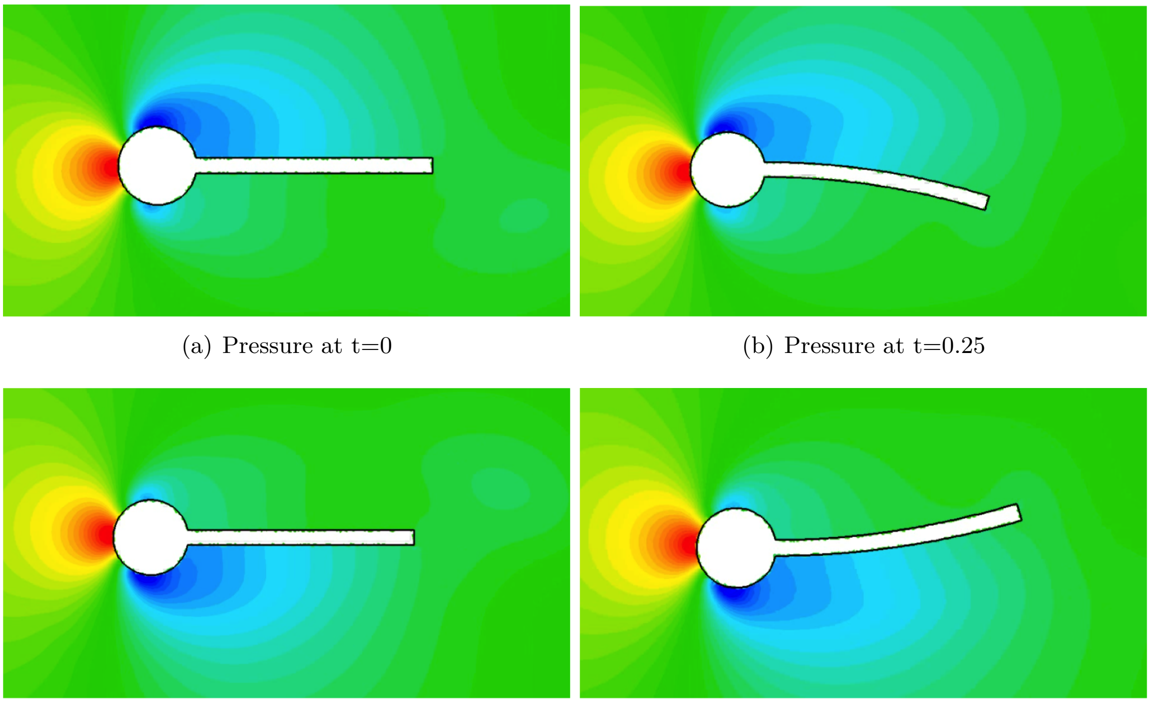
(a) Grid and embedded surface at t=0

(b) Grid and embedded surface at t=0.25

(c) Grid and embedded surface at t=0.5

(d) Grid and embedded surface at t=0.75

Figure 5-18: Mesh deformation and embedded surface of the cylinder with flapping beam.



(a) Pressure at t=0

(b) Pressure at t=0.25

(c) Pressure at t=0.5

(d) Pressure at t=0.75

Figure 5-19: Non-dimensional pressure around the cylider with flapping beam.

The Immersed Volume Method (2-2-3) is designed to operate on unstructured grids and it is also equipped with mesh adaptation algorithm that refine anisotropically the computational mesh around the moving surfaces using a levelset fuction. By using such mesh adaptation algorithm it is possible to control the mesh size in a non-body fitted grid by employing the information obtained form the level set function defined around the embedded surface. In other words the dimensions of the grid volumes can be controlled by looking at the level set function defined in the computational domain. In the case of the Embedded Boundary method it is possible to increase the size of the grid volumes inside and far from the embedded surface and at the same time refine only the region around such surface. By doing so the mesh size is dramatically reduced and the quality of the grid in the region where the embedded surface will be immersed is as high as possible.

