A Two-Dimensional Fluid-Structure Coupling Algorithm for the Interaction of High-Speed Flows with Open Shells

by

Daniel See Wai Tam

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2004

Author .....
Department of Aeronautics and Astronautics
May 20, 2004

Certified by .....
Raúl Radovitzky
Assistant Professor
Thesis Supervisor

Accepted by .....
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

A Two-Dimensional Fluid-Structure Coupling Algorithm for the Interaction of High-Speed Flows with Open Shells

by

Daniel See Wai Tam

Submitted to the Department of Aeronautics and Astronautics
on May 20, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

The design of future light aerospace structures will require numerical tools to accurately describe the strongly coupled dynamics of the interactions between a light structure and a flow surrounding it. Specific examples include inflatable structures and parachutes used as deceleration devices during planet entry. In this work, an algorithm for simulating the solid-fluid interactions of a highly-deformable open shell structure and a compressible fluid flow is presented. The main objective is to extend the algorithm previously presented by the authors to the case of open shell structures immersed in a fluid. For simplicity, we restrict our attention to the two-dimensional case. The computational strategy is based on combining an Eulerian finite volume formulation for the fluid and a Lagrangian formulation for the structural response. The coupling between the fluid and the solid response is achieved *via* a novel approach based on extrapolation and velocity reconstruction aided by level sets. The accuracy of the proposed approach is verified against exact solutions of supersonic flow past a rigid flat plate. The numerical results reproduce all the details of the flow field, including the—very important—forces on the structure. The versatility of the proposed coupling algorithm is demonstrated in simulations of supersonic flow past a highly-deformable infinite plate.

Thesis Supervisor: Raúl Radovitzky
Title: Assistant Professor

Acknowledgments

First, I would like to express my deep gratitude to Prof. Raúl Radovitzky, my advisor, for giving me the opportunity to come and study at MIT, which has been an enriching experience in many regards. I greatly appreciated his guidance and advices over the past two years.

The support of NNSA ASC Academic Alliance Program under contract no. W-7405-ENG-48, subcontract no. B523297 is gratefully acknowledged.

I would like to thank Pr. Ravi Samtaney for giving me the opportunity to work with his fluid code RM3D and for the support he provided.

I would like to thank all the professors and students from MIT, who have helped me in my research in a way or another and especially all the students from TELAC for their support, encouragements and for patiently listening to me talking french half of the time.

I would like to thank my friends from church and from the MIT graduate christian fellowship, for introducing me to american culture and of course for their help, support and prayers.

I will always be grateful to my friends on both sides of the Atlantic. They have always been there when I most needed them and their joy have illuminated many gray winter days. I would like to give special thank to Alex, Damien, Marie-Eve and Nami for their usefull advices concerning my work here at MIT.

I will be eternally grateful to my parents. Je vous dois bien plus que je ne saurai l' exprimer en quelques lignes, et vous serai à jamais reconnaissant de votre soutien, de votre amour et du merveilleux cadeau que vous m'avez fait: celui de vivre. I am also grateful to my sister Andrea and my brother Thomas for the joy of being part of the same family.

I could not end here without expressing my sincerest gratitude to the one without whom none of this would be and through which all things are possible: my father in heaven.

Contents

List of Figures

Chapter 1

Introduction

Current and future engineering applications involving the coupled mechanics of fluids and solid structures require robust and efficient algorithms to better understand the nature of their interactions and to provide quantitative answers to assist in design. Improved numerical models of fluid structure interactions (FSI) are essential for better understanding the coupled fluid-solid dynamics. For instance, in biomedical research the study of the coupled interactions between physiological flows with tissues is of highest interest in the study of cardiovascular diseases; in ocean engineering FSI is crucial in offshore structure and naval ship design; in civil engineering FSI has become critical in the design of structures providing increased survivability to blast loading, in aeronautical engineering FSI has always been key in aircraft design and it has become even more important in the development of new flight concepts based on morphing and flapping wings and also in the area of noise reduction. Usually, a specific mathematical model and concomitant numerical strategy is adopted depending on the flow regime and on the kind of structure involved.

In this work, we concentrate our attention on the interactions between highly deformable thin structures and high-speed flows. This focus is motivated by applications in the deployment of parachutes used as deceleration devices during atmosphere entry of interplanetary probes. In fact, the atmosphere entry phase and, more specifically, the deployment of the parachute is one of the critical points for a successful mission. Unfortunately, the canopy inflation process is not well understood and very

complex to model since the fluid and structure dynamics are tightly coupled and highly non-linear. Current and future interplanetary exploration missions demand the availability of efficient numerical tools for the design of these light structures. This work is part of the research effort to provide a robust and accurate means of modeling these interactions.

In recent years many different strategies have been proposed for the numerical simulation of coupled fluid-structure systems. In this section, a brief review of these methods is presented. In the interest of focus and conciseness, attention is restricted to those strategies considered most relevant to our specific problem. A typical FSI problem involves the dynamics of a structure with a potentially complex geometry which deforms in response to the forces exerted by the fluid. These forces result from the details of the transient flow field which is, in turn, affected by the evolving geometry of the structure as it deforms. A key element of the modeling of an FSI modeling strategy is an algorithm to impose the satisfaction of the conservation laws at the dynamically evolving interface. In particular this requires the fluid model to allow for the presence of an irregular solid interface inside the computational domain. The FSI approach depends strongly on the numerical models used to represent the solid and the fluid. Suitable algorithms for describing the flow and the deformation of the structure constitute basic and essential building blocks toward an effective fluid-structure coupling model.

In general, the dynamic deformation of solid structures is most adequately described in a Lagrangian framework, especially in the case that the deformations are large. The main advantage of the Lagrangian approach lies in its ability to naturally track the evolution of material properties associated with the material points as well as in the treatment of boundary conditions at material surfaces such as free boundaries or fluid-solid interfaces. In contrast to Eulerian approaches, boundary conditions are enforced at material surfaces *ab initio* and therefore require no special attention. In this work, we propose a lagrangian formulation to describe the large dynamic deformations of two-dimensional thin structures (rods) having both bending and membranal stiffness.

On the other hand, Lagrangian formulations are inadequate in the case of high-speed flows or flows involving significant vorticity due to the unavoidable mesh distortion incurred during deformation, which reduces the stable time step and the overall accuracy of the simulation. This problem can be partially remedied by the use of remeshing [39]. However, remeshing increases the complexity of the algorithm and of its implementation and suffers of robustness problems in the three-dimensional case. Therefore, Eulerian approaches in which the field equations are formulated in terms of spatial variables are better suited for most fluid flows. In this case the equations are discretized in space either on an unstructured triangular (tetrahedral) mesh in the two (three) dimensional case or on a structured cartesian grid. In the unstructured mesh approach, the flow is described in the Eulerian framework by recourse to a weak finite element formulation of the governing conservation equations, [35, 32, 34]. The main advantage of this approach is that it can accommodate complex geometries of flow domain without compromising the order of accuracy of the numerical solution. Adaptive remeshing can be used in this case as a means of improving the quality of the solution, [35, 34]. Unstructured mesh formulations have been extended to the case of moving flow boundaries, which enables the consideration of fluid-structure interaction problems. However, in the case of large deformations of the boundary the same mesh distortion problem described in the foregoing is incurred.

Approaches based on structured body-fitted meshes are a plausible alternative for dealing with irregular boundaries or immersed solid walls, [45]. The basic idea is to generate a curvilinear system of coordinates and to solve the system of partial differential equations on a uniform square through a conformal mapping. The system of equations needs to be first derived analytically in the transformed plane. This approach is not easily extensible to three dimensions and, therefore, has found limited interest.

In this work, we focus our attention on high-speed flows of inviscid fluids. We adopt a well established numerical approach for discretizing the governing Euler equations of compressible flows used in the computational gas dynamics community which is based on a finite volume formulation on a structured cartesian grid. The cartesian

grid discretization finds its appeal in its simplicity for formulating the fluxes of the conserved field variables without paying a penalty in accuracy due to element shape. The main difficulty with structured grid methods is in applying boundary conditions at an interface which does not fit the geometry of the structured grid as is almost always the case.

Different methods have been proposed to handle this issue. One of the basic strategies is based on the concept of cut cells. As the name suggests, in this approach the cells overlapping the boundary of the domain are cut and the specific geometry of the resulting cut-cell is considered when computing the fluxes on their boundaries to satisfy the conservation laws at the boundary. A particular challenge in this approach lies in maintaining the stability of the explicit time integration scheme without significantly reducing the time step size. Applying a conservative scheme on a cut cell that can be arbitrarily small due to the irregular boundary often leads to a serious deterioration of the stable time step given by the Courant-Friedrichs-Levy (CFL) stability condition, which seriously limits the can eventually have a serious impact on the stability of the scheme. The following strategies have been proposed in order to address the issue of stability associated with cut-cell approaches:

- *Cell merging techniques:* A first approach to handle the issue of small unstable irregular cut cells is to merge those cells which would cause reductions in the time step size to maintain stability, with larger neighboring cells. This is to ensure that control volumes are large enough to satisfy the CFL condition. Chiang et al. [8], Powell et al. [15] as well as Quirk, J. [37] have explored this method and coupled it to Adaptive Mesh Refinement (AMR see [38]) to increase resolution at the boundary. Although the merging of cells induces a certain loss of accuracy at the boundary, numerical results suggest that the method is globally second order accurate and first order accurate at the boundary.

- *Flow redistribution:* Collela et al. [4] proposed an alternative approach based on a redistribution procedure. A reference solution is computed at first, using fluxes computed by a second order scheme but ignoring the irregular boundary.

A correction is then computed in the cut cells and is only partially applied to these cells in order to keep stability. The remainder of the computed correction is then redistributed over the neighboring cells to keep the scheme conservative.

- *Large time step approach:* Another approach for cut cells has been proposed by LeVeque, R. and Berger, M. in [28, 29, 6]. This method relies on the large time step method developed by LeVeque [27]. The stable time step is based on the control volume of the regular cells and the large time step method is used at the irregular cut cells in order to compute the solution without losing stability. The scheme appears to be globally second order and somewhat better than first order at the boundary.

In spite of the success of these conservative methods in addressing the issue of stability associated with cut-cell approaches, increased implementation complexity, especially in three dimensions, has seriously limited their widespread adoption in production applications. Furthermore, the extension of these methods to the case of moving boundaries undergoing large displacements and eventually crossing grid points is particularly challenging and appears to remain elusive.

On the other hand, non-conservative methods based on extrapolation have been developed to solve Euler's equations on a structured grid. One example of particular interest for the present work is the Ghost Fluid Method (GFM) originally proposed by Fedkiw et al [18]. The GFM originated as an algorithm for handling multi-phase multi-fluid problems where interfaces are contact or shock discontinuities. The GFM method has subsequently been extended to model flows around irregular solid walls, [33, 14, 17, 2, 12]. This method is described in more detail in Chapter 3.

Another critical aspect in the formulation of effective FSI algorithms has to do with the ability of the fluid dynamics approach to deal with moving boundaries. This is particularly critical in the case of large-amplitude motions of the solid-fluid interface.

A commonly used approach is based on Arbitrary Lagrangian Eulerian (ALE) schemes[5, 3, 23]. This hybrid class of schemes is neither fully Eulerian nor La-

grangian. The unstructured mesh moves with a prescribed velocity $\vec{u}_a$ which is equal to zero far from the moving fluid-solid interface and equal to the velocity of the interface close to it. The conservation equations are expressed on this arbitrary moving mesh, thus the formulation reaches the Eulerian limit far enough from the interface when $\vec{u}_a = \vec{0}$ and the mesh follows the body motion at the interface $\vec{u}_a = \vec{u}_{interface}$, which is the compatibility condition. In the intermediate region between these two limits, the mesh moves with a prescribed velocity which at the same time respects the motion of the interface and minimizes the distortion of the grid. ALE methods allow a reasonably simple treatment of the fluid-solid interface, which is considered as Lagrangian, while limiting grid distortion as in Eulerian schemes. However, in order to define the arbitrary speed of the mesh $\vec{u}_a$ at the interface the solution often needs to be known a priori. For fully coupled systems, where the Lagrangian interface undergoes large deformations, severe mesh distortion cannot be avoided and the method often fails to give a solution. Another disadvantage of ALE approaches is the difficulty of its implementation and its integration in existing CFD codes.

Although perhaps not best suited for coupled systems involving ample structure motions, ALE methods have been successfully used in combination with turbulence models in fluid-solid interaction problems involving viscous flows. Farhat *et al.* have demonstrated fully coupled simulations of the aeroelastic interactions of a full scale F-16 using an ALE approach [19, 20, 16]. Also, Tezduyar et al. [43, 25, 44] have proposed a method (Deforming Spatial Domain/Stabilized Space Time) which is based on an ALE concept and have successfully applied it to problems of flows past deforming—albeit already inflated—parachutes. It bears emphasis that in all the applications presented by Farhat *et al.* and Tezduyar *et al.* the deformation of the structure is relatively small compared to the motions experienced by an inflating parachute, as is of interest in this work. As a notable exception, Benney et al presented a two dimensional simulation of the inflation of a parachute [5] using the methods developed in collaboration with Tezduyar. It is not clear that the same simulation can be done in three dimensions with this method due to the issues associated with mesh distortion and lack of robustness of remeshing algorithms.

An alternative to ALE methods is considered and developed in this thesis. The work presented aims at developing a method to simulate the coupled interactions of a high-speed flow and an immersed, highly deformable, thin, open structure. In order to avoid the complexity of generating unstructured meshes around moving interfaces, an Eulerian formulation based on a fixed cartesian grid is used to model the fluid dynamics. At the same time, the coupling strategy is expected to be robust enough to allow large displacements of immersed interfaces without affecting the numerical stability of the scheme. Another requirement for the sought algorithmic approach is a straightforward extension to three dimensions.

The approach we propose, here and subsequently referred to as the Ghost-fluid Eulerian Lagrangian (GEL) method, is an extension of previously presented algorithms to the case of thin open structures immersed in the fluid in which the details of the flow are relevant on both sides of the deforming structure. Previously proposed GEL coupling algorithms have focused on the interaction of compressible flows with bulk solids [33, 14] and thin shells [12]. The GEL algorithm couples the large-deformation Lagrangian finite element formulation and the Eulerian finite volume formulation adopted here. The coupling is accomplished *via* a novel technique based on level sets. At each time step, the distance function from the solid boundary is computed on the Eulerian grid. The resulting implicit representation of the fluid-shell boundary in the deformed configuration is used to enforce the conservation laws at the boundary between the fluid and the solid. In the case of thin structures, the GEL algorithm had the limitation that the shell had to be a closed manifold. This restriction was imposed by the assumption that the shell structure had a well-defined interior and exterior, which was the basis of the coupling algorithm based on level sets. In this work, we extend the algorithm [12] presented by Cirak and Radovitzky to the case of open thin shell structures interacting with a compressible flow. The extended approach retains the conceptual ideas of the ghost fluid method, but allows symmetrical treatment of both sides of an immersed thin boundary. Thus, proper boundary conditions are applied on each side of the interface. For simplicity, attention is restricted to the two-dimensional case.

This thesis is organized as follows: in Chapter 2, the formulation of the numerical models for the individual solid and fluid phases is presented. This chapter includes a detailed description of the special Lagrangian model of the large dynamic deformations of two dimensional rods and a review of the computational fluid dynamics approach. In chapter 3, we describe the Ghost-fluid Eulerian Lagrangian (GEL) method on which the coupling strategy relies before deriving the proposed algorithm extension that allows the imposition of adequate boundary conditions on a thin and immersed fluid-solid interface. Chapter 4 is devoted to the presentation of numerical simulations attesting to the feasibility and versatility of the proposed FSI algorithm as well as to the assessment of its accuracy. First, simulations corresponding to the supersonic flow past a very thin rigid boundary at different angles of attack are compared with the exact solutions. The purpose of these simulations is to demonstrate the ability of the proposed approach to describe the flow on both sides of a very thin structure. This chapter concludes with the presentation of the results corresponding to a fully coupled simulations of a supersonic flow initially normal to a flexible rod. In Chapter 5, a summary and conclusions of this work are presented.

Chapter 2

Formulation of numerical models for highly-deforming thin structures and high-speed flows

In this chapter, the numerical formulation and the details of the algorithms describing the dynamics of the individual solid and fluid phases are presented. As stated in the introduction given in Chapter 1, the success and effectiveness of a FSI algorithm depend critically on the suitability of the individual algorithms to describe the specific dynamics (flow or deformation) of each phase (fluid or solid). In the next chapter, we present the strategy and formulation for coupling these two individual formulations into an overall fluid-structure interaction algorithm.

2.1 A non-linear large displacements rod model

In this section the mathematical model, numerical formulation and algorithmic details corresponding to the structural model is presented. The focus of interest in this work is on the dynamic deformations of thin structures resulting from the unsteady interactions with a high speed flow. For simplicity, attention is restricted to the two dimensional case. The appropriate structural model, thus, corresponds to the large dynamic deformation of thin (one dimensional) structural elements which can with-

stand and transfer transverse loads through their bending and membrane stiffnesses. The case of an infinite plate can be conveniently considered within this modeling assumption by adding a plane strain condition. In the following, we derive a new lagrangian formulation of the nonlinear-elastic, dynamic deformations of thin rods. A shell formulation such as presented in [9, 10] would provide an adequate structural model in a three dimensional extension of the work presented in this thesis.

We consider a straight slender rod element, which in its undeformed configuration has a length L and uniform cross-sectional area A, moment of inertia I and mass density ρ. We assume that the rod deforms within a plane and that the principal direction of inertia for bending remains perpendicular to that plane throughout the motion. We refer the undeformed configuration of the rod element to orthonormal axes $(\mathbf{X_1}, \mathbf{X_2})$, where $\mathbf{X_1}$ is aligned with the axis of the rod, see Figure 2-1. In the finite element formulation proposed, the unknowns represent the physical displacements and rotation at extremity (node) $a = 1, 2$ of the rod element.

The rod element undergoes a motion consisting of a finite rotation, a finite uniform stretch and a small bending distortion. Let l be the length of the rod element after stretching. In order to facilitate the description of bending, the rotated and stretched configuration of the rod element is referred to orthonormal axes $(\mathbf{x_1}, \mathbf{x_2})$, where $\mathbf{x_1}$ is aligned with the rotated axis of the rod. Let w be the deflection of the rod normal to the rotated axis, i. e., in the $\mathbf{x_2}$-direction. A cross section of the rod is assumed to remain normal to the neutral axis, thus:

$$\theta \approx \frac{\partial w}{\partial x_1} \tag{2.1}$$

the attendant (small) rotation of the axis in the Euler-Bernouilli beam bending theory. With this kinematic assumption, the deformation mapping for the rod element is then:

$$x_1 \approx \frac{l}{L} X_1 - \theta(X_1) X_2 \tag{2.2}$$

$$x_2 \approx X_2 \tag{2.3}$$

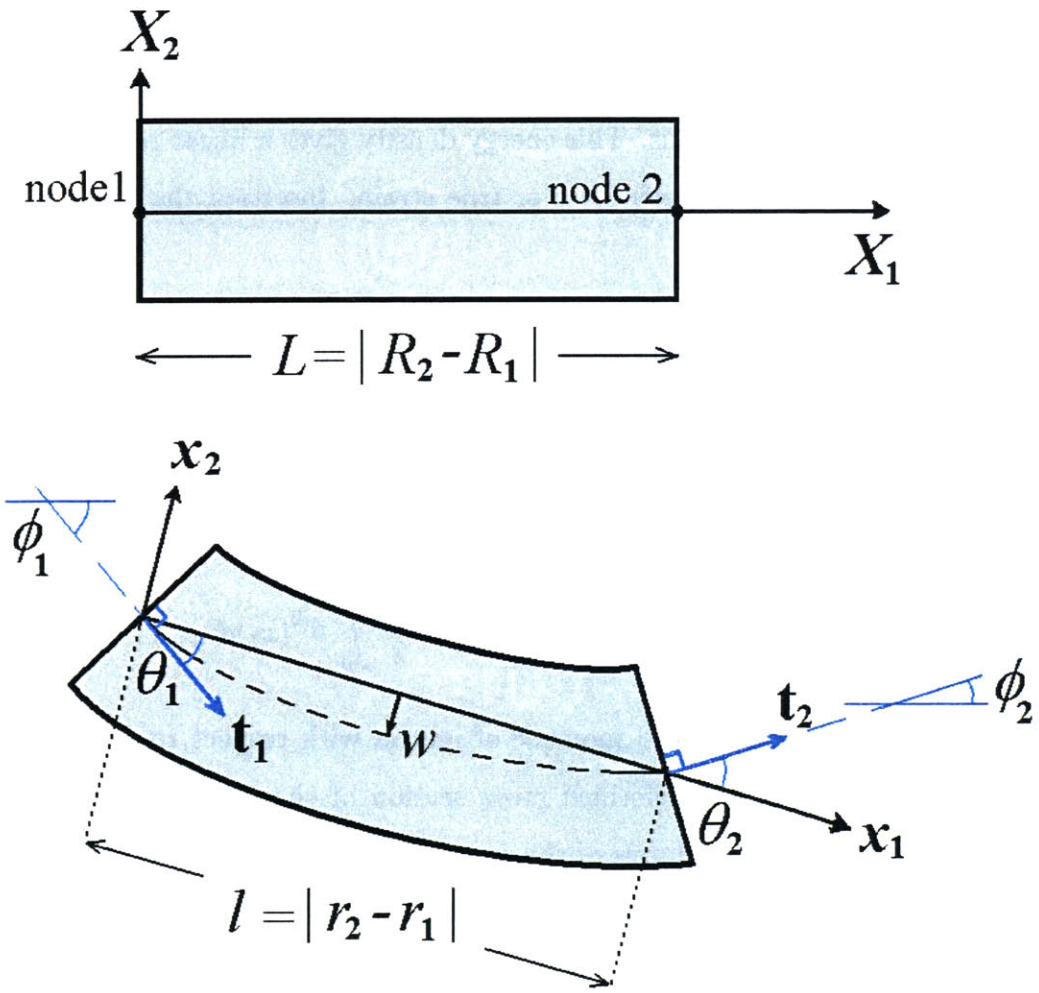


Figure 2-1: Kinematics of the rod model

The stretch of the longitudinal fibers of the rod element follows as

$$\lambda = \frac{\partial x_1}{\partial X_1} \approx \frac{l}{L} - \theta'(X_1)X_2 \tag{2.4}$$

