POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea Magistrale in
Ingegneria Aeronautica

Moving Reference Frame and Arbitrary Lagrangian Eulerian approaches
for the study of moving domains in Typhon

Relatore: Prof. Alberto Guardone

Co-relatore: Prof. Gilles Grondin

Tesi di Laurea di:

Alessandro Giacomo Maria GARDI
Matr. 711665

Anno Accademico 2010 – 2011.

Acknowledgments

My sincere thanks go firstly to Prof. A.Guardone for allowing me to attend to this significant international experience and for the huge patience he had with me along the foreign stay and moreover in the last periods after my return.

I'd also especially like to thank M G.Grondin and M J.Gressier for all their patience, their expertise, their precious hints and suggestions during the six months of my stay at the DEAP of ISAE-SUPAERO, where I learned to master my Fortran95 sourcecode development and testing skill, *nix shell scripting and take advantage of the powerful calculation systems available.
Thanks also to Mme Dabrainville, for assisting me in the research of the final project opportunity, and for explaining me all the necessary fulfillments and the students benefits at ISAE-SUPAERO.

Lastly, sincere thanks go to my parents, relatives and friends for having supported and put up with me in these critical periods.

Contents

Abstract

Among the various possible methods for treating moving domains in Computational Fluid Dynamics (CFD) studies, the Moving Reference Frame (MRF) and Arbitrary Lagrangian Eulerian (ALE) approaches are being extensively studied, developed and tested in Typhon, a three dimensional unstructured cell-centered finite volume parallel CFD solver written in Fortran(90/95) and freely available on the SourceForge open source web portal. Even if the selected and developed versions are among the basic and simplest availables, significant functionality improvements to the solver are achieved without invasive changes to the original code and causing a minimal computational burdening. The evalued test cases show positive correspondence to the known exact solutions, a part from expected surmountable anomalies that appear using particular numerical schemes. A considerable degree of freedom is also left for further developments.

Chapter 1

Introduction

1.1 The analysis of unsteady fluid flows around moving bodies and boundaries

The present work is aimed at extending the features of an existing computational fluid-dynamics solver (Typhon) to the solution of fluid flows in unsteady geometric domains, thus basically around moving bodies and boundaries. Topics of noticeable and recent interest such as the study on the trails of single or contra-rotating propellers and fans, the secondary flows inside turbomachines, the aeroelastic flutter phenomena of wings, rotorcraft blades or even buildings and chimneys are all examples requiring moving domain capabilities inside the fluid-dynamics solvers.
Despite the study of unsteady flows has always been of primary importance since the dawn of computational fluid-dynamics, the analysis of the flows around moving bodies and boundaries grew a noticeable spread only later on: even if several theoretical results for the analytical solutions have been already available since pre-war studies, the lack of an adequate scientific background on numerical approaches and moreover the limited computational speeds available have always been critical issues to overcome. Other potential issues were the the need of unsteady boundary conditions, for which adequate support by the numerical solver had to be developed, and even transient initial conditions, for which further preliminary calculations were necessary, thus improving the criticity of computational powers available.
Nowadays, methods for the treatment of moving domains are more and more commonly available both in commercial and academic solvers and so the study of fluid flows around or inside moving domains is finally widespread both in the research field and in the development and design fields. The support for unsteady geometries is therefore a more and more compelling requirement for the vast majority of both specialized and non-specialized solvers, and a noticeable interest is also gathered around the look for a best possible approach in terms of computational quickness, flexibility, robustness, reliability and overall simplicity.

1.2 Structure of the work

In the present introductory Chapter (1), we present the main approaches for treating moving domains in Computational Fluid-Dynamics solvers, some of their prominent advantages and drawbacks, the reasons behind the choices of Moving Reference Frame and Arbitrary Lagrangian-Eulerian methods as subjects of the this work, the global organization of the work and its objectives.

In Chapter 2 we present more in depth the Moving Reference Frame approach, its analytical derivations, its implementations and validations in Typhon.

In Chapter 3 we present more in depth the Arbitrary Lagrangian-Eulerian approach, its analytical derivations, its implementations and its validations in Typhon.

In Chapter 4 we present and comment the parameters and results of an analysis campaign on a NACA 0012 airfoil flutter, which serves as a prominent comparison case between Moving Reference Frame and Arbitrary Lagrangian-Eulerian approaches, and the experimental results.

Lastly, Chapter 5 is focused on the overall conclusions of the work.

1.3 Overview of the presently available numerical strategies

Considering all the possible variations, the number of proposed numerical methods to account for the domain movement available in the literature is noticeable. By restricting the search around some topical tasks, among which are the study of rotating propeller blades, turbomachines channels and wings in flutter, the possible numerical approaches are the following:

- a pure Lagrangian solver, with body movement achieved thanks to the assignment of strict boundary conditions. The theoretical literature behind is vast, and other main advantages are that the domain deformations and movement are inborn features and so is also the study of "free surface" problems. The drawbacks are the constant need for untangling, remeshing and remapping of the fluid field, even for steady domains, the extensive source code rewriting that would be needed due to the Eulerian typology of Typhon, the delicate and complex cell centered finite volume formulation and the degradation on the shape of the bodies/boundaries after several iterations;
- a Moving Reference Frame rigidly fixed on the moving body, for which the solver must correctly handle the additional non-inertial forces. The critical advantages are the very simple theoretical formulation, the minimal source

code interventions needed and the complete absence of mesh updates and remeshing steps. Among the drawbacks we must consider that only rigid movements can be achieved (at least in the presently selected and developed version of MRF), that the whole domain is moving so mutual relative movements are only available as boundary conditions, that the additional contributes are in the form of source terms, thus potentially introducing additional errors, and that the analysis input and outputs must be defined in the relative reference system, so transformation routines will be needed;

- an Arbitrary Lagrangian-Eulerian approach, that enables free mesh movement thanks to the introduction of additional flux terms in the solver to account for it. It statedly combines the advantages of Eulerian and Lagrangian approaches while attempting at minimizing their drawbacks. It is therefore credited as ideal for fluid-body interaction studies, but depending on the formulation can manifest great versatility and flexibility and can also neglect remeshing in selected cases. Recognized disadvantages are the vastness of the family of methods laying behind the same name, the usefulness of remeshing and remapping capabilities to prevent degraded performances and solution qualities, and the complexity of free surface modeling, contrary to the pure Lagrangian methods;

- an over-set grid method, such as "Chimera", that allows domain movement thanks to the relative movement of the grids, but requiring an intensive solver's sourcecode rewriting and a more complex preliminary treatment of the analysis cases, thus falling beyond the frames of the present work.

Among the discussed possible choices, the Moving Reference Frame and Arbitrary Lagrangian-Eulerian approaches have been selected, developed, validated and tested since they statedly offered the best features increase while being integrally compatible with the original solver's source code. As we'll promptly explain later on, limited versions of the Moving Reference Frame and Arbitrary Lagrangian-Eulerian extensions have been chosen and developed in the present work. More precisely, only a rigid non-inertial Moving Reference Frame has been adopted, avoiding the theoretical and practical complexity of a non-rigid frame, which could grant only limited further funcionality improvements. Concerning the Arbitrary Lagrangian-Eulerian extension, no remeshing and remapping capabilities have been introduced, thus restricting the field to small body deformations and movements, which are considered nonetheless significant funcionality improvements, yet remaining perfectly fit for further developments and extensions which could include any possible remeshing method.

1.4 Objectives and organization of the work

The objectives of the present work, ordered as the original planning, are:

- the review of the original Typhon source code;
- the analytical derivation of a Moving Reference Frame approach which combines versatility and feasibility;
- the development of the previously derived Moving Reference Frame approach inside Typhon's source code;
- the validation of the Moving Reference Frame module thanks to trivial test cases, for which an exact solution is well known;
- the analytical derivation of an Arbitrary Lagrangian-Eulerian approach, again combining versatility and feasibility;
- the development of the previously derived Arbitrary Lagrangian-Eulerian approach inside Typhon's source code;
- the validation of the Arbitrary Lagrangian-Eulerian approach thanks to trivial test cases, for which an exact solution is well known;
- an analysis campaign of flutter cases, for which both Moving Reference Frame and Arbitrary Lagrangian-Eulerian solutions are possible and experimental results are available, thus providing an overall comparison of the methods.

Part I

Theory and developments

Chapter 2

The Moving Reference Frame approach

2.1 Preliminary considerations

As previously stated in the general introduction, a very basic method providing a reliable way to account for moving geometries in the computational domain is known as the "Moving Reference Frame" approach, for which we actually move the whole study in a non-inertial relative reference system. Even if not mathematically complex, the Moving Reference Frame approach can already provide a significant functionality improvement to the CFD solver, since it enables a thorough study of fluid flows around bodies or boundaries in rigid motion, the latter being a three-dimensional accelerating or oscillating rotation or translation, while avoiding mesh deformations issues completely.

2.1.1 The non-inertial MRF for moving geometries

A consistent CFD study of rotating or oscillating bodies such as propellers and flutter studies for airfoils or finite three dimensional rigid wings can be easily performed by simply introducing the rigid motion as a parameter for the analysis of the fixed mesh containing the aforesaid body/boundary. The solver will in fact work in a relative observer's perspective, evaluating the fluid field in a single mesh representing a fixed domain, thus around fixed bodies and boundaries; those latter, thanks to the introduction of the non-inertial terms in the solver's equations, will actually correspond to moving geometries in the inertial perspective. The only significant limitation of the MRF for moving geometries is that a single global rigid law of motion must be inferred for all the bodies in the domain, that are actually rigid boundaries in the mesh. Some improvements may be obtained by using an expanded set of source terms for the fluid dynamics equations, including expansion/contraction terms that may account for some relative motion between the rigid boundaries, but the mathematical complexity of the MRF parameters

will raise quickly for very little practical advantage, so other methods may be more profitably used instead.

2.1.2 The non-inertial MRF for large translational velocities

The Moving Reference Frame approach counts another less manifest feature: a very large global component in fluid motion, like in "Hubble flows" and among high Mach number problems generally, is usually source of significant numerical errors in a traditional inertial reference frame study. The ratio between the thermal energy and the kinetic energy is in fact extremely small due to the superimposed global motion, so the numerical solution in a floating point environment will manifest large errors on the thermal energy and thus on the pressure field, compromising the quality of velocity and density solutions too. In the Moving Reference Frame approach, the global motion is completely detracted from the numerical computation, so that even very little local variations can be observed and precisely calculated. This aspect puts the Moving Reference Frame approach in maximum consideration for astrophysical and plasma nuclear magneto fluid-dynamics studies: rotating, expanding or contracting fluid flows in accelerating volumes, rotating ducts, collapsing stellar cores and pyrotechnic or supernovae explosions, or even in Inertial Confinement Fusion (ICF) problems can be successfully studied in a consistent, precise and much simpler way [1].

The aforesaid feature correspond, on the other hand, to another minor disadvantage for the MRF analysis of moving bodies: big domains will easily grow large fictitious advective components even for slow angular velocities, thus degrading the quality of the real solution in the absolute reference frame.

2.1.3 Basic concepts and setbacks of the MRF approach

In a Moving Reference Frame analysis, a full reference frame transformation is actually done and not a simple coordinate transformation like in Moving Mesh methods (among which the Arbitrary Lagrangian Eulerian can be included): both the solver inputs, the variables and the outputs becomes integrally relative, and not just in the sense they are functions of a new relative coordinate system, like in the Moving Mesh approaches. While conceptually more complicated, the development setbacks are pretty straightforward: developing a Moving Reference Frame extension to the solver will initially consist, as we will discuss further on, in the simple introduction of the non-inertial terms, such as fictitious forces and expansion/contraction effects, in the original Euler/Navier-Stokes equations. As already stated previously, by adopting this basic technique alone any mesh update is actually neglected, at least at the beginning: mathematical and development complexities and computational time will then be noticeably lower than in other methods. On the other hand, if analysis input parameters and output requirements are in the absolute reference frame, which is a very typical case, the development of transformations from/to the solver's now relative reference system will be needed. This

section will aim at the determination and explanation of the additional terms, the developments aspects in Typhon and at the review of the final results obtained by analyzing very simple cases for which a theoretical solution is known, thus providing a validation.

2.2 Governing equations

2.2.1 Derivation of the Navier-Stokes equations in a non-inertial reference frame

Considering a material point P seen from both an inertial frame "1" and a non-inertial frame "2", and accounting for both the translation and the rotation of frame "2", the following kynematical relations can be obtained:

$$\begin{aligned} & \boldsymbol{r}_{1P}(t) = \boldsymbol{r}_{12}(t) + \boldsymbol{r}_{2P}(t) \quad \Rightarrow \quad \boldsymbol{r}_{1P}(t) = \boldsymbol{r}_{12}(t) + \|\boldsymbol{r}_{2P}\|\hat{r}_{2P} \\ \Rightarrow \quad & \frac{d}{dt} \quad \Rightarrow \quad \boldsymbol{u}_{1P} = \boldsymbol{u}_{12} + \boldsymbol{u}_{2P} + \boldsymbol{\omega} \times \boldsymbol{r}_{2P} \\ \Rightarrow \quad & \frac{d}{dt} \quad \Rightarrow \quad \dot{\boldsymbol{u}}_{1P} = \dot{\boldsymbol{u}}_{12} + \dot{\boldsymbol{u}}_{2P} + 2\boldsymbol{\omega} \times \boldsymbol{u}_{2P} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \boldsymbol{r}_{2P} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r}_{2P} \end{aligned} \tag{2.1}$$

It is then convenient to rename the variables as follows:

$$\begin{aligned} \boldsymbol{u}_{1P} &= \boldsymbol{u} \\ \boldsymbol{u}_{2P} &= \widetilde{\boldsymbol{u}} \\ \boldsymbol{u}_{12} &= \boldsymbol{v} \end{aligned}$$

The material acceleration in a non-inertial reference frame can then be written as follows:

$$\dot{\widetilde{\boldsymbol{u}}} = \dot{\boldsymbol{u}} - \dot{\boldsymbol{v}} - 2\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}} - \boldsymbol{\omega} \times \boldsymbol{\omega} \times \widetilde{\boldsymbol{r}} - \dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}} \tag{2.2}$$

where:
$\dot{\boldsymbol{v}}$ is the frame "2" linear acceleration.
$2\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}$ is the "Coriolis acceleration".
$\boldsymbol{\omega} \times \boldsymbol{\omega} \times \widetilde{\boldsymbol{r}}$ is the "centrifugal acceleration".
$\dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}}$ is due to the frame "2" angular acceleration.

The additional fictitious forces that account for both the translation and the rotation of the non-inertial frame will then be:

$$-\widetilde{\rho}\dot{\boldsymbol{v}} - 2\widetilde{\rho}\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}} - \widetilde{\rho}\boldsymbol{\omega} \times \boldsymbol{\omega} \times \widetilde{\boldsymbol{r}} - \widetilde{\rho}\dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}} \tag{2.3}$$

that is, four negative contributes in the right hand side of the momentum equation.

By simply scalar multiplication for the velocity vector, the aforesaid additional momentum terms give birth, in turn, to subsequent energy terms:

$$-\widetilde{\rho}\widetilde{\boldsymbol{u}} \cdot \dot{\boldsymbol{v}} - 2\widetilde{\rho}\widetilde{\boldsymbol{u}} \cdot (\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}) - \widetilde{\rho}\widetilde{\boldsymbol{u}} \cdot (\boldsymbol{\omega} \times \boldsymbol{\omega} \times \widetilde{\boldsymbol{r}}) - \widetilde{\rho}\widetilde{\boldsymbol{u}} \cdot (\dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}}) \tag{2.4}$$

Original system of Navier-Stokes equations

The original system of Navier-Stokes equations adopted for the whole work takes the form:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \\ \dfrac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla P + \rho \boldsymbol{b} + \nabla \cdot \boldsymbol{\sigma} \\ \dfrac{\partial \rho e}{\partial t} + \nabla \cdot (\boldsymbol{u}(\rho e + P)) = \nabla \cdot (k \nabla T + \boldsymbol{u} : \boldsymbol{\sigma}) \end{cases} \tag{2.5}$$

The system of Euler equations will not be explicitly described since it can always be considered a sub-case of the Navier-Stokes system and all the subsequent analytical achievements are valid in both the cases.

