NOTE TO USERS

A Tight Two-Way Fluid-Structure Coupling for Aeroelastic Computations in the Time Domain

Claude Lepage

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

July 2004

Your file Votre référence
ISBN: 0-612-96962-2
Our file Notre référence
ISBN: 0-612-96962-2

Abstract

A Tight Two-Way Fluid-Structure Coupling for Aeroelastic Computations in the Time Domain

Claude Lepage, Ph.D.
Concordia University, 2004

A three-dimensional implicit finite element Euler/Navier-Stokes solver based on the Arbitrary Lagrangian-Eulerian formulation has been developed for fluid-structure interactions, moving and deforming bodies, and aeroelastic calculations. The flow solver is complemented by a built-in moving grid algorithm which allows the computational grid for the fluid to move simultaneously with the moving body, without the need for global mesh regeneration and solution interpolation. The flow solver is coupled to a stress solver in a tight fashion at each time step, although both solvers exist independently and are used as black boxes. The aerodynamic loads and the structural displacements are transferred between the two solvers in a conservative fashion using the concept of a virtual grid, which guarantees the conservative transfer of both the pressure and the viscous shear stresses and also lends itself to complex geometries featuring non-matching grids at the fluid-solid interface. The level of tightness of the coupling on the aeroelastic response is investigated and the efficiency and accuracy of different solution strategies are discussed. The technology has been validated on well documented aeroelastic configurations.

Acknowledgements

I am grateful to my thesis advisor, Professor W.G. Habashi, formerly at Concordia University and now at McGill University, for his guidance throughout the years and for providing the great research facilities and computing equipment at the CFD Laboratory. I am also thankful to Professor R. Bhat, of Concordia University, for accepting to become my co-supervisor after Professor Habashi acceded to a new position at McGill University.

My appreciation extends to Claude Lacombe of Pratt & Whitney Canada for the insightful discussions in the early stages of my research about the applications of aeroelasticity in an industrial context, in particular for turbomachinery components. I also wish to thank Dr. Siva Nadarajah of the McGill CFD Laboratory for his encouragements and assistance in the final stages of my thesis.

My doctoral work was funded by an NSERC post-graduate scholarship during the first two years. The sponsorship by Pratt & Whitney Canada, in parts, and the CFD Laboratory during the remaining years of my studies is greatly appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The design of modern aircraft, using lighter and thinner, hence more flexible materials, has stimulated the development of state-of-the-art numerical models for the accurate prediction of the aeroelastic phenomena that affect the aerodynamic performance, the maneuverability, and the controllability of the aircraft, as well as the internal maximum stresses occurring in its components. Typical aeroelastic evaluations of this nature include the study of external flows, such as the analysis of flutter over wings or the galloping of transmission cables, and, for internal flows, the vibration of turbomachinery blades. In the latter case, blade excitations, of large amplitude and traveling at the blade passing frequency, constitute an important phenomenon for which the interaction between rotor-stator rows can excite different structural modes in torsion and bending, leading to high-cycle-fatigue and to premature structural failure. The study of moving rigid or flexible bodies in an air stream, such as the deployment of flaps and ailerons on a wing, constitutes another class of fluid-structure interactions.

In general fluid-structure interaction cases, the aerodynamic forces influence the motion of the structure (or its deformation), which in return affects the characteristics

of the flow field. The role of aeroelasticity is to examine the limits of stability of the structure subject to wind-induced excitations. There are perhaps no better words than those of Smith and Kadambi [1] to describe flutter, here in the context of blade flutter: "Flutter occurs as a dynamic instability in which the blade motion becomes self-excited by extracting energy from the flowing fluid. This condition may develop when the aerodynamic forces become in-phase with and eventually lead the blade motion."

Traditionally, analytical methods based on the linearized equations for potential flow have been used in the study of aeroelastic problems of simple geometry, for example in the study of flow-induced oscillations of a rigid flat plate, based on quasi-unsteady linear aerodynamics. Still today, this approach prevails in the classroom to teach the fundamental mechanisms of flutter.

From a modern industrial perspective, the study of aeroelastic problems has taken a decisively computational turn, making use of state-of-the-art computational fluid dynamics (CFD) and computational structural dynamics (CSD) solvers to determine the stability limits of flow-induced vibrations for complex configurations such as aircraft and turbomachinery blades. The approach taken to solve the coupled problem mostly depends upon the capabilities of the flow and the stress solvers, while seeking a cost-effective balance between the turnaround analysis time and the available computing resources.

Today, boundary element methods are widely used in industrial aeroelastic simulations, in particular the doublet lattice method available in MSC/NASTRAN[1] as part of the aerodynamic solver. Such a method solves the aeroelastic problem in the frequency domain (eigenvalue analysis) and offers attractively low computational costs, although it is recognized that Euler and Navier-Stokes solvers better represent

[1]MSC/NASTRAN is a trademark of The MacNeal-Schwendler Corporation.

the true nonlinear behavior of the flow, critical in aeroelastic computations in the transonic regime. This latter class of aerodynamic solvers are nonlinear and necessitate a solution in the time domain, which can lead to prohibitive computational costs. The main goal of this thesis is to develop such a state-of-the-art aeroelastic solver based on the Euler and Navier-Stokes equations, with the fluid and the stress solvers constituting by themselves the building blocks for the elaboration of the aeroelastic system.

Aeroelastic calculations are still mostly used today, in the aerospace industry in general, as a design evaluation tool, rather than as a design tool per se. Computational aeroelasticity (CAE) enables the development of mechanisms for wing flutter suppression via active control surfaces [2], analyses over full fighter aircraft [3] and helicopter blades [4], and, for internal flows in turbomachines, the determination of the effects of engine noise on the vibrational modes of the fan blades. At this stage, it is perhaps reasonable to view the present method as a means to analyze the final design of a configuration, based on preliminary aeroelastic predictions of the doublet lattice method, for example, rather than to use such a tool in the design phase.

1.1 Literature Survey

As viewed from the CFD community, the achievements in computational aeroelasticity have paralleled the developments of the flow solvers, while essentially taking the availability of the stress solver for granted. A solution approach in the time domain is employed – the only feasible approach using a nonlinear flow solver. From the CSD world's viewpoint, emphasis is given to a solution approach in the frequency domain, exploiting the linearity of the stress solver and using linearized aerodynamics models, which retain the fundamental features of the air stream, but which may neglect the

fine details of the flow field. This thesis reflects the stance of the aerodynamicist: studying the aeroelastic problem in the time domain.

In aeroelastic computations, one is generally not interested in inelastic effects and in large displacements leading to the fracture of the structure — two phenomena which normally occur beyond the flutter point. Thermal effects, important for turbines, are negligible for compressors, for fans, and for external flows. Thus, for stress analysis, the theory of small displacements often suffices to accurately model the structure, as governed by the Navier equations for a linear isotropic and isothermal elastic material, with the use of specialized elements such as shells and plates. These equations are linear and hence are numerically easier and less expensive to solve for than the nonlinear Euler and Navier-Stokes equations governing the fluid. They are commonly solved using the Rayleigh-Ritz discretization method with selected bending and torsional modes (modal analysis).

The modeling of the fluid, on the other hand, requires more in-depth attention. In performing a wing flutter analysis, for example, the objective is to find the limits of stability for the entire envelope of flight, from take-off to cruise. At low speeds, a linearized theory can predict the flutter speed precisely, but at higher speeds, in the transonic regime, the compressible Euler or Navier-Stokes equations must be used to correctly capture the nonlinearities of the flow and to accurately determine the flutter boundary curve. It has been observed that the flutter dynamic pressure decreases as the Mach number increases until Mach = 1, reaching a minimum (the well-known transonic flutter dip), then increases sharply beyond Mach = 1 (see figure 1.1). The determination of this critical point is essential in establishing the safe flight envelope of the aircraft.