Assume now that the energy density per unit undeformed area of the rod is given by

$$W(\lambda) = E(1 - \lambda + \lambda \log \lambda) \tag{2.5}$$

where E is the Young's modulus. This energy density gives a linear relation between the nominal stress and the logarithmic or true strain. Inserting the rod kinematics (2.4) into (2.5) gives

$$W(\lambda) \approx E\left[1 - \frac{l}{L} + \frac{l}{L}\log\frac{l}{L}\right] + \frac{E}{2}\frac{L}{l}[\theta'(X_1)X_2]^2 \tag{2.6}$$

The strain energy of the rod is obtained by integrating equation (2.6) over the volume of the undeformed rod with the result

$$U \approx EA\,[L - l + l\log(l/L)] + \frac{EI}{2}\int_0^l \theta'^2(x_1)dx_1 \tag{2.7}$$

where A and I are the area and moment of inertia with respect to axis normal to the bending plane of the undeformed cross section of the rod. In evaluating the integral along the undeformed axis of the rod, a change of variables to the deformed configuration has been conveniently taken advantage of. We use hermitian cubic interpolation to represent w and its derivatives as a function of $\mathbf{x_1}$. This automatically satisfies the requirement of C^1 interelement continuity of w. Let

$$\theta(0) = \theta_1, \qquad \theta(l) = \theta_2 \tag{2.8}$$

In addition we have

$$w(0) = w(l) = 0 \tag{2.9}$$

From these conditions we obtain the following representations:

$$\frac{w(\xi)}{l} = [(1-\xi)^2\xi]\,\theta_1 + [\xi^2(\xi-1)]\,\theta_2 \tag{2.10}$$

$$\theta(\xi) = [(1-3\xi)(1-\xi)]\,\theta_1 + [\xi(3\xi-2)]\,\theta_2 \tag{2.11}$$

$$\theta'(\xi)\,l = [2(3\xi-2)]\,\theta_1 + [2(3\xi-1)]\,\theta_2 \tag{2.12}$$

where $\xi = \frac{x}{l}$. Inserting (2.12) into (2.7), the integral term in (2.7) can be evaluated analytically giving the element energy in the form

$$U \approx EA\,[L - l + l\log(l/L)] + \frac{2EI}{l}(\theta_1^2 + \theta_1\theta_2 + \theta_2^2) \tag{2.13}$$

In order to render (2.13) explicit in terms of nodal degrees of freedom, let $\mathbf{R_a}$ be the position vectors of nodes $a = 1, 2$ of the rod element in its undeformed configuration, and let $\mathbf{r_a}$ be the position vectors of nodes $a = 1, 2$ of the rod element in its deformed configuration, see Figure 2-1. In addition, let ϕ_a be the angle between the tangent to the rod and the $\mathbf{X_1}$-axis at nodes $a = 1, 2$ of the rod element. We have

$$L = |\mathbf{R_2} - \mathbf{R_1}|\,, \quad l = |\mathbf{r_2} - \mathbf{r_1}| \tag{2.14}$$

In addition, let

$$\mathbf{t_a} = (\cos\phi_\mathbf{a}, \sin\phi_\mathbf{a}), \qquad \mathbf{a = 1, 2} \tag{2.15}$$

be the unit tangent vector to the rod. Then

$$\theta_a \approx \frac{\mathbf{r_2} - \mathbf{r_1}}{l} \times \mathbf{t_a}, \qquad \mathbf{a = 1, 2} \tag{2.16}$$

Using these relations, the energy (2.13) can be expressed in terms of the nodal degrees of freedom $(\mathbf{r_a}, \phi_\mathbf{a})$, $a = 1, 2$, see appendix A.

At equilibrium, the potential energy of the rod is stationary:

$$\delta\Pi = \delta\Pi_{int} + \delta\Pi_{ext} = 0 \tag{2.17}$$

leading to the nonlinear system of algebraic equations:

$$\mathbf{F}_h^{int}(\mathbf{x}_h) = \frac{\partial \Pi_{int}}{\partial \mathbf{x}_h} = -\frac{\partial \Pi_{ext}}{\partial \mathbf{x}_h} = \mathbf{F}_h^{ext} \tag{2.18}$$

where $\mathbf{F}_h^{int}(\mathbf{x}_h)$ ($\mathbf{F}_h^{ext}$) represents the global finite element array of internal (external) forces and $\mathbf{x}_h$ is the finite element array of nodal generalized degrees of freedom. Explicit expressions for the array of internal forces are given in Appendix A. In the FSI interaction problems of interest in this work, the array $\mathbf{F}_h^{ext}$ represents the external nodal forces equivalent to the traction boundary conditions imposed on the rod structure by the flow. Since in this work an inviscid model is adopted for the fluid flow, we consider that the only aerodynamic force acting on the structure is due to the fluid pressure. The expression for $\mathbf{F}_h^{ext}$ is derived in a approximate but variationally consistent way, as follows:

$$F_{ia}^{ext} = \int \int_{So_2} -p\delta_{ij} N_a \mathbf{n}_j ds \tag{2.19}$$

where i is the degree of freedom, a the node, F_{ia}^{ext} the traction constraint on the ia^{th} degree of freedom of the rod element, δ_{ij} the Kronecker symbol, N_a the shape function of node a and p is the local value of the pressure exerted on the rod by the fluid and interpolated from the numerically computed flow field.

Under dynamic conditions, the weak formulation of the governing equations yields Hamilton's principle for continuous media. The Lagrangian $L = K - \Pi$ can be defined, with $\Pi = \Pi_{int} + \Pi_{ext}$ the potential energy introduced previously, and K the kinetic energy

$$K = \frac{\rho A L}{4}(|\mathbf{v}_1|^2 + |\mathbf{v}_2|^2) \tag{2.20}$$

where

$$\mathbf{v}_a = \dot{\mathbf{r}}_a, \qquad a = 1, 2 \tag{2.21}$$

are the nodal velocities. With this associated Lagrangian, Hamilton's principle can

be written in the following form:

$$\delta L = \delta K - \delta \Pi_{int} - \delta \Pi_{ext} = 0 \tag{2.22}$$

for a variationally admissible virtual displacement. A complete derivation of Hamilton's principle can be found in regular textbooks [40]. Since the virtual displacement is arbitrary, it follows that:

$$\mathbf{M}_h \ddot{\mathbf{x}}_h + \mathbf{F}_h^{int}(\mathbf{x}_h) = \mathbf{F}_h^{ext}(t) \tag{2.23}$$

where $\mathbf{M}_h$ is the mass matrix, $\ddot{\mathbf{x}}_h$ is the array of nodal accelerations, $\mathbf{F}_h^{int}(\mathbf{x}_h)$ is the array of internal forces and $\mathbf{F}_h^{ext}(t)$ array of external forces which depends on time t. As may be expected, rotational inertia effects play a negligible role in the dynamics of the rod and are thus ignored. For very thin rods, we thus adopt a lumped mass matrix approach derivable from the expression 2.20 for the kinetic energy of the element by recourse to Hamilton's principle. The lumped mass matrix is the diagonal matrix such that:

$$K = \frac{1}{2} \cdot \begin{pmatrix} \mathbf{v}_1 & \theta_1' & \mathbf{v}_2 & \theta_2' \end{pmatrix} \cdot \mathbf{M}_h \cdot \begin{pmatrix} \mathbf{v}_1 \\ \theta_1' \\ \mathbf{v}_2 \\ \theta_2' \end{pmatrix} \tag{2.24}$$

thus, it can be written

$$\mathbf{M}_h = \begin{pmatrix} \frac{\rho AL}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\rho AL}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\rho AL}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\rho AL}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.25}$$

The equations of motion are integrated in time using Newmark's family of algo-

rithms:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\dot{\mathbf{x}}_n + \Delta t^2\Big[\Big(\frac{1}{2} - \beta\Big)\ddot{\mathbf{x}}_n + \beta\ddot{\mathbf{x}}_{n+1}\Big] \tag{2.26}$$

$$\dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_n + \Delta t\big[(1-\gamma)\ddot{\mathbf{x}}_n + \gamma\ddot{\mathbf{x}}_{n+1}\big] \tag{2.27}$$

$$\mathbf{M}\ddot{\mathbf{x}}_{n+1} + \mathbf{F}^{int}(\mathbf{x}_{n+1}) = \mathbf{F}^{ext}(t_{n+1}) \tag{2.28}$$

where β and γ are the Newmark algorithm parameters.

For $\beta \neq 0$, a conventional implicit predictor-corrector algorithm [24] is adopted to solve the system of equations (2.26). The predictor is computed as follows:

$$\widehat{\mathbf{x}}_{n+1} = \mathbf{x}_n + \Delta t\dot{\mathbf{x}}_n + (\frac{1}{2} - \beta)\Delta t^2\ddot{\mathbf{x}}_n \tag{2.29}$$

$$\widehat{\dot{\mathbf{x}}}_{n+1} = \dot{\mathbf{x}}_n + (1-\gamma)\Delta t\ddot{\mathbf{x}}_n \tag{2.30}$$

The equation for the nodal accelerations (2.26) can be rewritten as the following nonlinear equation:

$$\mathbf{M}\frac{\mathbf{U}}{\beta\Delta t^2} + \mathbf{F}^{\mathbf{int}}(\widehat{\mathbf{x}}_{n+1} + \mathbf{U}) = \mathbf{F}^{\mathbf{ext}}{}_{n+1} \tag{2.31}$$

where $\mathbf{U} = \beta\Delta t^2\ddot{\mathbf{x}}_{n+1}$. A consistent linearization of this nonlinear algebraic equation about the predictor configuration leads to the computation of the tangent stiffness matrix:

$$\mathbf{K} = \frac{\partial \mathbf{F}_{\mathbf{int}}}{\partial \mathbf{x}}\bigg|_{\widehat{\mathbf{x}}_{\mathbf{n+1}}} \tag{2.32}$$

which enables a quadratic convergence of the Newton-Raphson algorithm used to obtain dynamic equilibrium at $t = t_{n+1}$. In the corrector step, updated values of the nodal fields are computed satisfying dynamic equilibrium at t_{n+1}:

$$\ddot{\mathbf{x}}_{n+1} = \frac{\mathbf{U}}{\beta\Delta t^2} \tag{2.33}$$

$$\dot{\mathbf{x}}_{n+1} = \widehat{\dot{\mathbf{x}}}_{n+1} + \frac{\gamma}{\beta\Delta t}\mathbf{U} \tag{2.34}$$

$$\mathbf{x}_{n+1} = \widehat{\mathbf{x}}_{n+1} + \mathbf{U} \tag{2.35}$$

It bears emphasis that for $\beta = 0$ the time integration can be done fully explicitly and does not require a non-linear equation to be solved. In this case, the integration is straightforward but the rotational inertia can no longer be neglected in the lumping of the matrix, since the mass matrix is required to be invertible. Given the focus of this work on long-term time integration problems, only the implicit Newmark algorithm has been used for it allows computations to be run over larger time steps.

2.2 A numerical model of compressible flow

2.2.1 Analytical equations modeling a compressible flow

In this section we summarize the formulation of the fluid solver which is based on references [42, 41]. Since we model the flow as compressible and inviscid, the governing equations correspond to the Euler equations of compressible flow. In the three-dimensional case, the numerical approximation of these equations is based on the following strong conservative form:

$$\frac{\partial U}{\partial t} + \frac{\partial \mathcal{F}(U)}{\partial x} + \frac{\partial \mathcal{G}(U)}{\partial y}) = 0 \tag{2.36}$$

where:

$$\begin{aligned} U &= \{\rho, \rho u, \rho v, \rho E\}^T \\ \mathcal{F}(U) &= \{\rho u, \rho u^2 + p, \rho uv, \rho(E + \frac{p}{\rho})u\}^T \\ \mathcal{G}(U) &= \{\rho v, \rho uv, \rho v^2 + p, \rho(E + \frac{p}{\rho})v\}^T \end{aligned}$$

In these expressions, ρ is the density, u and v are the Cartesian components of the velocity vector, p is the pressure, E is the specific total energy, U is the vector of conservative variables, $\mathcal{F}(U)$ and $\mathcal{G}(U)$ are the components of the flux vector. Euler's system of equation is closed with a perfect gas equation of state, which can be written in the following way:

$$p = (\gamma - 1)\rho e \tag{2.37}$$

where γ is the specific heat ratio and e is the specific internal energy with $E = e + \frac{1}{2}\|\mathbf{u}\|^2$

2.2.2 Finite volume formulation for the fluid dynamics

The continuum problem is discretized in space using a well established finite volume formulation. This numerical approach is based on discretizing the integral formulation of the conservation equations in the physical domain. It can also be seen as a finite difference discretization of the conservation laws written in the conservative form (2.36).

In Figure 2-2 the shaded region corresponds to the control volume for a two-dimensional problem on a cartesian grid. The spatially discretized scheme simply means that the variation of U over the control volume is equal to the inward flux.

$$\frac{\partial U_{i,j}}{\partial t} = \frac{\mathcal{F}_{i-\frac{1}{2},j} - \mathcal{F}_{i+\frac{1}{2},j}}{h} + \frac{\mathcal{G}_{i,j-\frac{1}{2}} - \mathcal{G}_{i,j+\frac{1}{2}}}{h} \tag{2.38}$$

Here $U_{i,j}$ more closely represents an average value of U over the $(i,j)^{th}$ control volume than the value of U at the point $M_{i,j}$ itself. This formulation has the advantage of being numerically conservative. More details may be found in standard textbooks [21, 26].

The fluxes at the cell interfaces are calculated using an upwind scheme, which is better suited to describe a system whose dynamics is governed by hyperbolic equations such as Euler's equations in the supersonic regime. This class of scheme takes into account the physical propagation of perturbations along the characteristics, which represents the physics of the system in a better way than symmetrical schemes (e.g. Lax-Wendroff schemes). A particularly efficient way of computing the fluxes is to solve a local Riemann problem at each cell interface, which provides an effective means of satisfying the appropriate jump conditions in the presence of discontinuities. Godunov first introduced this idea in 1959. More recent approaches by Osher and by Roe propose to solve the Riemann problem approximately and may also be formulated in terms of a general equation of state. These discretization schemes are first order in

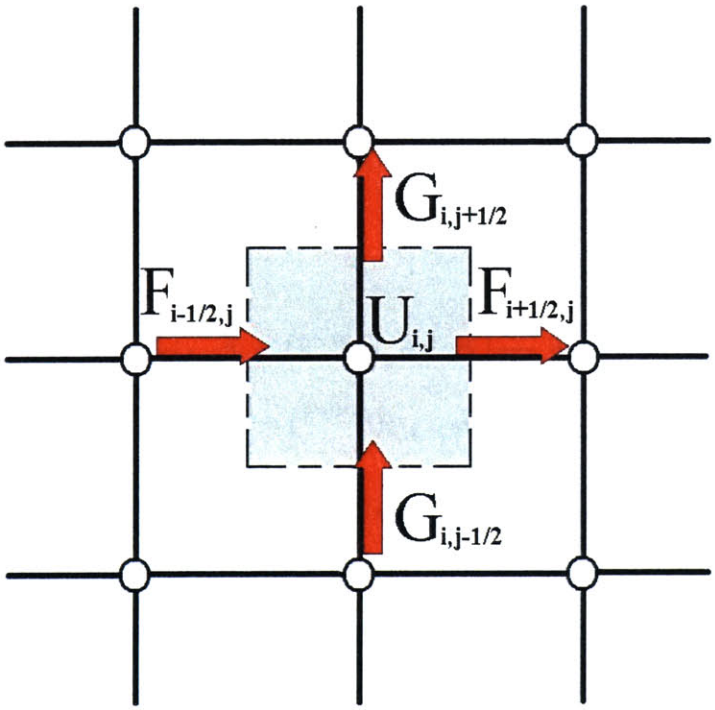


Figure 2-2: Finite Volume Method

space and may be taken as a starting point for the formulation of higher order schemes. In our case, second order accuracy is achieved via linear reconstruction with Van Leer type slope limiting, applied to projections in characteristic state space [30, 22, 26]. This method is often referred as the MUSCL approach (Monotone Upstream-centered Schemes for Conservation Laws).

The equations are then discretized in time and integrated explicitly using a second-order Runge-Kutta algorithm. This algorithm is briefly summerarized hereafter for a one-dimensional problem. First step:

$$U_{i,j}^{n+\frac{1}{2}} = U_{i,j}^{n} + \frac{\Delta t}{2h}\Big[\mathcal{F}_{i-1/2,j}(U^n) - \mathcal{F}_{i+1/2,j}(U^n) + \mathcal{G}_{i,j-1/2}(U^n) - \mathcal{G}_{i,j+1/2}(U^n)\Big] \tag{2.39}$$

Second step:

$$U_{i,j}^{n+1} = U_{i,j}^{n} + \frac{\Delta t}{h}\Big[\mathcal{F}_{i-1/2,j}(U^{n+\frac{1}{2}}) - \mathcal{F}_{i+1/2,j}(U^{n+\frac{1}{2}}) + \mathcal{G}_{i,j-1/2}(U^{n+\frac{1}{2}}) - \mathcal{G}_{i,j+1/2}(U^{n+\frac{1}{2}})\Big] \tag{2.40}$$

This fluid formulation was developed by Dr. R. Samtaney from the Plasma Physics Laboratory at Princeton University to model compressible flows and has been successfully used to study Richtmyer-Meshkov instabilities and the resulting compressible turbulent mixing. A two-dimensional version of this fluid solver, RM3D, provided by Dr. R. Samtaney, has been used and modified as part of the work of this thesis to include the support of the fluid-structure coupling algorithm described in the next chapter. More details of this formulation and its parallel implementation including adaptive mesh refinement capability may be found in Ref. [1].

Chapter 3

Eulerian-Lagrangian fluid-solid coupling algorithm

In the previous chapter the formulation of the dynamics of the flow and the structure as well as their corresponding numerical models were presented. In this chapter, the Eulerian-Lagrangian coupling algorithm that links the two dynamics together at the fluid-solid interface is described. The basic coupling strategy follows the Ghost-fluid Eulerian Lagrangian (GEL) method of references [33, 14, 12]. In this work, the GEL method is extended to the case of open and thin shells.

3.1 Ghost-fluid Eulerian Lagrangian method

In this section we review the basic ideas behind the GEL algorithm presented in the previously cited references.

3.1.1 Implicit representation of the fluid solid boundary on the fluid grid

Let Ω be the space occupied by the cartesian grid, Ω_F the domain occupied by the subset of Ω actually occupied by fluid, Ω_S the domain occupied by the solid and $\delta\Omega_{FS}$ the fluid-solid interface. Note that $\Omega_F \subseteq \Omega$ and $\Omega_S \subseteq \Omega$ and that for an open thin

shell $\Omega_S = \delta\Omega_{FS}$ since the solid has topologically no thickness in this case.