Navier-Stokes equations in a non-inertial reference frame

By introducing the formerly deferred additional non-inertial terms in both the momentum and the energy equations, the original system of Navier-Stokes equations becomes:

$$\begin{cases} \dfrac{\partial \widetilde{\rho}}{\partial t} + \widetilde{\boldsymbol{\nabla}} \cdot (\widetilde{\rho} \widetilde{\boldsymbol{u}}) = 0; \\ \dfrac{\partial \widetilde{\rho} \widetilde{\boldsymbol{u}}}{\partial t} + \widetilde{\boldsymbol{\nabla}} \cdot (\widetilde{\rho} \widetilde{\boldsymbol{u}} \otimes \widetilde{\boldsymbol{u}}) = -\widetilde{\boldsymbol{\nabla}} \widetilde{P} + \widetilde{\rho} \widetilde{\boldsymbol{b}} + \widetilde{\boldsymbol{\nabla}} \cdot \widetilde{\boldsymbol{\sigma}} \\ \qquad - \widetilde{\rho} \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \widetilde{\boldsymbol{r}}) \quad \Leftarrow \quad \text{centrifugal} \\ \qquad - 2 \boldsymbol{\omega} \times \widetilde{\rho} \widetilde{\boldsymbol{u}} \quad \Leftarrow \quad \text{Coriolis} \\ \qquad - \widetilde{\rho} \dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}} \quad \Leftarrow \quad \text{unsteadiness (rotational)} \\ \qquad - \widetilde{\rho} \dot{\boldsymbol{v}} \quad \Leftarrow \quad \text{translational inertia} \\ \dfrac{\partial \widetilde{\rho e}}{\partial t} + \widetilde{\boldsymbol{\nabla}} \cdot (\widetilde{\boldsymbol{u}} (\widetilde{\rho e} + \widetilde{P})) = \widetilde{\rho} \widetilde{\boldsymbol{u}} \cdot \widetilde{\boldsymbol{b}} + \widetilde{\boldsymbol{\nabla}} \cdot (k \widetilde{\boldsymbol{\nabla}} \widetilde{T} + \widetilde{\boldsymbol{u}} : \widetilde{\boldsymbol{\sigma}}) \\ \qquad - \widetilde{\rho} \widetilde{\boldsymbol{u}} \cdot (\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \widetilde{\boldsymbol{r}})) \\ \qquad - 2 \widetilde{\rho} \widetilde{\boldsymbol{u}} \cdot (\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}) \\ \qquad - \widetilde{\rho} \widetilde{\boldsymbol{u}} \cdot (\dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}}) \\ \qquad - \widetilde{\rho} \widetilde{\boldsymbol{u}} \cdot \dot{\boldsymbol{v}} \end{cases} \tag{2.6}$$

Before adopting the finite approximation form, a compulsory passage is the

following integral formulation:

$$\begin{cases} \displaystyle\int_V \frac{\partial\widetilde{\rho}}{\partial t}dV + \oint_S (\widetilde{\rho}\widetilde{\boldsymbol{u}})\cdot\hat{n}dS = 0; \\ \displaystyle\int_V \frac{\partial\widetilde{\rho}\widetilde{\boldsymbol{u}}}{\partial t}dV + \oint_S (\widetilde{\rho}\widetilde{\boldsymbol{u}}\otimes\widetilde{\boldsymbol{u}})\cdot\hat{n}dS = \int_V \widetilde{\rho}\widetilde{\boldsymbol{b}}dV + \oint_S (\widetilde{\boldsymbol{\sigma}} - \widetilde{P}\boldsymbol{I})\cdot\hat{n}dS \\ \qquad - \displaystyle\int_V \widetilde{\rho}\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\widetilde{\boldsymbol{r}})dV - 2\int_V \boldsymbol{\omega}\times\widetilde{\rho}\widetilde{\boldsymbol{u}}dV \\ \qquad - \displaystyle\int_V \widetilde{\rho}\dot{\boldsymbol{\omega}}\times\widetilde{\boldsymbol{r}}dV - \int_V \widetilde{\rho}\dot{\boldsymbol{v}}dV \\ \displaystyle\int_V \frac{\partial\widetilde{\rho}\widetilde{e}}{\partial t}dV + \oint_S (\widetilde{\boldsymbol{u}}(\widetilde{\rho}\widetilde{e}+\widetilde{P}))\cdot\hat{n}dS = \int_V \widetilde{\rho}\widetilde{\boldsymbol{u}}\cdot\widetilde{\boldsymbol{b}}dV + \oint_S (k\widetilde{\boldsymbol{\nabla}}\widetilde{T} + \widetilde{\boldsymbol{u}}:\widetilde{\boldsymbol{\sigma}})\cdot\hat{n}dS \\ \qquad - \displaystyle\int_V \widetilde{\rho}\widetilde{\boldsymbol{u}}\cdot(\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\widetilde{\boldsymbol{r}}))dV - \int_V \widetilde{\rho}\widetilde{\boldsymbol{u}}\cdot\dot{\boldsymbol{v}}dV \\ \qquad - \displaystyle\int_V \widetilde{\rho}\widetilde{\boldsymbol{u}}\cdot(\dot{\boldsymbol{\omega}}\times\widetilde{\boldsymbol{r}})dV - 2\int_V \widetilde{\rho}\widetilde{\boldsymbol{u}}\cdot(\boldsymbol{\omega}\times\widetilde{\boldsymbol{u}})dV \end{cases} \tag{2.7}$$

for which we took advantage of the well known Gauss' Divergence Theorem (cfr. appendix).

2.2.2 Finite approximation of the non-inertial MRF terms

In order to derive the development formulation for the Typhon solver, the last compulsory passage is the finite approximation of the MRF source terms. Inside the solver, all the geometric and calculation parameters and the fluid-dynamical unknowns $\widetilde{\rho}$, $\widetilde{\rho}\widetilde{\boldsymbol{u}}$ and $\widetilde{\rho}\widetilde{e}$ exist in their discrete values (arrays and matrices) only. High order extrapolation is not feasible at this point, and would seriously compromise the generality of the approach. A strong approximation is then introduced here: even if most of the parameters and variables, such as $\widetilde{\rho}$, $\widetilde{\rho}\widetilde{\boldsymbol{u}}$, $\widetilde{\rho}\widetilde{e}$ and $\widetilde{\boldsymbol{r}}$, manifest a dependence on the spatial position and thus on the radius, which would be perceptible even inside each single cell, the discrete form we adopt for the volume integrals for each cell will completely neglect the trend within the cells. This decision leads to the following discrete formulations:

$$\begin{aligned} -\int_V \widetilde{\rho}\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\widetilde{\boldsymbol{r}})dV &\longrightarrow -m_{cell}\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\widetilde{\boldsymbol{r}}_{cell}) \\ -2\int_V \boldsymbol{\omega}\times\widetilde{\rho}\widetilde{\boldsymbol{u}}dV &\longrightarrow -2V_{cell}\boldsymbol{\omega}\times(\widetilde{\rho}\widetilde{\boldsymbol{u}})_{cell} \\ -\int_V \widetilde{\rho}(\dot{\boldsymbol{\omega}}\times\widetilde{\boldsymbol{r}})dV &\longrightarrow -m_{cell}(\dot{\boldsymbol{\omega}}\times\widetilde{\boldsymbol{r}}_{cell}) \\ -\int_V \widetilde{\rho}\dot{\boldsymbol{v}}dV &\longrightarrow -m_{cell}\dot{\boldsymbol{v}} \end{aligned} \tag{2.8}$$

where the integrals are basically computed as $\int_V \widetilde{\rho} dV = m_{cell}$, so $\widetilde{\rho}(\widetilde{\boldsymbol{r}}, t) \simeq \widetilde{\rho}_{cell}(t)$ etc. It shall be noted, nonetheless, that Typhon's cell center, where every cell quantity is located, is actually a "volumetric barycenter", so this approximation error is slightly lower than if it was a simple algebraic average of the vertex positions.

2.2.3 Kinematics

By adopting relative dynamics, we have just discovered the appearance of new terms which are function of either the velocity or the acceleration, and yet even in the successive developments of transformation routines we'll realistically observe the contemporary presence of velocity and acceleration vectors. This fact alone has very important setbacks: as well known, angular acceleration vector is the first time derivative of the angular velocity vector, so the first one is completely known as soon as the second is. A single time derivation step links the two, but from the development point of view this isn't so trivial: computers don't have an inborn derivation capability. They can be "taught" derivation in two possible way:

- finite numerical derivation
- symbolic analytical derivation

Both of them are widely and successfully undertaken, but hide some drawbacks: the numerical derivation forcefully gives rise to computational errors that depend on the derivation algorithm, on the step and on the to-be-derived function but can't be fully overcome, whereas the symbolic analytical derivation requires a very extensive development, even if an already available mathematical library is going to be used, and this would fall seriously beyond the goals of the proposed work. Two remaining choices were left:

- the end user's supply of both the velocity and acceleration symbolic relations as analysis input parameters
- providing a limited set of kinematical cases for which the symbolic relations for both position, velocity and acceleration are hard-coded in their exact form

The first choice is flaw-vulnerable since no verification is performed on the correctness and coherence of the input functions, so problems difficult to discover and unexpected results may appear. The second choice was therefore pursued, also because it perfectly fitted the goals of the proposed work.

Since the flutter study was the original final objective of the whole work, separate and combined angular and translational oscillations were introduced. Constant angular rotation and linear translational acceleration were also included since they both provided for valuable results validation and for useful common study cases.

Constant translational acceleration

The linear translation case is pretty straightforward: the moving reference frame is undergoing a single constant acceleration vector that will develop its initial velocity vector. There's no dependence from the initial position, that is therefore omitted. We then obtain the well known uniformly accelerated motion:

$$\begin{aligned} \boldsymbol{s}(t) &= \boldsymbol{v}_0 t + \frac{1}{2}\boldsymbol{a}t^2; \\ \dot{\boldsymbol{s}}(t) &= \boldsymbol{v}_0 + \boldsymbol{a}t; \\ \ddot{\boldsymbol{s}}(t) &= \boldsymbol{a}; \end{aligned} \tag{2.9}$$

The only active MRF source terms will be $\widetilde{\rho}\dot{\boldsymbol{v}}$ in the momentum equation and its corresponding one in the energy equation.

Constant rotation

A constant initial angular velocity is the only free parameter for the constant rotational speed case. As before, there's no dependence from the initial position.

$$\begin{aligned} \boldsymbol{\vartheta}(t) &= \boldsymbol{\omega}t; \\ \dot{\boldsymbol{\vartheta}}(t) &= \boldsymbol{\omega}; \\ \ddot{\boldsymbol{\vartheta}}(t) &= 0; \end{aligned} \tag{2.10}$$

The only active MRF source terms will be $\widetilde{\rho}\boldsymbol{\omega} \times \boldsymbol{\omega} \times \widetilde{\boldsymbol{r}}$ and $2\widetilde{\rho}\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}$, that is, centrifugal and Coriolis forces in the momentum equation and their corresponding ones in the energy equation.

Oscillating translation

The translational oscillation case incorporates also the constant acceleration and initial velocity vectors. Again, there's no dependence from the initial position. The equations are as follows:

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{v}_0 t + \frac{1}{2}\boldsymbol{a}t^2 + A_{trns}\, sin\left(\frac{2\pi}{T_{trns}}t + \varphi_{trns}\right)\hat{\xi}; \\ \dot{\boldsymbol{s}} &= \boldsymbol{v}_0 + \boldsymbol{a}t + A_{trns}\,\frac{2\pi}{T_{trns}} cos\left(\frac{2\pi}{T_{trns}}t + \varphi_{trns}\right)\hat{\xi}; \\ \ddot{\boldsymbol{s}} &= \boldsymbol{a} - A_{trns}\,\frac{4\pi^2}{T_{trns}^2} sin\left(\frac{2\pi}{T_{trns}}t + \varphi_{trns}\right)\hat{\xi}; \end{aligned} \tag{2.11}$$

The free parameters are then the already met $\boldsymbol{v}_0$ and $\boldsymbol{a}$, the oscillation amplitude A_{trns}, period T_{trns}, phase φ_{trns} and direction $\hat{\xi}$.
Despite the complexity of the law of motion, the only MRF source term active is $\widetilde{\rho}\dot{\boldsymbol{v}}$ in the momentum equation and its corresponding one in the energy equation, like in the constant translational acceleration case.

Oscillating rotation

The rotational oscillation case includes the constant rotation case too. As usual, there's no dependence from the initial position. The equations are then:

$$\begin{aligned}
\boldsymbol{\vartheta} &= \boldsymbol{\omega}t &+ A_{rot} \quad & sin\left(\frac{2\pi}{T_{rot}}t + \varphi_{rot}\right)\hat{\zeta}; \\
\dot{\boldsymbol{\vartheta}} &= \boldsymbol{\omega} &+ A_{rot}\frac{2\pi}{T_{rot}} \quad & cos\left(\frac{2\pi}{T_{rot}}t + \varphi_{rot}\right)\hat{\zeta}; \\
\ddot{\boldsymbol{\vartheta}} &= &- A_{rot}\frac{4\pi^2}{T_{rot}^2} \quad & sin\left(\frac{2\pi}{T_{rot}}t + \varphi_{rot}\right)\hat{\zeta};
\end{aligned} \tag{2.12}$$

The free parameters are then the already met $\boldsymbol{\omega}$, the oscillation amplitude A_{rot}, period T_{rot}, phase φ_{rot} and axis $\hat{\zeta}$.
The active MRF source terms will be all the rotational ones, that is $\widetilde{\rho}\boldsymbol{\omega} \times \boldsymbol{\omega} \times \widetilde{\boldsymbol{r}}$, $2\widetilde{\rho}\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}$ and $\widetilde{\rho}\dot{\boldsymbol{\omega}} \times \widetilde{\boldsymbol{r}}$ and their corresponding ones in the energy equation.

Combined oscillation

The combined oscillation case includes both the translational and the rotational oscillations superimposed, thus retaining their original kynematical relations.
The free parameters are then $\boldsymbol{\omega}$, $\boldsymbol{v}_0$, $\boldsymbol{a}$, A_{rot}, A_{trns}, T_{rot}, T_{trns}, φ_{rot}, φ_{trns}, $\hat{\zeta}$ and $\hat{\xi}$.
All the derived MRF source terms will be active and this case will then be the heaviest, computationally speaking.

2.3 Implementations in Typhon

Several different steps had to be addressed during the development phase:

- creation of an "MRF" data container, to hold all the new parameters and any data useful along the calculation;
- recognition and assignment of the new MRF parameters from the analysis input files;
- introduction of the source terms in the solver's fluid dynamics equations;
- development of transformation routines to elaborate the input data for the source terms.

2.3.1 The MRF data structure

A FORTRAN structure, passed as an input/output argument for the solver routines, was considered the best choice, a part from being the standard solution in the whole Typhon sourcecode, and was therefore adopted. The mnu_mrf structure, defined in "$CFDTOOLS/Mesh/MESHMRF.f90$" contains thus all the useful and pertinent parameters and data, avoiding the proliferation of orphan variables in the subroutines. Particularly, it contains:

- the string "*name*";
- the integer parameters "*type*", "*input*" and "*output*", for the choice of the kynematical model and of the absolute/relative reference frame of the parameters respectively;
- the three-dimensional vectors "*velocity*" and "*acceleration*", for the constant linear acceleration and the oscillating translation cases;
- the three-dimensional vectors "*center*" and "*rot_axis*" and the scalar "*omega*", for the constant rotation and the oscillating rotation cases;
- the three-dimensional vector "*trn_dir*" and the scalars "*trn_period*", "*trn_ampl*" and "*trn_phi*" for the oscillating translation;
- the scalars "*rot_period*", "*rot_ampl*" and "*rot_phi*" for the oscillating rotation case.

2.3.2 MRF Input parameters reading and processing

In Typhon, all the analysis parameters are gathered in a text file called "main.rpm" or eventually in children parameter files linked from it, in the analysis folder. Each category is normally associated with a "BLOCK", so a "BLOCK:MRF" was promptly created to contain all the MRF parameters. The source file "$def\_mrf.f90$"

contained in "$SOURCE/PARAM$" provides parsing and some preliminary computation on the analysis parameters. An example of main.rpm MRF section is as follow:

$$
\begin{array}{ll}
BLOCK : MRF & \\
\quad NAME & =oscillating\_airfoil \\
\quad TYPE & =COMBINED\_OSCILLATION \\
\quad INPUT & =ABSOLUTE \\
\quad OUTPUT & =RELATIVE \\
\quad CENTER & =(0., 0., 0.) \\
\quad AXIS & =(0., 0., -1.) \\
\quad CENTER\_ACCELERATION & =(0., 0., 0.) \\
\quad CENTER\_VELOCITY & =(0., 0., 0.) \\
\quad OMEGA\_RPM & =0. \\
\quad ROT\_AMPL & =0.420001336 \\
\quad ROT\_PERIOD & =0.245700245 \\
\quad ROT\_PHI & =0. \\
\quad TRANS\_AMPL & =0.005334 \\
\quad TRANS\_PERIOD & =0.245700245 \\
\quad TRANS\_DIR & =(0., 1., 0.) \\
\quad TRANS\_PHI & = -3.0805 \\
ENDBLOCK &
\end{array}
\tag{2.13}
$$

Both scalars and three-dimensional vectors are present, but no symbolic relation of time or space can be inserted: by introducing the limited set of kynematical relations, this degree of freedom has been forcefully eliminated.

2.3.3 MRF Source terms

In Typhon, all the right hand side contributes to the Euler / Navier-Stokes equations are gathered in the field of "residuals", which includes separate mass and energy scalar residuals and momentum vector residuals. This field is reset to zero at the beginning of each computational iteration, and successively grows thanks to the contributes of both source terms and flux terms. The previously derived finite approximations of the Moving Reference Frame souce terms are already in the right form to be added to the residuals field, and this task is being done by the "$calc\_source\_mrf(umesh, field, mrf, curtime)$" subroutine, located in the "$calc\_source\_mrf.f90$" source file inside the "$SOURCE/EQNS$" folder.

2.3.4 Transformation routines for the MRF

Leaving the end users free to choose whether to provide absolute or relative initial and boundary conditions and whether obtain absolute or relative outputs, required the creation of several transformation subroutines, that silently process the available data every time it is needed. The subroutines "$mrfvel\_abs2rel(mrf, time, pos, velocity)$", "$mrfvel\_rel2abs(mrf, time, pos, velocity)$" and "$mrfpos\_rel2abs(mrf, time, vertex)$" provide respectively velocity transformation from absolute to relative reference frames, velocity transformation from relative to absolute reference frames and positions transformation from relative to absolute reference frames; they are all located in the "$MESHMRF.f90$" source file inside the "$CFDTOOLS/Mesh$" folder.