We tested this mesh adaptation algorithms for the grid generation around a very complex 3D surface: the F22 Raptor fighter. As it is possible to observe in Fig.5-22, the fighter is represented by 9 separate embedded surfaces (8 control surfaces and the aircraft body). A level set fuction is defined for each surface and the informations are than used to define the mesh density in the background CFD grid. The fine boundary layer mesh is prescribed only in the vicinity of the embedded surface and the grid volumes that lie inside the surface are imposed to be as large as possible.

The mesh obtained with the above mentioned algorithm is presented in Fig.5-21

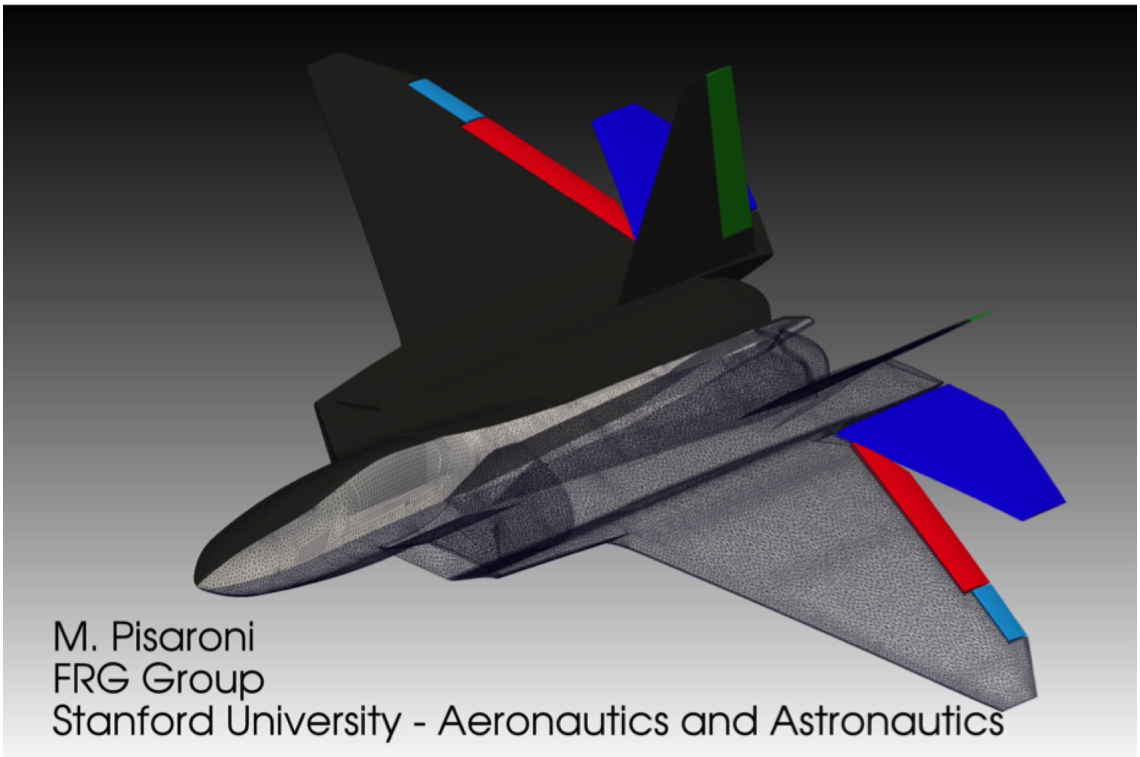
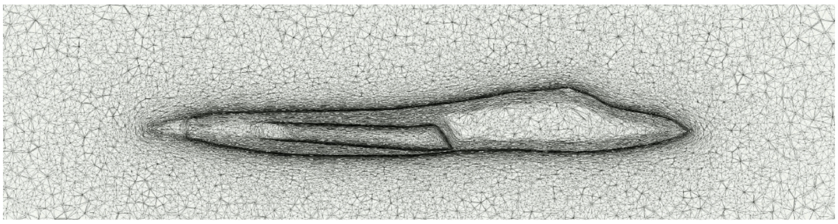
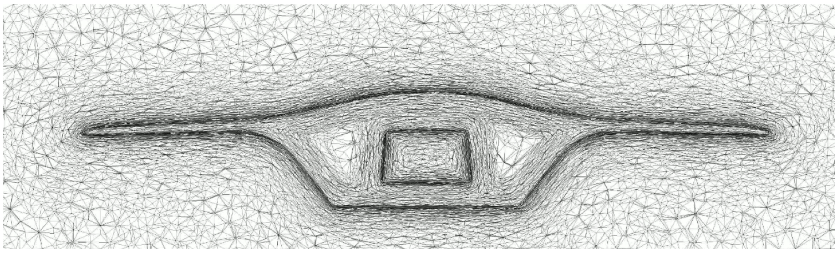