The aerodynamic computational models with aeroelastic applications to *external flows* have evolved, in parallel with the evolution of computing power, from the tran-

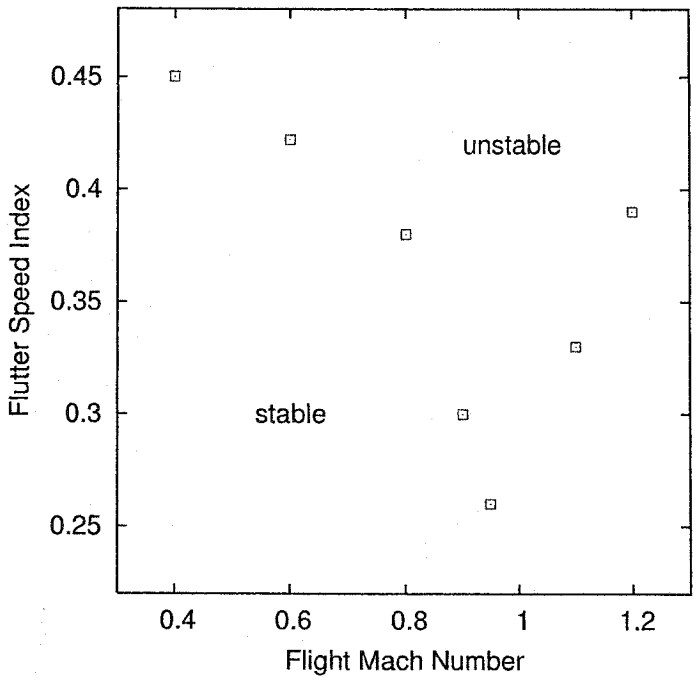


Figure 1.1: Representation of a Wing Flutter Boundary Curve (Qualitative).

sonic small-disturbance (TSD) theory based on the potential flow equations [5, 6], to the compressible Euler equations [5, 7, 8, 9], and finally to the compressible thin-layer and turbulent Navier-Stokes equations [10, 11, 12]. These developments have progressed from 2-D to 3-D steady, to time-linearized, and then to fully unsteady models, using structured and unstructured grids, for external flows over airfoils, wings, and aircraft configurations.

The transonic small-disturbance and full potential flow equations, widely used to study inviscid flows, are, however, deficient despite their attractively low computational requirements. In particular, for the very popular linearized TSD equations, the deformations of the structure are imposed via the surface flow tangency condition $\mathbf{v} \cdot \mathbf{n} = 0$, which takes the form

$$\frac{\partial \Phi}{\partial y} = \frac{1}{U_\infty} \frac{DY}{Dt}, \tag{1.1}$$

where Φ is the perturbation velocity potential ($\phi = U_\infty(x+\Phi)$ is the velocity potential) and $Y(\mathbf{x}, t)$ describes the surface of the deforming body [13]. Consequently, the computational domain does not change as the body deforms, limiting the theory to thin slender wings.

More sophisticated models make use of the Euler and Navier-Stokes equations which incorporate nonlinear effects. While these aerodynamic models can account for rotational (vortical) flows and can capture shock waves, the computational requirements are greatly magnified and must therefore be justified. Unlike the TSD equations, the structural deformations are accounted for directly in the Euler and Navier-Stokes equations, which are in this case most commonly recast using the Arbitrary Lagrangian-Eulerian (ALE) formulation. True aeroelastic computations with two-way feedback are carried out: the aerodynamic forces produce an instantaneous deformation of the structure and these flow-induced displacements alter the flow. Since the deformations are effected in the computational (physical) domain, some sort of remeshing of the domain must take place such that the computational domain follows the motion or the distortion of the structure. It is recalled that no remeshing of the domain is required for the TSD theory. In spite of this added complication, bodies of arbitrary shape and complexity can be now treated.

Fewer publications exist on aeroelastic analyses of *internal flows* through cascades and turbomachines, in comparison to the numerous publications reported for external flows. A review paper by Fransson and Verdon [14] describes numerous standard configurations for two-dimensional cascades. Transonic flows through two-dimensional oscillating cascades are reported in [15, 16] for potential flow methods (linearized unsteady potential, vortex panel method,...), in [17, 18] for Euler calculations, in [19, 20] for viscous-inviscid interaction calculations using an integral boundary layer model, and in [21] for thin-layer Navier-Stokes calculations. Applica-

tions to three-dimensional cascades are presented in [22, 23, 24] for Euler calculations coupled to a finite element stress solver and in [25, 26] for Navier-Stokes calculations.

More advanced aeroelastic calculations of three-dimensional turbomachinery configurations are hindered by the level of complexity of the physical problem (see Rao [27] for a detailed discussion). In particular, rotor-stator interactions are at the source of the pressure disturbances which excite the blades and, to analyze such a problem, substantial extensions to the CFD solver are required to model rotating and non-rotating components simultaneously [28, 29]. For the study of blade vibrations in turbomachinery, it is sufficient to consider only one blade and to use cyclic symmetry to obtain the solution for the entire blade row, in the particular case where all blades vibrate following the same pattern (fundamental mode). However, for more complex vibrational modes, the entire row of blades must be analyzed, thus multiplying the size of the problem and the solution time by the number of blades in a row. In multistage axial, centrifugal, or mixed jet engines, rotor-stator interactions mandate modeling several rows of blades. These notably computationally intensive unsteady simulations, in particular for turbulent Navier-Stokes calculations, have been, in many cases, beyond the scope of the computing resources available to most industries and even major research laboratories.

1.2 Scope of Research

It is apparent from the conducted literature review that advanced three-dimensional CFD codes for Euler and Navier-Stokes aeroelastic calculations have been developed in a number of research laboratories. The code ENSAERO, based on the finite difference method (FDM) and developed at the NASA Ames Research Center by the team headed by G.P. Guruswamy, has been used at Lockheed Martin Aeronautical

Systems for the flutter analysis of wings and aircraft [12]. The finite element edge-based aeroelastic code AU3D [25], developed at Imperial College, is tailored to the analysis of flows in turbomachines [26]. (The use of the finite element method, unlike the finite difference method, offers significant advantages from the aspect of grid generation of complex geometries using unstructured meshes.) Another substantial effort in aeroelasticity is by the group headed by C. Farhat at the University of Boulder, Colorado, with applications to external flows with high-speed flight maneuvers of fighter aircraft [30]. Significant contributions have also been made by this group in the development and study of the theoretical properties of algorithms in computational aeroelasticity (convergence, stability, accuracy, robustness) [31, 32, 33, 34, 35]. Finally, R. Löhner's team has achieved success in general fluid-structure problems involving moving bodies, such as detonation problems, the deployment of parachutes, the launching of missiles, and in vascular flow applications [36, 37, 38].

A common denominator of the aforementioned works is that they all rely on in-house CFD solvers. On the other hand, the use of commercial stress packages, in particular MSC/NASTRAN, is more common. A second consensus among these state-of-the-art aeroelastic solvers is the use of implicit schemes, which are unconditionally stable and allow for the use of large time steps for accurate cost-effective calculations. Significant efforts have also been committed in parallelizing these solvers to reduce the overall turnaround analysis time.