2.4 Validation cases and results

This section is aimed at testing and validating the robust and reliable behavior of the previously derived additional non-inertial source terms. For all the following analyses, the time integration step is automatically calculated by Typhon as the smallest one which can grant global stability on the whole domain complying with the Courant-Friedrichs-Levy condition on the smallest cell with a unary CFL parameter. Whenever laminar viscous Navier-Stokes calculations are pursued, gaseous air in standard conditions is the adopted fluid.

It shall furtherly be noted that since the solver behavior with different size meshes, spatial schemes (numerical fluxes) and time-marching schemes was not mastered at this point, a vast analysis campaign, counting almost all possible combinations, was pursued. This level of detail will not be reached in the ALE validation and flutter analysis, since the present MRF results are valid in all cases, so only the best combinations will be retained.

2.4.1 The accelerating/decelerating rectangles and cuboids

The first trivial test case established is aimed at checking the correct operativity of the translational fictitious force term. The mesh used are a 2D rectangle, measuring 1 x 0.1 meters - 100 x 10 cells, and a 3D rectangular cuboid (parallelepiped), measuring 2 x 3 x 8 meters - 6 x 9 x 24 cells.

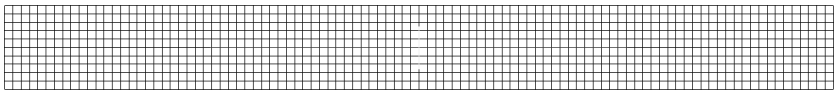

Figure 2.1: The 2D rectangle mesh

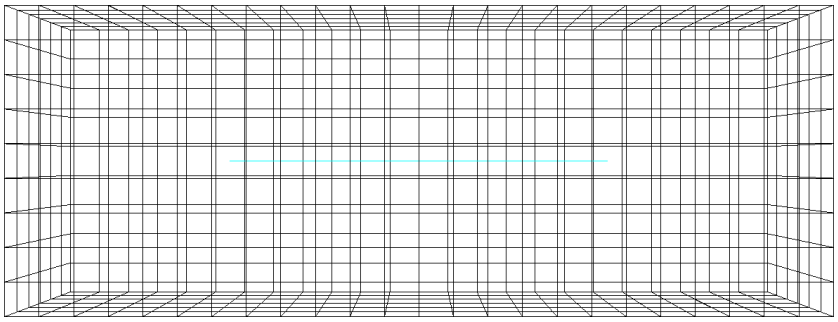

Figure 2.2: The 3D rectangular cuboid mesh

Different tests have been conducted, either starting from a null frame velocity and accelerating, or starting from a uniform arbitrary velocity and decelerating. The initial condition is always a uniform field at rest in the whole domain, so that steadiness of the theoretical solution is assured. An "extrapolation" condition is chosen for the inflow and outflow boundary, so that the solver would not

force the solution along the time span, thus leaving to float freely according to the MRF source term. Both "symmetry" (slipping walls) and "extrapolation" side boundaries conditions are investigated, the latter being interesting for checking the development of spurious non-axial velocity components. All the possible combinations of spatial numerical fluxes and time integration methods both for viscous (Navier-Stokes) and inviscid (Euler) analysis are considered too. The correctness of the results is so high that no comparison table is provided: after a one-second time span simulation, the spurious non-axial velocity components and the residual error on the axial velocity are always ten or more orders of magnitude smaller than the characteristic quantities. The resulting figures below can be considered the worst-case and are then included only for the sake of clarity.

2D rectangle

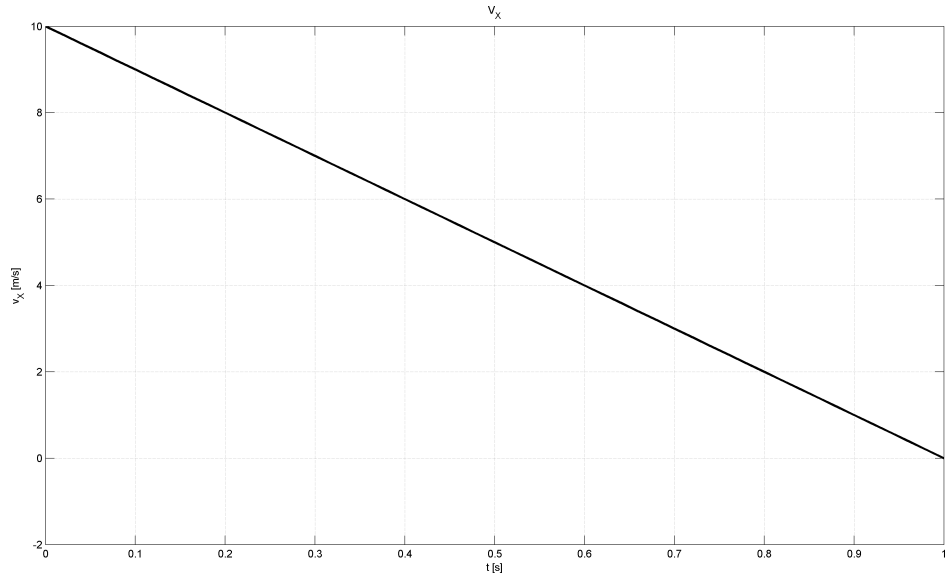

Figure 2.3: 2D rectangle - Vx

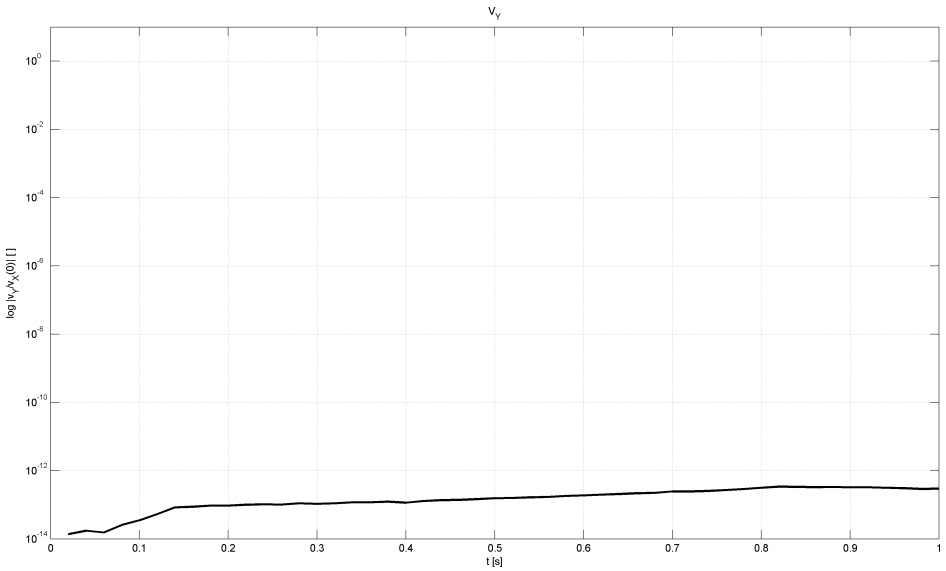

Figure 2.4: 2D rectangle - Vy

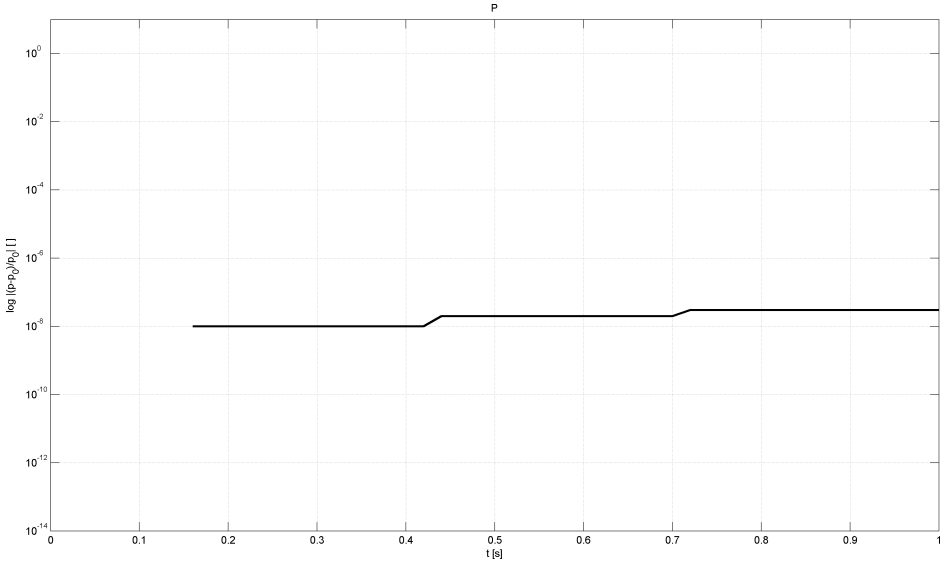

Figure 2.5: 2D rectangle - P

3d rectangular cuboid

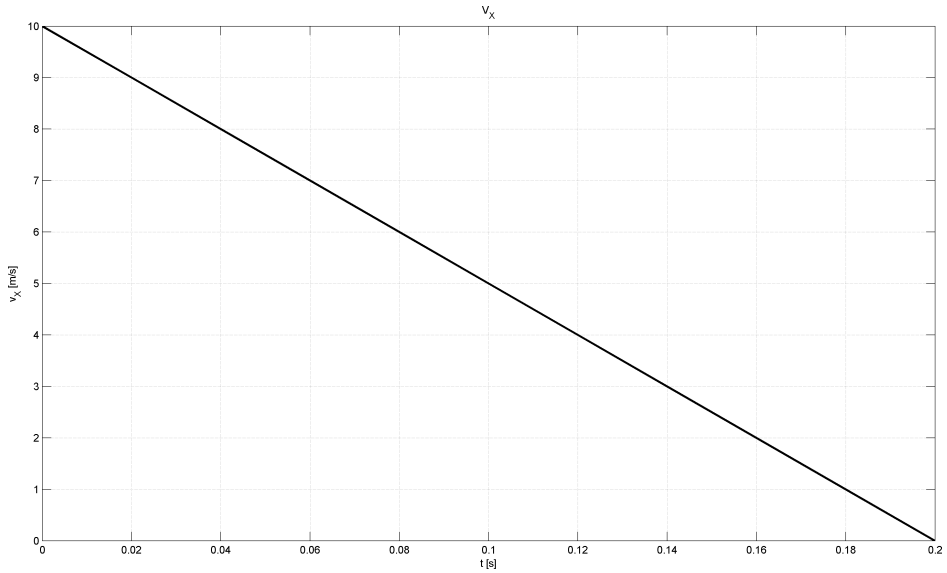

Figure 2.6: 3D rectangular cuboid - Vx

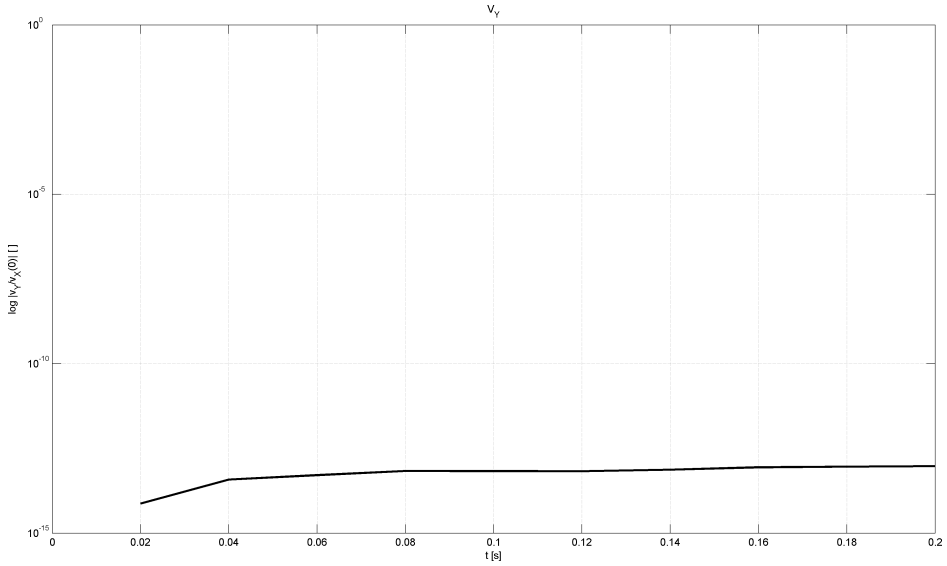

Figure 2.7: 3D rectangular cuboid - Vy

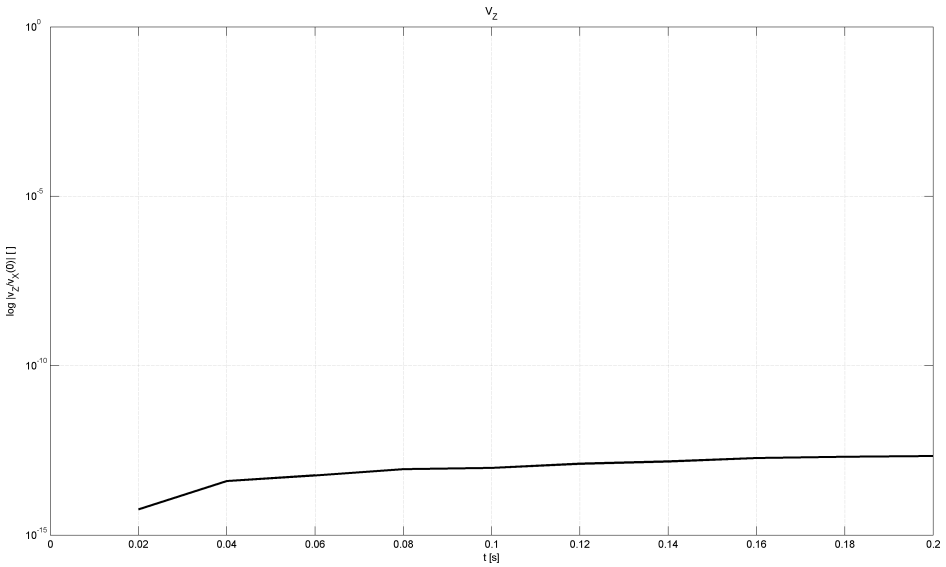

Figure 2.8: 3D rectangular cuboid - Vz

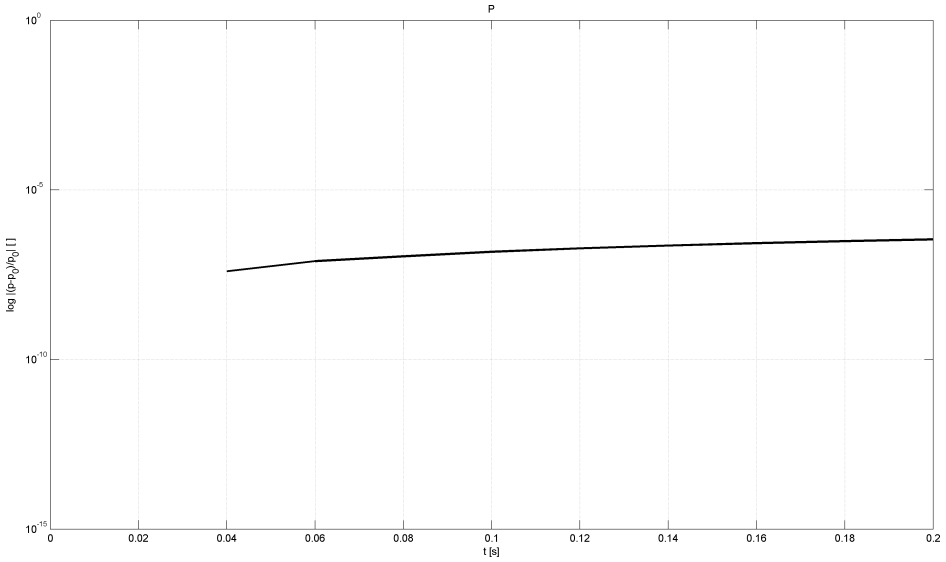

Figure 2.9: 3D rectangular cuboid - P

Small inhomogeneities in the input meshes, which are manifest in an ADFviewer investigation, can be ascribed among the possible responsibles of the residual errors. As a conclusion of this test campaign, the translational inertia term is highly robust and reliable, showing no dependence on time integration algorithm, spatial numerical flux, mesh size and refinement, and very little dependence on the analysis time span.

2.4.2 The steady circular annulus

The most exhaustive test campaign is bound to the evaluation of the correct behavior of Coriolis and centrifugal forces. The constant rotation case is selected, and thus the study has to make use of an axisymmetric domain: several 2D circular annulus are adopted, an internal radius of one meter and an external radius of 10 meters were selected for the principal test campaign, but different scalings have also been investigated. Different meshing patterns (triangles, quadrangles) and parameters (12, 24 and 64 circle-wise partitions, 3, 4, 8, 12, 24, 30 and 32 radial partitions) are considered.