The geometry of the fluid-solid interface $\delta\Omega_{FS}$ is described implicitly by the zeroth level set of the signed distance function φ. The function φ is discretized on the cartesian grid and computed over the entire domain Ω as the minimum distance from one point of Ω to the interface $\delta\Omega_{FS}$

$$\varphi_{i,j} = \varphi(x_i, y_j) = \min_{P \in \delta\Omega_{FS}} \|\overrightarrow{PM_{i,j}}\| \tag{3.1}$$

where $M_{i,j}$ is the grid point of coordinate (i, j). Since the mesh geometry of the two solvers do not match, the level set function φ provides an efficient means of locating the interface which is simply identified by the following equation:

$$\varphi(x, y) = 0 \tag{3.2}$$

The gradient of phi ($\vec{\nabla}\varphi$) is further computed. $\vec{\nabla}\varphi$ is orthogonal to contours of φ-constant value and is directed towards increasing values of the φ function. Thus it represents the normal to the boundary on $\delta\Omega_{FS}$:

$$\vec{n} = \frac{\vec{\nabla}\varphi}{\|\vec{\nabla}\varphi\|} \tag{3.3}$$

The gradient of φ is used for extrapolation purposes as we will see later in the chapter. The overall complexity of the algorithm used to compute φ is of order of $\mathcal{O}(m \cdot n)$ in the two-dimensional problem addressed here, where m represents the number of points in the discretization of $\delta\Omega_{FS}$ and n the number of point in the cartesian discretization of Ω. This is because for the m elements of the discretized interface, the distance from the rod segment to all of the n grid points needs to be evaluated in order to compute the minimum distance function φ. Mauch [31] has proposed an algorithm to compute the minimum distance to the boundary at each grid point whose computational complexity is reduced to $\mathcal{O}(m + n)$. Thus the computation of φ does not constitute a computational bottleneck. A three dimensional extension of the algorithm presented in this work would clearly necessitate the advantageous use

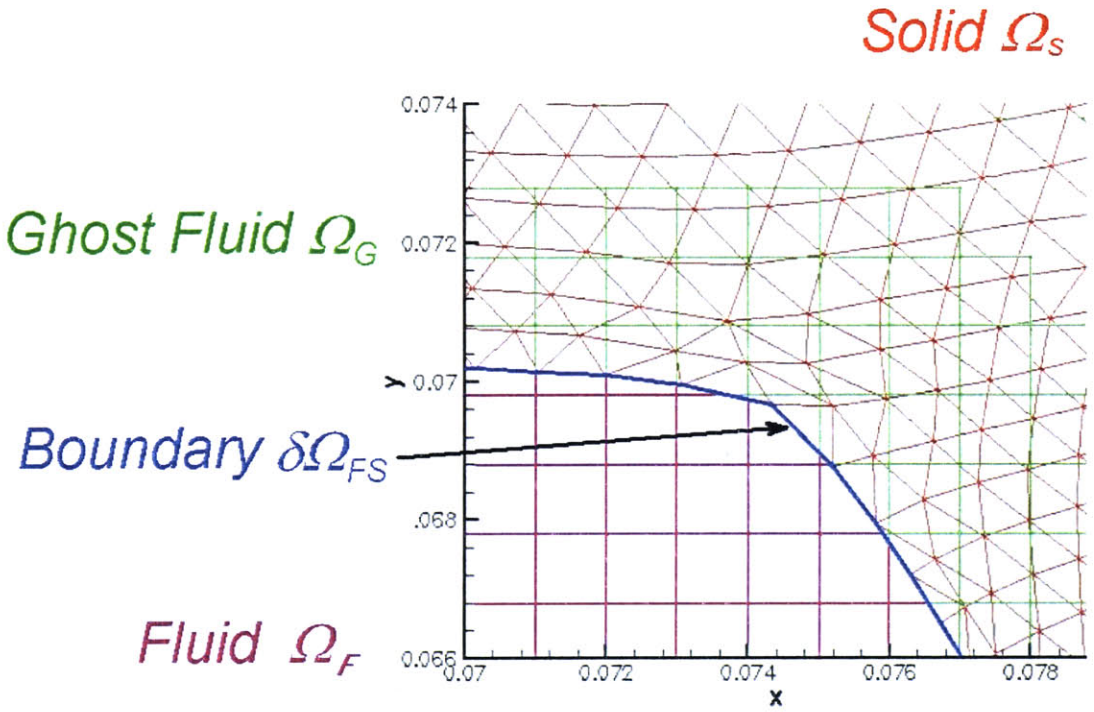


Figure 3-1: Eulerian-Lagrangian representation of the physical domain

of Mauch's algorithm, as it has extensively been done in [33, 14, 12].

3.1.2 Imposition of boundary conditions at the fluid-solid interface

The application and exchange of boundary conditions between the fluid and the solid dynamics is crucial to the success of the coupling method. Here and subsequently, the interface is modeled as a moving solid wall that is impermeable, non-reactive and adiabatic (no heat flux through the interface).

Imposition of boundary conditions on the flow

The following physical conditions are applied on the inviscid fluid: no mass flux, continuity of the normal velocities, free-slip condition for the tangential velocities and continuity of the normal stress. In the GEL method, boundary conditions are applied by extrapolation and velocity field reconstruction.

Extrapolation step: Primitive variables of the flow (density ρ, velocity components u, v, and pressure p) are extrapolated from the flow domain Ω_F to a thin layer of cell Ω_G, on the other side of the interface $\delta\Omega_{FS}$, see Figure 3-1. This narrow band of cells Ω_G accross the interface is called the *ghost* region. The thickness of this region is defined by a parameter φ_0, which determines the additional number of cell layers needed to store the extrapolated values of the flow field that are necessary to impose the relevant boundary conditions. φ_0 depends on the stencil of the Eulerian numerical scheme. It is chosen such that each computational grid point of Ω_F has a complete stencil and such that the values in the first row of ghost cells are updated. In fact for $\varphi_{i,j} \leq 1 \cdot \Delta x$ the cell could be crossed by $\delta\Omega_{FS}$ over the time step and the ghost value would become a real value. Thus the solution needs to be updated on this first layer of ghost cells. The extrapolation is done by advecting the values of the flow field next to the boundary into the ghost region. This step is carried out by integrating the following advection equation

$$\frac{\partial q}{\partial \tau} + \vec{n} \cdot \vec{\nabla} q = 0 \tag{3.4}$$

where q is the extrapolated quantity and $\vec{n}$ the normal to the interface $\delta\Omega_{FS}$. A first order upwind scheme is used for the spatial discretization of equation 3.4, and the equation is integrated in pseudo time steps until the ghost cells have been populated with the required extrapolated values.

Velocity reconstruction step: These non-physical extrapolated values in Ω_G are then reconstructed in order to apply implicitly the proper conservation laws at the boundary by reflecting type conditions. The final field values in the ghost cells are derived from the previous extrapolation step as follows,

$$\begin{pmatrix} \rho_G \\ \vec{V}_G \\ p_G \end{pmatrix} = \begin{pmatrix} \rho_E \\ \left[(2\vec{V}_W - \vec{V}_E)\cdot\vec{n}\right]\vec{n} + [\vec{V}_E\cdot\vec{t}]\vec{t} \\ p_E \end{pmatrix} \tag{3.5}$$

where the subscript E represents the field variables extrapolated from Ω_G, and $\vec{V}_W$ is the extrapolated value of the interface velocity. This reflective type extrapolation captures $\delta\Omega_{FS}$ as a contact discontinuity, with continuous pressure and normal velocity across the interface. Free slip boundary condition is applied since no constraint is added to the tangential velocity component $\vec{V}_E \cdot \vec{t}$. Finally the normal velocity of the fluid $[\vec{V}_W - \vec{V}_E] \cdot \vec{n}$ is simply reflected with $-[\vec{V}_W - \vec{V}_E] \cdot \vec{n}$ in the reference frame attached to $\delta\Omega_{FS}$ which yields $(2\vec{V}_W - \vec{V}_E) \cdot \vec{n}$ in the global reference frame. Thus the normal velocity of the flow relative to the fluid-solid interface is forced to zero at $\delta\Omega_{FS}$ corresponding to the non-penetration boundary condition at the interface.

Computation of the solution: The solution is then computed without using any specialized numerical operators at the interface in Ω_F and in the first layer of ghost cells in Ω_G.

Boundary condition on the structure

In this model, the load applied on the structure is only due to the pressure of the surrounding flow; no viscous drag is considered. Thus, traction boundary conditions are applied on the thin structure at the interface. The pressure field is linearly inter-

polated from the fluid domain and the corresponding ghost region, to each element of the finite element discretization of the rod corresponding to $\delta\Omega_{FS}$. The pressure is considered to be constant along an element of the rod. The vector of external nodal forces is derived from the expression (2.19) and is variationally consistent. A force normal to the rod element of value $-pl/2$ is applied at each of the two extremities of the element. Also, moments of $-pl^2/12$ and $pl^2/12$ are applied respectively at the two extremities, see Figure 3-2.

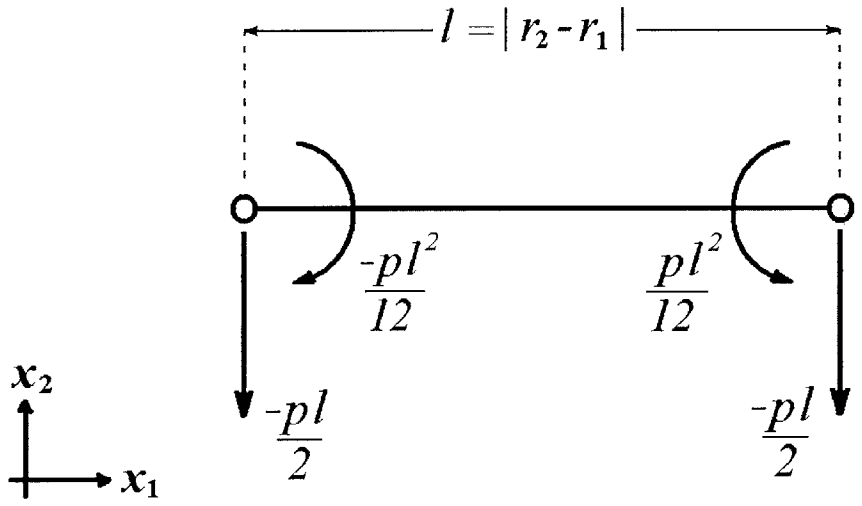


Figure 3-2: Pressure forces of a rod element

3.1.3 Algorithmic steps in fluid-structure interaction simulations using the GEL method

In this section, the sequence of steps involved in a fluid structure simulation using the GEL coupling algorithm is described. The algorithmic steps described below can be taken as guidelines for the implementation of the GEL coupling algorithm.

1. *Increment time step*

2. *Exchange information between the fluid and the solid solver.* The solid solver communicates the position and the velocity of each node of the interface $\delta\Omega_{FS}$

to the fluid solver. The fluid solver communicates the pressure field around the interface to the solid solver.

3. *Set boundary conditions.* The boundary conditions are set at the beginning of the time step. Since the boundary is moving, the level set function φ needs first to be recomputed entirely. Using φ, the pressure field is linearly interpolated from Ω_F to $\delta\Omega_{FS}$ and the ghost region Ω_G is populated with extrapolated and reconstructed values from Ω_F as described previously.

4. *Compute a stable time step.* A stable time step Δt_F is computed for the fluid by simply using the CFL condition, without any more constraints. A stable time step Δt_S is also computed for the structure if an explicit time integration is used for the structure. In general, one typically has $\Delta t_S \ll \Delta t_F$. If an implicit time integration is used, a stable time step does not need to be computed. The final time step Δt is taken equal to: $\Delta t = min(\Delta t_F, \Delta t_S)$.

5. *Integrate in time the equations of the dynamics.* The solid and the fluid dynamics are solved and integrated in time. As described in the previous chapter, the fluid approach is integrated using a second order Runge-Kutta scheme. The structure on the other hand is integrated using implicit Newmark algorithm where the values $\beta = 0.5$ and $\gamma = 1$ are adopted for the Newmark parameters. These values are chosen to allow for some numerical dissipation in order to reach steady state in the simulations of interest in this work. Although it has not been done in this work, this step can be optimized by using parallel implementation. Also, here the two solutions are simply integrated separately and independently. For partitioned FSI solvers, more sophisticated time integration strategies could be investigated to reach better accuracy. Arienti et al. [2] have briefly explored this possibility.

6. *Repeat until time period is reached*

3.1.4 Particular details of previous GEL algorithm

The GEL method has been successfully used to solve a variety of fully coupled FSI problems. Meiron et al. [33] proposed this method to simulate the dynamic response of materials to detonation-wave loading; Cirak and Radovitzky used the GEL method to model the inflation of an airbag [11].

For both of these examples, the implementation of the GEL algorithm relies on the fact that a well defined exterior fluid domain can be defined, either because the flow is considered to be relevant only on one side of the interface or because the solid has a non negligible thickness, see fig.(3-3). In these cases we have $\Omega_F \bigcap \Omega_G = \emptyset$, which means that the real flow region Ω_F and the ghost region Ω_G do not overlap. Thus,

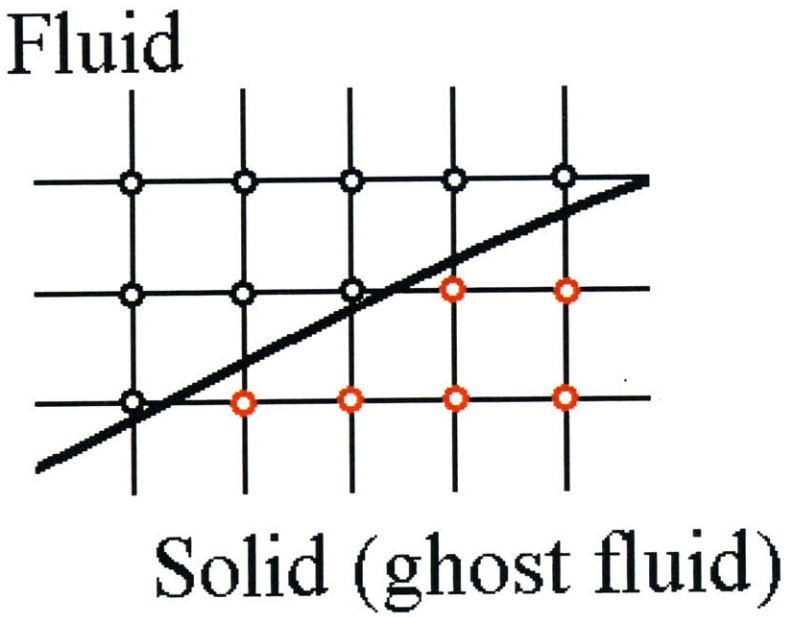


Figure 3-3: Interior and Exterior of the flow domain

the exterior and interior of the evolving flow domain can by identified by assigning a

sign to the level set function φ such that:

$$\varphi(x) \begin{cases} < 0, & \text{if x} \in \Omega_F; \\ = 0, & \text{if x} \in \delta\Omega_{FS}; \\ > 0, & \text{if x} \in \Omega_G. \end{cases} \tag{3.6}$$

With this sign convention, the normal $\vec{n}$ to the interface $\delta\Omega_{FS}$ (3.3) is oriented from the Eulerian fluid domain to the Lagrangian solid.

The extrapolation step is in this case straightforward. Since the ghost region does not overlap with the physical fluid domain, the ghost cells are populated and the extrapolated values are kept in the same array as the real values. Results of fully coupled simulations can be found in [12, 11]. The GEL algorithm gives particularly good results when simulating interactions between the flow and a highly flexible structure undergoing very large displacements.

3.2 Extension of the GEL fluid-solid coupling algorithm to open thin structures

3.2.1 Limitations of the existing GEL formulation

The scope of application of the algorithm proposed by Cirak and Radovitzky [12] excludes fluid-structure interaction problems in which the details of the flow matter on both sides of an immersed open boundary. As originally formulated, it cannot apply two different boundary conditions on different sides of the interface, as it is required for applications involving either open shells or fluid flow on the interior and exterior of a closed shell. It can model the inflation of an airbag since fluid is only considered to be inside the airbag, but it cannot properly model the inflation of an open immersed boundary such as a parachute, for instance.

The main limitation of the previous formulation is that it is based on the explicit assumption that the flow is relevant on only one side of the fluid-solid interface $\delta\Omega_{FS}$. Recall that, in order to apply boundary conditions, values from the interior physical

fluid domain Ω_F are extrapolated to a narrow band of cells Ω_G in the exterior domain called the ghost cells and stored in the same array as the real values of the flow. This was possible because in the algorithm as conceived originally, these exterior cells did not play any role beyond being placeholders for the extrapolated fields responsible of enforcing the boundary conditions. This implementation convenience introduces an asymmetry in the treatment of the two sides of the interface.

In the limiting case of an infinitely thin open boundary, $\delta\Omega_{FS} = \Omega_S$ and the fluid covers the entire domain, which implies that there is no exterior to the fluid domain. Thus:

$$\Omega_F \bigcap \Omega_G \neq \emptyset \tag{3.7}$$

Our objective in this work is to extend the use of the GEL approach to the case of open shell structures immersed in a fluid and/or to the case of a closed shell containing a fluid in its interior, which at the same time is immersed in fluid. For simplicity, we restrict our attention to the two-dimensional case. However, special importance is given to guaranteeing that the algorithmic modifications proposed are extensible to three dimensions in a straightforward manner.

3.2.2 Modifications to the GEL Algorithm

In the proposed extension of the GEL method, the basic steps of the coupling algorithm used by Cirak *et al* are retained. Boundary conditions on the flow are still applied by extrapolation *via* level sets and by velocity reconstruction, and traction boundary condition on the structure are still applied by interpolating the pressure at the boundary.

The main modification of the extrapolation step addresses the basic observation that $\Omega_F \bigcap \Omega_G \neq \emptyset$, which implies that around the boundary a grid point of the computational domain lies simultaneously in the real flow region and in the ghost fluid region. In fact, Ω_G corresponds now to the whole region around the interface $\delta\Omega_{FS}$, since each one of the two sides of $\delta\Omega_{FS}$ needs its corresponding ghost region on the other side of the boundary.

From this simple observation, one can see that a more involved representation and implementation to compute and store the values of the flow variables in the real flow and the ghost region at the same time is required. In this work the use of auxiliary data arrays, called the *ghost* arrays has been required in order to support the coexistence of ghost and real flow variables at grid points near the fluid solid interface $\delta\Omega_{FS}$. These *ghost* arrays are used to store the extrapolated and reconstructed ghost values, see fig.(3-4). Thus for each of the six variables of the conservative formulation of the Euler equations, two superposed matrices are used: the first one stores the real values of the conservative variables on both sides of the free boundary and at each point of the fluid grid Ω_F, the second one stores the values of the extrapolated conservative variables in Ω_G. Using this dual data structure, values from each side of $\delta\Omega_{FS}$ are symmetrically extrapolated to the corresponding other side. The fields are then reconstructed in the entire ghost region using the GEL method.

The extremities of the immersed interface require a more careful treatment. For open shells, the ghost region surrounds the interface. In the center region of Ω_G, *ghost* arrays store non-physical *ghost* values, which are used to apply boundary conditions implicitly. At the extremities of $\delta\Omega_{FS}$, in the small part of Ω_G beyond the extremities of $\delta\Omega_{FS}$, the fluid flows from both sides of the interface mix and the sides of the boundary are no longer defined. This region needs special attention. This is done by filling the *ghost* arrays with the *real* values of the flow. Thus in this small region, the same values are stored in the *real* and the *ghost* arrays. This enables to translate into the numerical model the fact that at the extremities of $\delta\Omega_{FS}$ the flows on each side of the interface no longer remain separated by the structure and merge.

During the computation of the solution the conservative variables need to be transformed into the primitive variables, for which *ghost* arrays also have to be introduced. In general, every time a quantity is needed simultaneously in the real and the ghost region, *ghost* arrays have been used in the implementation for simplicity. It is important to note that the generalization of the GEL method to thin and open shells has a cost: the memory space required for the computation is multiplied by two. In principle, the number of ghost cells scales with $\mathcal{O}(\frac{l}{\Delta x})$ where l represents the length

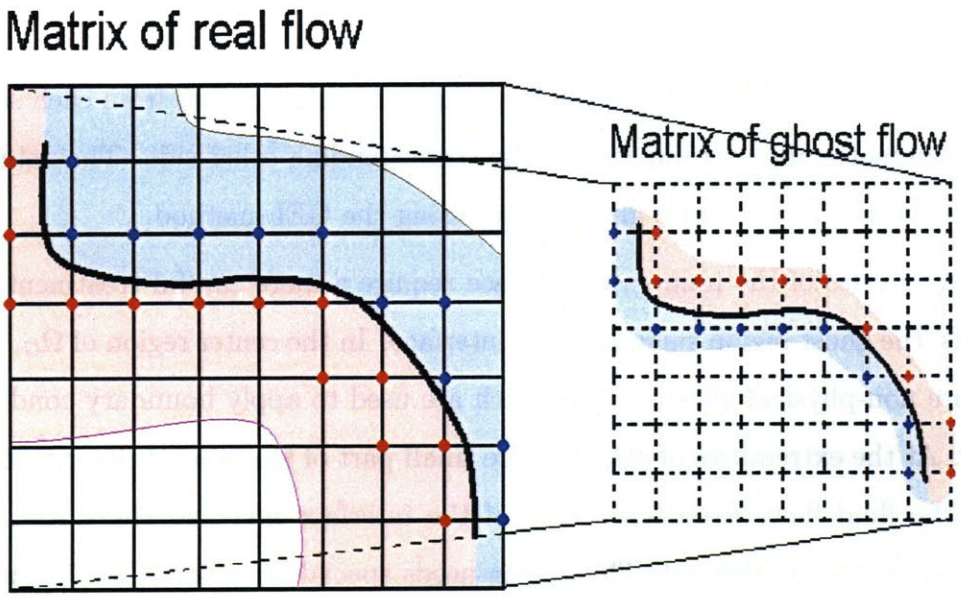


Figure 3-4: Real and Ghost arrays

of $\delta\Omega_{FS}$ and Δx the characteristic size of the structured grid. Thus, by indexing the ghost cells and by using sparse arrays and other data structures, it should be possible to reduce the extra memory needed. This has not been done in this work.

Once the ghost region has been populated with the appropriate reconstructed flow variables, the solution is computed across the entire fluid domain. However, in the extended algorithm grid points that are located close to the interface $\delta\Omega_{FS}$ need a special treatment. The use of dual data structures implies that the consistent real and corresponding ghost values are used to properly compute the fluxes coming from the interface. In practice, this simply means that when computing a flux at a grid point A, if a grid point B of the numerical stencil lies on the other side of $\delta\Omega_{FS}$ then values from the *ghost* arrays have to be used at that point. This condition is simple and does not complicate the implementation significantly, but it relies on an efficient means of knowing whether or not two points close to the boundary lie on the same side of $\delta\Omega_{FS}$.

Although the zero level set of the unsigned φ-function computed for open boundaries still implicitly determines the location of the interface in the fluid grid, it no longer enables the determination of the side of the interface on which a given fluid cell lies. Using the experience from the previous approaches, a local sign is assigned to the level set function φ on each side of the interface $\delta\Omega_{FS}$. In this case the sign no longer indicates whether a cell is in the solid or in the fluid domain but on which side of the boundary it is. This sign is only relevant on those cells around the interface, for which values from both the *real* and the *ghost* arrays may be needed to compute the solution. The sign of φ is thus meaningful for $|\varphi_{i,j}| \leq \varphi_r$, where φ_r is a parameter depending on the numerical stencil, and has no meaning elsewhere. For the grid point $A(x_A)$ where the flux is computed, a simple criterion for $B(x_B)$ to be on the other side of the interface is:

$$|\varphi(x_A)| \leq \varphi_r \text{ and } \varphi(x_A) \cdot \varphi(x_B) \leq 0 \tag{3.8}$$

The first condition verifies that A is sufficiently close to the interface for the stencil

to have possible grid points on the other side, the second that A and B are on two different sides. Two methods have been considered in order to determine the side of the interface and assign a sign to the level set function φ.