The primary goal of the thesis is to develop a cost-effective fluid-structure coupling algorithm which links an implicit finite element three-dimensional unsteady Navier-Stokes flow solver to a finite element stress solver for general fluid-structure interactions of rigid and elastic bodies moving in air, in view of aeroelastic calculations of internal and external flows. Such systems are to be simultaneously solved for the fluid and the structure in a fully coupled time-accurate fashion. In the process,

the issues to be addressed are *the importance of the structural deformations on the fluid solution* and *the need for a tight coupling between the two solvers to obtain a stable and time-accurate solution.* Innovative ideas are presented for *the smoothing of the displacements in the mesh movement algorithm*, for *the inclusion of the viscous shear stresses in the calculation of the aerodynamic forces*, and finally for *the transfer of the aerodynamic loads in a conservative fashion across the flow and the stress grids when the two grids do not necessarily match at the fluid-solid interface.*

At the basis of this work is the three-dimensional implicit finite element Navier-Stokes solver, FENSAP, developed at the CFD Laboratory by a team of research associates and graduate students. Substantial contributions of the author include the parallelization of the flow solver on shared-memory architectures, in addition to the implementation of the time-marching capabilities for the simulation of unsteady flows, followed by extensions to support moving grids for aeroelastic calculations. Furthermore, a rudimentary but complete stress code based on the Navier equations of linear elasticity was developed by the author. In some cases, simplified structural models described by systems of ordinary differential equations are used. Details about the two in-house solvers are given in chapters 2 and 3, with emphasis given to the aspects pertaining to the aeroelastic capabilities of the codes.

In the first issue to be addressed, the effects of the structural deformations on the flow solution will be studied, as opposed to neglecting the motion of the structure, albeit small, in the calculation of the unsteady aerodynamic forces. An approach based on the Arbitrary Lagrangian-Eulerian (ALE) formulation [39, 40, 41, 42] is selected in which the aerodynamic mesh adjusts itself dynamically, without remeshing, to the deforming structure. The implementation of the ALE formulation requires modifications to an existent unsteady Eulerian flow solver to add the mesh movement (ALE) terms to the governing equations and to provide a mechanism to move the

interior points of the grid such that it follows the motion of the structure. No re-interpolation of the flow solution is required since the time derivative in the equations for the fluid is modified to account for the time evolution of the flow quantities along the trajectory of the grid points, while maintaining the accuracy in space and time.

While the use of the ALE formulation is the most common approach used in the field of aeroelasticity for treating deforming structures, an alternative approach is the space-time weighted-residual Galerkin finite element formulation, popularized by Tezduyar *et al.* [43, 44, 45], in which the shape functions are taken to vary in space and time. Time differentiation is performed via the shape functions, as opposed to using semi-discrete multi-step or Runge-Kutta formulas for the temporal terms. The space-time discretization permits for time-dependent deformable elements, without explicitly adding terms in the equations (as for the ALE formulation). This concept is not investigated in this research for practical reasons. As mentioned previously, this work on aeroelasticity was initiated based on an already existing flow solver and it was concluded that the space-time formulation proposed by Tezduyar was not amenable to the structure and the philosophy of the FENSAP code. A comparative study of the two approaches would, however, be of great value.

Both approaches mentioned above require the implementation of a grid movement algorithm in order that the mesh for the air follows the movement of the structure. A possible simplification, which short-circuits the need for grid movement, is to impose "transpiration" velocity boundary conditions on the surface of the moving body, as given by the speed of the body, as would be done in the linearized potential flow theory for thin bodies (recall equation (1.1)). That is, instead of specifying the displacements on the body (and having to move the grid), a correction to the flow velocity of the fluid on the surface of the body is imposed to account for the deforming body. The approach has been applied to the Euler equations by modifying the surface normal, as

given by the structural dynamics solver and used in the calculation of the fluid fluxes at the wall, without having to actually deflect the surface and move the grid [46]. This approximation, which is suitable only for small displacements, is not extendible to general fluid-structure interactions for which the imposed motion of the structure can be of large amplitude. The approach is thus not considered in this work.

In the second issue to be addressed, a Navier-Stokes simulation is necessary to accurately represent the viscous flow phenomena present in turbomachines, such as vortical and separated flows, in order to quantify the importance of the viscous effects, including stall, on the flow and on the ensuing response of the blades. In addition, a Navier-Stokes solution accounts for the complete viscous stress tensor acting on the surface of the blades, as opposed to only considering the surface pressure forces, provided that these viscous shear stresses are fed to the stress solver. Although the technology will not be demonstrated in this thesis for applications in turbomachinery, the concepts hereby developed for viscous flows will still be directly applicable.

Finally, in the third issue, the tightness of the fluid-structure coupling is investigated. In early studies, time-dependent flow pressures on stationary grids were fed to a stress solver and the consequent aeroelastic behavior analyzed over several cycles of oscillations. While state-of-the-art independently developed flow and stress solvers were used without changes, there is, in such a loose coupling, no continuous two-way feedback between the fluid and the structure. To consider the effects of the deformation of the structure on the unsteady aerodynamic forces, the motion of the structure must be fed back to the flow solver, assuming capabilities to treat deforming structures, and new time-dependent aerodynamic forces can then be computed over several cycles. This process can be iterated; however, such a loose coupling, performed over a large time interval (several cycles of oscillations), requires that the unsteady calculations be repeated at every coupling instance, leading to prohibitive

computational costs, without guarantees that a time-accurate solution of the coupled system has been attained.

An implicit coupling algorithm is proposed in which the two systems are solved in a segregated fashion, but with the structural and the aerodynamic information exchanged in a tight two-way coupling as often as at every Newton iteration of the flow solver at each time step. Performed in this fashion, such a tight coupling with the stress solver incurs no significant additional costs to the fluid solver since the nonlinear fluid problem, solved using an implicit method, will converge in about the same number of iterations per time step, with or without coupling. Repeated calculations at the same time step are thus avoided by continuously updating the position of the structure while iterating on the nonlinear fluid problem, resulting in a very cost-efficient direct approach. Convergence of the two systems at each time step guarantees time-accurate solutions.

The coupling of the two solvers is controlled by the coupling driver – an external interface responsible for managing the execution of the flow and the stress solvers, on one hand, and transferring the aerodynamic loads and the structural displacements from one grid to the other, on the other hand. The two solvers thus do not communicate directly with one another, but do so through the intermediary of the driver. This segregated approach promotes the independent development of the two codes and their execution in serial or in parallel, subject to minor modifications to enable inter-code communications for efficiency purposes. The coupling driver is complemented by an interpolation module for the transfer of the aerodynamic loads and of the displacements across the grids. Unfortunately, algorithmic complexities can arise when interpolating data across non-matching grids and care must be taken to ensure conservation of the instantaneous total energy of the system. The generalization of the interpolation module using the concept of a virtual grid eliminates the need for

matching grids and also broadens the range of applications to structures of arbitrary complexity. Moreover, the use of unstructured grids is compatible with the use of mesh adaptation for enhancing the quality of the flow solution.

1.3 Outline of Thesis

The mathematical and the numerical models for the flow and the stress solvers are described in chapters 2 and 3, respectively. The fluid flow is modeled by the Euler or laminar Navier-Stokes equations based on the ALE formulation, discretized using the implicit weak-Galerkin finite element method on unstructured grids. Extensions to the flow solver to account for the structural displacements are detailed. They encompass the extension of the governing equations for the fluid using the ALE formulation, on one hand, and the grid movement scheme to smooth the CFD grid locally to conform to the deformations of the structure, on the other hand. Further capabilities to calculate the viscous shear stresses are implemented in the solver. All details about the flow solver are given in chapter 2.

The displacements are obtained by solving the Navier equations by the Rayleigh-Ritz method, assuming small displacements and a linear isotropic material. The time discretization is based on the $\beta-$Newmark method. The linear nature of the Navier equations allows for the use of a modal analysis based on the natural modes of vibration of the system, using the superposition principle. The modal analysis is a very efficient and attractive approach for time-marching analyses, since the governing partial differential equations can be reduced to a small system of decoupled ordinary differential equations. However, it requires the solution of a generalized eigenvalue problem to initially obtain the natural frequencies and the corresponding eigenvectors of the system. The details about the stress solver are provided in chapter 3.