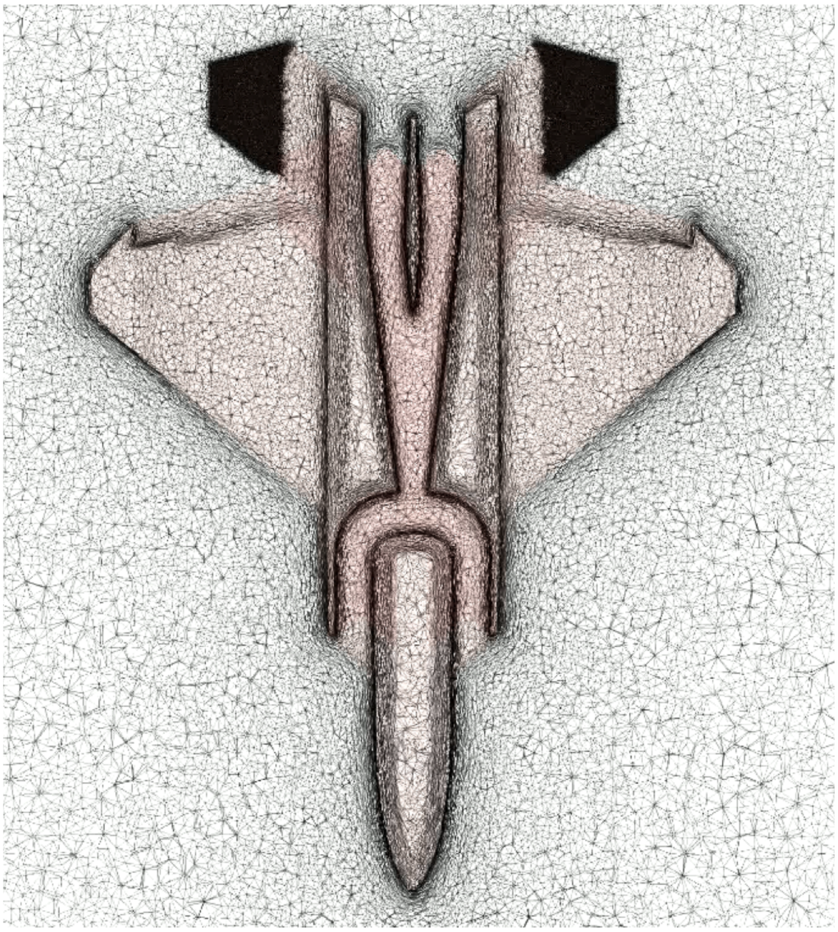
Figure 5-20: F22 Raptor geometry and embedded surface grid.



(a) Side view



(b) Front view



(c) Top view

Figure 5-21: Different view of the non-body fitted mesh obtained using the mesh adaptation algorithm.