The first method that has been implemented is a travel algorithm. As a reference point $M_{i,j}$ travels on the structured grid along the interface, the local sign is assigned to the cells next to the boundary. Starting at a grid cell in the immediate neighborhood of one extremity of the boundary, $M_{i,j}$ travels along the cartesian mesh and goes from one cell to the other by turning clock wise for example. When $M_{i,j}$ crosses the interface it then changes its direction of travel and starts to move counter clock wise. Following this path the travelling point $M_{i,j}$ visits all the points in the interface's neighborhood as shown on Figure 3-5. At the same time a consistent sign is applied to the cells around $\delta\Omega_{FS}$.

This is a very efficient means of assigning the sign to φ because it only requires to visit those cells around the interface, the number of which is of order of $\mathcal{O}(\frac{l}{\Delta x})$ as has been previously noted. A potential limitation of this approach is that its extension to the three-dimensional case may not be straightforward. Therefore, another method based on the orientation has been considered.

A key observation is that, although the boundaries under consideration no longer required to be closed, they are still orientable manifolds, i.e., the boundaries have two unequivocally identifiable sides which can be conveniently assigned to the adjacent fluid domain. In the two-dimensional case, the interface is oriented by parameterizing the curve. Once an orientation is chosen for the two-dimensional space, a tangential direction $\vec{t}$ and a normal direction $\vec{n}$ to the curve can be locally defined Figure 3-6, thus determining two sides on the curve.

In our implementation of this idea, the sign is determined when φ is computed. As we have seen in section 3.2.2, in order to compute φ, the distances between the N points of the cartesian grid and the M elements of the discretized interface need to be calculated. When the closest element, i.e., the one defining the actual distance of the grid point to the solid interface is found, its tangent $\vec{t}_m$ and normal $\vec{n}_m$ vectors endowed by the boundary parameterization are defined. The distance from a grid

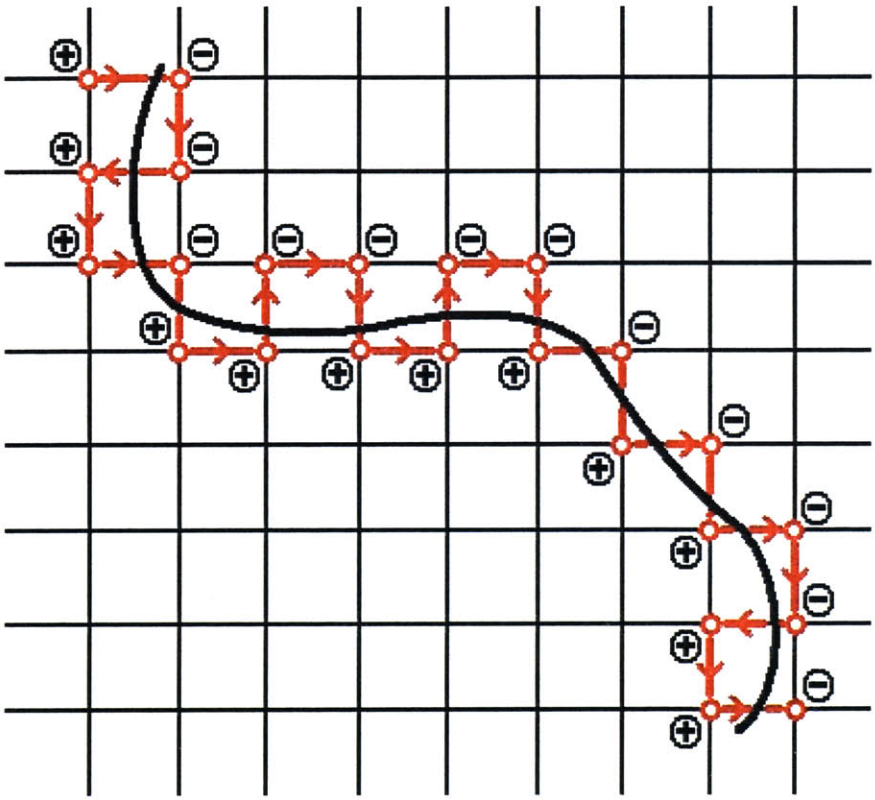


Figure 3-5: The travel algorithm

point $M_{i,j}$ of the cartesian grid to the element m is given a positive sign if the $(i,j)^{th}$ cell lies in the half plane defined by $\vec{n}_m$ and with a negative sign if it lies in the other half plane Figure 3-6.

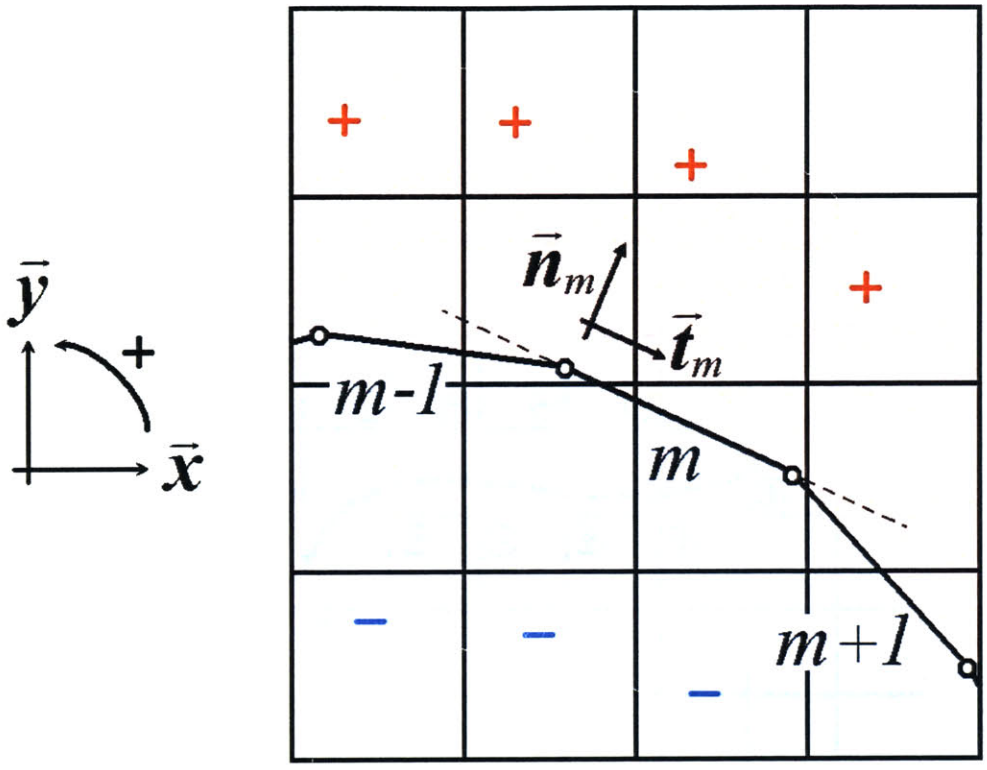


Figure 3-6: Pseudo sign defined by the orientation of the interface

This method enables the computation of a *pseudo* signed level set function and is extensible to the three-dimensional case in a straightforward manner. Therefore, this is the method used in the final implementation of the extended GEL method.

3.2.3 Algorithmic steps in the extended GEL algorithm

In the extended algorithm, the same steps described in Section 3.1.3 are followed. We briefly summarize those steps and address the main differences. For the fluid dynamics, boundary conditions are imposed by extrapolating the flow variables symmetrically using an auxiliary data structure to hold ghost values and by reconstructing

the extrapolated values in order to enforce the non-penetration boundary condition on both sides of the interface. A stable time step is computed using the CFL condition to ensure stability. The solution is then computed in the entire physical domain with the regular numerical scheme. Far enough from the boundary, the cell fluxes are computed using real values of the flow, while close to the boundary, they are computed using the real and relevant ghost values. The pseudo sign of the distance function enables to indicate whether or not two points are on the same side of the boundary by simply looking at the sign of the distance function. The cell fluxes coming from the boundary are thus computed using the real values and the required corresponding ghost values on the other side of the boundary.

For the structure dynamics, pressure fields are interpolated from each flow region to the corresponding side of the interface $\delta\Omega_{FS}$. The appropriate traction boundary conditions are thus applied on the two sides of $\delta\Omega_{FS}$, using the formulation described in section 2.1.

A final aspect of the coupling algorithm that requires special consideration is those grid points that are crossed during the computation by the deforming solid boundary. A careless treatment of the boundary conditions at these points would lead to a coupling algorithm in which the flow field on either side of the boundary would be polluted by the flow field on the other side. During the computation, the grid points crossed by the interface no longer remain on the same side of the interface and, therefore, require a special treatment. For coupled systems involving compressible flows, the values of the flow variables are discontinuous across the boundary and change drastically from one side of $\delta\Omega_{FS}$ to the other. Typically, one side is a high pressure region and the other one is a low pressure region. In the GEL algorithm these two regions should only influence each other through the motion of the interface and the values of the actual flow field on one side should not affect the flow on the other side. In fig.(3-7) the crossed points are colored in green. The flow variables from the region below the interface colored in pink in fig.(3-7) have to be extrapolated to the green points. To this end, the values of the real flow variables are simply replaced by the extrapolated ghost values at the same point immediately after the regular GEL

extrapolation step, without increasing the overall complexity of the implementation. It bears emphasis that in cut cell approaches this aspect of moving boundaries leads

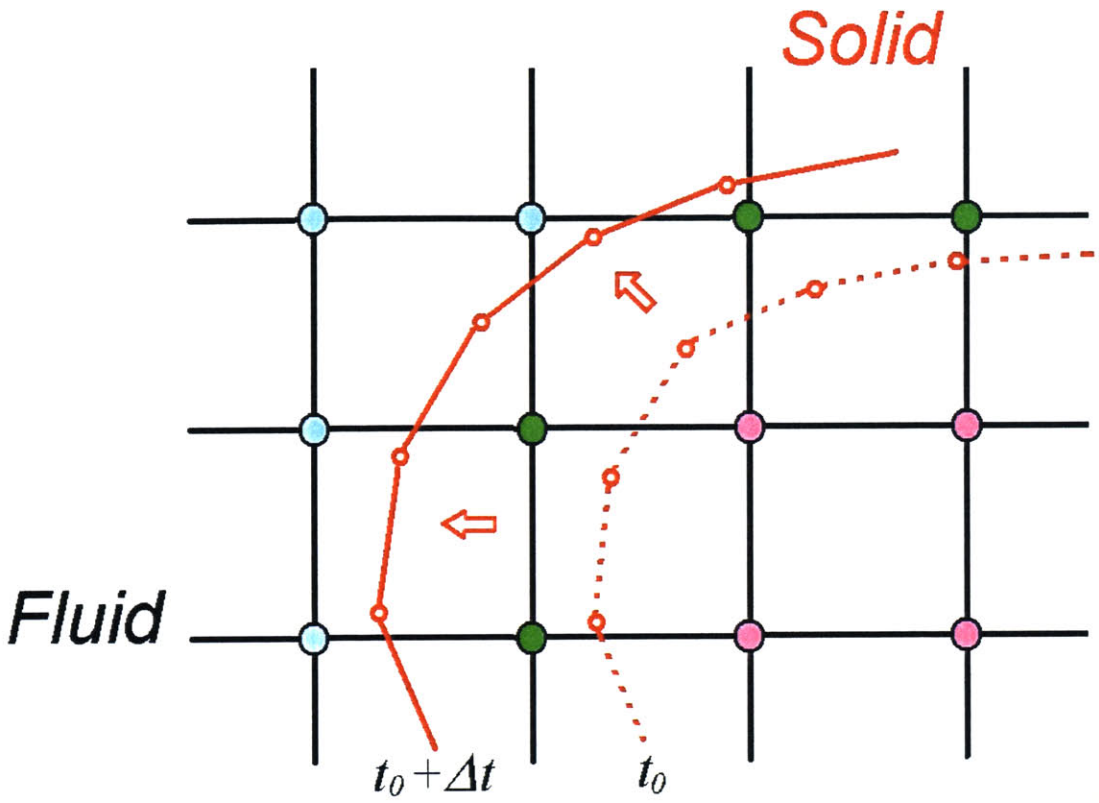


Figure 3-7: Free motion of the interface in the fluid domain

to an additional complexity in the formulation and the implementation in order to satisfy the conservative aspect of the scheme.

The pseudo sign of the level sets provides a very convenient way to keep track of these crossed points. One can detect if a grid point was crossed by the interface by simply looking for changes in time of the sign of the level set function. In fact, once an orientation of the interface is chosen as described in section 3.2.2, the sign of the distance function on each side of the boundary does not change during the computation. This means that the local sign around the interface is consistent between two time steps. Thus if at a grid point $M_{i,j}$ a change of sign occurs between two time steps in the value of the level set function $\varphi_{i,j}$, this point remains no longer on the

same side of the immersed boundary which means that the interface has crossed this grid point.

The implementation of this additional step does not require many changes in the algorithm. At the beginning of the time step the level set function φ is computed and is given a *pseudo* sign. At each grid point $M_{i,j}$ inside the ghost region Ω_G, the sign of the current level set function $\varphi_{i,j}$ is compared to the sign of the function at the previous time step. At the points where a change of sign occurs the conservative variables stored in the *ghost* arrays are transferred to the real flow arrays. The flow variables are then extrapolated/reconstructed, and the solution is computed using the GEL procedure as described previously.

Chapter 4

Numerical Results

In this chapter we verify the correctness of each individual component of the FSI algorithm and we present numerical results attesting to the versatility of the overall approach. The feasibility of the algorithm proposed to compute the signed level set function is demonstrated in Section 4.1 by solving Laplace's Equation with a Dirichlet boundary condition imposed on a thin immersed open boundary. This effectively amounts to an alternate efficient formulation of an immersed boundary method, see for example [47]. In Section 4.2, simulations are conducted to verify the ability of the GEL algorithm to properly represent supersonic flows past fixed thin open boundaries. In this section, numerical results are compared to exact solutions of supersonic flows past inclined planes. Particular emphasis is given to the correct computation of the forces exerted by the flow on the boundary. Finally, in Section 4.3 we exercise the fully coupled FSI algorithm in the simulation of supersonic flow past an initially-flat highly-deformable rod demonstrating the feasibility and versatility of the overall approach for simulating complex FSI problems.

4.1 An immersed boundary method for Laplace's Equation based on level sets

The purpose of this section is to demonstrate the effective use of level sets as convenient means for applying boundary conditions on boundaries immersed in cartesian grids. For simplicity, the model equation of choice is Laplace's equation.

4.1.1 Model formulation and numerical approach

Laplace's equation is a very well known and used analytical problem. This partial differential equation can be written as

$$\triangle \Psi = 0 \tag{4.1}$$

which takes the following form in a two-dimensional space:

$$\frac{\partial^2}{\partial x^2}\Psi(x,y) + \frac{\partial^2}{\partial y^2}\Psi(x,y) = 0 \tag{4.2}$$

Although this partial differential equation is elliptical while the compressible Euler equations are hyperbolic equations, this problem is still very useful to solve because it is analytically rather simple. Also solutions of this PDE are known for being remarkably smooth and regular.

In our case Laplace's Equation is solved on a domain Ω spatially discretized in a cartesian mesh. Boundary conditions are applied at the boundary $\delta\Omega$ of the domain Ω and on an immersed and open boundary $\delta\Omega_{IB}$. Different methods have been proposed to solve this problem and further generalized to be applied to a larger class of problems [47]. The solution approach proposed here focuses on the use of level sets as vehicles to apply boundary conditions in immersed interfaces, as a prologue to the more complex problems of FSI of interest in this thesis. This step was also instrumental in the development of some of the implementation details necessary for the extended GEL method, including the auxiliary data structure and the *pseudo*

signed level set function φ.

Laplace's Equation is solved using a finite difference scheme. The system is inverted using an over relaxed iterative method. On a regular grid point $M_{i,j}$ a second order centered discretization of equation (4.2) yields:

$$\Psi_{i+1,j} + \Psi_{i,j+1} + \Psi_{i-1,j} + \Psi_{i,j-1} - 4 \cdot \Psi_{i,j} = 0 \tag{4.3}$$

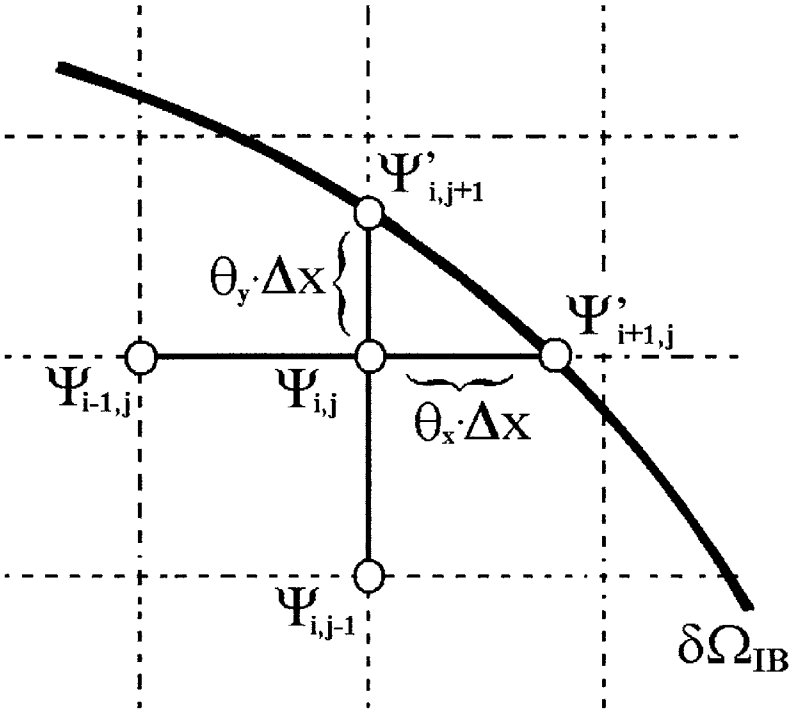


Figure 4-1: Laplace's Equation with an immersed boundary

At the boundary, the Laplacian operator can be expressed using the irregular stencil. The notation used is described on Figure 4-1. Δx is the characteristic size of a grid cell. If $\delta\Omega_{IB}$ intersects the cartesian grid between $M_{i,j}$ and $M_{i+1,j}$, $\theta_x \Delta x$ is the distance between $M_{i,j}$ and the intersection. θ_y is defined the same way. The function Ψ is prescribed on $\delta\Omega_{IB}$ which sets the boundary condition on the immersed

interface. $\Psi'_{i+1,j}$ is the prescribed value of Ψ at the intersection between $\delta\Omega_{IB}$ and $[M_{i,j}, M_{i+1,j}]$ and $\Psi'_{i,j+1}$ the value at the intersections with $[M_{i,j}, M_{i,j+1}]$

The scheme is a first order scheme in this case and can be written:

$$\frac{2 \cdot \Psi'_{i+1,j}}{\theta_x(1+\theta_x)} + \frac{2 \cdot \Psi'_{i,j+1}}{\theta_y(1+\theta_y)} + \frac{2 \cdot \Psi_{i-1,j}}{1+\theta_x} + \frac{2 \cdot \Psi_{i,j-1}}{1+\theta_y} - \frac{2 \cdot (\theta_x+\theta_y) \cdot \Psi_{i,j}}{\theta_x \cdot \theta_y} = 0 \qquad (4.4)$$

More details can be found in textbooks [36, 13]. Equation (4.4) can be rewritten using the regular scheme, see Figure 4-2:

$$\widehat{\Psi}_{i+1,j} + \widehat{\Psi}_{i,j+1} + \Psi_{i-1,j} + \Psi_{i,j-1} - 4 \cdot \Psi_{i,j} = 0 \qquad (4.5)$$

where:

$$\widehat{\Psi}_{i+1,j} = \frac{4 \cdot \theta_x \cdot \theta_y}{(\theta_x+\theta_y) \cdot (1+\theta_x)} \left[\Psi_{i-1,j} + \frac{\Psi'_{i+1,j}}{\theta_x} \right] - \Psi_{i-1,j} \qquad (4.6)$$

$$\widehat{\Psi}_{i,j+1} = \frac{4 \cdot \theta_x \cdot \theta_y}{(\theta_x+\theta_y) \cdot (1+\theta_y)} \left[\Psi_{i,j-1} + \frac{\Psi'_{i,j+1}}{\theta_y} \right] - \Psi_{i,j-1} \qquad (4.7)$$

Even though this step does not constitute an extrapolation step *stricto sensu*, the methodology used to solve Laplace's equation is the same as in the extended GEL method. $\widehat{\Psi}_{i+1,j}$ and $\widehat{\Psi}_{i,j+1}$ are stored in a *ghost* array and the solution is then computed with the same numerical scheme, but using the relevant real and ghost values at the interface. This requires the use of the auxiliary data structure and the *pseudo* signed φ-function.

The solution of the discretized Laplace problem (4.3) yields a system of N equations, where N is the number of grid cells in the computational domain. This system has been solved using Successive Over Relaxation (SOR) method.