Chapter 4 focuses on the aspects of the fluid-structure coupling. The coupling driver is a stand-alone code which invokes, as black boxes, the flow and the stress solvers. In particular, the coupling driver is responsible for synchronizing the action of the two solvers and for extracting the aerodynamic loads and the structural displacements on the surface of the structure and passing them as inputs to the solvers. Since the surface grids for the air and the structure usually do not match, the grid for the air being in general finer, the coupling driver is complemented by a surface interpolation module which interpolates, from one grid surface to the other, the structural displacements in a nodal fashion and transfers the aerodynamic loads in a face-based conservative manner.

Numerical examples are given in chapter 5 to demonstrate the capabilities of the algorithms and to verify the methodologies. Many of the configurations presented are two-dimensional, for practical reasons, given the long computational times. Three-dimensional results are shown for the case of wing flutter.

The contributions of the thesis are summarized in chapter 6, along with the conclusions. Future developments, beyond the scope of the thesis, are outlined in this final chapter.

Chapter 2

Description of the Flow Solver

The basic equations describing the flow model are presented in this chapter, as well as the numerical method employed for their discretization and their solution. Special attention is devoted to the aspects of the algorithms which are specific to fluid-structure interaction problems, such as the extension of the governing equations for the fluid, based on the Arbitrary Lagrangian-Eulerian formulation, the implementation of the mesh movement algorithm, and the calculation of the aerodynamic loads.

The model for the fluid is based on the three-dimensional compressible unsteady Navier-Stokes equations. Throughout this work, the appellation "Navier-Stokes" or "Euler" equations shall refer to the system consisting of the continuity, the momentum, and the energy equations. In the limit of negligible viscosity, the Euler equations are recovered and, in many cases, they are solved instead of the Navier-Stokes equations, given their lower computational costs. Despite the neglected viscous effects, the Euler equations can account for shock waves, which constitute the dominant nonlinear effect in predicting the transonic dip phenomenon of the flutter boundary curve. At this stage in the development of the flow solver, unsteady high-speed turbulent flow calculations are not available.

The general formulation of the flow solver FENSAP is given in section 2.1. The use of the Arbitrary Lagrangian-Eulerian formulation, which permits to include the effects of the deformation of the structure on the flow solution, is stated in section 2.2. In section 2.3, the stabilized weak-Galerkin finite element discretization of the Navier-Stokes equations is given. The flow solver is complemented by the grid movement algorithm, the focal point of section 2.4. Finally, in section 2.5, details are given relative to the calculation of the aerodynamic loads on the surface of the structure.

2.1 Flow Governing Equations

The compressible Navier-Stokes system of equations are non-dimensionalized by introducing the dimensionless variables

$$\begin{gathered} x' = \frac{x}{l_\infty}, \quad y' = \frac{y}{l_\infty}, \quad z' = \frac{z}{l_\infty}, \quad t' = \frac{tU_\infty}{l_\infty}, \\ p' = \frac{p}{\rho_\infty U_\infty^2}, \quad u' = \frac{u}{U_\infty}, \quad v' = \frac{v}{U_\infty}, \quad w' = \frac{w}{U_\infty}, \quad T' = \frac{T}{T_\infty}, \\ \rho' = \frac{\rho}{\rho_\infty}, \quad \mu' = \frac{\mu}{\mu_\infty}, \quad \kappa' = \frac{\kappa}{\kappa_\infty}, \quad C_p' = \frac{C_p}{C_{p_\infty}}, \end{gathered} \tag{2.1}$$

where l_∞ is the characteristic length of the physical problem and where the physical quantities U_∞, T_∞, ρ_∞, μ_∞, κ_∞, and C_{p_∞} are, respectively, the free stream flow speed, the temperature, the density, the dynamic viscosity, the thermal conductivity, and the specific heat capacity of air, at infinity. Omitting the primes $'$, the non-dimensional form of the equations becomes, in the Cartesian framework,

$$\begin{aligned} &\frac{d\rho}{dt} - \frac{d\mathbf{x}}{dt} \cdot \nabla\rho + (\rho v_j)_{,j} = 0 \\ &\frac{d(\rho v_i)}{dt} - \frac{d\mathbf{x}}{dt} \cdot \nabla(\rho v_i) + (\rho v_i v_j)_{,j} = -p_{,i} + \frac{1}{Re_\infty}(\tau_{ij})_{,j}, \quad i = 1,2,3, \\ &\rho C_p \left(\frac{dT}{dt} - \frac{d\mathbf{x}}{dt} \cdot \nabla T + v_j T_{,j} \right) = (\gamma_\infty - 1) M_\infty^2 \left(\frac{dp}{dt} - \frac{d\mathbf{x}}{dt} \cdot \nabla p + v_j p_{,j} \right) \\ &\qquad + \frac{1}{Re_\infty Pr_\infty}(\kappa T_{,j})_{,j} + \frac{(\gamma_\infty - 1) M_\infty^2}{Re_\infty} \Phi \end{aligned} \tag{2.2}$$

where

$$\tau_{ij} = \mu\left(v_{i,j} + v_{j,i}\right) + \lambda\delta_{ij}v_{k,k}, \qquad (2\mu + 3\lambda = 0), \tag{2.3}$$

$$\Phi = \mu\left[2\left(\epsilon_{xx}^2 + \epsilon_{yy}^2 + \epsilon_{zz}^2\right) + \left(\epsilon_{xz}^2 + \epsilon_{xy}^2 + \epsilon_{yz}^2\right)\right] + \lambda\left(\nabla\cdot\mathbf{v}\right)^2, \tag{2.4}$$

$$\epsilon_{ij} = \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}, \tag{2.5}$$

and the Mach, the Reynolds, and the Prandtl numbers are defined, respectively, as

$$M_\infty = \frac{U_\infty}{\sqrt{\gamma R T_\infty}}, \qquad Re_\infty = \frac{\rho_\infty l_\infty U_\infty}{\mu_\infty}, \qquad Pr_\infty = \frac{\mu_\infty C_{p_\infty}}{\kappa_\infty}. \tag{2.6}$$

The Euler equations are obtained in the limiting case when $Re_\infty \to \infty$, $\mu \to 0$, and $\kappa \to 0$.

The system is closed by the equation of state (non-dimensional)

$$p = \frac{\rho T}{\gamma_\infty M_\infty^2} \tag{2.7}$$

for an ideal gas. The thermodynamic properties of the gas are taken to vary with the temperature, according to Sutherland's laws for the dynamic viscosity (dimensional)

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{3/2}\left(\frac{T_\infty + 110.4\,\mathrm{K}}{T + 110.4\,\mathrm{K}}\right) \tag{2.8}$$

and the thermal conductivity (dimensional)

$$\frac{\kappa}{\kappa_\infty} = \left(\frac{T}{T_\infty}\right)^{3/2}\left(\frac{T_\infty + 133.7\,\mathrm{K}}{T + 133.7\,\mathrm{K}}\right). \tag{2.9}$$

The coefficient of specific heat capacity at constant pressure, $C_p(T)$, is obtained from tabulated data from Black and Hartley [47] for an ideal gas, as plotted in figure 2.1 in the range $100\,\mathrm{K} < \mathrm{T} < 1000\,\mathrm{K}$. The ratio of specific heats, $\gamma = C_p/C_v$, is obtained from C_p and $C_v = C_p - R$, where $R = 287\,\mathrm{J/(kg\cdot K)}$ is the gas constant.