The above presented grid was not fully tested at realistic flow conditions because of the lack of reliable data and because of the computational load required to simulate the high Reynolds flow around such large 3D fighter. However few calculations were done to test the Embedded Boundary method with a grid obtained with the mesh adaptation algorithm. The following figure shows the streamlines around and inside the duct of the fighter computed using the Embedded Boundary method equipped with the PhysBAM - Collision-based approach (4-2-2) to track the embedded moving surfaces and the local reconstruction algorithm (4-4-1) to compute the flow induced loads on the embedded surface.

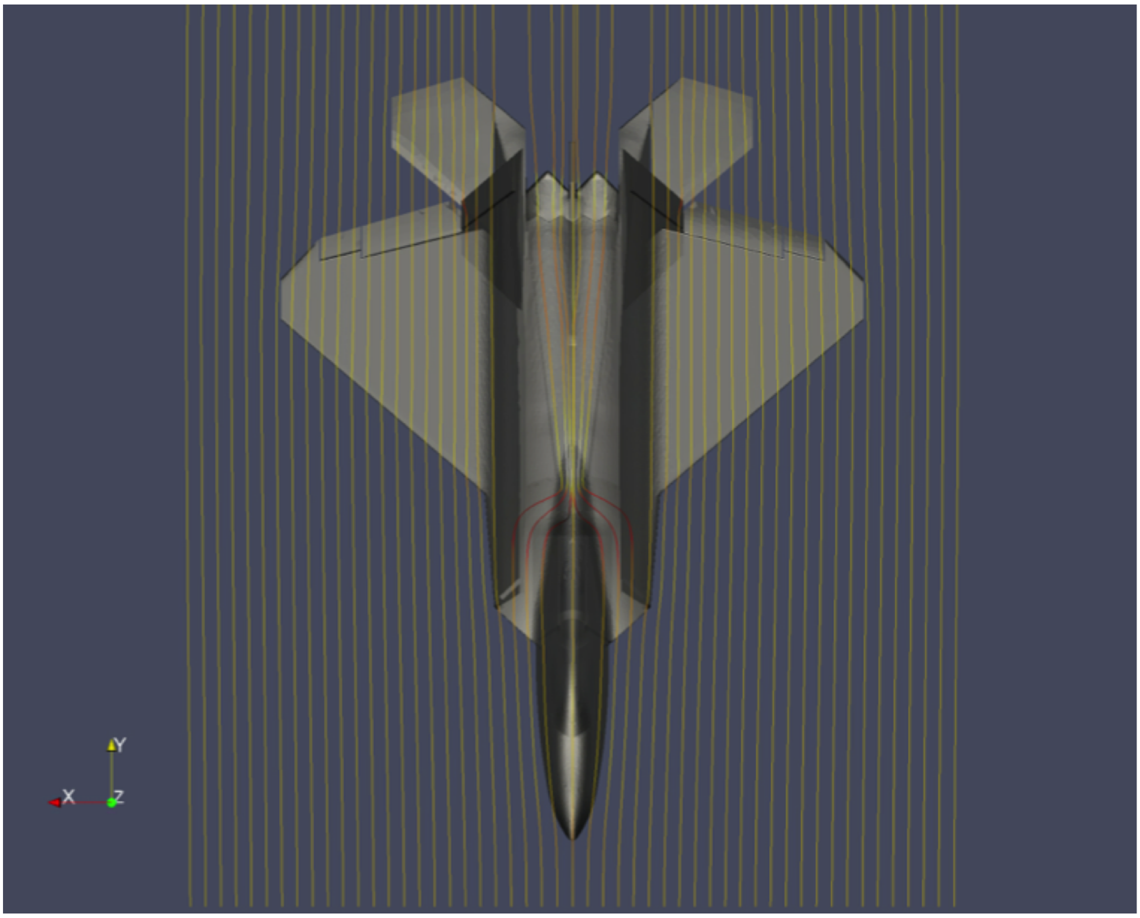


Figure 5-22: F22 Raptor geometry and embedded surface grid.