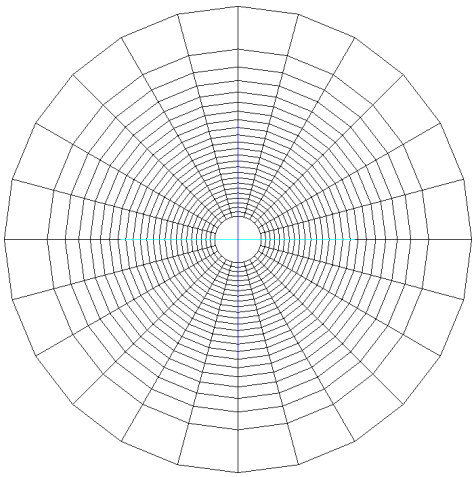

Figure 2.10: 24x24 2D circular annulus mesh

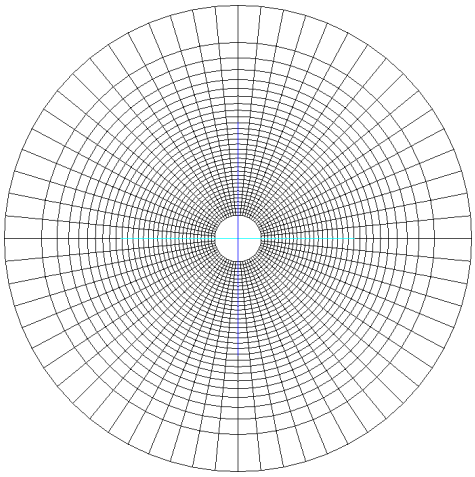

Figure 2.11: 64x32 2D circular annulus mesh

A steady exact solution is again the starting crucial point for detecting anomalies in the operativity of the source terms: a fluid at rest in the absolute frame is again assumed, so the initial conditions are chosen accordingly. A constantly rotating

reference frame with several possible angular speeds is selected, its rotation center and axis coincident with the domain ones ensure steadiness. The calculated solution is left freely floating along the timespan, again thanks to slipping walls (symmetry) boundary conditions, which don't force in the exact solution. The following time trend of velocity magnitude and pressure are examples of the calculation results which will be used as input for the following post-processing campaign.

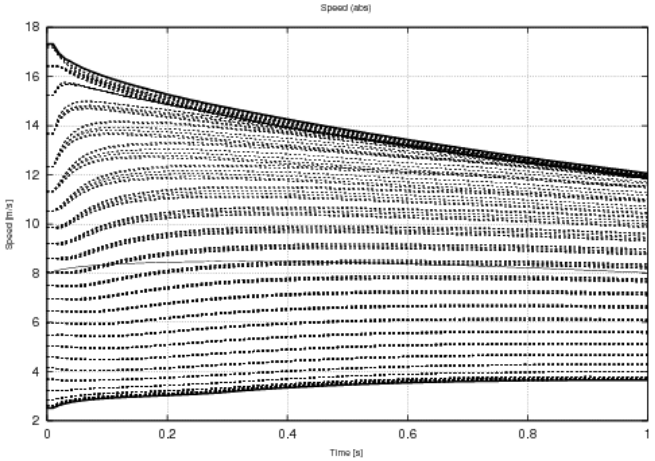

Figure 2.12: Velocity magnitudes for an HLLE with MUSCL analysis of a 24x24 circular annulus

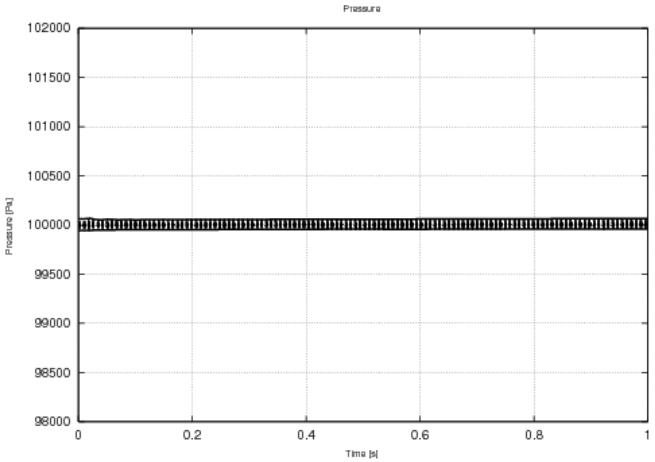

Figure 2.13: Pressures for an HLLE with MUSCL analysis of a 24x24 circular annulus

The maximum and minimum velocity magnitudes in the domain could detect two typical phenomena involved in the numerical analysis, that are source of errors which have to be discerned from source terms operativity errors. The first is the numerical dissipation and is characterized by the overall reduction of both the

maximum and the minimum velocity magnitudes in the domain. The second is the numerical diffusion and is characterized by the the velocities and pressures getting closer to the domain averages, so it is manifest each time the minimum velocity magnitude is increasing towards the maximum magnitude. The inviscid analysis campaign provided the following trends:

2D 24x24 circular annulus

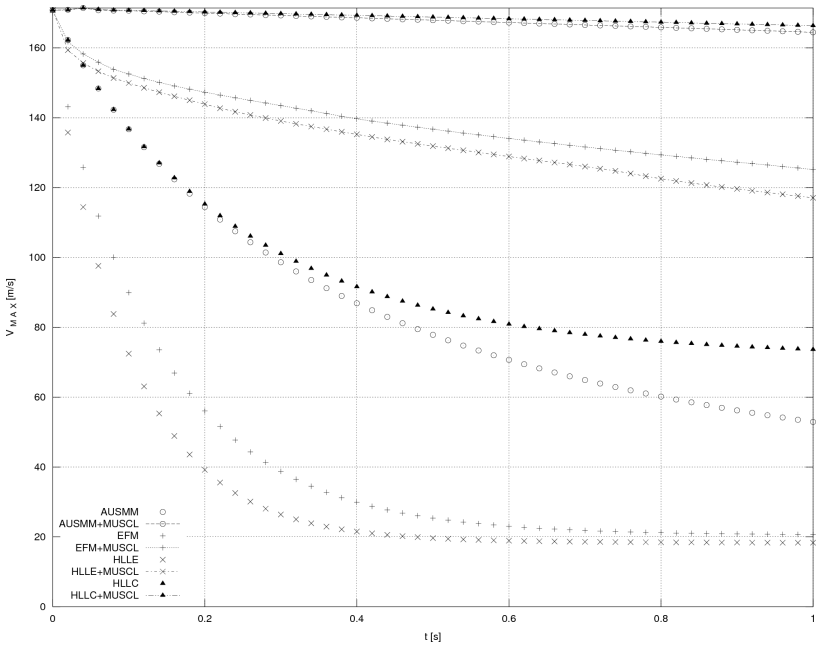

Figure 2.14: Maximum velocity trends in a 24x24 circular annulus

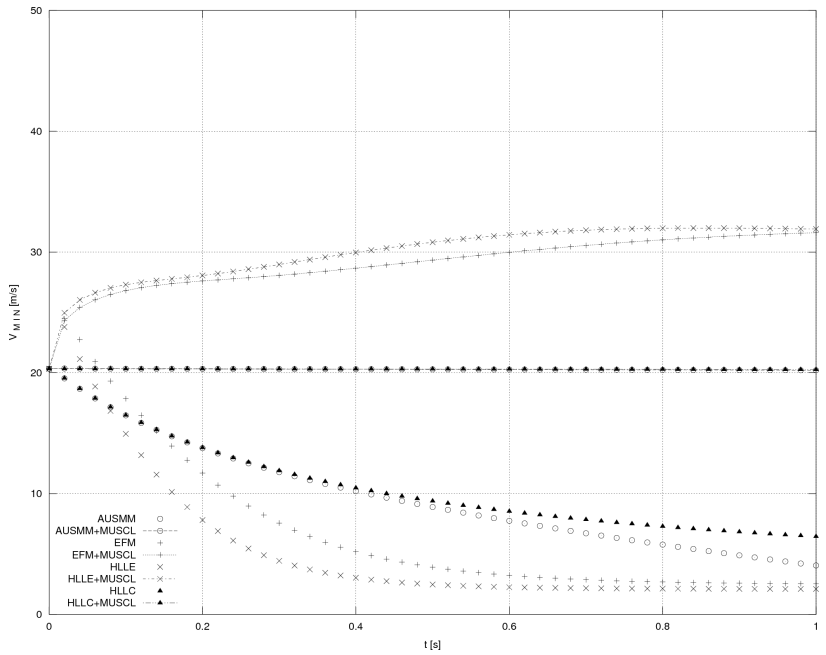

Figure 2.15: Minimum velocity trends in a 24x24 circular annulus

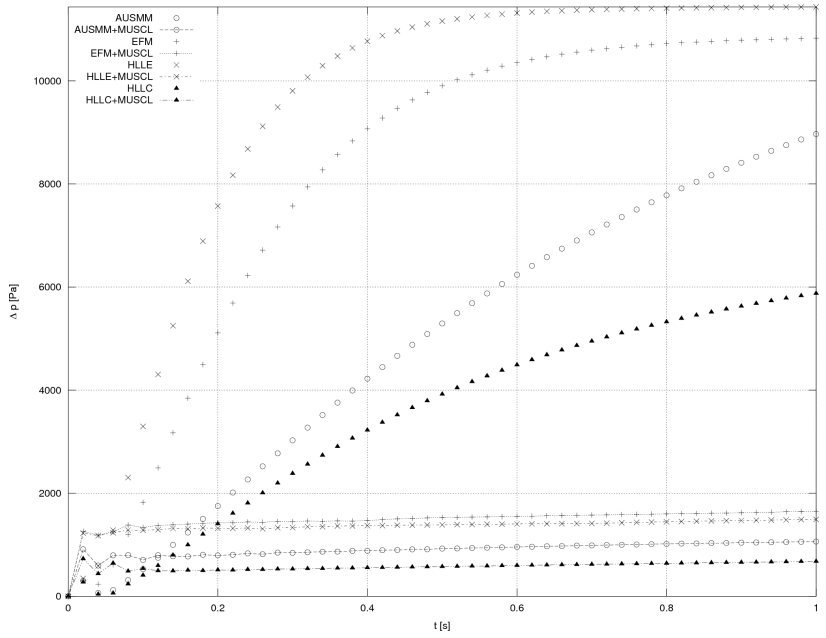

Figure 2.16: Pressure drops in a 24x24 circular annulus

2D 64x32 circular annulus

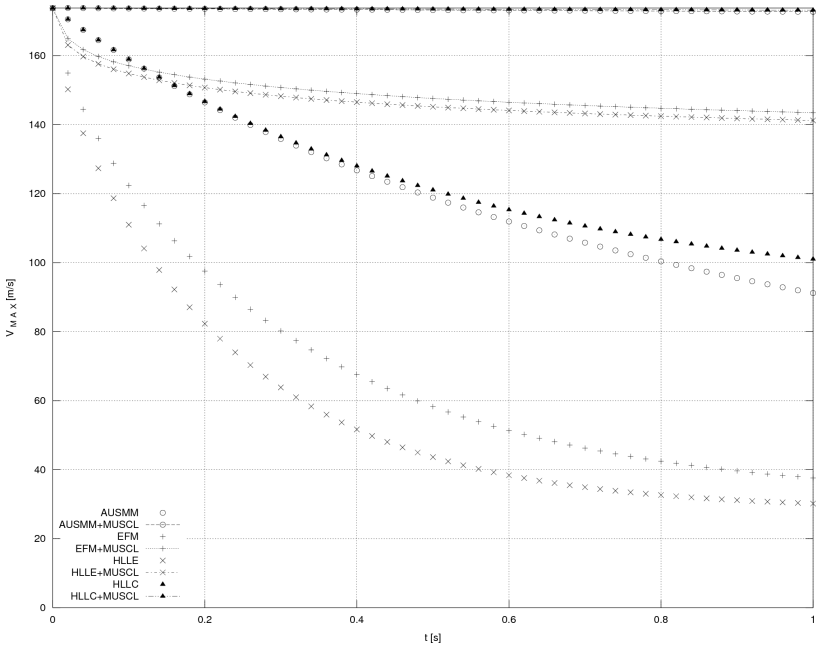

Figure 2.17: Maximum velocity trends in a 64x32 circular annulus

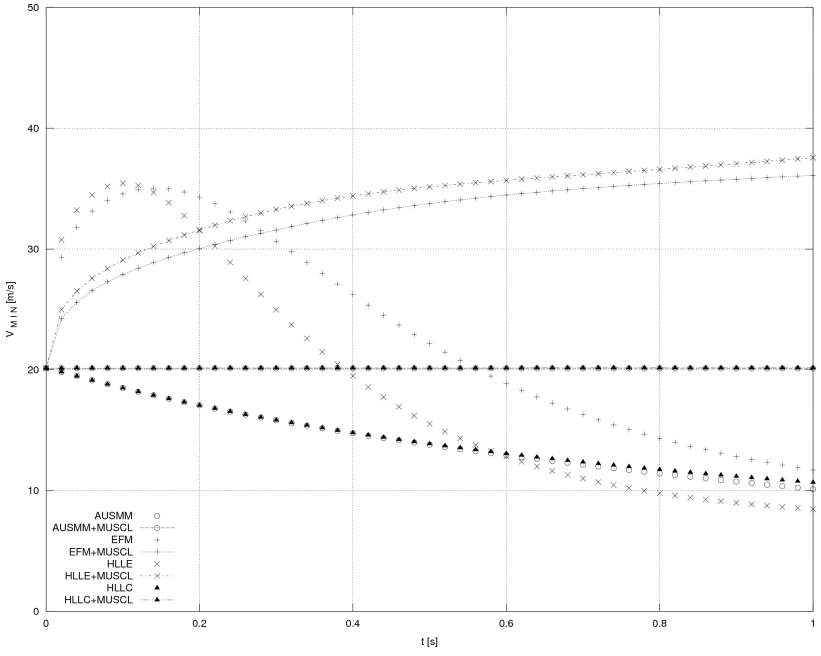

Figure 2.18: Minimum velocity trends in a 64x32 circular annulus

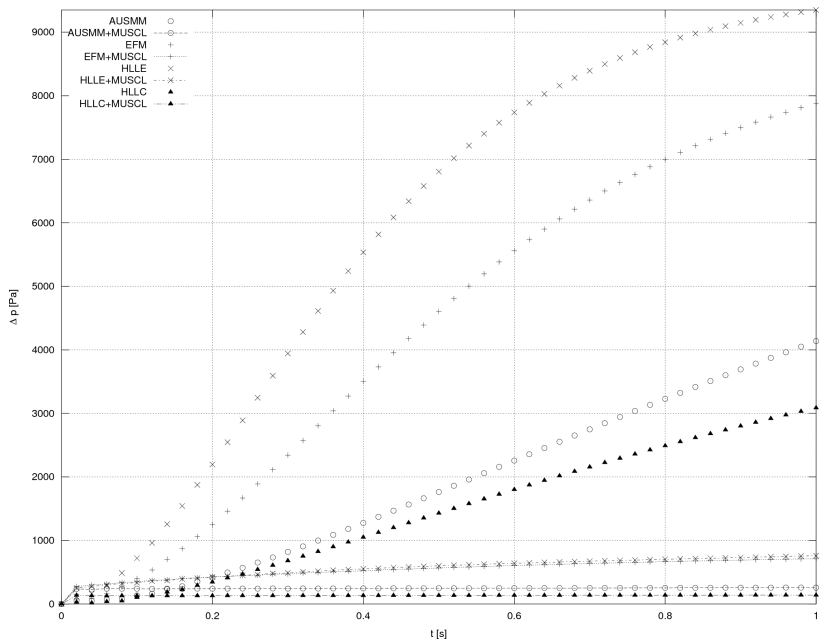

Figure 2.19: Pressure drops in a 64x32 circular annulus

Interestingly, in some cases both the numerical dissipation and numerical diffusion are significant and working against each other to steer the Minimum velocity in opposite directions [figure 2.18]. From the inviscid analysis above it can already be deduced that AUSMM and HLLC numerical schemes with MUSCL higher order extrapolations give very robust results, even for extremely long time-spans (not shown). Viscous analysis campaign is then pursued with different scaling of the already introduced meshes. The results are averaged and organized in tables. Three different kind of errors are provided. The first one is a ratio between the lost velocity and the initial velocity magnitudes. The second one is a ratio between the accumulated pressure and the original pressure and shows how much of the lost velocity was actually transformed in an average pressure increase in the domain: if this increase is significant, then we are not facing dissipation, but instead an incorrect behavior of the MRF source terms, since the trend is restoring "inertial" conditions; low figures indicate instead that a strong numerical dissipation, which source terms are not supposed to face, is present, thus exonerating the MRF source terms. The third one is a ratio between the calculated "delta" pressure along the radius and the one corresponding to the exact solution if the fluid is rotating together with the reference frame, which is of course the wrong case: this indicator is somehow halfway before the aforesaid two, since it depends on the dissipation effects, but still a bit less than the velocity magnitude. The target, in order to validate the MRF source terms, is having all the three errors, and the second one particularly, reasonably low, below the ten percent threshold at worst. The results are split between the first-order analysis campaign and the MUSCL higher order analysis, since they are markedly different.