This system consists of 9 equations in 9 variables: the pressure p, the velocity components v_1, v_2, v_3, the temperature T, the density ρ, the dynamic viscosity μ,

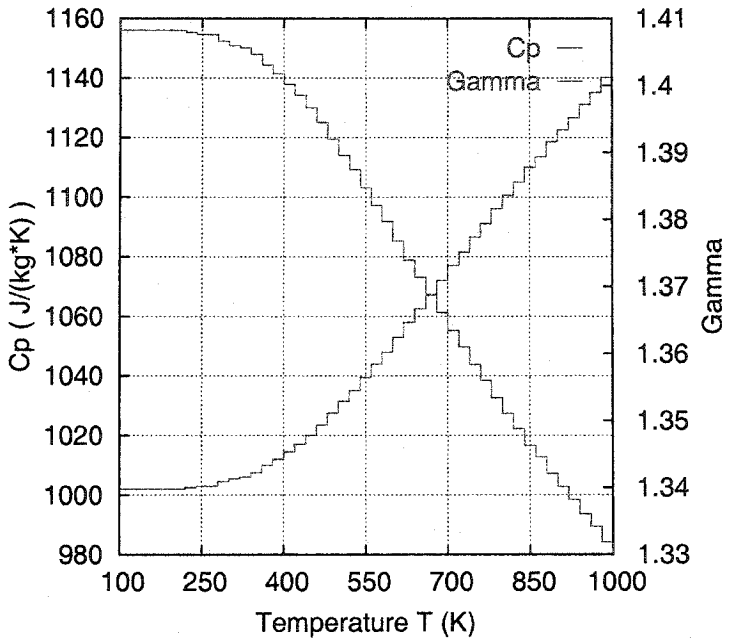


Figure 2.1: C_P and γ as Functions of Temperature T [47].

the thermal conductivity κ, and the specific heat at constant pressure C_p. In the solution process, the pressure, the velocity, and the temperature are first obtained; then, the thermodynamic variables ρ, μ, κ, and C_p are updated. In some special cases, such as steady inviscid flows without heat transfer, the total enthalpy may be taken as being constant throughout the flow field. In this case, the temperature is obtained from the definition of the total enthalpy instead of solving the energy equation. However, in aeroelastic applications and unsteady flows involving moving bodies, this assumption is not justified since energy is being exchanged between the fluid and the structure.

For flows characterized by subsonic inflow conditions, the free stream velocity and the temperature are imposed at the inlet and a static pressure is imposed at the far exit. For supersonic inflow conditions, all variables are imposed at the inlet. At a wall, either a temperature profile or a heat flux rate is imposed, in addition to the

no-slip condition for viscous flows or the flow tangency condition for inviscid flows.

The non-dimensionalization of the equations is important from the numerical point of view. Looking ahead, for an implicit discretization, a system of linear equations must be solved. The matrix being large and sparse, the system is solved using a preconditioned iterative solver of the conjugate gradient family. The effectiveness of the preconditioner, needed to provide convergence of the system, is closely related to the scaling of the variables. In particular, it is desired that the primary variables solved for, here p, v_1, v_2, v_3, and T, be of the same order of magnitude $\mathcal{O}(1)$.

Moreover, it is important to be aware of the non-dimensionalization of these variables in order to recover the correct physical quantities when extracting from the solution file the aerodynamic forces needed for the coupling with the stress solver. Similarly, the displacements must be converted to the proper units for the flow solver.

2.2 Arbitrary Lagrangian-Eulerian Formulation

The governing equations (2.2) for the fluid have been extended to account for moving grids using the Arbitrary Lagrangian-Eulerian (ALE) formulation [39, 40]. In essence, the partial derivative with respect to time of any quantity f has been replaced in the equations by the ALE time derivative, given by

$$\frac{\partial f}{\partial t} = \frac{df}{dt} - \frac{d\mathbf{x}}{dt} \cdot \nabla f, \tag{2.10}$$

where it is understood that $\frac{d}{dt}$ denotes the time derivative taken along the trajectory of a grid node moving at the speed $\frac{d\mathbf{x}}{dt}$.

It must be clearly understood, within the ALE framework, that the equations for the fluid are still defined in an inertial frame of reference. The moving/deforming

bodies are immersed in the fluid domain which is at rest relative to a far-field observer. For applications in turbomachinery, the use of the ALE formulation is applicable in the same fashion with $\frac{d\mathbf{x}}{dt}$ being the relative grid velocity in the rotating frame of reference.

By taking the grid velocity field $\frac{d\mathbf{x}}{dt}$ equal to the fluid velocity $\mathbf{v}$, the classical Lagrangian description of flow is obtained. The purely Lagrangian framework is deficient in the sense that the mesh can become highly tangled when the nodes move at the velocity of the fluid following the trajectories of the fluid particles. On the other hand, $\frac{d\mathbf{x}}{dt} = 0$ on a stationary grid and the Eulerian description of flow is recovered. In the Eulerian framework, on moving meshes, the domain must be remeshed and the solution must be interpolated at every time step to account for the deformation of the computational fluid domain which conforms to the moving structure. In the ALE formulation, the nodes on the body move at the speed of the moving and deforming body, with the interior grid points moving in an arbitrary fashion, but in such a way as to prevent mesh tangling. Further, the flow variables are advanced in time along the nodal trajectories without the need for an explicit interpolation step.

In finite volume implementations, the new term $\frac{d\mathbf{x}}{dt} \cdot \nabla f$ is lumped with the convective term $\nabla \cdot (f\mathbf{v})$ to take the form $\nabla \cdot \left[f \left(\mathbf{v} - \frac{d\mathbf{x}}{dt} \right) \right]$ ($f = \rho$ for continuity, $f = \rho\mathbf{v}$ for momentum, $f = \rho H$ for energy). Although a source vector arises from this formulation, this formulation is required for the conservative calculation of the unsteady fluxes on the moving grid, with the fluid moving through the cell at a relative convective velocity $\mathbf{v} - \frac{d\mathbf{x}}{dt}$. In the present finite element context, the modification to the time derivative using equation (2.10) provides a simpler implementation.

Finally, the boundary conditions of the flow solver are extended to account for moving boundaries. For the Navier-Stokes equations, the no-slip boundary condition $\mathbf{v} = 0$ at a wall becomes $\mathbf{v} = \frac{d\mathbf{x}}{dt}$. For the Euler equations, the flow tangency

boundary condition $\mathbf{v} \cdot \mathbf{n} = 0$ (no-penetration) becomes $\mathbf{v} \cdot \mathbf{n} = \frac{d\mathbf{x}}{dt} \cdot \mathbf{n}$. In the weak-Galerkin finite element formulation, described in the next section, the latter condition is naturally imposed via the contour integral of the continuity equation. Any deformation or motion of the structure is thus reflected in the flow boundary conditions and will be accounted for in the flow solution.

2.3 Finite Element Formulation

A semi-segregated solution approach is adopted for the solution of the primitive variables p, v_1, v_2, v_3, T. This choice of solution variables facilitates the imposition of velocity and pressure boundary conditions at the inlet and the exit, in comparison to using the conserved variables ρ, ρv_1, ρv_2, ρv_3, ρe, where e is the energy of the fluid. The continuity and the momentum equations are first solved together in a strongly coupled fashion for p, v_1, v_2, v_3. The temperature T is then obtained from the energy equation, either from the assumption of constant total enthalpy (valid when the flow is steady with no moving bodies or when there is no heat transfer) or by solving the full energy equation using the weak-Galerkin finite element method. Finally, the thermodynamic variables ρ, C_p, μ, and κ are obtained from the equation of state for an ideal gas, tabulated data for C_p as a function of T, and Sutherland's laws for the dynamic viscosity and the thermal conductivity, respectively.