4.1.2 Simulation results

Using the methodology described in the previous section, Laplace's equation has been solved on a two-dimensional space discretized in 400×400 cells. On the exterior boundary of the computational domain, the solution is set to zero. Dirichlet bound-

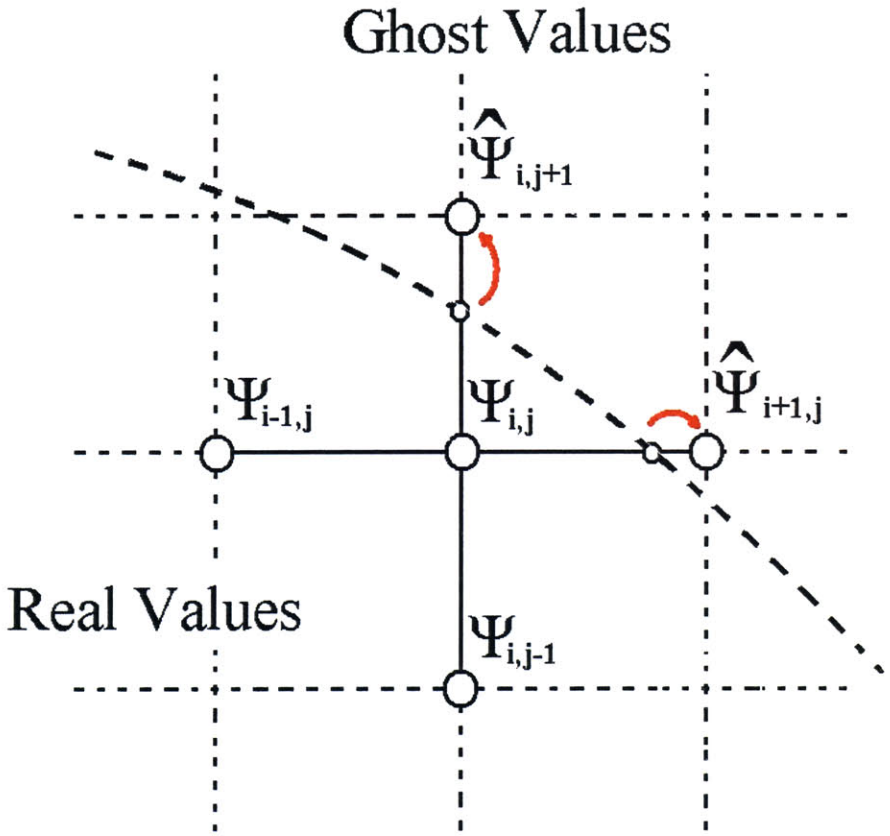


Figure 4-2: Laplace's Equation at the immersed boundary

ary conditions are prescribed on a thin, immersed and open boundary of arbitrary geometry. In these simulations, Figures 4-3 and 4-4, a constant value Ψ_0 is imposed on the thin boundary. It bears emphasis that the solution is computed properly on each side of the boundary. The strict discontinuity of the gradient $\vec{\nabla}\Psi$ of the function Ψ across the boundary shows that there is no pollution of the solution from one side of the interface to the other one.

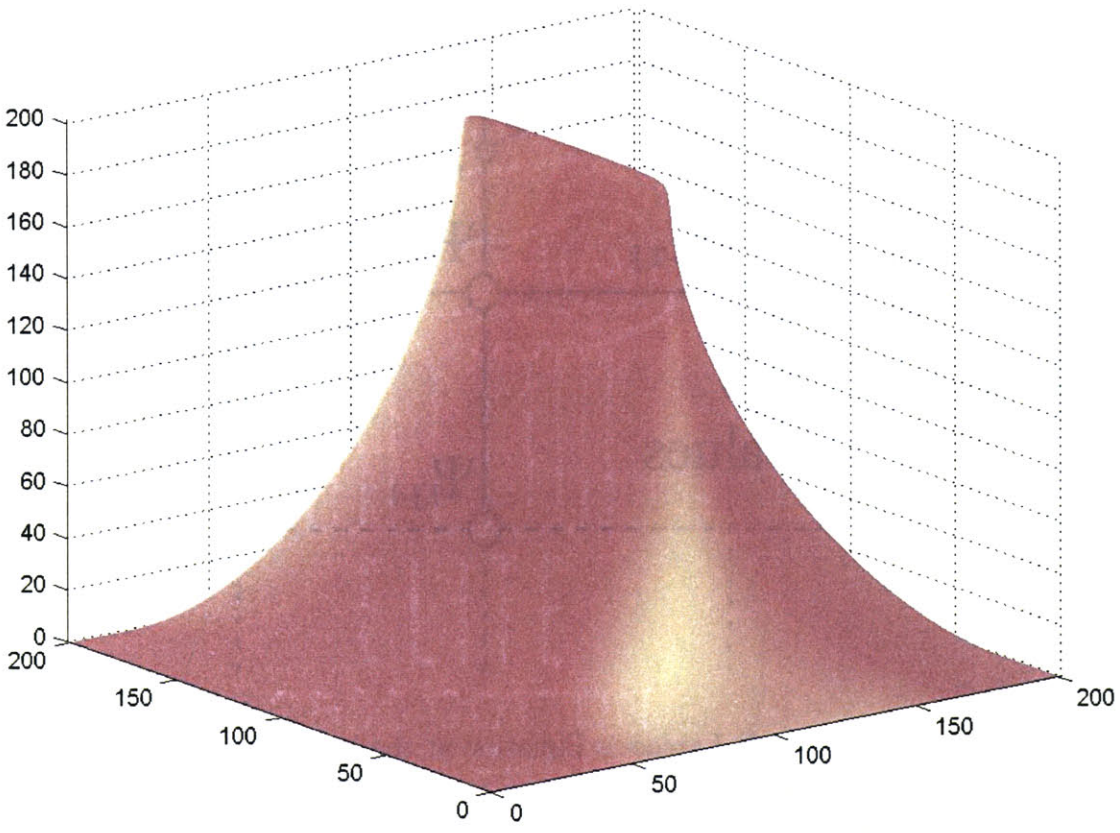


Figure 4-3: Solution of Laplace's Equation with boundary condition applied on a thin immersed boundary (1)

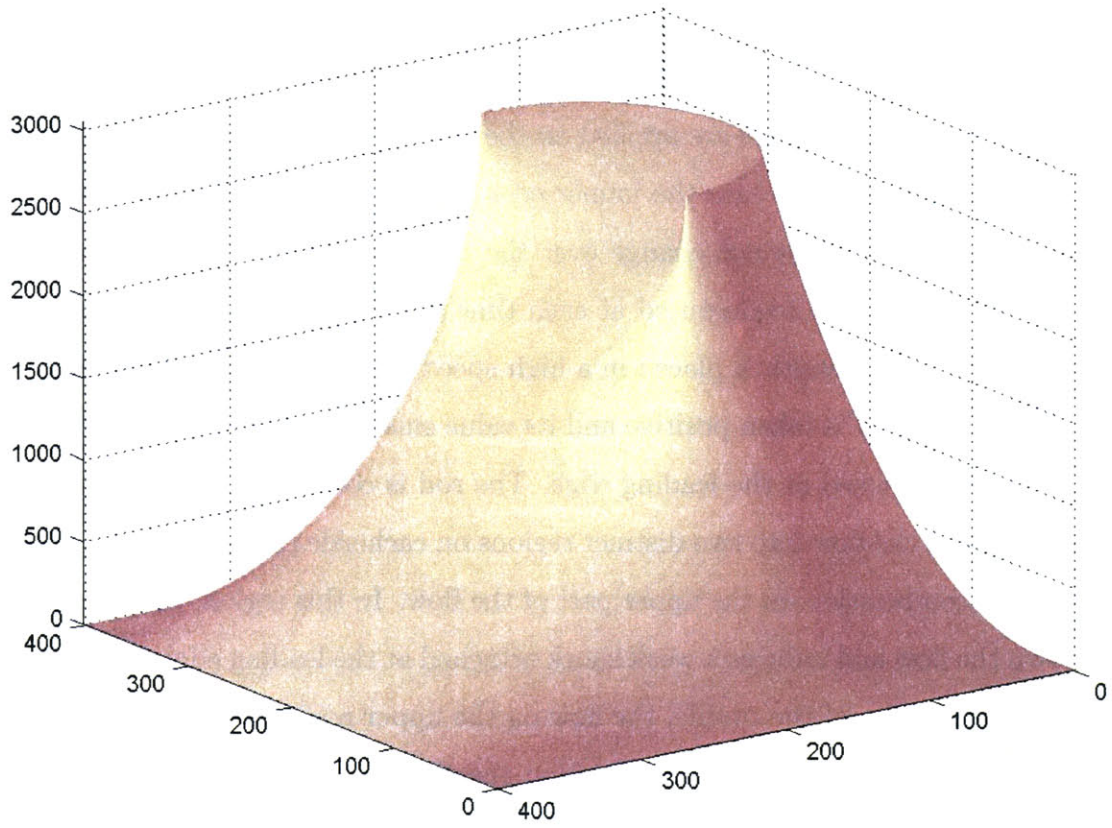


Figure 4-4: Solution of Laplace's Equation with boundary condition applied on a thin immersed boundary (2)

4.2 Verification tests

4.2.1 Simulation of a supersonic flow past a rigid plate at different angles of attack

The Euler compressible fluid solver RM3D has been adapted using the dual data structure and *pseudo* signed φ-function to compute flows around thin, open and immersed profiles. The extended formulation has been used on a test case for which an analytical solution is known. This was done to verify that the appropriate conservation boundary conditions are applied on the two sides of the infinitely thin immersed boundary. In this test case the immersed interface is rigid and fixed. Therefore the location of $\delta\Omega_{FS}$ does not change over the computation and the level set function does not need to be recomputed at each time step. The solid profile corresponds to a straight rod of length L placed in a high speed flow at an angle of attack α, Figure 4-5. The angle α is taken positive and its value small enough for a compression shock to remain attached at the leading edge. The rod is completely immersed in the fluid and divides the flow into two distinct regions on each side of the interface. The profile induces compression on the upper part of the flow. In this region, the solid wall slows down the flow and induces a weak shock attached at the leading edge of the profile. At the trailing edge of the profile, the flow on the upper region is re-accelerated through an expansion wave. In the lower region the flow is accelerated around the leading edge through an expansion wave. It is then slowed down by a weak shock attached at the trailing edge of the profile. The basic features of this flow are sketched in Figure 4-5.

4.2.2 Derivation of the analytical solution

In this section, we summarize the derivation and main results of the analytical solution for the flow around the straight profile described previously. For this geometry, the inviscid supersonic flow has to be calculated around a dihedron. A concave dihedron initiates a shock while a convex dihedron initiates an expansion wave. The solution

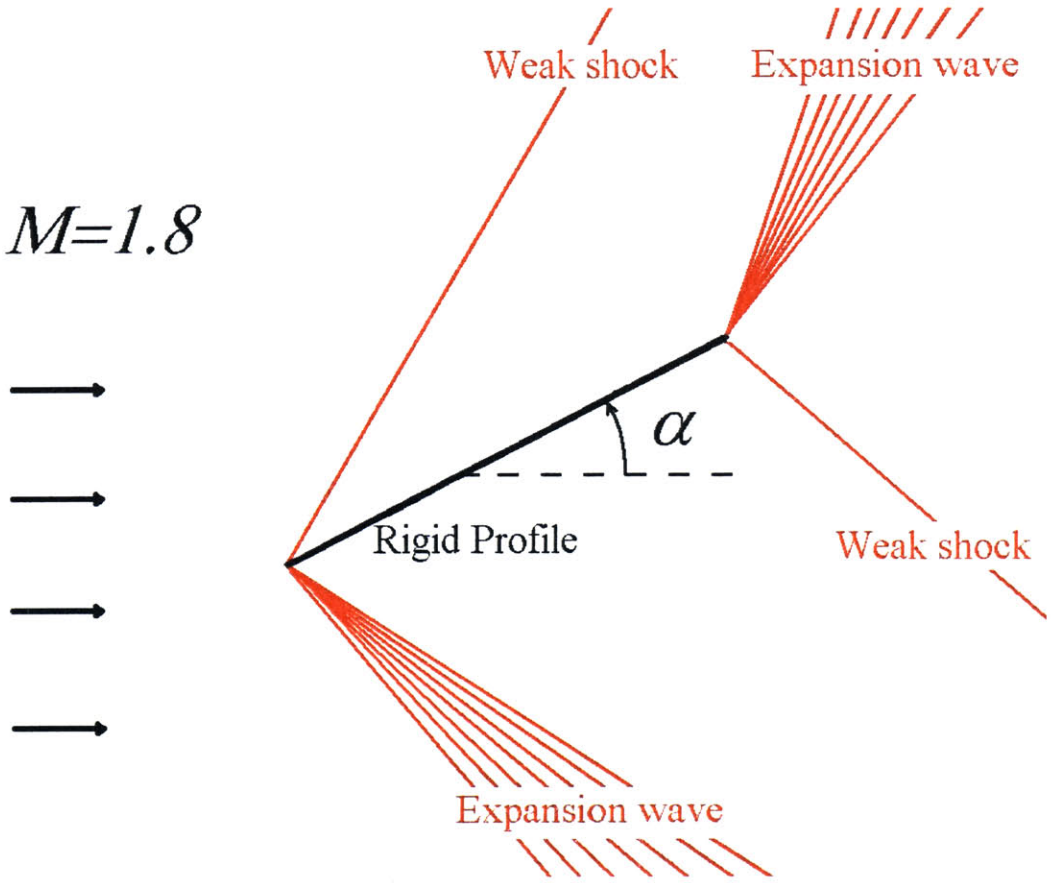


Figure 4-5: Schematic of the simulation run for the static case

for these two cases is derived hereafter.

Weak shock at a concave dihedron

Consider a concave dihedron that induces in the flow a deflection of angle θ, see Figure 4-6. The dihedron initiates a discontinuity in the flow, which can be identified as a shock. This means that the tangential velocity to the discontinuity is continuous through it but that the normal velocity is discontinuous: in Figure 4-6 $V_{t_0} = V_{t_1}$ and $V_{n_0} \neq V_{n_1}$. The direction of the shock makes an angle σ with the upstream velocity $\vec{V}_0$.

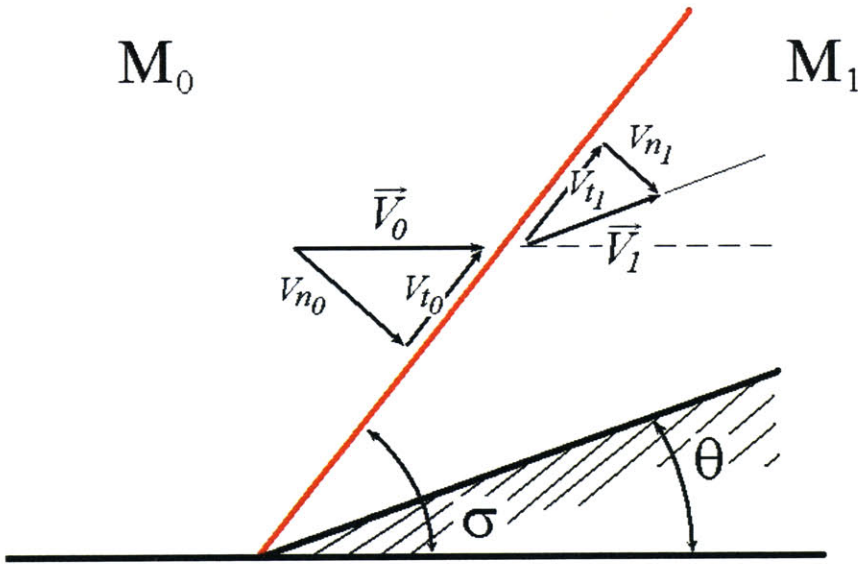


Figure 4-6: Supersonic flow at a concave dihedron

The derivation of mass, momentum and energy conservation laws with the presence of a shock wave yields the Rankine-Hugoniot system of equations, which characterizes the discontinuity. More details on this derivation can be found in standard textbooks [7, 46]. From the Rankine-Hugoniot equations one can deduce the following

non-linear equation:

$$\frac{\tan(\sigma-\theta)}{\tan\sigma} = \frac{2}{\gamma+1}\frac{1}{M_0^2\sin^2\sigma} + \frac{\gamma-1}{\gamma+1} \tag{4.8}$$

which links the deflection angle θ, the shock angle σ and the upstream Mach number M_0. Following this, all the values of the flow variables can be computed behind the shock. For example:

$$\frac{p_1}{p_0} = \frac{2\gamma}{\gamma+1}M_{n_0}^2 - \frac{\gamma-1}{\gamma+1} \tag{4.9}$$

$$M_{n_1}^2 = \frac{1+\frac{\gamma-1}{2}M_{n_0}^2}{\gamma M_{n_0}^2 - \frac{\gamma-1}{2}} \tag{4.10}$$

where $M_{n_0} = M_0 \cdot \sin\sigma$ is the upstream normal Mach number and $M_{n_1} = M_1 \cdot \sin(\sigma-\theta)$ is the normal Mach number behind the shock wave. The analytical solution for a weak shock wave has been computed with Matlab and used for verification of simulation results.

Expansion wave around a convex dihedron

Consider a convex dihedron that induces in the flow a deflection of angle θ, see Figure 4-7. Writing Euler's system of equations in curvilinear coordinates, it can be shown that for a supersonic flow $M > 1$ the PDE system is hyperbolic [7, 46]. In this case the solution can be derived using the method of characteristic curves. The derivation is briefly summarized in the following.

At each point of the flow, two characteristic curves a C^+ and a C^- intersect, see Figure 4-7. The C^+ curve makes an angle μ with the streamline and the C^- makes an angle $-\mu$ with the streamline, where $\mu = \arcsin\frac{1}{M}$. The Riemann invariants λ^+ and λ^- defined as:

$$\lambda^+ = \omega - \theta \tag{4.11}$$

$$\lambda^- = \omega + \theta \tag{4.12}$$

remain constant respectively along the characteristic curves C^+ and C^-. In this expression $\omega(M)$ is the Prandtl-Meyer function defined as:

$$\omega(M) = \mu + \sqrt{\frac{\gamma+1}{\gamma-1}} \arctan\left[\sqrt{\frac{\gamma-1}{\gamma+1}} \cot\mu\right] - \frac{\pi}{2} \tag{4.13}$$

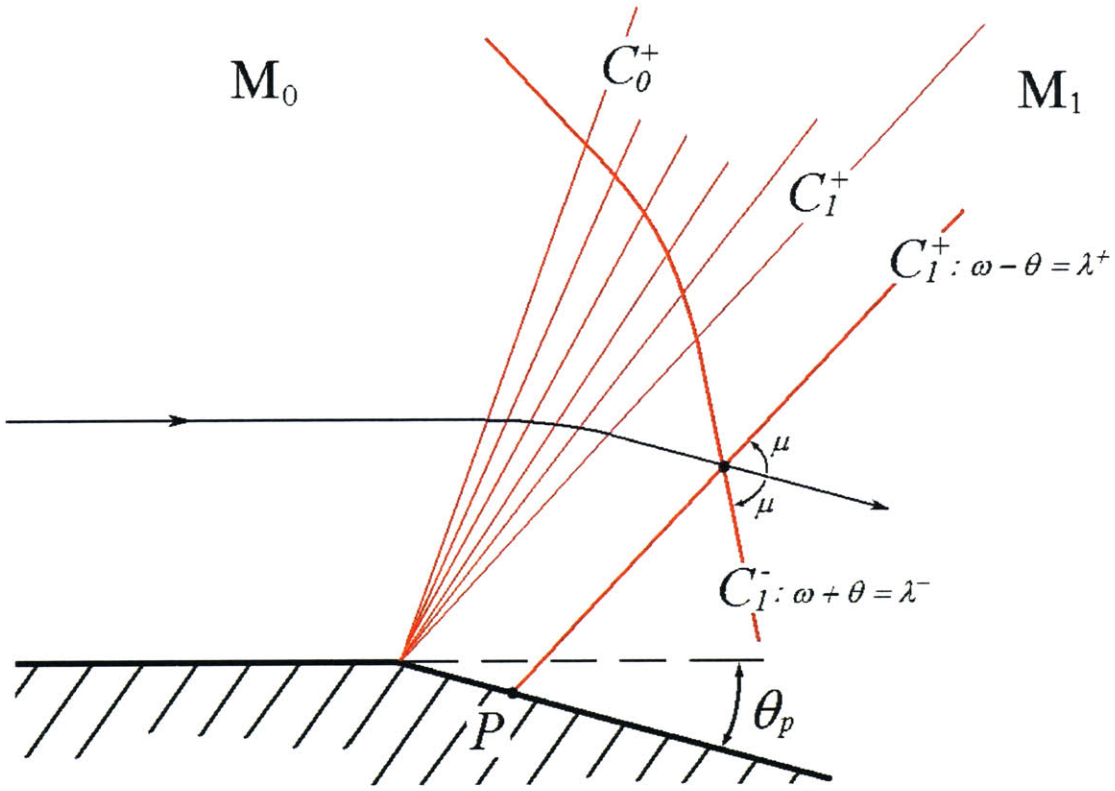


Figure 4-7: Supersonic flow around a convex dihedron

The resolution of the system is straightforward. At a point M of the flow, the Riemann invariant λ^+ is identified at P, where P is the intersection of the C^+ curve going through M with the solid wall, see Figure 4-7. λ^- is identified in the upstream flow and we thus have:

$$\omega - \theta = \omega_P - \theta_P = \lambda^+ \tag{4.14}$$

$$\omega + \theta = \omega_0 + \theta_0 = \lambda^- \tag{4.15}$$

By further using the information carried along the C^+ curve at P, the system can be solved completely:

$$\theta = \theta_P \tag{4.16}$$

$$\omega = \omega_0 + \theta_0 - \theta_P \tag{4.17}$$

Inverting the Prandtl-Meyer function, the Mach number can be computed from the value of ω given by the previous analysis. Since the entropy remains constant along streamlines their is no loss in stagnation pressure $p_{i_0} = p_{i_1}$ and the pressure drop due to the expansion wave can be written as a function of the up and downstream Mach numbers:

$$\frac{p_1}{p_0} = \frac{p_1}{p_{i_1}} \cdot \frac{p_{i_0}}{p_0} = \left(\frac{1 + \frac{\gamma-1}{2} M_0^2}{1 + \frac{\gamma-1}{2} M_1^2} \right)^{\frac{\gamma}{\gamma-1}} \tag{4.18}$$

The non-linear equation 4.13 has been solved using Matlab and the analytical results for the expansion wave have been used for verification of the simulation results.