Mesh geometry	Numerical flux	Higher order	$\lvert\frac{v_{end}-v_0}{v_0}\rvert$ [%]	$\lvert\frac{p_{end}-p_0}{p_0}\rvert$ [%]	$\lvert\frac{\Delta p_{mrf}}{\Delta p_{inertial}}\rvert$ [%]
tri (delanuay)	EFM	-	73.39	12.39	57.54
tri (delanuay)	AUSMM	-	60.42	8.01	38.87
tri (delanuay)	HLLC	-	49.72	8.38	26.94
tri (patch)	EFM	-	75.31	9.96	57.51
tri (patch)	AUSMM	-	59.53	6.17	33.51
tri (patch)	HLLC	-	53.20	6.04	25.46
12x03	EFM	-	78.68	17.17	37.05
12x03	AUSMM	-	84.48	8.97	65.74
12x03	HLLC	-	58.13	10.91	19.96
12x04	EFM	-	81.23	15.22	45.27
12x04	AUSMM	-	86.92	9.45	71.97
12x04	HLLC	-	61.03	9.99	24.74
12x08	EFM	-	87.88	11.91	62.59
12x08	AUSMM	-	89.28	8.98	68.58
12x08	HLLC	-	67.79	8.32	36.24
12x12	EFM	-	91.36	11.57	70.61
12x12	AUSMM	-	88.61	8.93	68.28
12x12	HLLC	-	71.43	8.11	42.59
12x24	EFM	-	95.59	11.42	80.50
12x24	AUSMM	-	86.62	8.92	68.35
12x24	HLLC	-	76.74	8.10	51.97
24x24	EFM	-	90.18	10.94	78.01
24x24	AUSMM	-	76.58	7.73	56.52
24x24	HLLC	-	68.48	7.28	43.87
24x24 (lin)	EFM	-	82.87	10.77	54.24
24x24 (lin)	AUSMM	-	81.72	7.99	58.28
24x24 (lin)	HLLC	-	62.60	7.22	34.71
24x30	EFM	-	85.19	10.47	59.16
24x30	AUSMM	-	79.67	7.93	57.85
24x30	HLLC	-	63.79	7.18	36.89
64x32	EFM	-	67.37	8.48	38.25
64x32	AUSMM	-	56.02	5.39	30.70
64x32	HLLC	-	47.89	5.20	21.25

Among the first order viscous results, even the finest meshes, together with the best spatial numerical fluxes, manifest large errors, although the second kind is always reasonably smaller. The trend is nonetheless positive, since all the errors decrease with the improving mesh complexity, but these results are still too scarce for a validation.

Mesh geometry	Numerical flux	Higher order	$\lvert\frac{v_{end}-v_0}{v_0}\rvert$ [%]	$\lvert\frac{p_{end}-p_0}{p_0}\rvert$ [%]	$\lvert\frac{\Delta p_{mrf}}{\Delta p_{inertial}}\rvert$ [%]
tri (delanuay)	EFM	MUSCL	86.07	16.82	30.31
tri (delanuay)	AUSMM	MUSCL	36.55	18.95	43.09
tri (delanuay)	HLLC	MUSCL	56.79	19.87	35.40
tri (patch)	EFM	MUSCL	30.36	7.10	17.50
tri (patch)	AUSMM	MUSCL	19.11	5.75	20.34
tri (patch)	HLLC	MUSCL	11.40	5.99	18.18
12x03	EFM	MUSCL	61.71	7.19	26.50
12x03	AUSMM	MUSCL	15.73	2.88	17.29
12x03	HLLC	MUSCL	10.90	2.74	11.40
12x04	EFM	MUSCL	60.39	5.65	23.74
12x04	AUSMM	MUSCL	20.22	2.75	16.11
12x04	HLLC	MUSCL	10.70	2.28	11.99
12x08	EFM	MUSCL	27.75	1.91	13.34
12x08	AUSMM	MUSCL	9.69	1.51	14.66
12x08	HLLC	MUSCL	6.81	1.29	12.23
12x12	EFM	MUSCL	16.56	1.31	12.63
12x12	AUSMM	MUSCL	5.82	1.38	14.28
12x12	HLLC	MUSCL	6.38	1.18	11.99
12x24	EFM	MUSCL	5.89	1.27	12.90
12x24	AUSMM	MUSCL	3.11	1.38	14.19
12x24	HLLC	MUSCL	4.22	1.25	12.61
24x24	EFM	MUSCL	28.18	0.44	4.38
24x24	AUSMM	MUSCL	1.14	0.40	4.52
24x24	HLLC	MUSCL	0.71	0.38	3.92
24x24 (lin)	EFM	MUSCL	28.47	1.37	7.15
24x24 (lin)	AUSMM	MUSCL	2.72	0.67	5.03
24x24 (lin)	HLLC	MUSCL	1.37	0.56	3.74
24x30	EFM	MUSCL	24.21	1.23	6.64
24x30	AUSMM	MUSCL	2.59	0.65	4.85
24x30	HLLC	MUSCL	1.46	0.53	3.69
64x32	EFM	MUSCL	29.79	0.49	3.77
64x32	AUSMM	MUSCL	0.32	0.09	0.93
64x32	HLLC	MUSCL	0.12	0.08	0.80

The MUSCL higher order intra-cell extrapolation is responsible of a significant improvement in the quality of the results: excluding the very dissipative EFM numerical scheme, we can deduce that starting from a 24x24 mesh definition all the errors for AUSMM and HLLC analysis are significantly below the ten percent threshold. The errors in the 64x32 2D circular annulus are even three orders of magnitude smaller than the characteristic quantities. These results, together with the corresponding inviscid ones, can then prove the correct operativity of the centrifugal and Coriolis source terms: even a markedly coarse mesh, with cell edges lengthing up to 37.5 centimeters, is able to provide excellent results, provided a MUSCL higher order extrapolation is available. Significantly finer meshes, at least twice as detailed, make possible abandoning the MUSCL higher order extrapolation, retaining comparable results quality.

2.4.3 The oscillating circular annulus

The last test campaign on the Moving Reference Frame developments is aimed at investigating the correct operativity of the remaining additional source term: the angular acceleration inertia $\widetilde{\rho}\dot{\widetilde{\boldsymbol{\omega}}} \times \widetilde{\boldsymbol{r}}$. The studied domains are still the previously introduced circular annulae. A fluid at rest in the absolute frame is again selected, assuring absolute steadiness of the theoretical solution, the initial conditions have been depicted accordingly. Symmetry boundary conditions are again the best choice to avoid forcing the exact solution, thus leaving the results floating freely along the timespan. Inviscid and viscous cases and several oscillation periods and amplitudes are being investigated, but the results are always superimposables. The following plots are thus a few examples of the post-processing output.

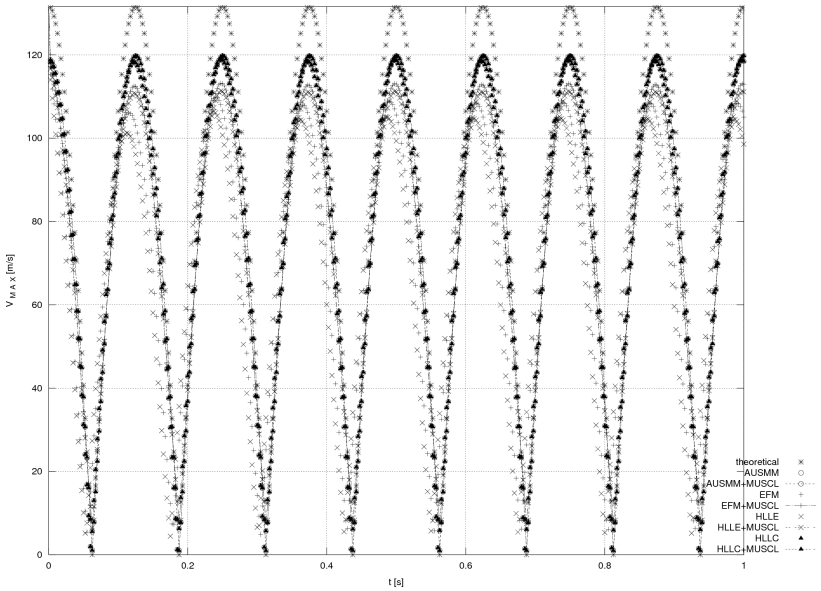

Figure 2.20: Maximum velocity trends in an oscillating 24x24 circular annulus

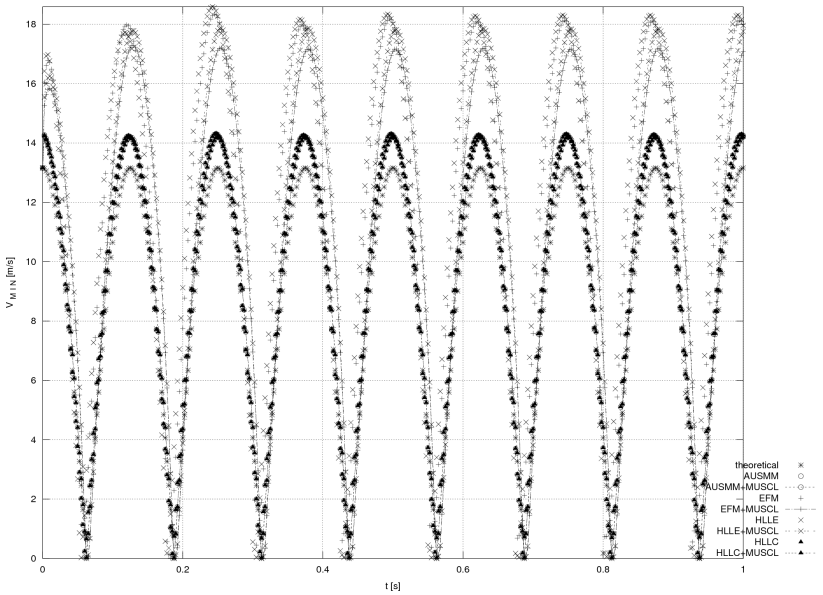

Figure 2.21: Minimum velocity trends in an oscillating 24x24 circular annulus

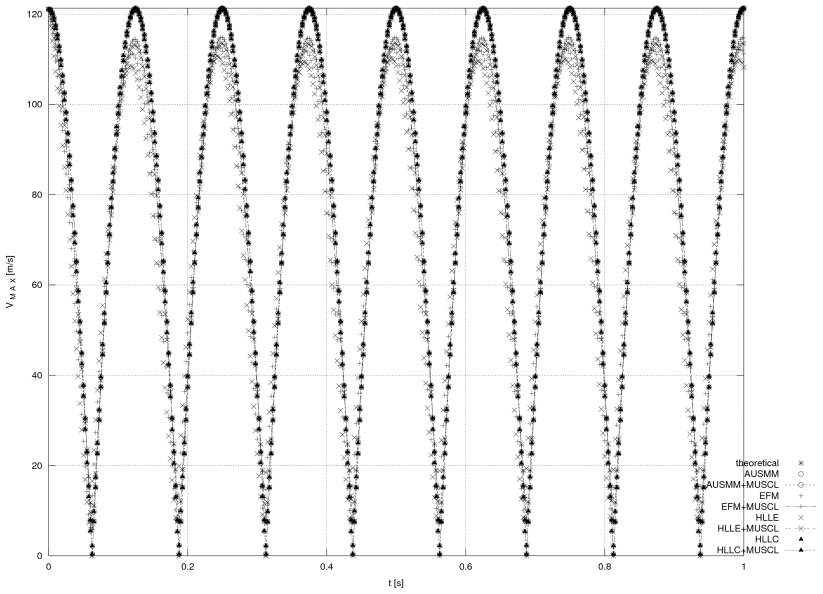

Figure 2.22: Maximum velocity trends in an oscillating 24x24 circular annulus

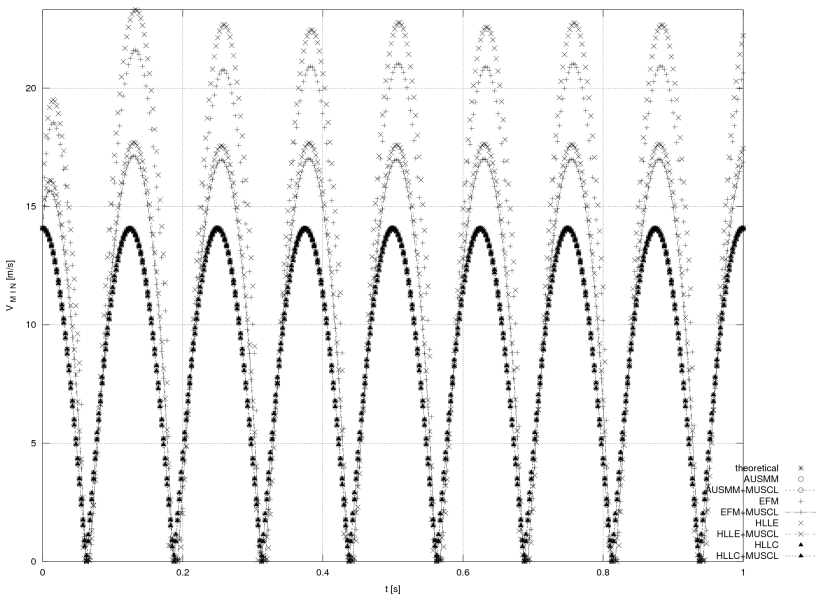

Figure 2.23: Minimum velocity trends in an oscillating 24x24 circular annulus

The steadiness of the oscillations, both in term of amplitude and period, is the aim of this test campaign. Steadiness of the period is always robustly achieved even for extremely long lasting analyses and coarse meshes, and can be explained since the periodicity of the oscillation is forced by the previously introduced MRF kynematics, which are symmetrical around the zero and thus leave no margin for a change of frequency or de-biasing of the solution. The only anomaly detected is the slight phase anticipation present in HLLE and EFM analyses, mostly around the peaks: it can be ascribed among the effects of higher numerical viscosity that characterize these spatial fluxes. The steadiness of the oscillations amplitude, instead, is never completely reached in the first one or two complete periods, both on the outer border (maximum velocity) and on the inner border (minimum velocity) of the domain. This anomaly can be seen as the second-order counterpart of the first-order unsteadiness already met in the constant rotation test campaigns: the velocity magnitude is suffering numerical dissipation and diffusion and thus evolves along the time, compromising the periodicity of the initial rotational oscillations. Results are nonetheless very precise even for the coarse meshes shown, provided an HLLC or AUSMM spatial flux with MUSCL higher order extrapolation are adopted. The operativity of the angular acceleration inertia source term has thus been proved robust and reliable.

2.5 Conclusions

The introduced, developed and tested Moving Reference Frame approach has shown excellent feasibility if compared to the extremely small source code and computational burdening. The detected sources of error can be traced to the strong finite approximation of the theoretical source terms and the significant numerical dissipation and diffusion of some spatial numerical schemes. The mesh refinement requirements are nonetheless absolutely in line with the solver's original ones, and the freedom of choices on all the calculation methods and parameters is untouched. Initial and Boundary conditions can be both provided in the absolute or the relative reference frames and thus don't require further pre-processing by the user. The limited set of kynematical cases prevents parameters flaws and is already capable of handling the most common real cases of rigid moving bodies or boundaries, and can be still expanded at wish.

Chapter 3

The Arbitrary Lagrangian Eulerian approach

3.1 Preliminary considerations

The basic idea of detaching from both the classical Eulerian and Lagrangian approaches can be traced back to the late 1960s, but took quite a considerable time to spread, be rigorously theorized and widely studied. The primal objective of the later called Arbitrary Lagrangian Eulerian approach was to combine the advantages of both the original approaches, while minimizing their drawbacks, thus aiming at becoming the new standard for fluid-dynamical codes.

3.1.1 Eulerian solvers advantages and drawbacks

The most common fluid-dynamics codes are still Eulerian-based and will likely remain so. The reasons are quite straightforward: the Eulerian algorythms are the most stable, efficient, fast and robust way to solve fixed domains even in presence of severe unsteadynesses, non-linearities and even several coupled problems. Large distortions of the continuum motion can be handled easily, provided that an adequate mesh resolution is considered, which is often the main issue. Eulerian dynamics by themselves are on the contrary completely unable to account for mesh movement and thus have the very strong limitation of being applicable only to fixed domains, with fixed boundaries.

3.1.2 Lagrangian solvers advantages and drawbacks

Lagrangian solvers were instead widely used for solid mechanics, for which they grew comparable stability, efficiency, quickness and robustness. Their fluid-dynamical counterparts were a very promising way to extend the numerical solvers scope beyond the fixed domains: their cells following the continuum motion were naturally able to account for free surfaces modeling and, with a minimal efforts, could also handle moving bodies and boundaries modeling, going well beyond the

fixed domains frontier. Despite these powerful and promising features, they quickly proved to require an extremely complex implementation to the fluid-dynamical codes: contrary to the solid mechanics, particles of fluid don't have fixed linkings with their neighbours and thus travel freely in the domain. A finite representation of the domain is then naturally subject to large distorsions for which no theoretical solution exist: no matter how fine the mesh is, the cells will always overlap and tangle, growing the well known "hourglass" cells issue. A very frequent remeshing, which in turn require remapping, is needed so the solver's efficiency and speed are severely compromised. By going further down in the details, difficulties arise also for the vertex advancement scheme, since vertex movement must be deducted by interpolation in cell-centered solvers and this is often source of additional errors depending on the considered problem, so a plethora of different solutions were proposed. Another major concern is the potential worsening of both fixed and moving rigid boundaries along medium and long simulations: rigid walls can be modeled only as boundary conditions, and thus there is no warranty on their resolution and shape after several iterations, so the remeshing algorythm will probably have to care of that too.