The weak-Galerkin finite element formulation is obtained by integrating the governing equations with respect to a weight function W, taken to be the same as the shape functions N interpolating the solution, and integrating the spatial terms by parts using the divergence theorem. This effectively transfers the order of differentiation from the spatial terms (convective and diffusive) to the weight function, hence allowing for the use of lower-order elements. In this context, linear shape functions

are used in space, but since equal order of interpolation is used for all nodal degrees of freedom, artificial viscosity is added to the system for stabilization purposes. In particular, the regularization of the continuity equation by the pressure allows the continuity and the momentum equations to be solved in a coupled manner in terms of the pressure and the velocity when the solution to the energy equation is lagged. The transformed integral equations are:

Continuity:

$$\begin{aligned} &-\int_{\mathcal{V}} W\left\{\frac{d\rho}{dt}-\frac{dx_j}{dt}\frac{\partial\rho}{\partial x_j}\right\}d\mathcal{V}+\int_{\mathcal{V}} W_{,j}\rho v_j d\mathcal{V}-\int_{\mathcal{S}} W\rho v_j n_j d\mathcal{S}= \\ &\int_{\mathcal{V}} W_{,j}\left\{h^{\alpha}\left[\left(\epsilon_1^p+\epsilon_2^p\right)p_{,j}-\epsilon_2^p\hat{p}_{,j}\right]\right\}d\mathcal{V}-\int_{\mathcal{S}} W\left\{h^{\alpha}\left[\left(\epsilon_1^p+\epsilon_2^p\right)p_{,j}-\epsilon_2^p\hat{p}_{,j}\right]n_j\right\}d\mathcal{S} \end{aligned} \tag{2.11}$$

Momentum, $i=1,\ldots,3$*:*

$$\begin{aligned} &-\int_{\mathcal{V}} W\left\{\frac{d\rho v_i}{dt}-\frac{dx_j}{dt}\left(\rho\frac{\partial v_i}{\partial x_j}+v_i\frac{\partial\rho}{\partial x_j}\right)\right\}d\mathcal{V} \\ &+\int_{\mathcal{V}} W_{,j}\left\{\rho v_j v_i+\delta_{ij}p-\frac{\mu}{Re_\infty}\left(v_{i,j}+v_{j,i}-\frac{2}{3}\delta_{ij}v_{k,k}\right)\right\}d\mathcal{V} \\ &-\int_{\mathcal{S}} W\left\{\rho v_j v_i+\delta_{ij}p-\frac{\mu}{Re_\infty}\left(v_{i,j}+v_{j,i}-\frac{2}{3}\delta_{ij}v_{k,k}\right)\right\}n_j d\mathcal{S}= \\ &\int_{\mathcal{V}} W_{,j}\left\{h^{\alpha}\left[\left(\epsilon_1^v+\epsilon_2^v\right)v_{,j}-\epsilon_2^v\hat{v}_{,j}\right]\right\}d\mathcal{V}-\int_{\mathcal{S}} W\left\{h^{\alpha}\left[\left(\epsilon_1^v+\epsilon_2^v\right)v_{,j}-\epsilon_2^v\hat{v}_{,j}\right]n_j\right\}d\mathcal{S} \end{aligned} \tag{2.12}$$

Energy:

$$\begin{aligned} &-\int_{\mathcal{V}} W\left\{\rho C_p\left[\frac{dT}{dt}+\left(v_j-\frac{dx_j}{dt}\right)T_{,j}-(\gamma_\infty-1)M_\infty^2\left(\frac{dp}{dt}+\left(v_j-\frac{dx_j}{dt}\right)p_{,j}\right)\right]\right\}d\mathcal{V} \\ &-\int_{\mathcal{V}}\left\{\frac{\kappa}{Re_\infty Pr_\infty}W_{,j}T_{,j}+\frac{(\gamma_\infty-1)M_\infty^2}{Re_\infty}W\Phi\right\}d\mathcal{V}+\int_{\mathcal{S}}\frac{\kappa}{Re_\infty Pr_\infty}WT_{,j}n_j d\mathcal{S}= \\ &\int_{\mathcal{V}} W_{,j}\left\{h^{\alpha}\left[\left(\epsilon_1^T+\epsilon_2^T\right)T_{,j}-\epsilon_2^T\hat{T}_{,j}\right]\right\}d\mathcal{V}-\int_{\mathcal{S}} W\left\{h^{\alpha}\left[\left(\epsilon_1^T+\epsilon_2^T\right)T_{,j}-\epsilon_2^T\hat{T}_{,j}\right]n_j\right\}d\mathcal{S} \end{aligned} \tag{2.13}$$

The ALE time derivative $\frac{df}{dt}$ of a dependent variable is taken along the trajectory of the moving node; for example, the first-order implicit discretization is

$$\frac{df}{dt}^{n+1} = \frac{f(\mathbf{x}^{n+1}, t^{n+1}) - f(\mathbf{x}^n, t^n)}{t^{n+1} - t^n}. \tag{2.14}$$

High-order implicit discretizations in time are straightforward to construct, since no interpolation of the solution on the meshes at previous time levels is required.

The 1^{st}, 2^{nd}, and 3^{rd} order formulas are summarized in table 2.1 using variable time steps, although uniform time steps are most commonly used. In practice, only the 2^{nd} order formula is used, since the 1^{st} order one is too diffusive and the 3^{rd} order one is highly dispersive. The same formulas are employed in calculating both the time derivatives of the physical flow variables (ρ, p, v_i, T) and the grid velocity $\frac{d\mathbf{x}}{dt}$.

In the stabilization terms,

$$\nabla \hat{g} = \sum_{j=1}^{ndp} N_j \left(\nabla \hat{g}\right)_j \tag{2.15}$$

where

$$\left(\nabla \hat{g}\right)_j = \frac{1}{\tilde{V}_j} \sum_{je=1}^{supp(j)} V_{je} \sum_{k=1}^{ndp} \nabla N_k|_{cent(je)}\, g_k^{je} \tag{2.16}$$

for any flow variable g. The coefficients ϵ_1^p, ϵ_1^v, ϵ_1^T, ϵ_2^p, ϵ_2^v, and ϵ_2^T control the amount of first- and second-order numerical dissipation in the continuity, momentum, and energy equations, respectively. The term h^α represents a local length scale h of the element, raised to the power α ($\alpha = 0$ or 1). The formulation allows for varying artificial viscosity coefficients, using a detector based on solution gradients to switch from first- to second-order numerical dissipation [48].

In the solution strategy, the discretized governing equations are solved iteratively, using Newton's method, in terms of the incremental changes Δp, Δv_1, Δv_2, Δv_3,

Order	$f'(t^{n+1})$
1	$\frac{f^{n+1}-f^n}{\Delta t} + \mathcal{O}(\Delta t)$
2	$\frac{(3+\alpha)}{(2+\alpha)}\frac{f^{n+1}}{\Delta t} - \frac{(2+\alpha)}{(1+\alpha)}\frac{f^n}{\Delta t} + \frac{1}{(2+\alpha)(1+\alpha)}\frac{f^{n-1}}{\Delta t} + \mathcal{O}\left((2+\alpha)\Delta t^2\right)$
3	$\frac{11+7\alpha+3\beta+\alpha^2+\alpha\beta}{(2+\alpha)(3+\alpha+\beta)}\frac{f^{n+1}}{\Delta t} - \frac{(2+\alpha)(3+\alpha+\beta)}{(1+\alpha)(2+\alpha+\beta)}\frac{f^n}{\Delta t}$ $+\frac{3+\alpha+\beta}{(2+\alpha)(1+\alpha)(1+\beta)}\frac{f^{n-1}}{\Delta t} - \frac{2+\alpha}{(3+\alpha+\beta)(2+\alpha+\beta)(1+\beta)}\frac{f^{n-2}}{\Delta t}$ $+\mathcal{O}\left((2+\alpha)(3+\alpha+\beta)\Delta t^3\right)$

Table 2.1: Implicit Multi-Step Time Discretization Formulas: $\Delta t = t_{n+1} - t_n$, $t_n - t_{n-1} = (1+\alpha)\Delta t$, $t_{n-1} - t_{n-2} = (1+\beta)\Delta t$.