Chapter 6

Conclusions

This thesis presents an Embedded Boundary method computational framework for the solution of fluid-structure interaction problems involving large and complex structural motions and deformations developed at Stanford University by Prof. Farhat research group. The Embedded Boundary method solves compressible viscous and inviscid flows around moving and deforming structures on non body-conforming CFD grids. The proposed method is equipped with two robust computational algorithms for tracking the discrete embedded interface with respect to arbitrary non body-conforming CFD grid and an algorithm to compute the loads on the structure based on the local reconstruction of the embedded surface.

This method is gaining popularity because it looks a perfect candidate to simulate the flow around complex geometries, moving parts and highly deformable materials that are nowadays used in many aerospace systems and that cannot be modeled using the existing body-fitted mesh methods like the ALE approach.

The flexibility of the method in treating moving structures is demonstrated with the solution of several challenging problems in the fields of aeronautics and an hybrid methodology based on the motion and deformation of the background grid is presented to reduce the grid size needed to simulate the flow around moving and deforming embedded structures. In addition a strategy to design an optimal Eulerian CFD grid for Embedded Boundary method using the level set method is presented and tested for the complex geometry of the F22 Raptor fighter.

Bibliography

[1] J. Su; R. Fedkiw A. Robinson-Mosher; T. Shinar; J. Gretarsson. Two-way coupling of fluids to rigid and deformable solids and shells. *SIGGRAPH 2008 - ACM TOG 27*, pages 46.1–46.9, 2008.

[2] T. C. S. Rendall; C. B. Allen. Unified fluid–structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559, June 2008.

[3] P. R. Spalart; S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA Paper*, 92(0439), 1992.

[4] J. T. Batina. Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA Journal*, 29(3):327–333, 1991.

[5] C. Farhat; A. Rallu; K. Wang; T. Belytschko. Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 84:73–107, 2010.

[6] C. Farhat; P. Geuzaine; G. Brown. Application of a three-field nonlinear fluid- structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers & Fluids*, 32(3):3–29, 2003.

[7] R. Yurkovich; D. Liu; P. Chen. State-of-the-art of unsteady aerodynamics for high performance aircraft. *AIAA Paper No. 2001-0428, 39th AIAA Aerospace Sciences Meeting & Exhibit, Reno NV*, January 2001.

[8] O.A. Kandil; H.A. Chuang. Unsteady vortex-dominated flows around ma-neuvering wings over a wide range of Mach numbers. *AIAA Paper No. 88–0317, 26th AIAA Aerospace Sciences Meeting, Reno NV*, January 1988.

[9] E. Hachem; T. Kloczko; H. Digonnet; T. Coupez. Stabilized finite element solution to handle complex heat and fluid flows in industrial furnace using the immersed volume method. *International Journal for Numerical Methods in Fluids*, 68(1):99–121, 2012.

[10] C. Farhat. CFD on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization. *International Journal of Computational Fluid Dynamics*, 19(8):595–603, November 2005.

[11] Christoph Degand; C. Degand; C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(3-4):305–316, February 2002.

[12] K. Wang; A. Rallu; J.F. Gerbeau; C. Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 67:1175–1206, 2011.

[13] K. Wang; J. Grétarsson; A. Main; C. Farhat. Computational algorithms for tracking dynamic fluid–structure interfaces in embedded boundary methods. *International Journal for Numerical Methods in Fluids*, 70(4):515–535, 2012.

[14] E. Guendelman; A. Selle; F. Losasso; R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. *SIGGRAPH 2005 - ACM TOG 24*, pages 973–981, 2005.

[15] R. Fedkiw. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *Journal of Computational Physics*, 175:200–224, 2002.

[16] Y.H. Tseng; J.H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192:593–623, 2003.