3.1.3 Arbitrary Lagrangian Eulerian features

We have just seen that simplicity, stability, robustness, efficiency and quickness are the most sought after advantages of the Eulerian solvers, whereas domain movements, deformations and free surfaces modeling are the topical features of the Lagrangian codes. Both flow movement through the mesh and mesh movement must then be allowed, so in Arbitrary Lagrangian Eulerian frames the cells will neither stay still in their absolute position nor fully inconditionally follow the fluid in its most perverse dynamics. A considerable degree of freedom is still left: various mesh movement criteria, relying both on geometric considerations and on flow considerations, have been developed and proved succesful. If the domain deformations are not severe, remeshing can be completely neglected. ALE is also naturally suited for fluid-body interaction studies, since it enables a transition between the Eulerian frame of the asympthotic fluid and the purely Lagrangian fluid mechanics computation inside the body. Versatility and flexibility can be then considered further unexpected major advantages of the Arbitrary Lagrangian Eulerian approach. All these strong advantages don't come free: the complexity of ALE implementation, even if smaller than the purely Lagrangian one, is still considerable. Difficulties arise, in particular, both in the vertex movement criteria itself and in how the additional flux terms will account for it. The mesh update algorythm is also closely affected by the time integration method used by the solver, since all the calculated quantities must be coherent with it. The behavior of the additional term will depend also on whether a higher order interpolation will be used by the solver or not.

3.2 Governing equations

For the development of Arbitrary Lagrangian Eulerian extensions inside Typhon, the [2] analytical treatment has been strictly followed. ALE contributes are thus derived as additional flux terms in the system of Euler equations, for the inviscid case. These analytical results are however directly extendable to the viscous Navier-Stokes equations.

3.2.1 Derivation of the fluid dynamics equations

The Lagrangian point of view

By adopting a Lagrangian perspective, we deal with just two reference systems:
$\underline{\boldsymbol{X}}$ = material coordinates, bound to the moving material particles;
$\underline{\boldsymbol{x}}$ = spatial coordinates, made of fixed spatial (immaterial) points.

Considering any law of motion of the kind

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X}, t), \quad t = t$$

(in which the same physical time is measured in both systems) we can introduce the "mapping", that is an application

$$\begin{array}{rccc} \varphi & : & R_{\boldsymbol{X}} & \longrightarrow & R_{\boldsymbol{x}} \\ & & (\boldsymbol{X}, t) & \longmapsto & (\boldsymbol{x}, t) = \varphi(\boldsymbol{X}, t) \end{array}$$

which allows us to constantly relate the motion of the material particles (and so of the moving reference system) to the fixed spatial frame. The Jacobian matrix of the foresaid mapping will then be:

$$\frac{\partial(\boldsymbol{x}, t)}{\partial(\boldsymbol{X}, t)} = \frac{\partial\varphi(\boldsymbol{X}, t)}{\partial(\boldsymbol{X}, t)} = \left\{ \begin{array}{cc} \left[\begin{array}{c} \\ \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \\ \\ \end{array} \right] & \begin{array}{c} \vdots \\ \boldsymbol{v} \\ \vdots \end{array} \\ \ldots \mathbf{0}^T \ldots & 1 \end{array} \right\} \tag{3.1}$$

where we introduced the material particles speed

$$\boldsymbol{v}(\boldsymbol{X}, t) = \left. \frac{\partial \boldsymbol{x}}{\partial t} \right|_{\boldsymbol{X}} \tag{3.2}$$

In order to ensure that the mapping will always be reversible [..], we'll have to verify that

$$det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}\right) > 0$$

(then the Jacobian will be surely non-singular and φ^{-1} will exist defined).

The ALE point of view

In the Arbitrary Lagrangian Eulerian perspective we have to introduce a third coordinate system:
$\underline{\boldsymbol{X}}$ = material coordinates, bound to the moving material particles;
$\underline{\boldsymbol{\chi}}$ = "reference grid" coordinates, bound to the computational mesh;
$\underline{\boldsymbol{x}}$ = spatial coordinates, made of fixed spatial (immaterial) points.

His place between the former two is intentional, because normally this reference frame should fall somewhere halfway between the non-moving spatial frame and the ever-moving material frame. Since the reference systems are now three, there will be two additional mappings:

$$\begin{aligned}(\boldsymbol{\chi}, t) \quad &\longmapsto \quad \phi(\boldsymbol{\chi}, t) = (\boldsymbol{x}, t) \\ (\boldsymbol{X}, t) \quad &\longmapsto \quad \psi^{-1}(\boldsymbol{X}, t) = (\boldsymbol{\chi}, t)\end{aligned}$$

with the following Jacobians:

$$\frac{\partial(\boldsymbol{x}, t)}{\partial(\boldsymbol{\chi}, t)} = \frac{\partial \phi(\boldsymbol{\chi}, t)}{\partial(\boldsymbol{\chi}, t)} = \begin{Bmatrix} \begin{bmatrix} & \\ \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}} & \\ & \end{bmatrix} & \begin{matrix} \vdots \\ \hat{\boldsymbol{v}} \\ \vdots \end{matrix} \\ \dots \mathbf{0}^T \dots & 1 \end{Bmatrix} \tag{3.3a}$$

$$\frac{\partial(\boldsymbol{\chi}, t)}{\partial(\boldsymbol{X}, t)} = \frac{\partial \psi^{-1}(\boldsymbol{X}, t)}{\partial(\boldsymbol{X}, t)} = \begin{Bmatrix} \begin{bmatrix} & \\ \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{X}} & \\ & \end{bmatrix} & \begin{matrix} \vdots \\ \boldsymbol{w} \\ \vdots \end{matrix} \\ \dots \mathbf{0}^T \dots & 1 \end{Bmatrix} \tag{3.3b}$$

where we introduced the following speeds:

$$\hat{\boldsymbol{v}}(\boldsymbol{\chi}, t) = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{\chi}} = \text{absolute speed of the mesh.} \tag{3.4a}$$

$$\boldsymbol{w}(\boldsymbol{X}, t) = \left.\frac{\partial \boldsymbol{\chi}}{\partial t}\right|_{\boldsymbol{X}} = \text{particles speed relative to the mesh "in the mesh' perspective"} \tag{3.4b}$$

The correct composition would be:

$$\boldsymbol{v} = \hat{\boldsymbol{v}} + \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}} \boldsymbol{w}$$

Thus, for the following steps, it is wise to define another speed:

$$\boldsymbol{c} = \boldsymbol{v} - \hat{\boldsymbol{v}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}} \boldsymbol{w} \tag{3.5}$$

c is the convective speed of the particles relative to the mesh, meseaured in the fixed reference frame.

The fundamental ALE relation

As a first step, we start working on gradients of generic functions in the various coordinate systems. Hence, the gradient of

$$f^{**}(\boldsymbol{X},t) = f^{*}(\boldsymbol{\chi},t) = f^{*}(\psi^{-1}(\boldsymbol{X},t),t) = f^{*}\cdot\psi^{-1} \tag{3.6}$$

will be

$$\frac{\partial f^{**}}{\partial(\boldsymbol{X},t)} = \frac{\partial f^{*}}{\partial(\boldsymbol{\chi},t)}\cdot\frac{\partial\psi^{-1}}{\partial(\boldsymbol{X},t)} \tag{3.7}$$

Expliciting the single terms:

$$\left\{\frac{\partial f^{**}}{\partial \boldsymbol{X}} \quad \frac{\partial f^{**}}{\partial t}\right\} = \left\{\frac{\partial f^{*}}{\partial \boldsymbol{\chi}} \quad \frac{\partial f^{*}}{\partial t}\right\}\begin{Bmatrix} \left[\begin{array}{c} \\ \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{X}} \\ \\ \end{array}\right] & \begin{array}{c}\vdots\\ \boldsymbol{w}\\ \vdots\end{array} \\ \ldots\,\mathbf{0}^{T}\,\ldots & 1 \end{Bmatrix} \tag{3.8}$$

that is equal to the following system:

$$\Rightarrow \quad \begin{cases} \dfrac{\partial f^{**}}{\partial \boldsymbol{X}} = \dfrac{\partial f^{*}}{\partial \boldsymbol{\chi}}\dfrac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{X}} \\ \dfrac{\partial f^{**}}{\partial t} = \dfrac{\partial f^{*}}{\partial t} + \dfrac{\partial f^{*}}{\partial \boldsymbol{\chi}}\boldsymbol{w} \end{cases} \qquad \left(\boldsymbol{w} = \left.\frac{\partial \boldsymbol{\chi}}{\partial t}\right|_{\boldsymbol{X}}\right) \tag{3.9}$$

Since, as previously stated,

$$\boldsymbol{c} = \boldsymbol{v} - \hat{\boldsymbol{v}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}}\boldsymbol{w}$$

it is finally possible to write:

$$\Rightarrow \left.\frac{\partial f^{**}}{\partial t}\right|_{\boldsymbol{X}} = \frac{\partial f^{*}}{\partial t} + \frac{\partial f^{*}}{\partial \boldsymbol{\chi}}\boldsymbol{w} = \frac{\partial f^{*}}{\partial t} + \frac{\partial f^{*}}{\partial \boldsymbol{x}}\boldsymbol{c} = \left.\frac{\partial f^{*}}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{\nabla} f\cdot\boldsymbol{c} \tag{3.10}$$

This is a fundamental relation for the ALE develpment, since it allows the expression of the time-dependant terms of the conservation laws in the new arbitrarily moving mesh reference system.

3.2.2 ALE formulation of the differential Euler equations

The common differential form of the Euler equations is:

$$\begin{cases} \dfrac{d\rho}{dt} = \left.\dfrac{\partial \rho}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{v}\cdot\boldsymbol{\nabla}\rho = -\rho\boldsymbol{\nabla}\cdot\boldsymbol{v} \\ \rho\dfrac{dv}{dt} = \rho\left(\left.\dfrac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{x}} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v}\right) = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} \\ \rho\dfrac{dE}{dt} = \rho\left(\left.\dfrac{\partial E}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{v}\cdot\boldsymbol{\nabla}E\right) = \boldsymbol{\nabla}\cdot(\boldsymbol{\sigma}\cdot\boldsymbol{v}) + \boldsymbol{v}\cdot\rho\boldsymbol{b} \end{cases} \tag{3.11}$$

By using the previously derived relation (3.10), we can easily derive the ALE formulation of the Euler equations, in differential form:

$$\begin{cases} \left.\frac{\partial \rho}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{c}\cdot\boldsymbol{\nabla}\rho = -\rho\boldsymbol{\nabla}\cdot\boldsymbol{v} \\ \rho\left(\left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{\chi}} + (\boldsymbol{c}\cdot\boldsymbol{\nabla})\boldsymbol{v}\right) = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} \\ \rho\left(\left.\frac{\partial E}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{c}\cdot\boldsymbol{\nabla}E\right) = \boldsymbol{\nabla}\cdot(\boldsymbol{\sigma}\cdot\boldsymbol{v}) + \boldsymbol{v}\cdot\rho\boldsymbol{b} \end{cases} \tag{3.12}$$

also known as "quasi-Eulerian form", and eventually the internal energy balance:

$$\rho\frac{de}{dt} = \boldsymbol{\sigma} : \boldsymbol{\nabla}^S\boldsymbol{v} \quad \Rightarrow \quad \rho\left(\left.\frac{\partial e}{\partial t}\right|_{\boldsymbol{\chi}} + c\cdot\boldsymbol{\nabla}e\right) = \boldsymbol{\sigma} : \boldsymbol{\nabla}^S\boldsymbol{v} \tag{3.13}$$

(where $\boldsymbol{\nabla}^S\boldsymbol{v} = \frac{1}{2}(\boldsymbol{\nabla}\boldsymbol{v} + \boldsymbol{\nabla}^T\boldsymbol{v})$)
Thanks to the same relations, deduction of the accelerations is also possible.

$$\boldsymbol{a} = \left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{X}} \quad \Rightarrow \quad \boldsymbol{a} = \left.\frac{\partial \boldsymbol{v}}{\partial t}\right|_{\boldsymbol{\chi}} + \boldsymbol{c}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \tag{3.14}$$

ALE formulation of the integral Euler equations

The simplest way to develop the integral Euler equations in ALE formulation is by using Reynolds' transport and Green's theorems.

The Green's theorem states that:

$$\oint_{\delta\Omega} \boldsymbol{F}\cdot\boldsymbol{n}dS = \int_{\Omega} \boldsymbol{\nabla}\cdot\boldsymbol{F}dV \tag{3.15}$$

whereas the Reynold's transport theorem is:

$$\left.\frac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \int_{V_t} f(\boldsymbol{x},t)dV = \int_{V_t} \left.\frac{\partial f(\boldsymbol{x},t)}{\partial t}\right|_{\boldsymbol{x}} dV + \int_{S_t} f(\boldsymbol{x},t)\hat{\boldsymbol{v}}\cdot\boldsymbol{n}dS \tag{3.16}$$

By rewriting the Reynold's transport theorem for the quantities $(\rho, \rho\boldsymbol{v}, \rho E)$ we get

$$\begin{cases} \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \int_{V_t} \rho dV = \int_{V_t} \left.\frac{\partial \rho}{\partial t}\right|_{\boldsymbol{x}} dV + \int_{S_t} \rho\hat{\boldsymbol{v}}\cdot\boldsymbol{n}dS \\ \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \int_{V_t} \rho\boldsymbol{v}dV = \int_{V_t} \left.\frac{\partial \rho\boldsymbol{v}}{\partial t}\right|_{\boldsymbol{x}} dV + \int_{S_t} \rho\boldsymbol{v}\hat{\boldsymbol{v}}\cdot\boldsymbol{n}dS \\ \left.\frac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \int_{V_t} \rho E dV = \int_{V_t} \left.\frac{\partial \rho E}{\partial t}\right|_{\boldsymbol{x}} dV + \int_{S_t} \rho E\hat{\boldsymbol{v}}\cdot\boldsymbol{n}dS \end{cases} \tag{3.17}$$

Then, by substituting the first term of the right hand side with (3.11) we can shortly obtain:

$$\begin{cases} \left.\dfrac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \displaystyle\int_{V_t} \rho dV + \int_{S_t} \rho \boldsymbol{c} \cdot \boldsymbol{n} dS = 0 \\ \left.\dfrac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \displaystyle\int_{V_t} \rho \boldsymbol{v} dV + \int_{S_t} \rho \boldsymbol{v} \boldsymbol{c} \cdot \boldsymbol{n} dS = \int_{V_t} (\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \rho \boldsymbol{b}) dV \\ \left.\dfrac{\partial}{\partial t}\right|_{\boldsymbol{\chi}} \displaystyle\int_{V_t} \rho E dV + \int_{S_t} \rho E \boldsymbol{c} \cdot \boldsymbol{n} dS = \int_{V_t} \Big(\boldsymbol{v} \cdot \rho \boldsymbol{b} + \boldsymbol{\nabla} \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{v}) \Big) dV \end{cases} \tag{3.18}$$

3.3 Implementations in Typhon

The tasks which had to be accomplished are:

- creation of the "ALE" data container, to hold all the new input parameters and any data useful along the calculation;
- recognition and assignment of the new ALE parameters from the analysis input files;
- introduction of the flux terms in the solver's fluid dynamics equations;
- development of the mesh update algorythm.

3.3.1 The ALE data structure

Similarly to the MRF case, a FORTRAN structure proved to be the optimal choice. The "*mnu_ale*", defined in "*SOURCE/PARAM/MENU_ALE.f*90" has been thus designed with the following members:

- the string "*name*";
- the integer parameter "*type*" for the mesh movement criteria or algorythm;
- the symbolic relations "*movement_x*", "*movement_y*" ,"*movement_z*" and "*movement_θ*", for the mesh movement;
- the string "*moving_body*", for the "*BODY*" mesh movement algorythm;
- the integer "*idboco_body*", which will correspond to the previously described moving body;
- the three-dimensional vector "*body_center*", which will be calculated from the moving body nodes;
- the reals "*body_maxradius*" and "*closest_boundary*" and the allocatable real array "*weight*" which are internal parameters for the body mesh movement algorythm;
- the allocatable three-dimensional vector arrays "*original_vertex*", "*old_facecentres*" and "*face_velocity*", needed for the time dependent parameters in the flux termes.

3.3.2 ALE input parameters reading and processing

Similarly to the MRF case, a new "*BLOCK : ALE*" has been introduced in the *main.rpm* analysis input parameters file, to include all the possible ALE inputs. The parsing of that block was handled by the subroutine contained in the

$PARAM/def\_ale.f90$ source file, which also accounted for preliminary treatment of the parameters. An example of the main.rpm ALE section is as follow:

$$
\begin{aligned}
&BLOCK : ALE \\
&\quad NAME && = oscillating\_airfoil \\
&\quad TYPE && = BODY \\
&\quad MOVING\_BODY && = CRVS \\
&\quad BODY\_CENTRE && = (0., 0., 0.) \\
&\quad BODY\_MOVEMENT\_X && = 0. \\
&\quad BODY\_MOVEMENT\_Y && = 0.008128 * SIN(27.8344874 * T - 3.0892) \\
&\quad BODY\_MOVEMENT\_Z && = 0. \\
&\quad BODY\_MOVEMENT\_THETA && = -0.001047198 \\
&\quad -0.026005 * SIN(27.8344874 * T) \\
&ENDBLOCK
\end{aligned}
\tag{3.19}
$$

3.3.3 ALE flux terms

As previously described in the MRF section, all the right hand sides contributes in the Navier-Stokes equations, be they sources or fluxes, are evaluated as residuals of each analysis iteration, that are initialized at zero each time. For the ALE flux terms, this task is being done by the "$calc\_flux\_ale(defsolver, nflux, ista, umesh, cg\_l, cg\_r, QL, QR, flux, calc\_jac, jacL, jacR)$" subroutine, located in the "$calc\_flux\_ale.f90$" source file inside the "$SOURCE/EQNS$" folder. As manifest from the subroutine parameters, the higher order extrapolation of the fluid-dynamics state on the cell faces is used if available. The Jacobian of the flux is also used if provided, but was considered beyond the scopes of the present work and thus postponed.