ΔT of the solution. From the Newton iteration k to $k+1$, at time t^{n+1}, the solution is advanced according to:

$$\begin{aligned} p_{(k+1)}^{n+1} &= p_{(k)}^{n+1} + \Delta p, \\ v_{1(k+1)}^{n+1} &= v_{1(k)}^{n+1} + \Delta v_1, \\ v_{2(k+1)}^{n+1} &= v_{2(k)}^{n+1} + \Delta v_2, \\ v_{3(k+1)}^{n+1} &= v_{3(k)}^{n+1} + \Delta v_3, \\ T_{(k+1)}^{n+1} &= T_{(k)}^{n+1} + \Delta T. \end{aligned} \tag{2.17}$$

Once the solution has converged at the time step $n+1$, after a certain number of Newton iterations $k = 1, 2, \ldots$, the solution is advanced to the next time step $n+2$. The importance of iterating within a time step comes from the fact that the Navier-Stokes equations are nonlinear and that the scheme is implicit. This iterative

strategy will be exploited during the coupling of the CFD code with the stress solver (see chapter 4).

The *local* influence matrix for an element, after the Newton linearization of the fully coupled fluid problem, takes the form

$$\begin{pmatrix} C_p & C_{v_1} & C_{v_2} & C_{v_3} & C_T \\ M_{v_1,p} & M_{v_1,v_1} & M_{v_1,v_2} & M_{v_1,v_3} & M_{v_1,T} \\ M_{v_2,p} & M_{v_2,v_1} & M_{v_2,v_2} & M_{v_2,v_3} & M_{v_2,T} \\ M_{v_3,p} & M_{v_3,v_1} & M_{v_3,v_2} & M_{v_3,v_3} & M_{v_3,T} \\ E_p & E_{v_1} & E_{v_2} & E_{v_3} & E_T \end{pmatrix} \begin{pmatrix} \Delta p \\ \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \\ \Delta T \end{pmatrix} = - \begin{pmatrix} r_p \\ r_{v_1} \\ r_{v_2} \\ r_{v_3} \\ r_T \end{pmatrix}. \tag{2.18}$$

The grid velocity term $\frac{d\mathbf{x}}{dt} \cdot \nabla f$ is omitted in the construction of the Jacobian matrix arising from the Newton linearization, since it has minimal effects on the global convergence rate. It is emphasized that this new term $\frac{d\mathbf{x}}{dt} \cdot \nabla f$ is included in the residuals, so neglecting it in the matrix affects only the convergence rate and not the accuracy. The coefficients of the influence matrix, as well as supplementary details about the discretization procedure, are available in the FENSAP documentation [49].

The right-hand-side vector represents the residuals of the equations, which, when driven to zero, give the solution to the stabilized discrete equations of the weak-Galerkin problem (including artificial dissipation). The common solution procedure consists of gradually decreasing the coefficients of artificial dissipation, using the solution obtained at one cycle as the restart solution for the next cycle. The ultimate goal is to minimize the amount of numerical viscosity added to the original equations, such as to approach the true physical solution of these equations, which has no artificial dissipation.

In the semi-segregated strategy, the pressure and the velocity field are first ob-

tained by solving the global system

$$\begin{pmatrix} K_p & K_{v_1} & K_{v_2} & K_{v_3} \\ K_{v_1,p} & K_{v_1,v_1} & K_{v_1,v_2} & K_{v_1,v_3} \\ K_{v_2,p} & K_{v_2,v_1} & K_{v_2,v_2} & K_{v_2,v_3} \\ K_{v_3,p} & K_{v_3,v_1} & K_{v_3,v_2} & K_{v_3,v_3} \end{pmatrix} \begin{pmatrix} \Delta p \\ \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \end{pmatrix} = - \begin{pmatrix} r_p \\ r_{v_1} \\ r_{v_2} \\ r_{v_3} \end{pmatrix} \tag{2.19}$$

resulting from the assembly of the elements. Next, the temperature is obtained by solving the system

$$(K_T)(\Delta T) = -(r_T). \tag{2.20}$$

It must be noted that the coefficients of the approximate Jacobian matrix in the $\Delta p - \Delta v$ system are derived assuming a frozen total enthalpy state, in accordance with the assumption of constant total enthalpy, even when the energy equation is solved in terms of the temperature. Such a linearization does not seem to adversely affect the global convergence rate of the nonlinear problem when the energy equation is solved in a segregated manner.

The linear systems which arise from the assembled systems (2.19) and (2.20) yield large sparse matrices that are solved using a preconditioned Generalized Minimal Residual (GMRES) iterative solver, with the choice of the preconditioner depending upon the problem at hand. In time-marching applications, a diagonal preconditioner often proves to be a cost-effective strategy since the presence of the unsteady terms, proportional to $1/\Delta t$ on the main diagonal of the matrix, favorably enhances the spectral properties of the matrix and thus improves the convergence rate of the linear solver. An Incomplete LU factorization (ILU) can also be used as a preconditioner, but the extra robustness of the ILU preconditioner is in general not needed for time-marching problems with small time-steps. In comparison to a diagonal preconditioner, the ILU preconditioner requires much more memory for its storage and the application of the preconditioner by the inversion of the LU triangular blocks is costlier.

For a steady-state solution, as stated previously, the overall solution strategy consists of solving the system of discretized equations by driving the residuals to zero while gradually reducing the amount of artificial viscosity. This steady-state solution is then used as the starting solution for the unsteady flow problem, for a moving body problem or for an aeroelastic configuration, using the smallest artificial viscosity coefficient at the last cycle of the steady-state solution.

The accuracy of the solution can be enhanced by adapting the computational grid. On unstructured grids, using triangular (in 2-D) or tetrahedral (in 3-D) elements, the mesh adaptation tools developed at the CFD Laboratory provide mesh refinement and coarsening, edge swapping, and nodal displacements [50, 51, 52]. The mesh adaptation process is external to the flow solver and is invoked between artificial viscosity cycles during the calculation of the steady-state solution. After several solution-adaptation cycles, an adapted and eventually optimal grid can be found for the steady problem. However, this process being a static one, new difficulties are encountered when applying the technology to unsteady flows. First, the cost of external mesh adaptation is quite high, since the changes in the topology and the connectivity of the grid at every time step require the CFD code to repeat its initialization. Second, errors are introduced from the interpolation of the solution on the previous grid to the newly adapted grid. While these errors are washed out during the steady-state mesh adaptation process, by repeated solution-adaptation steps, they can only be washed out in an unsteady approach by determining the optimal grid at every time step. This process would require several steps of adaptation within a single time step. Lastly, the interpolation of the solution across grids at different time levels, needed in order to apply high-order multi-step time-integration formulas as listed in table 2.1, introduces errors in the solution and can prevent time accuracy.

In some of the applications presented in chapter 5, mesh adaptation has been

used for the steady-state solution only. When the displacements of the structure are small, the grid adapted for the steady-state solution can remain adequate for accurate aeroelastic calculations. Despite that it is not optimal at every instant in time, the adapted grid still yields, at a cheaper cost, much more accurate results than an unadapted one.

2.4 Grid Movement Algorithm

The role of the mesh movement algorithm is to adjust the computational grid for the fluid such that it follows smoothly the deformation of the structure, as depicted in figure 2.2. Clearly, without mesh movement of the interior grid points, the elements near a moving wall would deform and, eventually, the presence of highly distorted or inverted elements would lead to the breakdown of the flow solver. Also, from the calculated movement of the grid, the nodal velocity field $\frac{d\mathbf{x}}{dt}$ is computed before being fed into the governing equations (2.2) for the fluid.