[17] T. E. Noll; J. M. Brown; M. E. Perez-Davis; S. D. Ishmael; G. C. Tiffany; M. Gaier. Investigation of the helios prototype aircraft mishap,. *Mishap Report*, Volume I, January 2004.

[18] I. Malcevic; O. Ghattas. Dynamic-mesh finite element method for lagrangian computational fluid dynamics. *Finite Elements in Analysis and Design*, 38:965–982, 2002.

[19] C. Farhat; P. Geuzaine; C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174:669–694, 2001.

[20] R. Verzicco; J. Mohd-Yusof; P. Orlandi; D. Haworth. Large Eddy Simulation in complex geometric configurations using boundary body forces. *AIAA Journal*, 38(3):427–433, 2000.

[21] R. Mittal; G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[22] R. Verzicco; M. Fatica; G. Iaccarino; P. Moin; B. Khalighi. Large Eddy Simulation of a road vehicle with drag-reduction devices. *AIAA Journal*, 40(12):2447–2455, 2002.

[23] H. Udaykumar; R. Mittal; P. Rampunggoon; A. Khanna. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174:345–380, 2001.

[24] C. Farhat; T.Y Lin. Transient aeroelastic computations using multiple moving frames of reference. *AIAA Paper No. 90–3053, AIAA 8th Applied Aerodynamics Conference. Portland, Oregon, 20–22 August,*, 1990.

[25] C. Farhat; M. Lesoinne; N. Maman. Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids*, 21:807–835, 1995.

[26] F. Cirak; R. Deiterding; S. Mauch. Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to shocks and detonations. *Computers and Structures*, 85:1049–1065, 2007.

[27] J. Mohd-Yosuf. Combined immersed boundary/B-spline methods for simulation on complex gemoetries. *Annual Research Briefs, Center for turbulence research*, 1997.

[28] R. Radovitzky; M. Ortiz. Lagrangian finite element analysis of newtonian fluid flows. *International Journal for Numerical Methods in Engineering*, 43:607–619, 1998.

[29] R. Lohner; J.D. Baum; E.L. Mestreau; D. Sharov; C. Charman; D. Pelessone. Adaptive embedded unstructured grid methods. *AIAA-03*, 1116, 2003.

[30] R. Glowinski; T.W. Pan; A.J. Kearsley; J. Periaux. Numerical simulation and optimal shape for viscous flow by a fictitious domain method. *International Journal for Numerical Methods in Fluids*, 20:695–711, 2005.

[31] R. Glowinski; T.W. Panb; J. Périauxc. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 151(1-2):181–194, January 1998.

[32] C.S. Peskin. Flow patterns around heart valves: a digital computer method for solving the equation of motion. *PhD thesis*, 1972.

[33] C.S. Peskin. The fluid dynamics of heart valves: experimental, theoretical, and computational methods. *Annual Review of Fluid Mechanics*, 14:235–259, 1982.

[34] C.S. Peskin. The immersed boundary method. *Acta Numerica*, pages 479–517, 2002.

[35] C. Farhat; J.F. Gerbeau; A. Rallu. FIVER: A finite volume method based on exact two-phase Riemann problems and sparse grids for multi-material flows with large density jumps. *Journal of Computational Physics*, 231:6360–6379, 2012.

[36] C. L. Bottasso; D. Detomi; R. Serra. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4244–4264, October 2005.

[37] H. Udaykumar; R. Mittal; W. Shyy. Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids. *Journal of Computational Physics*, 153:535–574, 1999.

[38] D. Goldstein; R. Handler; L. Sirovich. Modeling a No-Slip Flow Boundary with an External Force Field. *Journal of Computational Physics*, 105(2):354–366, 1993.

[39] A. Gilmanov; F. Sotiropoulos. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207:457–492, 2005.

[40] T. J. R. Hughes; W. K. Liu; T. Zimmerman. Lagrangian-eulerian finite element formulation for incompressible viscous flow. *Computer Methods in Applied Mechanics and Engineering*, 29:239–249, 1981.