3.3.4 The mesh update algorythms

Among the various possible mesh movement solutions available, which are also widely undertaken, two of the simplest have been developed since they perfectly fit the scopes of the present work. The first and most straight-forward is the "$GLOBAL$" movement, for which the final user must provide continuous symbolic relations of space and time for the three independent components of the mesh movement function inside the *main.rpm* parameter file. The heaviest burden is left on the final user's shoulders, but the degree of freedom granted by this approach is considerable. The $ale\_meshupdate(umesh,defsolver,gradcond\_computed,curtime,dt)$ subroutine, located in the $MESHALE.f90$ source file inside the $SOURCE/MGRID$ folder handles the evaluation of the space and time symbols for each vertex and

each time step, successively fulfills the actual mesh movement, calls geometric subroutines to recalculate face areas and centers, cell volumes and centers and finally calculates the resulting faces velocities, that are later used by the flux terms.
An example of a *main.rpm* ALE parameters block with the *GLOBAL* movement, extensively using the *step*() function, is as follow:

$$
\begin{aligned}
&BLOCK : ALE \\
&\qquad NAME && = moving\_mesh \\
&\qquad TYPE && = GLOBAL \\
&\qquad MESH\_MOVEMENT\_X && = 0. \\
&\qquad MESH\_MOVEMENT\_Y && = 0.3 * SIN(31.4159 * T) * (1 - Y) * (STEP(X) * \\
&\qquad\qquad && \quad * STEP(1 - X) * X + STEP(X - 1.001)) \\
&\qquad MESH\_MOVEMENT\_Z && = 0. \\
&ENDBLOCK
\end{aligned}
\tag{3.20}
$$

The second solution available is the "*BODY*" movement, for which an interpolation between a user-selected moving body and the remaining fixed boundaries is assumed. The previously introduced *ale_meshupdate*() subroutine does, orderly

- allocates and assigns the original vertexes and old face centers and allocates face velocities vector;
- finds the user inputted moving body among the mesh boundaries and stores its boundary faces indexes;
- calculates the *body_center*, the *body_maxradius* and the *closest_boundary* by cycling on all the body vertexes and the remaining boundaries vertexes;
- calculate the "weight" on each mesh non-boundary vertex for the interpolation, comparing the distance from the body center with the max radius and the closest boundary;
- evaluates the space and time symbols for the four continuous symbolic relations of X, Y, Z, θ body movement;
- fulfills the actual mesh movement;
- calls the geometric subroutines to recalculate face areas and centers, cell volumes and centers;
- calculates the resulting faces velocities.

3.4 Validation cases and results

In order to check the correct behavior of ALE flux terms, tests for both the "static" effects and the "dynamic" ones have been pursued. The principal difference among the two kinds is that in the first case the mesh movement is slow compared to the flow velocities studied, so the flow tends to evolve like a series of "snapshots" and the mesh movement itself doesn't cause noticeable variations on the solutions; in the second case, the mesh movement speed is comparable to the fluid flow ones and thus, thanks to the introduction of the correct wall velocity boundary condition, has a direct and noticeable influence on the flow field results. A two-dimensional contracting/expanding piston domain was included in the dynamic validation cases, but had to be successively discarded since cell dimensions and time step requirements for a stable solutions were too strict for a useful test campaign, yet the trends were promising. The two-dimensional flutter presented in the next section is thus the only dynamic validation case described, whereas a supersonic flow inside a two-dimensional oscillating duct (wedge geometry) provide for the static validation case.

As already mentioned in the MRF validation cases introduction, this time a single combination will be evalued for the computational enquiry, taking benefit of the best possible combination of numerical fluxes (spatial schemes) and time-marching schemes. The previous results of the MRF validational campaign are, in fact, still perfectly valid.

3.4.1 Supersonic flow inside an oscillating two-dimensional duct

Theory

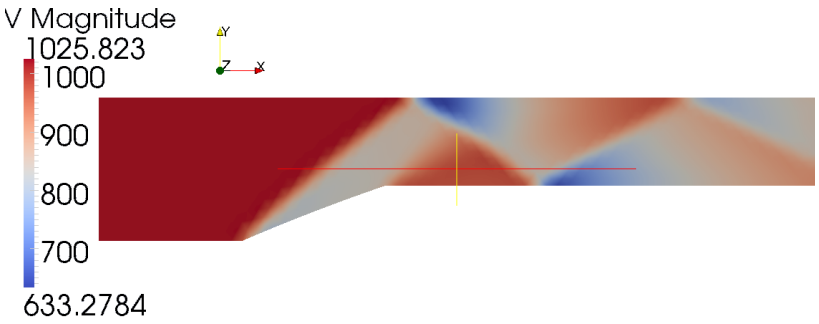

Figure 3.1: 2d wedge shock solution in concave attitude

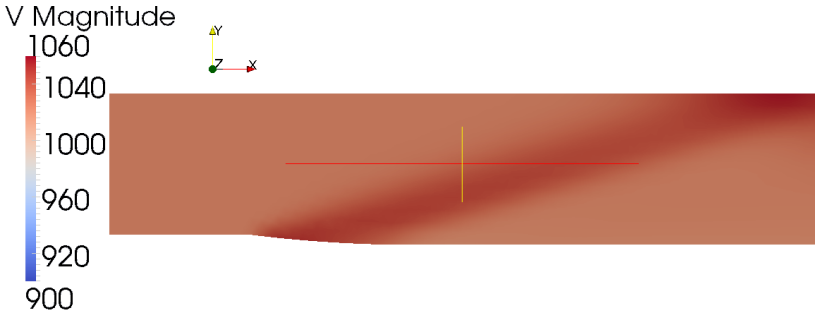

Figure 3.2: 2d wedge expansion solution in convex attitude

A supersonic flow entering a two-dimensional duct with variable geometry in time undergoes either oblique shocks or an expansion fan. Theoretical results are available thanks to the oblique shock and Prandtl-Meyer relations, so a validation case for ALE is promptly available.

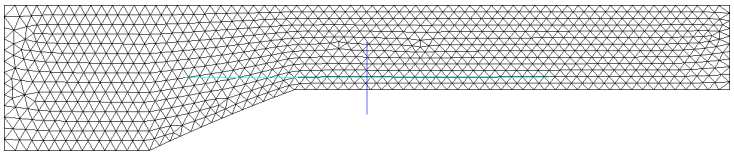

Figure 3.3: 2d wedge mesh in a concave attitude

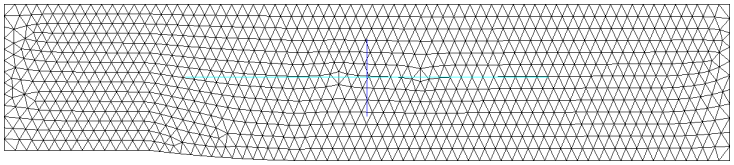

Figure 3.4: 2d wedge mesh in a convex attitude

Numerical treatment

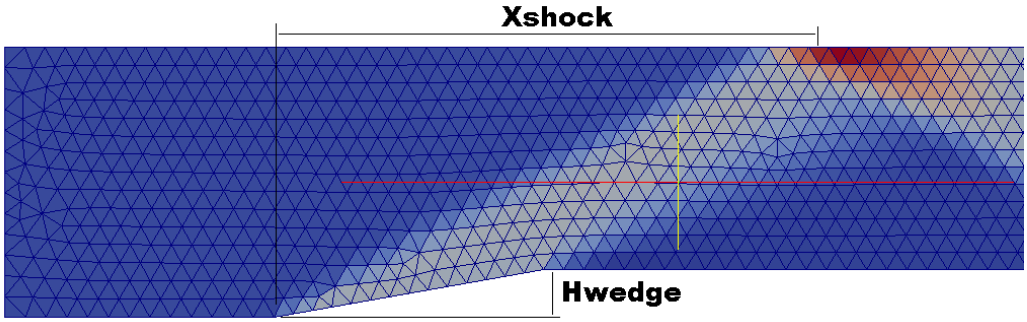

Figure 3.5: Parameters

The position of the oblique shock/expansion fan position on the roof (upper boundary) $X_{shock}(t)/X_{fan}(t)$ was object of the study: theoretical oblique shock/-expansion fan solutions are available and described below and can be very easily compared to the numerical results. Those latter have been gathered from the calculation outputs by considering two different ways to locate the shock: for the first one, the shock is where the maximum drop of fluid-dynamics state solutions is found, whereas for the second one, the shock is positioned where the flow solutions differ the most from the original asympthotic conditions. The arithmetical averaging between these two definitions provided smoother and more precise results, since mesh coarseness prevented a good curve to be deduced from either of them. Wedge height $H_{wedge}(t)$ data has been also gathered from the results, since needed for the analytical solution.

Analytical oblique shock treatment

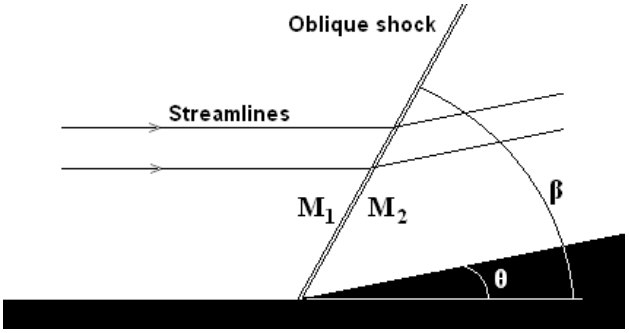

Figure 3.6: Oblique shock characteristic parameters (courtesy of wikipedia.org)

The theoretical solution of the oblique shock is quite straightforward: first of all, we know that

$\gamma = 1.4$,

$M_1 = 3.0$ and

$\theta(t) = atan\left(H_{wedge}(t)\right)$

and that $\delta = 1$ corresponds to the weak shock solution, which is the sought one.

We can thus evaluate

$$\lambda(t) = \sqrt{(M_1^2 - 1)^2 - 3(1 + \frac{\gamma - 1}{2}M_1^2)(1 + \frac{\gamma + 1}{2}M_1^2)tan^2\theta(t)} \tag{3.21}$$

and

$$\chi(t) = \frac{(M_1^2 - 1)^3 - 9(1 + \frac{\gamma-1}{2}M_1^2)(1 + \frac{\gamma-1}{2}M_1^2 + \frac{\gamma+1}{4}M_1^4)tan^2\theta(t)}{\lambda^3(t)} \tag{3.22}$$

and finally derive $x_{shock}(t)$ from:

$$1/x_{shock}(t) = tan\beta(t) = \frac{M_1^2 - 1 + 2\lambda(t)cos(\frac{4\pi\delta + acos\chi(t)}{3})}{3(1 + \frac{\gamma-1}{2}M_1^2)tan\theta(t)} \tag{3.23}$$

Analytical expansion fan treatment

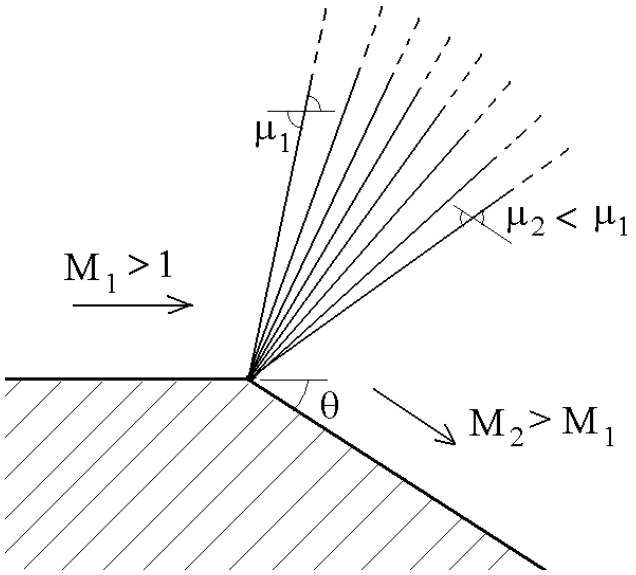

Figure 3.7: Prandtl-Meyer expansion fan characteristic parameters (courtesy of wikipedia.org)

For the expansion fan, the Prandtl-Meyer theory is promptly available. With the same foresaid datas, we start calculating

$$\nu_1(M_1) = \sqrt{\frac{\gamma+1}{\gamma-1}} \cdot atan\sqrt{\frac{\gamma-1}{\gamma+1}(M_1^2-1)} - atan\sqrt{M_1^2-1} \tag{3.24}$$

and then, since

$$\nu_2(M_2) = \theta + \nu_1(M_1) \tag{3.25}$$

we implicitly solve the following relation to obtain M_2:

$$\nu_2(M_2) = \sqrt{\frac{\gamma+1}{\gamma-1}} \cdot atan\sqrt{\frac{\gamma-1}{\gamma+1}(M_2^2-1)} - atan\sqrt{M_2^2-1} \tag{3.26}$$

Now, since

$$\begin{aligned} \mu_1 &= asin\frac{1}{M_1} \\ \mu_2 &= asin\frac{1}{M_2} \end{aligned} \tag{3.27}$$

we may finally desume:

$$1/x_{fan} = tan\beta = tan\frac{\mu_1+\mu_2+\theta}{2} \tag{3.28}$$

Results comparison

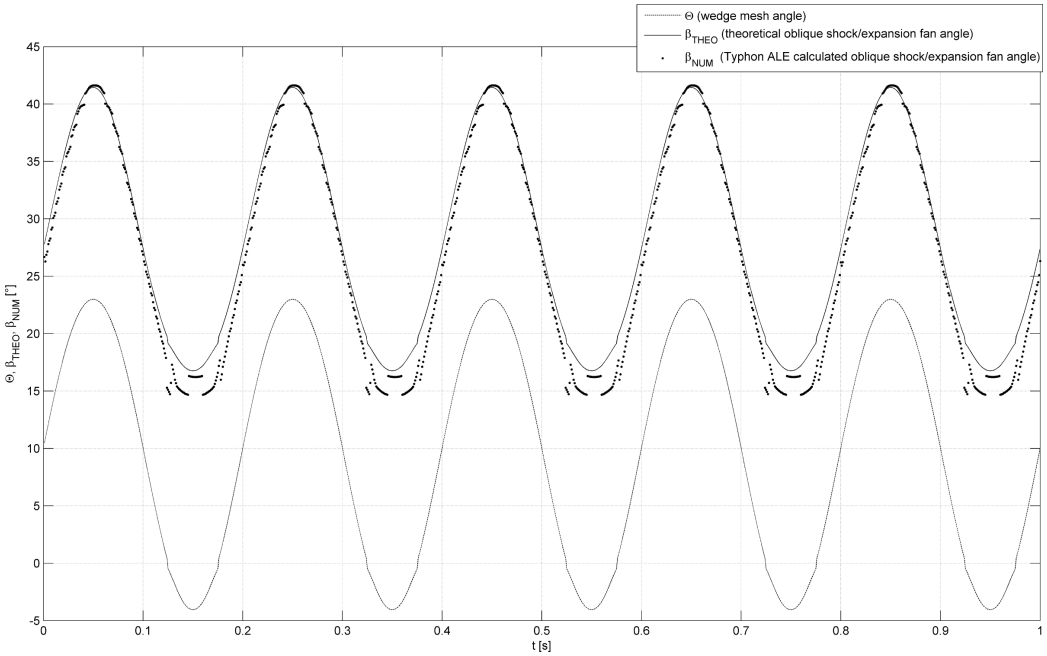

Figure 3.8: 2d Oscillating wedge results

By understanding that the expansion fan position was not easily identified at very small negative θ angles and by allowing some jagging/aliasing of the numerical results due to the foresaid coarseness of the mesh, the numerical and analytical results do perfectly correspond. This noticeable result warrants the correct behavior of the solver in completely different transient geometries.

3.5 Conclusions

Even if not as powerful as many of the Arbitrary Lagrangian-Eulerian extensions available in the literature and in the present day CFD solvers, which also allow remeshing and remapping for significant domain movements, our developed and tested ALE extension has shown excellent feasibility with only moderate source-code interventions and computational burdening. Again caution must be put when choosing the spatial numerical scheme, but to a lesser extent compared to the MRF case. The limitations of the ALE extension itself are less strict than the solver ones. The only further requirement is thus the adoption of an adequate mesh that will tolerate the given movement without macroscopic degradation, and providing enough distance between the moving body and the remaining boundaries in case the "*BODY*" mesh movement interpolation algorythm is going to be used.

Part II

Analysis and results

Chapter 4

The two-dimensional airfoil flutter

4.1 Two degrees of freedom flutter

The extensive test campaign on a NACA0012 rectangular wing flutter available in [6] is the framework for the final validation and comparison between the Moving Reference Frame analysis, the Arbitrary Lagrangian Eulerian study and the published experimental results.

As already mentioned in the MRF and ALE validation cases introduction, again a single combination will be evalued for the computational enquiry, taking benefit of the best possible combination of numerical fluxes (spatial schemes) and time-marching schemes. The previous results of the MRF validational campaign are, in fact, still perfectly valid.