4.2.3 Numerical results and verification

A set of simulations corresponding to different angles of attack has been computed and verified against the analytical solution. The Mach number adopted in these simulations is $M_\infty = 1.8$ and the initial properties of the gas are: $p_\infty = 1.0atm$, $\rho_\infty = 1.293 \frac{kg}{m^3}$, and $\gamma = 1.4$.

An example of the flow field resulting from the simulations and corresponding to an angle of attack $\alpha = 15$ degrees is shown in Figure 4-8. The different colors correspond to values of the pressure normalized by the upstream pressure, thus $\frac{p}{p_\infty} = 1.0$ upstream of the profile. The fluid domain is discretized with 400×680 fluid cells. The axes in the figure are scaled in grid points. The computed solution shows an excellent agreement with the analytical solution. In particular, all the features of the flow, including shocks and expansion waves have the right intensity. Figure 4-9 is a plot of $Y(X) = \frac{P(X,290)}{P_\infty}$ and shows a section of the pressure field along grid cells of Y-coordinate 290. As it can be observed in this figure, the upper and lower regions of

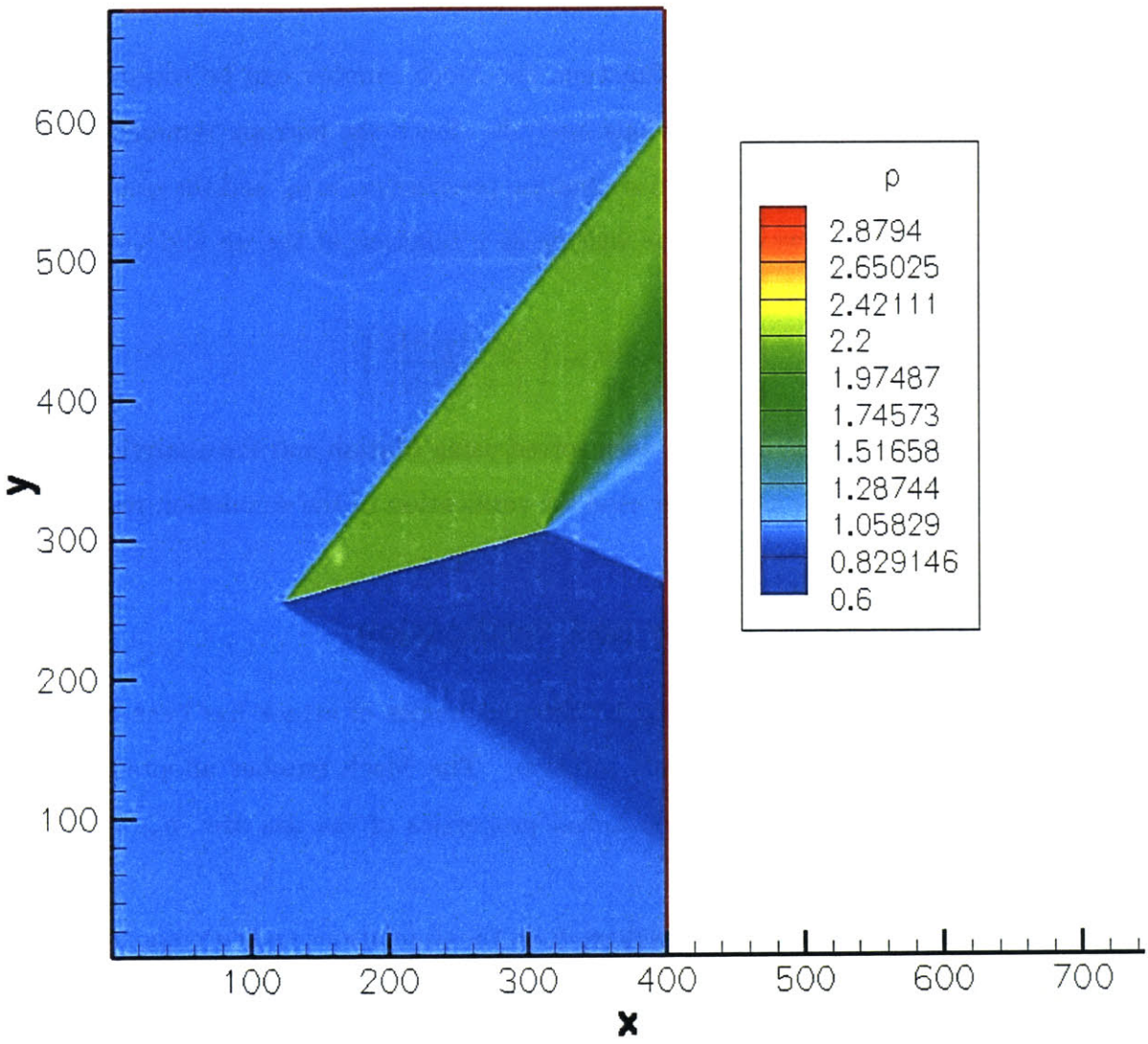


Figure 4-8: Pressure contours of a supersonic flow around a rod

the flow field are computed correctly.

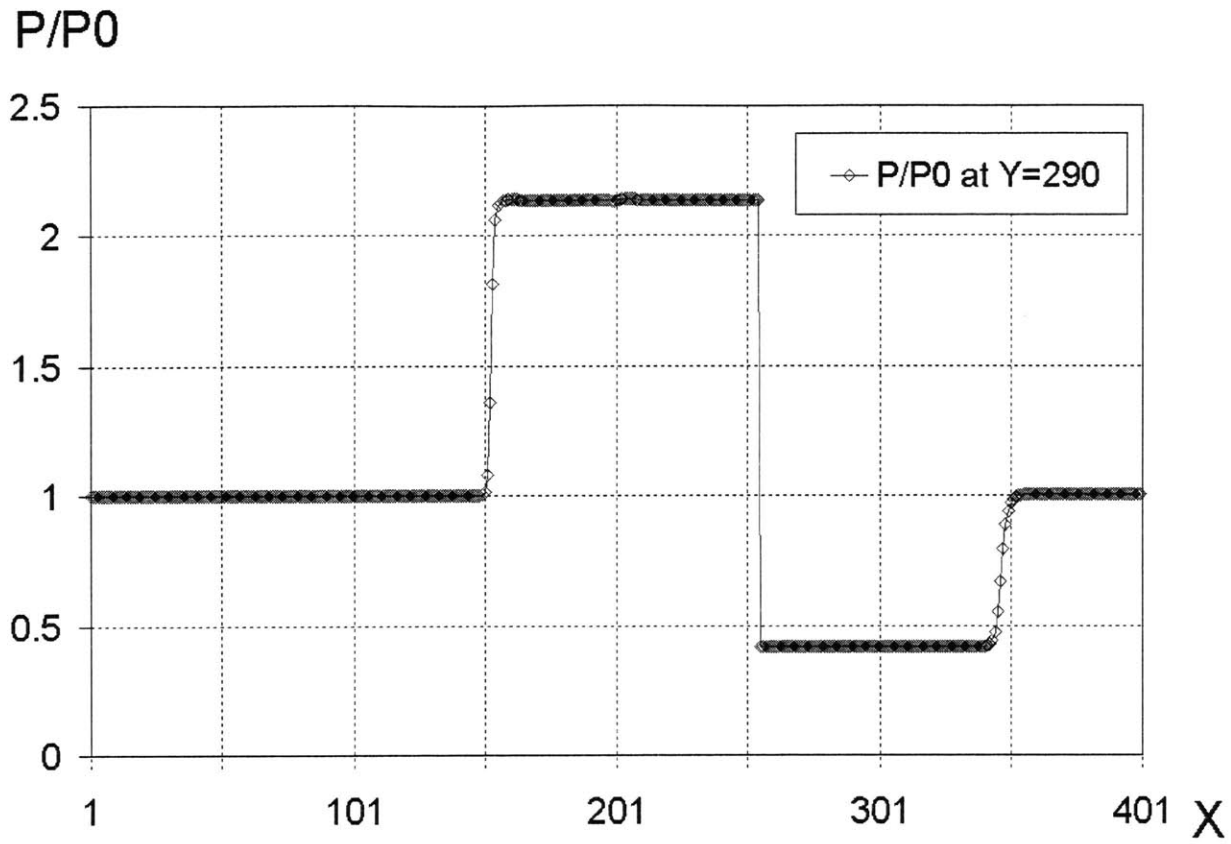


Figure 4-9: $Y(X) = \frac{P(X,290)}{P_\infty}$

- $X \in [0, 155]$: This uniform region corresponds to the upstream region in which $p/p_\infty = 1.0$.

- $X = 155$: The discontinuity of the flow corresponds to the compression shock attached at the leading edge, through which the upper part of the flow is first slowed down.

- $X \in [155, 255]$: The uniform pressure corresponds to the upper region of the flow behind the weak shock.

- $X = 255$: The discontinuity corresponds to the discontinuity of the pressure across the profile. It bears emphasis that there is no pollution from one side of

the boundary to the other. The two regions of the flow are properly isolated and influence each other only through the motion of the interface, which is fixed in this case.

- $X \in [255, 345]$: The uniform pressure corresponds to the lower part of the flow which has been accelerated through the expansion wave around the leading edge.

- $X = 345$: The lower part of the flow is slowed down and compressed through a weak shock at the trailing edge.

- $X \in [345, 400]$: This uniform region in which $p/p_\infty = 1.0$ corresponds to the flow behind the profile.

Similar simulations have been run for a range of angles of attack α from 5 to 18 degrees. For angles of attack over 18 degrees and for an upstream mach number of $M_\infty = 1.8$ the shock at the leading edge of the profile detaches. In this case the nature of the flow features changes so that the comparison to an analytical solution is no longer trivial.

For the range of α under consideration, the values of the pressure in the flow region behind the weak shock and the expansion wave initiated at the leading edge can be computed analytically as a function of α as described in section 4.2.2. Figure 4-10 shows the comparison between the numerically computed and the exact values of the pressure region behind the shock. As expected the compression becomes more severe and the pressure behind the shock increases as α increases. The very good agreement is noteworthy. A similar comparison of the pressure behind the expansion wave is given in Figure 4-11. As α is increased, the flow undergoes a stronger acceleration and the pressure behind the expansion wave decreases. The agreement is equally good.

It can be concluded from these simulations that the algorithm proposed applies appropriate boundary conditions on both sides of the thin profile and results in correct pressure distributions caused by the flow on the boundary. Therefore, it can be expected that this, in turn, will result in appropriate traction boundary conditions on the structure in coupled simulations.

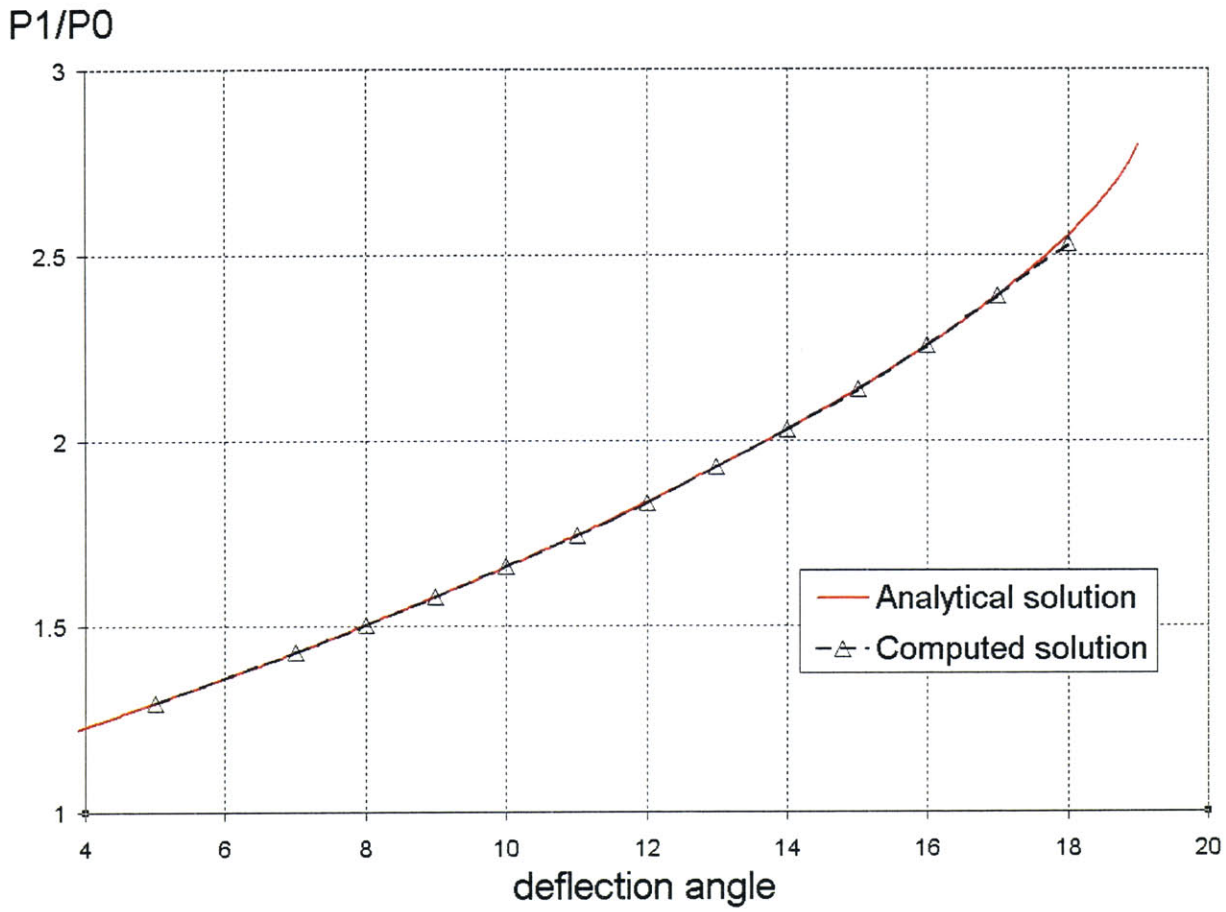


Figure 4-10: Pressure in the flow behind the shock

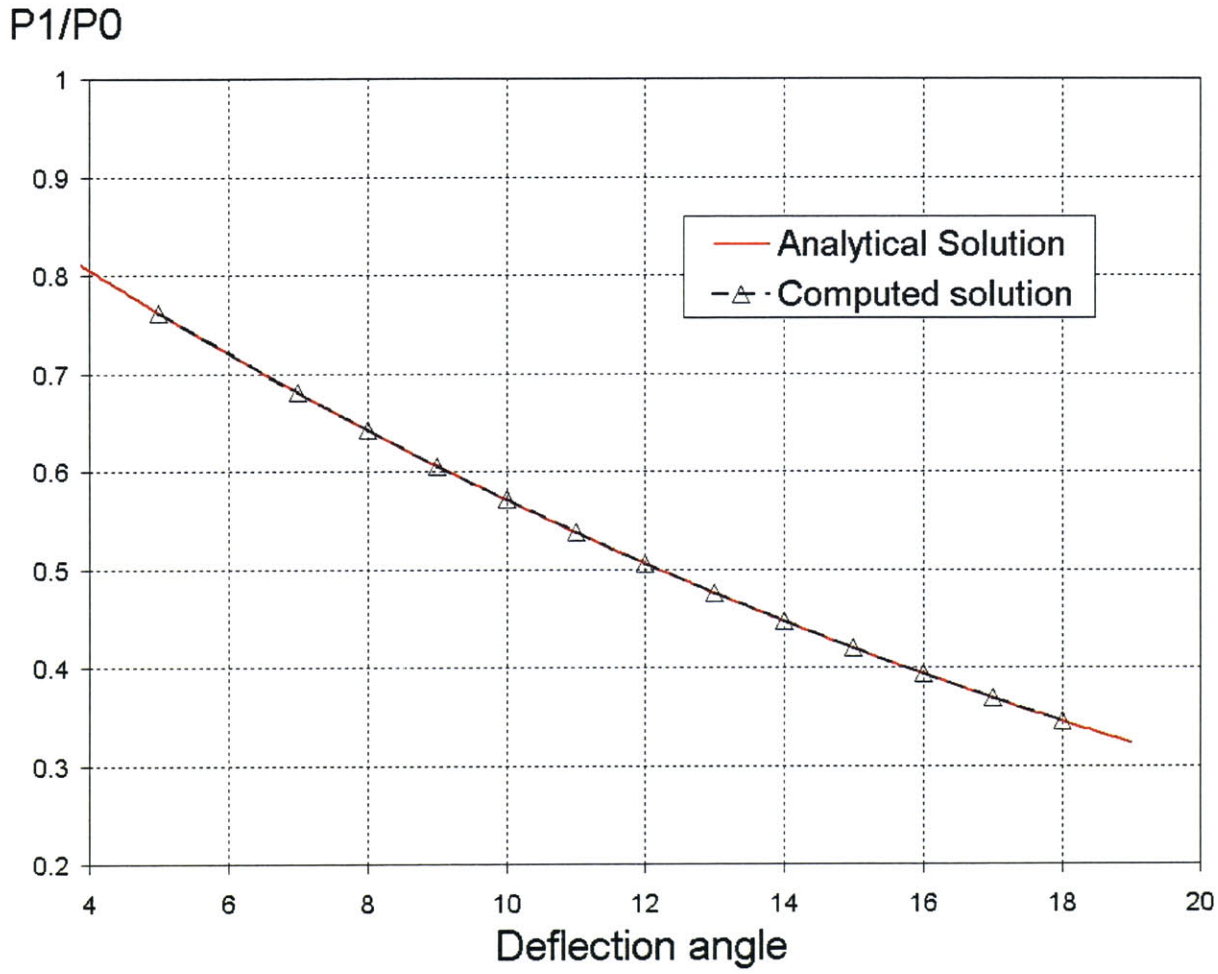


Figure 4-11: Pressure in the flow behind the expansion wave

4.3 Applications to fully-coupled fluid-structure interaction problems

4.3.1 Simulations of supersonic flow past a highly-deformable thin rod

In the previous section we demonstrated the ability of the extended algorithm to simulate supersonic flows around thin open but fixed interfaces. In this section, we investigate the ability of the method to provide adequate descriptions of fully-coupled fluid-structure interaction problems. To this end, the finite element rod model described in section 2.1 has been implemented and coupled with the fluid code using the algorithm described in chapter 3. The resulting computer code has been applied to the simulation of a supersonic flow past an initially-flat two-dimensional rod made of an elastic fabric with a Young's Modulus of $E = 6.0 \cdot 10^9 Pa$, mass density of $\rho = 1000.0 \frac{kg}{m^3}$. The length of the rod is $1.0m$, its thickness $3.0 \cdot 10^{-3}m$, its cross sectional area $A = 1.0 \cdot 10^{-3}m^2$ and its moment of inertia $I = 2.25 \cdot 10^{-9}m^4$. 50 hermite cubic finite elements are used in the discretization.

The flow at infinity has the following initial properties: pressure of $p_\infty = 1.0atm$, mass density of $\rho_\infty = 1.293 \frac{kg}{m^3}$, and $\gamma = 1.4$. The flow's Mach number is $M_\infty = 2.0$. The size of the computational fluid domain is $5.20m \times 9.60m$, discretized with 260×480 fluid cells. A schematic of the simulation set up is shown in Figure 4-12.