4.1.1 Case definition

According to the NASA paper [6, tab. 5], airflows at several different Machs are investigated. For each of them, a dynamic pressure, a density, a mean angle of attack, a natural flutter frequency, magnitudes and phases of angle of attack and vertical position oscillations are published and given to the solver's MRF and ALE extensions. The exact choices, in details, are:

- a two-dimensional mesh of quadrangulars with an external shape of a rounded diamond is assumed: the diamond grants certainty on the inflow and outflow conditions during MRF calculations. The mesh is displayed down below;
- Euler dynamics are always assumed;
- the spatial numerical scheme is always HLLC with MUSCL higher order extrapolation, since they proved to be the best combination overall;
- the time marching scheme is a simple Forward Euler, for the sake of computational speed;

- the time step was chosen according to the Typhon's $GLOBAL\_STABILITY$ criterium, for which a CFL of 1 is evaluated on every domain cell and the strictest time step is then assumed;
- a steady calculation to a convergence of 10^{-5} is run beforehand, to provide for high quality initial conditions for the unsteady alalyses;
- two full periods of the flutter oscillations are always calculated and the second one is the sole investigated, to exclude eventual initial trasient phenomena;
- the boundary conditions adopted, coherently with the inviscid Euler analyses, are $SYMMETRY$ (slipping wall) on the airfoil, $SUBSONIC\_INLET$ on the inflow and $SUBSONIC\_OUTLET$ on the outflow;
- three different Mach numbers have been particularly studied: 0.3 , 0.51 , 0.82;
- the numeric results are then gathered in terms of pressure coefficients (c_P) over the airfoil surface;
- the numerical and experimental pressure coefficients c_P can finally be translated into lifting coefficients c_L which can be compared.

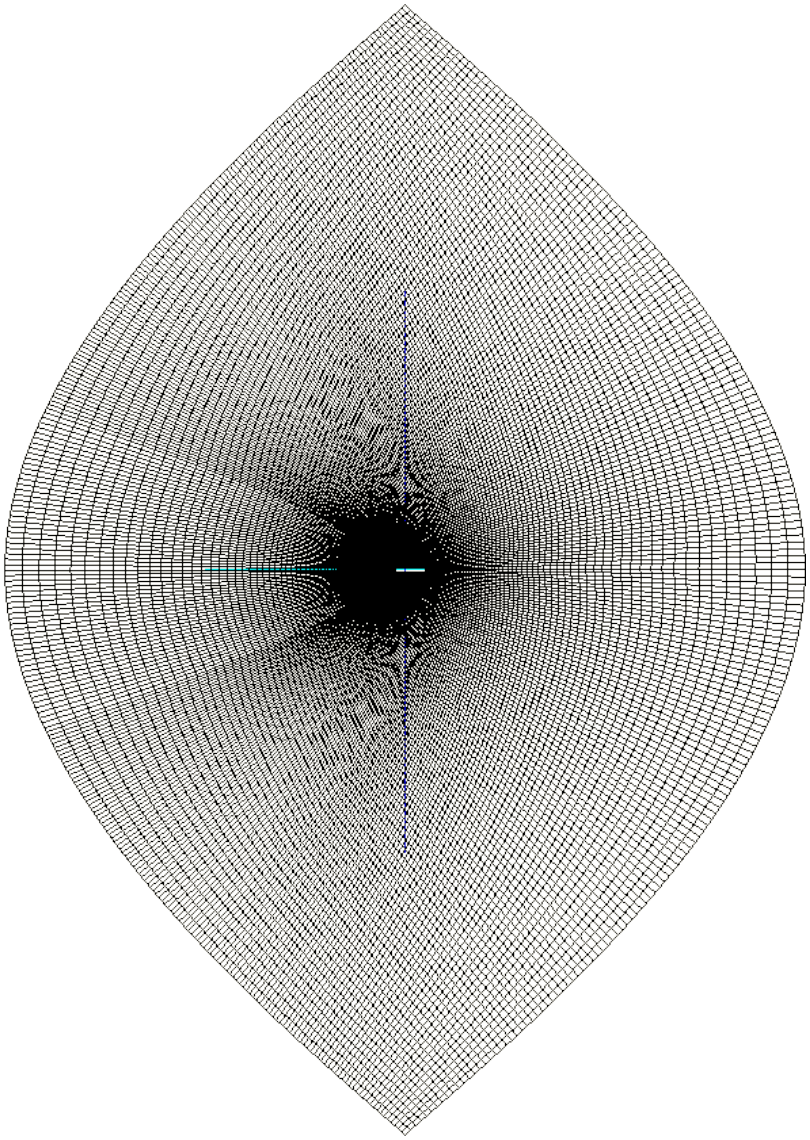

Figure 4.1: 2d mesh used for the MRF and ALE flutter calculations

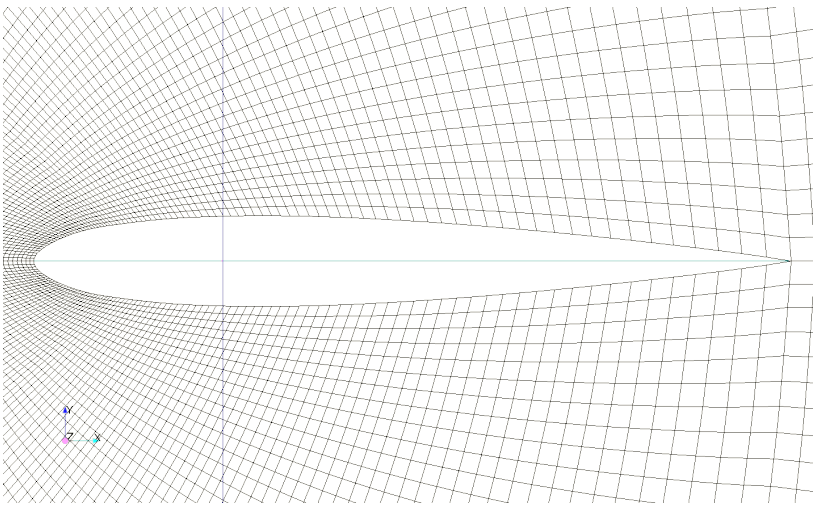

Figure 4.2: NACA0012 airfoil detail of the 2d mesh used for the MRF and ALE flutter calculations

4.1.2 Numerical treatment

For a compressible flow, the pressure coefficient c_P is defined as:

$$c_P = \frac{2}{\gamma M_{\text{inf}}^2} \left(\frac{p}{p_{\text{inf}}} - 1 \right) \tag{4.1}$$

now since M_{inf}, γ and p_{inf} are all known analysis input values, the pressure field p is the only unknown that has to be retrieved from the analysis results.

For the construction of the $c_L - \alpha$ chart, the sole missing relation is

$$c_L = \int_0^1 \left(c_{P_L}(x) - c_{P_U}(x) \right) dx \tag{4.2}$$

that is, the integral along the adimensionalised chord of the difference between the pressure coefficient on the lower surface and the pressure coefficient on the upper surface. This relation is of course useful for both the experimental and the numerical data, provided that data fitting (interpolation) is previously made.

4.1.3 Results

The gathered numerical and experimental results are being plotted in the same charts, to compare the magnitudes and trends.

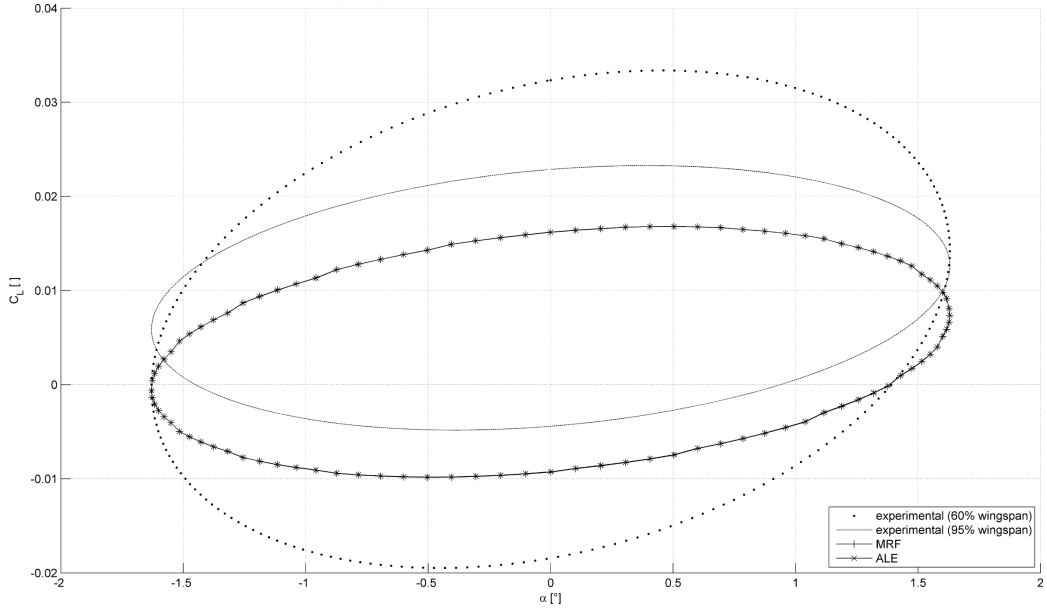

Figure 4.3: $c_L - \alpha$ charts of experimental, MRF and ALE results for the NACA0012 flutter at Mach=0.3

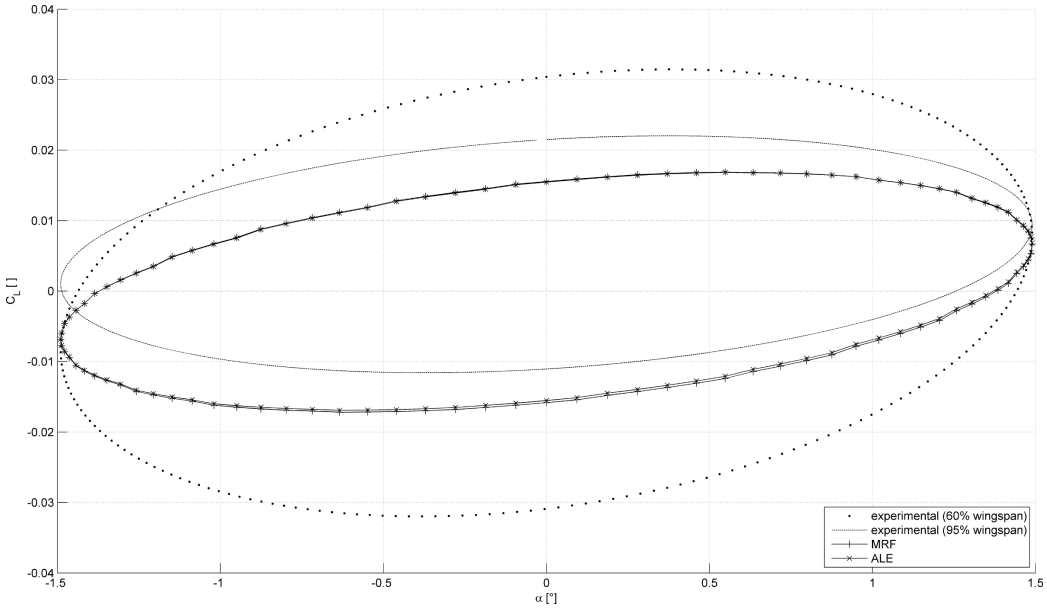

Figure 4.4: $c_L - \alpha$ charts of experimental, MRF and ALE results for the NACA0012 flutter at Mach=0.51

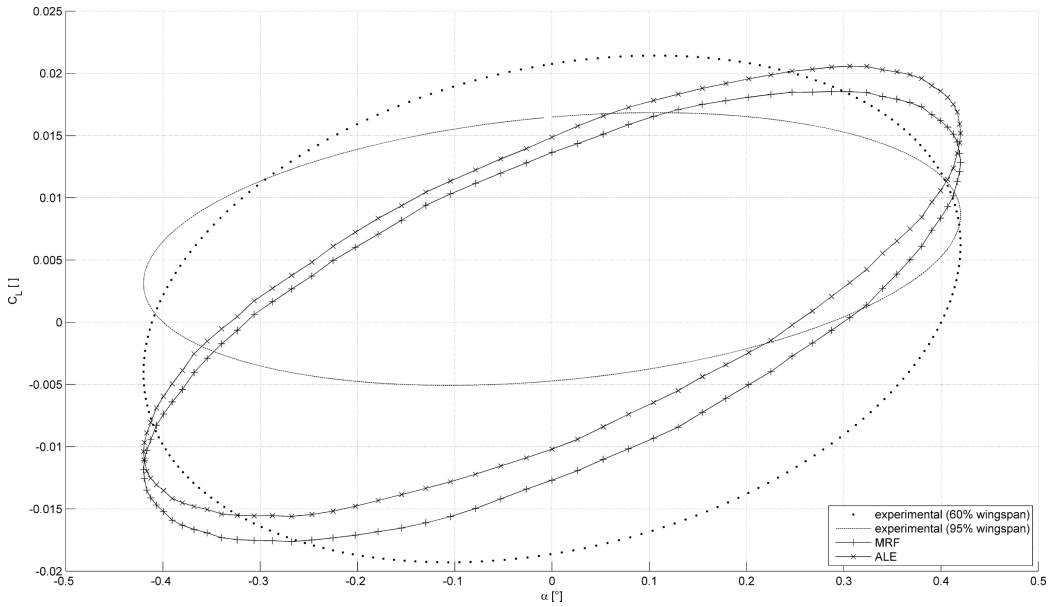

Figure 4.5: $c_L - \alpha$ charts of experimental, MRF and ALE results for the NACA0012 flutter at Mach=0.82

4.2 Conclusions

The plotted results manifest good coherence in the orders of magnitude among experimental and numerical results and, excluding the fact that the MRF calculation at Mach=0.82 has been wrongly made with null average angle of attack, manifest also exceptional correspondence between ALE and MRF solutions. The noticeable difference in the trends and shapes between experimental and numerical results can be explained firstly by the fact that the numerical analysis has been done on a two-dimensional airfoil, whereas the real phenomenum is three-dimensional (and this fact is manifest alone by comparing the different experimental results between 60% and 95% wingspan), and possibly because, due to the mesh coarseness, Euler inviscid dynamics have been preferred to the Navier-Stokes viscous ones, and slipping wall boundary condition has been preferred to the non-slipping one. Slight errors in the input case parameters both due to experimental errors or to the conversions to the international metric system may also have played a role, since very little variations on magnitude and frequency of the oscillations can drastically change the flutter phenomenum. The differences are considered nonetheless reasonable, since they also fall in-between the experimental extreme values.

Chapter 5

Overall conclusions

All the original objectives presented in the introduction could be reasonably achieved: Moving Reference Frame and Arbitrary Lagrangian-Eulerian extensions to the Typhon finite-volume cell-centered CFD solver have been mathematically derived, developed, extensively tested and successfully compared.

The results manifest significant coherence among the two and are perfectly comparable with the experimental results available. Numerical errors are markedly low even in the Moving Reference Frame cases, which, as stated, is working via source-terms, theorically and traditionally more affected by errors.

The end users don't have new significant restrictions for the use of the new moving geometries extensions apart from the original solver's ones. The additional case definitions follow the original Typhon traditions and require only the basic necessary parameters. The computational times are only sligthly greater for the ALE case, whereas are almost unchanged for the MRF case: this little difference is due to the mesh movement algorythm, which is nonetheless one of the simplest available.

Countless further improvements are worth and possible, since each approach holds a plethora of additional features, but they fall beyond the scope of the present work, which nontheless established the basis for all and every of them.

Part III

Bibliography

Bibliography

[1] (2006) Alexei Y. Poludnenko, Alexei M. Khokhlov - *"Computation of fluid flows in non-inertial contracting, expanding, and rotating reference frames"* - Journal of Computational Physics

[2] (2004) J. Donea, A. Huerta, J.-Ph. Ponthot and A. Rodrìguez-Ferran - *"Arbitrary Lagrangian-Eulerian methods"* - Encyclopedia of Computational Mechanics - edited by Erwin Stein, René de Borst and Thomas J.R. Hughes - Copyright 2004 John Wikey & Sons, Ltd.

[3] (2008) Pierre-Henri Maire - *"A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes"* - Journal of Computational Physics

[4] (2010) D. Muffo, G. Quaranta, A. Guardone, P. Mantegazza - *"Interface Velocity Consistency in time-accurate flow simulations on dynamic meshes"*

[5] (2009) Alberto Guardone and Giuseppe Quaranta - *"Failure of geometric conservation law for non-uniform flow simulations on dynamic meshes"* - International Conference on Computational Methods for Coupled Problems in Science and Engineering

[6] (1992) J.A.Rivera Jr, B.E.Dansberry, M.H.Durham, R.M.Bennett and W.A.Silva - *"Pressure Measurements on a rectangular wing with a NACA0012 airfoil during conventional flutter"* - NASA Technical Memorandum 104211

[7] (1995) Jean-Marc Moschetta and D.I. Pullin - *"A robust low diffusive kinetic scheme for the Navier-Stokes/Euler equations"* - Journal of Computational Physics

[8] (2007) Hiroaki Nishikawa, Keiichi Kitamura - *"Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers"* - Journal of Computational Physics

[9] (1994) Murali Beddhu, Min-Yee Jiang, Lafayette K. Taylor and David L. Whitfield - *"Towards computations of ocean flows using Navier-Stokes Equations"*

[10] (2009) Matthias Preisig, Thomas Zimmermann - *"Free-surface fluid dynamics on moving domains"* - Computer methods in applied mechanics and engineering

[11] (2009) Marx Chhay and Aziz Hamdouni - *"A new construction for invariant numerical schemes using moving frames"* - Comptes Rendus Mechanique

[12] François Dubois - *"Partial Riemann problem, boundary conditions, and gas dynamics"* - Artificial Boundaries and Layers, Domain Decomposition. Applications to Large Scale Computers

Compiled on September 19, 2011.