At first, the structure is held fixed until steady-state conditions are reached in the flow. A strong shock slows down the flow upstream of the rigid straight rod. The highly deformable structure is then released except at the extremities, which are constrained to move in the x direction, and starts inflating under the pressure of the flow, inducing complex interactions between the flow and the thin structure, see Figure 4-12. Figures 4-13 and 4-14 show pressure isocontours of the flow and the deformed rod at successive time steps over a time range of approximately $25ms$. Figures 4-15 and 4-16 show a detailed view of the pressure field and the deformation of the rod at the same times. Figure 4-15-a shows the initial steady-state flow around

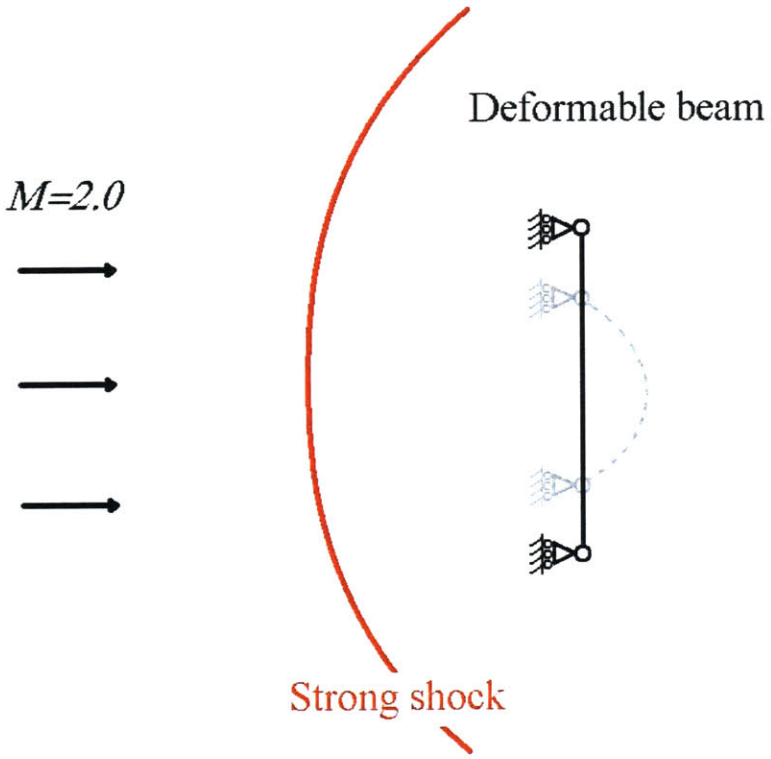
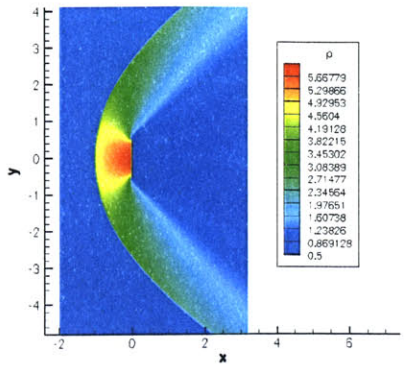
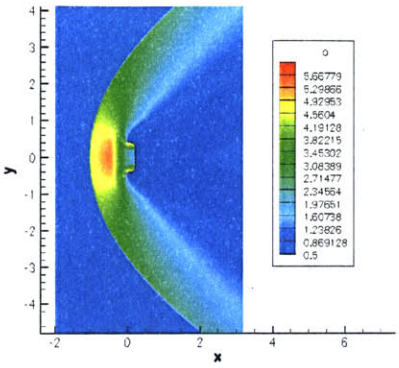


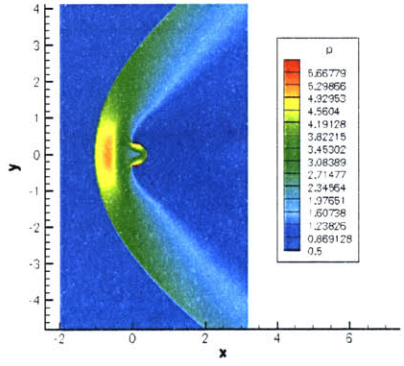
Figure 4-12: Schematic of the coupled simulation



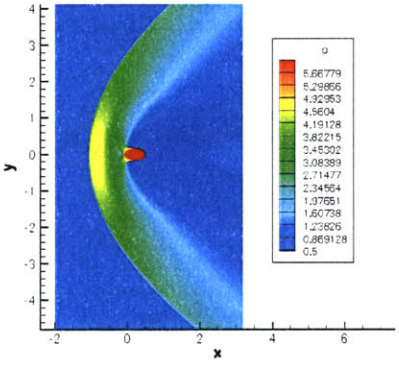
(a) **Step 0 after** $0.00ms$



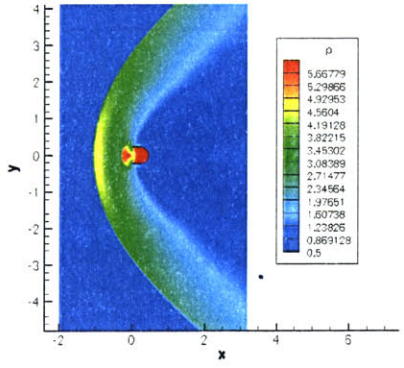
(b) **Step 150 after** $1.20ms$



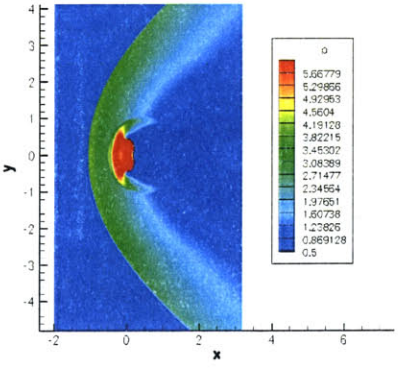
(c) **Step 250 after** $2.00ms$



(d) **Step 350 after** $2.80ms$

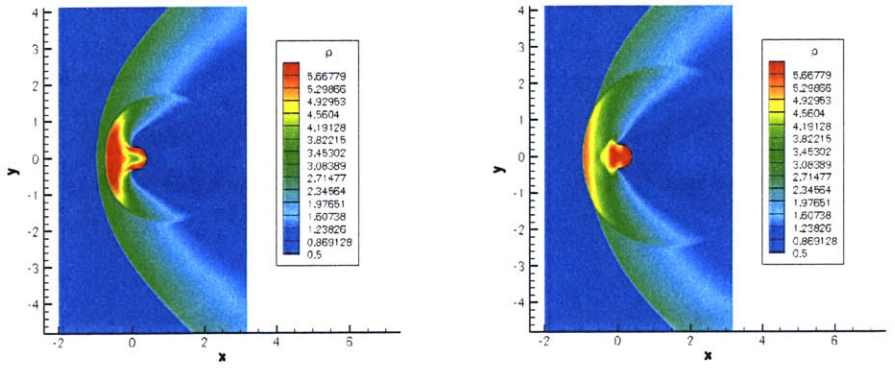


(e) **Step 450 after** $3.60ms$



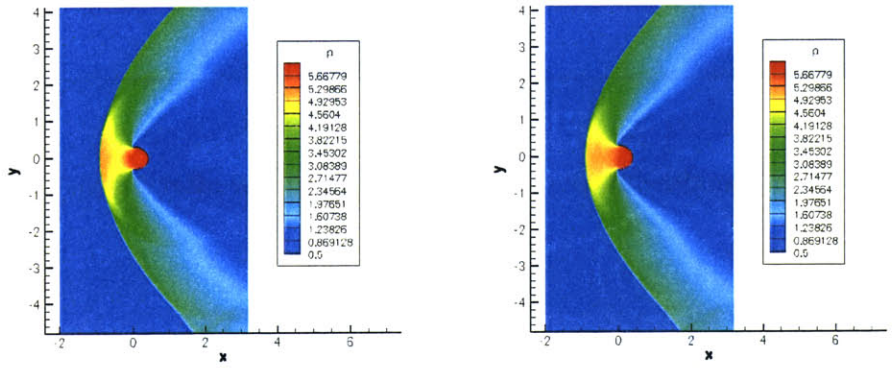
(f) **Step 650 after** $5.20ms$

Figure 4-13: Sequence of snapshots of pressure contours in the simulation of supersonic flow past a highly-deformable rod



(a) **Step 850 after** 6.80ms

(b) **Step 1050 after** 8.40ms



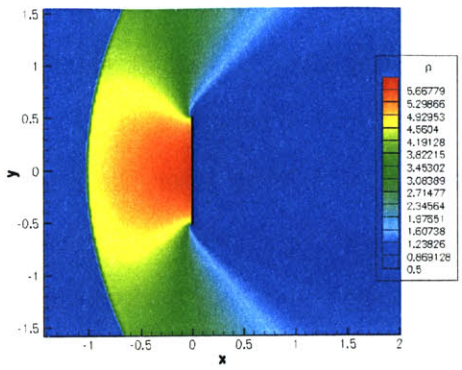
(c) **Step 2050 after** 16.40ms

(d) **Step 3050 after** 24.40ms

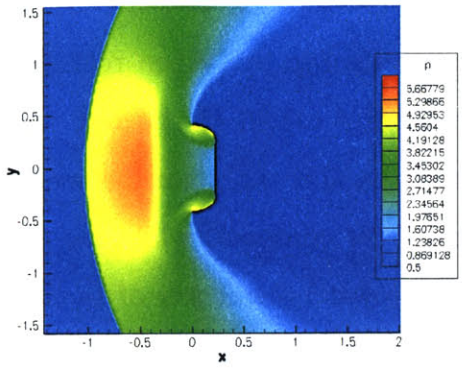
Figure 4-14: Sequence of snapshots of pressure contours in the simulation of supersonic flow past a highly-deformable rod (continued)

the straight rod. A strong shock has formed in front of the profile. In the region upstream of the rod, the pressure is very high $p/p_\infty \geq 5.0$. It is low $p/p_\infty \leq 1.0$ in the downstream region. When released, the rod starts to bend at the extremities, while the center region of the rod remains flat and accelerates rigidly under the pressure of the flow, as it can be seen at time $t = 1.20ms$ after the release of the structure. This continues while the slower flexural wave propagates from the rod extremities toward the center of the beam. As expected, the backwards motion of the rod releases an expansion wave, which travels upstream towards the strong shock. This induces a decrease in the upstream pressure see Figure 4-15-b/c. The bending motion propagates from the extremities of the initially flat rod and converges at the center of the rod around $t = 2.80ms$. The maximum deflection of the center of the rod is reached as the structure is completely inflated. The pressure rebuilds in the upstream region inside the inflated structure, see Figure 4-15-d. At $t = 3.60ms$, the expansion wave, first initiated by the backwards motion of the rod, reaches the strong shock, as it can be observed on Figure 4-15-e. This changes the shape of the shock, which starts to move downstream, thus following with some delay the initial backwards motion of the rod.

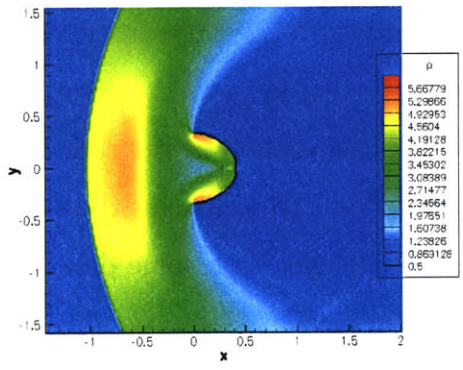
No physical dissipation effects have been included within the finite element model of the rod. The structure thus tends to restitute the stored elastic energy to the surrounding flow. This can be best seen in Figure 4-15-f. As elastic energy is released, the rod pushes the upstream flow and generates a compression wave. On Figure 4-16-a the compression wave shed by the rod is traveling upstream. When this compression wave reaches the strong shock around $t = 8.40ms$, the shock is pushed forwards again following, with some delay, the motion of the rod, see Figure 4-16-b. In this simulation, the choice of the Newmark parameters, defined by equations (2.26), induces numerical dissipation over the time integration. Therefore, the coupled system undergoes large oscillations before slowly converging to a steady state, in the inflated configuration. These oscillations consist of the two phases described previously. The structure first inflates and stores elastic energy and then restitutes the energy to the flow as it moves forward. The shape and location of the shock is



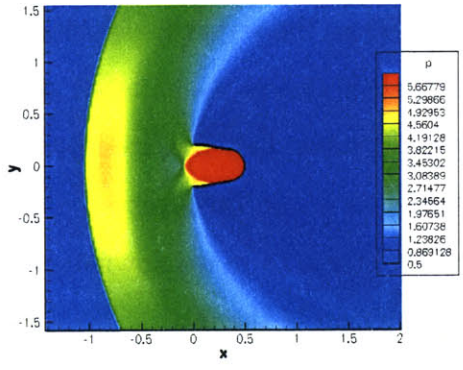
(a) **Step 0 after** $0.00ms$



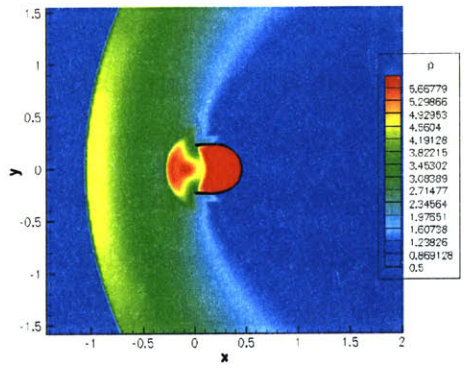
(b) **Step 150 after** $1.20ms$



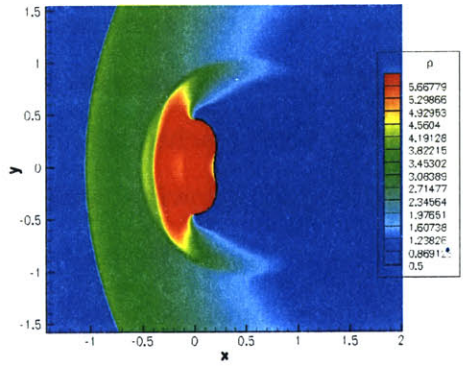
(c) **Step 250 after** $2.00ms$



(d) **Step 350 after** $2.80ms$

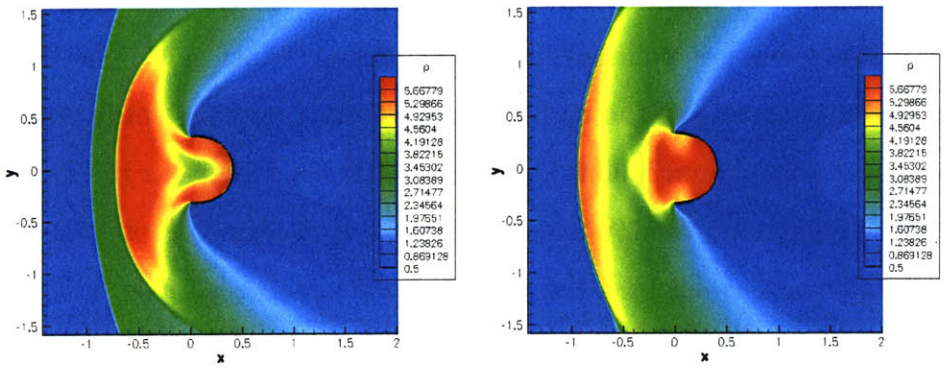


(e) **Step 450 after** $3.60ms$



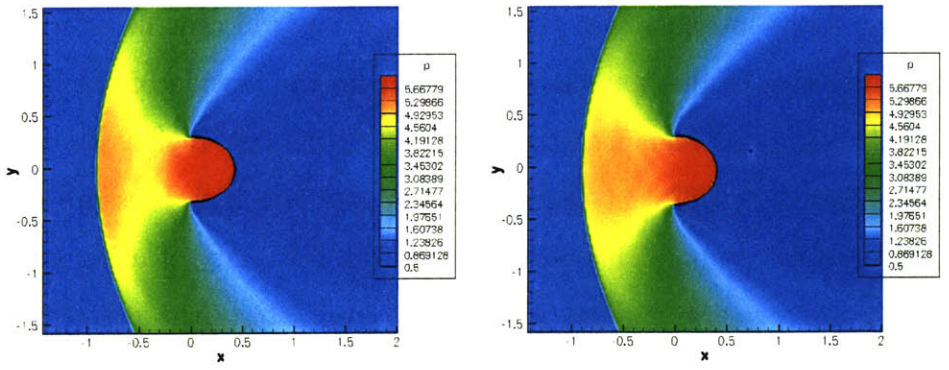
(f) **Step 650 after** $5.20ms$

Figure 4-15: Sequence of snapshots of pressure contours in the simulation of supersonic flow past a highly-deformable rod (detailed view)



(a) **Step 850 after** $6.80ms$

(b) **Step 1050 after** $8.40ms$



(c) **Step 2050 after** $16.40ms$

(d) **Step 3050 after** $24.40ms$

Figure 4-16: Sequence of snapshots of pressure contours in the simulation of supersonic flow past a highly-deformable rod (detailed view, continued)

coupled to the motion of the rod, thus demonstrating the strong coupling between the compressible flow and the deforming solid. The strong shock moves back and forth under the action of the expansion and compression waves successively generated by the rod in motion. Figure 4-16-c/d the coupled system has reached steady state at time $t = 24.40ms$ after the structure was first released.

These figures clearly show that in the coupled unsteady solution, the flow on either side of the interface is not polluted by the flow on the other side. The flow regions upstream and downstream of the profile remain distinct over the entire computation. The pressure remains discontinuous through the interface, between the high pressure upstream flow region and the low pressure downstream region. From this succession of snapshots it can be seen that the two flow regions on each side of the interface do not influence one another in a direct manner but rather indirectly, through the motion of the structure.

Also as the rod undergoes very large displacements, the grid points of the fluid domain are crossed without affecting the coupling of the dynamics. Grid cells neither attract nor repel the interface. On the contrary, many grid cells lie alternatively on one or the other side of the boundary depending on the motion of the interface. It bears emphasis that this remains true, even though the flow regions on each side of the interface have very different characteristics: one is a high pressure region, the other a low pressure region. This demonstrates the feasibility of using a structured Eulerian grid to represent the flow domain. It also attests the correctness of the procedure used to apply the boundary conditions.

Chapter 5

Summary

The objective of the present work was to develop a numerical tool able to simulate the coupling interactions between high-speed flows and highly deformable thin structures. The approach developed here couples a Lagrangian model for the structure dynamics and an Eulerian model for the fluid dynamics. The two dynamics have been modeled separately and then coupled and integrated within a partitioned fluid-structure interaction algorithm. On the one hand, an existing Euler compressible fluid approach models the high-speed flow. On the other hand, a finite element approach has been developed to model the large deformations of a two-dimensional rod.

We proposed an algorithm based on the extension of the Ghost-fluid Eulerian Lagrangian method. In this approach, non-penetration boundary conditions at the immersed interface are applied, at each time iteration, on the flow. Traction boundary conditions are applied on the structure by interpolating the pressure field at the interface. This method has proved to be very robust when dealing with very large displacements of the structure.

The previously implemented algorithm was restricted to thick solid structures and closed boundaries in which a well-defined exterior to the fluid domain exists. A new algorithm has been derived to enable the coupling of the flow with a thin structure. The implementability of the new method has been tested by successfully solving Laplace's equation with boundary conditions imposed on a thin, immersed and open interface. The extended GEL method has been used to model the interface

between the two dynamics. The method was verified against the analytical solution corresponding to the supersonic flow past a flat, infinitely thin, rigid plane at different angles of attack. The numerical solution is shown to reproduce the analytical solution both in the compressed region on the leading side of the plane as well as in the rarefied region on the trailing side of the plane. It bears emphasis that no pollution of the solution across the infinitely thin boundary is observed in the numerical simulation, thus attesting to the ability of the proposed approach to properly describe the flow on both sides of a thin interface without any bias.

As an example of application of the resulting FSI algorithm to a fully coupled problem, we simulated the supersonic flow past a highly deformable rod at a normal angle of attack. The initially flat structure inflates under the pressure of the flow and undergoes large displacements. The simulation shows that a complex pattern of highly unsteady coupled interactions are set in motion between the flow and the structure, leading to the wobbling motion of the structure until it comes to rest in its final inflated configuration. These fully coupled simulations demonstrates that the approach is robust when dealing with large displacements. Even though the accuracy of the method at the boundary is not expected to be better than first order, the algorithm does a remarkable job at capturing the fluid-structure interactions in the large displacement regime. For problems of interest here, the deployment of parachutes for example, this approach enables us to model the deformation of the canopy during its inflation.

One of the main objectives of this work was to develop a coupling method on a two-dimensional problem that could be extended to the three-dimensional case in a straightforward manner. Numerical models of the structure and fluid dynamics have been already developed in the three-dimensional case. The version of the fluid solver RM3D used in this work is simply the two-dimensional version of the existing three-dimensional solver. Also, as mentioned before, a three-dimensional shell model has been developed and successfully used to simulate the inflation of airbags. Throughout this work, special care has been given to the extensibility of the partitioned FSI interface to the three-dimensional case. This extension is not compromised

by mesh generation issues since we chose, from the beginning, an approach based on a structured grid. In fact, the previous GEL formulation has already been used to solve three-dimensional coupled problems. The GEL approach developed in this work introduces simple but effective ideas to deal with open and immersed shells. The dual data structure, using *ghost* and *real* arrays, can be used just as well for the three-dimensional problem. In fact this idea has been implemented directly in the framework of RM3D and should not require much more work to be carried out to the three-dimensional version of the same solver. The related boundary treatment, by bilateral extrapolation and reconstruction of the flow fields on each side of the interface, should be directly transferrable to the three-dimensional case as well; same holds for the interpolation of the pressure fields on each side of the interface, which is required in order to apply the correct traction condition on the thin structure.

The entire methods relies on the computation of a *pseudo* signed level set function, which is needed to locate effectively the interface. As discussed previously, an effective and computationally affordable algorithm is available to compute the three-dimensional level set function. The critical point is the determination and the assignment of the local *pseudo* sign. In principle, the method developed in this work can be used in three dimensions. A unique orientation can be defined at the surface of a three-dimensional shell thus identifying the two distinct sides of the interface. This has to be done carefully though, in order not to turn this step of the algorithm into a computational bottleneck. The treatment of the extremities of the shell should be done with the same care as in the two-dimensional case. With this said, the developed approach should be extendable to fully coupled three-dimensional problems and provide an efficient tool to model complex coupled systems of interest involving high-speed flows and highly deformable thin structures.

Appendix A

Formulation of the beam model

A.1 Strain energy of a rod element

The energy (2.13) can be expressed in terms of the nodal degrees of freedom. We further add the following notations:

$$\mathbf{R_1} = (X_{11}, X_{21}) \tag{A.1}$$

$$\mathbf{R_2} = (X_{12}, X_{22}) \tag{A.2}$$

$$\mathbf{r_1} = (x_{11}, x_{21}) \tag{A.3}$$

$$\mathbf{r_2} = (x_{12}, x_{22}) \tag{A.4}$$

Using the fomulas (2.16), (2.15), (2.15) and the introduced notations the energy can be expressed:

$$\begin{aligned} U \approx EA & \left[|\mathbf{R_2} - \mathbf{R_1}| - |\mathbf{r_2} - \mathbf{r_1}| + |\mathbf{r_2} - \mathbf{r_1}| \log \left(\frac{|\mathbf{r_2} - \mathbf{r_1}|}{|\mathbf{R_2} - \mathbf{R_1}|} \right) \right] \\ & + \frac{2EI}{|\mathbf{r_2} - \mathbf{r_1}|^3} [((x_{12} - x_{11}) \sin\phi_1 - (x_{22} - x_{21}) \cos\phi_1)^2 \\ & + ((x_{12} - x_{11}) \sin\phi_1 - (x_{22} - x_{21}) \cos\phi_1) ((x_{12} - x_{11}) \sin\phi_2 - (x_{22} - x_{21}) \cos\phi_2) \\ & + \frac{2EI}{|\mathbf{r_2} - \mathbf{r_1}|^3} [((x_{12} - x_{11}) \sin\phi_2 - (x_{22} - x_{21}) \cos\phi_2)^2] \end{aligned}$$

where E is the Young modulus, A the cross section of the rod and I the moment of inertia.

A.2 Expression of the internal Forces

Recall from section 2.1

$$F_{ia}^{int}(\mathbf{X}_h) = \frac{\partial \Pi_{int}}{\partial x_{ia}} \tag{A.5}$$

where $i = 1..3$ is the degree of freedom, $a = 1..2$ the node, F_{ia}^{int} the internal force on the $ia = 1..6$ degree of freedom of the rod. In this case, there are three degrees of freedom per node: translation in the X_1 direction, translation in the X_2 direction and a bending rotation of angle ϕ:

$$\begin{aligned}
F_1^{int} &= \frac{\partial \Pi_{int}}{\partial x_{11}} \\
F_2^{int} &= \frac{\partial \Pi_{int}}{\partial x_{21}} \\
F_3^{int} &= \frac{\partial \Pi_{int}}{\partial \phi_1} \\
F_4^{int} &= \frac{\partial \Pi_{int}}{\partial x_{12}} \\
F_5^{int} &= \frac{\partial \Pi_{int}}{\partial x_{22}} \\
F_6^{int} &= \frac{\partial \Pi_{int}}{\partial \phi_2}
\end{aligned}$$

We introduce the following notations:

$$\begin{aligned}
L &= \sqrt{(X_{12} - X_{11})^2 + (X_{22} - X_{21})^2} \\
l &= \sqrt{(x_{12} - x_{11})^2 + (x_{22} - x_{21})^2} \\
\theta(1) &= (x_{12} - x_{11}) \sin \phi_1 - (x_{22} - x_{21}) \cos \phi_1 \\
\theta(2) &= (x_{12} - x_{11}) \sin \phi_2 - (x_{22} - x_{21}) \cos \phi_2
\end{aligned}$$

Notations for the first derivatives of these quantities:

$$\delta l(1) = \frac{\partial l}{\partial x_{11}} = -\frac{x_{12} - x_{11}}{l}$$

$$\delta l(2) = \frac{\partial l}{\partial x_{21}} = -\frac{x_{22} - x_{21}}{l}$$

$$\delta l(3) = \frac{\partial l}{\partial \phi_1} = 0$$

$$\delta l(4) = \frac{\partial l}{\partial x_{12}} = \frac{x_{12} - x_{11}}{l}$$

$$\delta l(5) = \frac{\partial l}{\partial x_{22}} = \frac{x_{22} - x_{21}}{l}$$

$$\delta l(6) = \frac{\partial l}{\partial \phi_2} = 0$$

$$
\begin{aligned}
\delta\theta(1,1) = \quad & \frac{\partial\theta(1)}{\partial x_{11}} = && -\sin\phi_1 \\
\delta\theta(1,2) = \quad & \frac{\partial\theta(1)}{\partial x_{21}} = && \cos\phi_1 \\
\delta\theta(1,3) = \quad & \frac{\partial\theta(1)}{\partial \phi_1} = && (x_{12}-x_{11})\cos\phi_1 + (x_{22}-x_{21})\sin\phi_1 \\
\delta\theta(1,4) = \quad & \frac{\partial\theta(1)}{\partial x_{12}} = && \sin\phi_1 \\
\delta\theta(1,5) = \quad & \frac{\partial\theta(1)}{\partial x_{22}} = && -\cos\phi_1 \\
\delta\theta(1,6) = \quad & \frac{\partial\theta(1)}{\partial \phi_2} = && 0.0 \\
\delta\theta(2,1) = \quad & \frac{\partial\theta(2)}{\partial x_{11}} = && -\sin\phi_2 \\
\delta\theta(2,2) = \quad & \frac{\partial\theta(2)}{\partial x_{21}} = && \cos\phi_2 \\
\delta\theta(2,3) = \quad & \frac{\partial\theta(2)}{\partial \phi_1} = && 0.0 \\
\delta\theta(2,4) = \quad & \frac{\partial\theta(2)}{\partial x_{12}} = && \sin\phi_2 \\
\delta\theta(2,5) = \quad & \frac{\partial\theta(2)}{\partial x_{22}} = && -\cos\phi_2 \\
\delta\theta(2,6) = \quad & \frac{\partial\theta(2)}{\partial \phi_2} = && (x_{12}-x_{11})\cos\phi_2 + (x_{22}-x_{21})\sin\phi_2
\end{aligned}
$$

Using this notation the expression of F_{ia}^{int} for $ia = 1..6$ can be written:

$$
\begin{aligned}
F_{ia}^{int} = & EA\delta l(ia)\ln\left(\frac{l}{L}\right) \\
& - \frac{6EI}{l^4}\delta l(ia)\left[\theta(1)^2 + \theta(1)\theta(2) + \theta(2)^2\right] \\
& + \frac{2EI}{l^3}\left[\delta\theta(1,ia)(2\theta(1)+\theta(2)) + \delta\theta(2,ia)(2\theta(2)+\theta(1))\right]
\end{aligned}
$$

A.3 Expression of the stiffness matrix

We introduce the following notations for the second derivatives of these quantities:

$$\begin{aligned}
\delta^2 l(1,1) &= \frac{\partial^2 l}{\partial^2 x_{11}} &&= \frac{1}{l} - \frac{(x_{12}-x_{11})^2}{l^3} \\
\delta^2 l(1,2) &= \frac{\partial^2 l}{\partial x_{21}\partial x_{11}} &&= -\frac{(x_{12}-x_{11})(x_{22}-x_{21})}{l^3} \\
\delta^2 l(1,4) &= \frac{\partial^2 l}{\partial x_{12}\partial x_{11}} &&= -\delta^2 l(1,1) \\
\delta^2 l(1,5) &= \frac{\partial^2 l}{\partial x_{22}\partial x_{11}} &&= -\delta^2 l(1,2) \\
\delta^2 l(2,1) &= \frac{\partial^2 l}{\partial x_{11}\partial x_{21}} &&= -\frac{(x_{12}-x_{11})(x_{22}-x_{21})}{l^3} \\
\delta^2 l(2,2) &= \frac{\partial^2 l}{\partial^2 x_{21}\partial x_{21}} &&= \frac{1}{l} - \frac{(x_{22}-x_{21})^2}{l^3} \\
\delta^2 l(2,4) &= \frac{\partial^2 l}{\partial x_{12}\partial x_{21}} &&= -\delta^2 l(2,1) \\
\delta^2 l(2,5) &= \frac{\partial^2 l}{\partial x_{22}\partial x_{21}} &&= -\delta^2 l(2,2) \\
\delta^2 l(4,1) &= \frac{\partial^2 l}{\partial x_{11}\partial x_{12}} &&= \delta^2 l(1,4) \\
\delta^2 l(4,2) &= \frac{\partial^2 l}{\partial x_{21}\partial x_{12}} &&= \delta^2 l(2,4) \\
\delta^2 l(4,4) &= \frac{\partial^2 l}{\partial^2 x_{12}} &&= \delta^2 l(1,1) \\
\delta^2 l(4,5) &= \frac{\partial^2 l}{\partial x_{22}\partial x_{12}} &&= \delta^2 l(1,2) \\
\delta^2 l(5,1) &= \frac{\partial^2 l}{\partial x_{11}\partial x_{22}} &&= \delta^2 l(1,5) \\
\delta^2 l(5,2) &= \frac{\partial^2 l}{\partial x_{21}\partial x_{22}} &&= \delta^2 l(2,5) \\
\delta^2 l(5,4) &= \frac{\partial^2 l}{\partial x_{12}\partial x_{22}} &&= \delta^2 l(2,1) \\
\delta^2 l(5,5) &= \frac{\partial^2 l}{\partial^2 x_{22}} &&= \delta^2 l(2,2)
\end{aligned}$$

All the other $\delta^2 l(ia, jb)$ for $ia = 1..6$ and $jb = 1..6$, are equal to zero.

$$\delta^2\theta(1,1,3) = \frac{\partial^2\theta(1)}{\partial\phi_1\partial x_{11}} = -\cos\phi_1$$

$$\delta^2\theta(1,2,3) = \frac{\partial^2\theta(1)}{\partial\phi_1\partial x_{21}} = -\sin\phi_1$$

$$\delta^2\theta(1,3,1) = \frac{\partial^2\theta(1)}{\partial x_{11}\partial\phi_1} = -\cos\phi_1$$

$$\delta^2\theta(1,3,2) = \frac{\partial^2\theta(1)}{\partial x_{21}\partial\phi_1} = -\sin\phi_1$$

$$\delta^2\theta(1,3,3) = \frac{\partial^2\theta(1)}{\partial^2\phi_1} = -(x_{12}-x_{11})\sin\phi_1 + (x_{22}-x_{21})\cos\phi_1$$

$$\delta^2\theta(1,3,4) = \frac{\partial^2\theta(1)}{\partial x_{12}\partial\phi_1} = \cos\phi_1$$

$$\delta^2\theta(1,3,5) = \frac{\partial^2\theta(1)}{\partial x_{22}\partial\phi_1} = \sin\phi_1$$

$$\delta^2\theta(1,4,3) = \frac{\partial^2\theta(1)}{\partial\phi_1\partial x_{12}} = \cos\phi_1$$

$$\delta^2\theta(1,5,3) = \frac{\partial^2\theta(1)}{\partial\phi_1\partial x_{22}} = \sin\phi_1$$

$$\delta^2\theta(2,1,6) = \frac{\partial^2\theta(2)}{\partial\phi_2\partial x_{11}} = -\cos\phi_2$$

$$\delta^2\theta(2,2,6) = \frac{\partial^2\theta(2)}{\partial\phi_2\partial x_{21}} = -\sin\phi_2$$

$$\delta^2\theta(2,4,6) = \frac{\partial^2\theta(2)}{\partial\phi_2\partial x_{12}} = \cos\phi_2$$

$$\delta^2\theta(2,5,6) = \frac{\partial^2\theta(2)}{\partial\phi_2\partial x_{22}} = \sin\phi_2$$

$$\delta^2\theta(2,6,1) = \frac{\partial^2\theta(2)}{\partial x_{11}\partial\phi_2} = -\cos\phi_2$$

$$\delta^2\theta(2,6,2) = \frac{\partial^2\theta(2)}{\partial x_{21}\partial\phi_2} = -\sin\phi_2$$

$$\delta^2\theta(2,6,4) = \frac{\partial^2\theta(2)}{\partial x_{12}\partial\phi_2} = \cos\phi_2$$

$$\delta^2\theta(2,6,5) = \frac{\partial^2\theta(2)}{\partial x_{22}\partial\phi_2} = \sin\phi_2$$

$$\delta^2\theta(2,6,6) = \frac{\partial^2\theta(2)}{\partial^2\phi_2} = -(x_{12}-x_{11})\sin\phi_2 + (x_{22}-x_{21})\cos\phi_2$$

All the other $\delta^2\theta(k, ia, jb)$ for $k = 1..2$, $ia = 1..6$ and $jb = 1..6$, are equal to zero.

The stiffness matrix is needed for the time integration of the structure formulation, see Chapter 2, Section 2.1:

$$K_{ia,jb} = \frac{\partial F_{jb}^{int}}{\partial x_{ia}} = \frac{\partial^2 \Pi_{int}}{\partial x_{ia} \partial x_{jb}} \tag{A.6}$$

where $ia = 1..6$ and $jb = 1..6$. Using the notations just introduced the expression for $K_{ia,jb}$ can be written:

$$\begin{aligned}
K_{ia,jb} =& EA\delta^2 l(ia, jb) \ln\left(\frac{l}{L}\right) + \frac{EA}{l}\delta l(ia)\delta l(jb) \\
&+ \frac{24EI}{l^5}\delta l(ia)\delta l(jb)\left[\theta(1)^2 + \theta(1)\theta(2) + \theta(2)^2\right] \\
&- \frac{6EI}{l^4}\delta^2 l(ia, jb)\left[\theta(1)^2 + \theta(1)\theta(2) + \theta(2)^2\right] \\
&- \frac{6EI}{l^4}\delta l(ia)\left[\delta\theta(1, jb)(2\theta(1) + \theta(2)) + \delta\theta(2, jb)(2\theta(2) + \theta(1))\right] \\
&- \frac{6EI}{l^4}\delta l(jb)\left[\delta\theta(1, ia)(2\theta(1) + \theta(2)) + \delta\theta(2, ia)(2\theta(2) + \theta(1))\right] \\
&+ \frac{2EI}{l^3}[2\delta\theta(1, ia)\delta\theta(1, jb) + \delta\theta(1, ia)\delta\theta(2, jb) \\
&\quad + 2\delta\theta(2, ia)\delta\theta(2, jb) + \delta\theta(2, ia)\delta\theta(1, jb)] \\
&+ \frac{2EI}{l^3}\left[\delta^2\theta(1, ia, jb)(2\theta(1) + \theta(2)) + \delta^2\theta(2, ia, jb)(2\theta(2) + \theta(1))\right]
\end{aligned}$$

Bibliography

[1] ASCI Alliance Center for the Simulation of Dynamic Response of Materials, FY00 Annual Report. URL: http:// www.cacr.caltech.edu/ ASAP/onlineresources/publications/, 2000.

[2] M. Arienti, P. Hung, and J. Morano, E.and Shepherd. A level set approach to eulerian-lagrangian coupling. *Journal of Computational Physics*, 185:213–251, 2003.

[3] K. Bathe, H. Zhang, and S. Ji. Finite element analysis of fluid flows fully coupled with structural interactions. *Computers and Structures*, 72:1–16, 1999.

[4] B. Bell, P. Colella, W. Crutchfield, R. Pember, and M Welcome. An adaptive cartesian grid method for unsteady compressible flow in irregular region. *Journal of Computational Physics*, 120:278–304, 1995.

[5] R. Benney and K. Stein. Computational fluid-structure model for parachute inflation. *Journal of Aircraft*, 33(4):730–736, 1996.

[6] M. Berger and R. LeVeque. An adaptive cartesian mesh algorithm for the euler equations in arbitrary geometries. In *Proceedings, AIAA 9th Computational Fluid Dynamics Conference*, pages 1–7, Buffalo, NY, June 14-16 1989. American Institute of Aeronautics and Astronautics.

[7] A. Bonnet and J. Luneau. *Théorie de la Dynamique des Fluides.* Cépaduès Editions, Toulouse, France, 1989.

[8] L. Chiang, B. van Leer, and K. Powell. Simulation of unsteady inviscid flow on an adaptively refined cartesian grid. *AIAA, technical Report*, 1992.

[9] F. Cirak and M. Ortiz. Fully c^1-conforming subdivision elements for finite deformation thin-shell analysis. *Internat. J. Numer. Methods Engrg.*, 51:813–833, 2001.

[10] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: A new paradigm for thin-shell finite-element analysis. *Internat. J. Numer. Methods Engrg.*, 47(12):2039–2072, 2000.

[11] F. Cirak and R. Radovitzky. A general algorithm for coupling lagrangian-shell with eulerian-fluid formulations. In *Proceedings of the IUTAM Symposim on Integrated Modeling of Fully Coupled Fluid-Structure Interactions Using Analysis, Computations and Experiments*, New Brunswick, NJ, June 1-6 2003. International Union of Theoretical and Applied Mechanics.

[12] F. Cirak and R. Radovitzky. A new fluid-shell coupling algorithm based on level sets. In *Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Norfolk, VA, April 7-10 2003. American Institute of Aeronautics and Astronautics.

[13] Stephen H. Crandall. *Engineering Analysis, A survey of Numerical Procedures.* McGraw-Hill, New York, 1956.

[14] J. Cummings, M. Aivazis, R. Samtaney, R. Radovitzky, S. Mauch, and D. Meiron. A virtual test facility for the simulation of dynamic response in materials. *Journal Of Supercomputing*, 23(1):39–50, 2002.

[15] D. De Zeeuw and K. Powell. An adaptively refined cartesian mesh solver fo the euler equations. *Journal of Computational Physics*, 104:56–68, 1993.

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

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

[18] R.P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152:457–492, 1999.

[19] P. Geuzaine, G. Brown, C. Harris, and C. Farhat. Aeroelastic dynamic analysis of a full f-16 configuration for various flight conditions. *AIAA JOURNAL*, 41:363–371, 2003.

[20] P. Geuzaine, C. Grandmont, and C. Farhat. Design and analysis of ale schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *Journal of Computational Physics*, 191:206–227, 2003.

[21] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1, Fundamentals of Numerical Discretization. John Wiley and Sons, 1988.

[22] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 2, Computational Methods for Inviscid and Viscous Flows. John Wiley and Sons, 1990.

[23] H. Hu, N. Patankar, and M. Zhu. Direct numerical simulations of fluid-solid systems using the arbitrary eulerian-lagrangian technique. *Journal of Computational Physics*, 169:427–462, 2001.

[24] Thomas J. R. Hughes. *The Finite Element Method.* Dover, 2000.

[25] V. Kalro and T. Tezduyar. A parallel 3-d computational method for fluid-structure interactions in parachutes systems. *Computational Methods Applied Mechanics and Engineering*, 190:321–332, 2000.

[26] Culbert B. Laney. *Computational Gasdynamics.* Cambridge University Press, 1998.

[27] R. LeVeque. A large time step generalization of godunov's method for system of conservation laws. *SIAM Journal on Numerical Analysis*, 22:1051–1073, 1985.

[28] R. LeVeque. Cartesian grid methods for flow in irregular regions. *Numerical Methods in Fluid Dynamics, III*, pages 375–382, 1988.

[29] R. LeVeque. High resolution finite volume methods on arbitraty grids via wave propagation. *Journal of Computational Physics*, 78:36–63, 1988.

[30] Randall J. LevVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[31] S. Mauch. A fast algorithm for computing the closest point and distance transform. *Preprint, http://www.acm.caltech.edu/~seanm/software/cpt/cpt.html*, 2001.

[32] D. Mavriplis. Accurate multigrid solution of the euler equations on unstructured and adaptive meshes. *AIAA Journal*, 28(2):213–221, 1990.

[33] D. Meiron, R. Radovitzky, and R. Samtaney. The virtual test facility: An environment for simulating the nonlinear dynamic response of solids under shock and detonation wave loading. In *Proceedings of the Sixth U.S. National Congress on Computational Mechanics*, Dearborn, MI, 2001. U.S. Association for Computational Mechanics.

[34] J. Peraire, Peiró, and K. Morgan. Adaptive remeshing for three-dimensional compressible flow computations. *Journal of Computational Physics*, 103:269–285, 1992.

[35] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72:449–466, 1987.

[36] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in Fortran, Second Edition.* Cambridge University Press, 1992.

[37] J. Quirk. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers Fluids*, 23(1):125–142, 1994.

[38] J. Quirk. A parallel adaptiv grid algorithm for computational shock hydrodynamics. *Applied Numerical Mathematics*, 20:427–453, 1996.

[39] R. Radovitzky and M. Ortiz. Lagrangian finite element analysis of newtonian fluid flows. *International Journal For Numerical Methods In Engineering*, 43(4):607–617, 1998.

[40] J. N. Reddy. *Energy Principles and Variational Methods in Applied Mechanics, Second Edition.* John Wiley and Sons, Hoboken, NJ, 2002.

[41] R. Samtaney and D. I. Meiron. Hypervelocity richtmyer-meshkov instability. *Physics of Fluids*, 9(6):1783–1803, 1997.

[42] R. Samtaney and N. J. Zabusky. Circulation deposition on shock-accelerated planar and curved density-stratified interfaces: models and scaling laws. *Journal of Fluid Mechanics*, 269:45–78, 1994.

[43] K. Stein, R. Benney, V. Kalro, T. Tezduyar, T Leonard, and M. Accorsi. Parachute fluid-structure interactions: 3-d computation. *Computer Methods in Applied Mechanics and Engineering*, 190:373–386, 2000.

[44] T. Tezduyar, Y. Osawa, K. Stein, R. Benney, V. Kumar, and J. McCune. Numerical methods for computer assisted analysis of parachute mechanics. In *Proceedings of 8th International Conference on Numerical Methods in Continuum Mechanics*, Liptovsky Jan, Slovakia, 2000. U.S. Association for Computational Mechanics.

[45] J. Thompson and U. Warsi. Boundary-fitted coordinates systems for numerical solution of partial differential equations - a review. *Journal of Computational Physics*, 47:1–95, 1982.

[46] P. Thompson. *Compressible Fluid Dynamics.* McGraw-Hill, New York, 1972.

[47] A. Wiegmann and K.P. Bube. The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions. *SIAM J. Numer. Anal.*, 37(3):827–862, 2000.