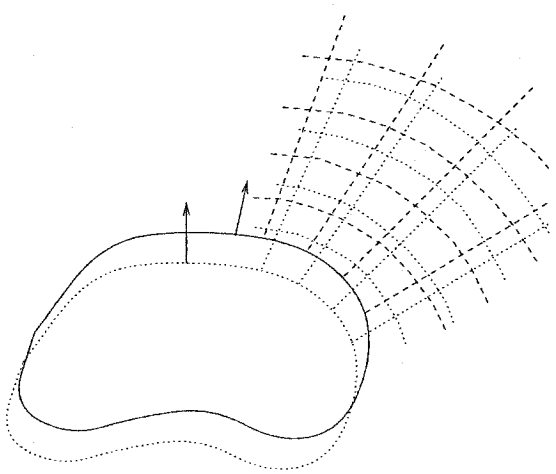


Figure 2.2: Representation of the Movement of the Grid Following the Motion of a Body.

A number of grid movement strategies are reported in the literature. The mesh is moved either using an algebraic mapping, using a smoothing operator, or using global remeshing or unsteady mesh adaptation. The use of an algebraic mapping, for example a rigid-body rotation or a translation, is very efficient, but is specific to simple geometries and known motions. The use of global remeshing or unsteady mesh adaptation is, naturally, the most robust technique, but for three-dimensional complex geometries, the cost of these operations is simply prohibitive, since any change in the topology of the mesh requires a full re-initialization of the data structures of the implicit flow solver. Moreover, global remeshing requires the interpolation of the solution from the old grid to the new one, thus leading to possible losses in the conservation of the flow variables (mass, momentum, energy). Nonetheless, it is acknowledged that global remeshing is the most robust approach and that it has been successfully used with large deformations [36, 37], but its cost cannot be justified in the context of small displacements in aeroelasticity.

Neither of the two options discussed above are optimal, from the point of view of generality and efficiency, and a strategy based on a grid smoothing operator is exploited and extended in the present work. In essence, the displacements in the interior of the fluid domain are smoothed by solving Laplace's equation $\nabla^2 u = 0$ for each of the three components of the displacements, in a buffer zone around the body (see figure 2.3) [53]. The boundary conditions are of the Dirichlet type. The grid displacements are prescribed on the surface of the moving body, as given by the structural code. Zero displacements are imposed in the "far-field", away from the body. Once the displacements have been obtained over the entire grid, the new position of the nodes is simply determined as $\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{u}$.

The idea of grid smoothing using a Laplacian operator as in the above is not new. Indeed, Anderson *et al.* [54] give an introductory overview of the concepts of elliptic

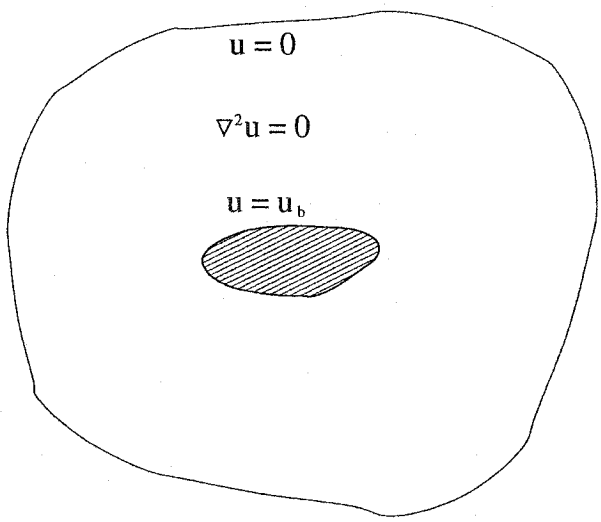


Figure 2.3: One-Zone Mesh Movement Strategy.

grid generation using Laplace's equation. An algebraic, yet totally equivalent, counterpart to the differential approach is the spring analogy [55], which is, in principle, an explicit Jacobi point-relaxation discretization of Laplace's equation consisting of averaging the displacements around each node until convergence.

The use of Laplacian smoothing retains the original topology of the grid. Unlike global remeshing or unsteady mesh adaptation, there are no incurred losses in accuracy due to solution interpolation from the previous to the next grid since there is simply no solution interpolation step in the proposed smoothing scheme. The Laplacian smoothing is generally applicable to the most complex geometries, including multiple bodies, without increasing the level of difficulty, even in three dimensions. Finally, it is emphasized that the smoothing operations are performed on the displacements. The concentration of grid points and the stretching of cells resulting from anisotropic mesh adaptation are not destroyed by the mesh movement smoothing operations. The explanation is simple: the Laplacian smoother is applied to the relative nodal displacements, not to the absolute nodal positions.

In this work, the mesh movement algorithm is embedded inside the flow solver, as opposed to implementing it from within the fluid-structure coupling driver. In this way, the communication costs across the coupling driver and the flow solver can be minimized, since only the structural displacements at the nodes on the surface of the structure are passed to the flow solver, rather than passing the displacements at all the nodes of the fluid grid. In addition, the mesh movement algorithm can directly benefit from the parallelism strategies implemented in the flow solver. In contrast, the coupling driver requires little computing resources and can run on a single-processor work station.

In the same spirit as the flow solver FENSAP, an implicit finite element discretization of the Laplacian operator is preferred for its robustness, as opposed to using an explicit approach. Perhaps one drawback of the use of the differential approach for mesh smoothing is the fact that a large system of equations must be solved. However, since each component of the displacement vector is solved separately, the memory storage for the matrices is not significant and, in fact, the memory from the implicit Navier-Stokes solver can be reused. Moreover, the resulting matrices are symmetric positive definite and a classical conjugate-gradient solver with diagonal preconditioning suffices to solve the systems efficiently.

For an insightful physical interpretation of elliptic grid smoothing, an analogy can be drawn to the Navier equations of linear elasticity for small displacements, given as:

$$\nabla^2 u + \frac{1}{1-2\nu}\frac{\partial \epsilon}{\partial x} = 0, \quad \nabla^2 v + \frac{1}{1-2\nu}\frac{\partial \epsilon}{\partial y} = 0, \quad \nabla^2 w + \frac{1}{1-2\nu}\frac{\partial \epsilon}{\partial z} = 0, \quad (2.21)$$

where $\epsilon = \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z}$ and ν is the Poisson ratio. If the displacement field is "divergence free" ($\epsilon = \nabla \cdot \mathbf{u} = 0$), in analogy to an incompressible fluid, the system reduces to solving Laplace's equation for each of the three, now decoupled,

components u, v, w of the displacement field, akin to treating the air as an isotropic linear elastic medium. This is exactly the smoothing procedure previously outlined.

For small displacements, typical of aeroelastic deformations, the above strategy, termed a *one-zone* mesh movement strategy, is generally adequate to smooth the displacements without introducing mesh degeneracies. However, even if the displacements indeed appear small relative to the reference length of the structure, they can be quite large relative to the local length scale of the grid, in particular for elements near a wall in high Reynolds number flow calculations or near a stagnation point or singularity in Euler flows. For large displacements, absolute or relative, the one-zone approach is deficient and soon leads to mesh degeneracies usually occurring near salient corners of the geometry, such as at a sharp trailing edge, as seen in figure 2.4. Further, nonlinearities are introduced by repeatedly smoothing the grid on the displaced grid. As illustrated for a pitching airfoil, after several periods of oscillations of the periodic motion, the grid can become highly distorted near the trailing edge, making it unsuitable for accurate flow calculations (see figure 2.4c).

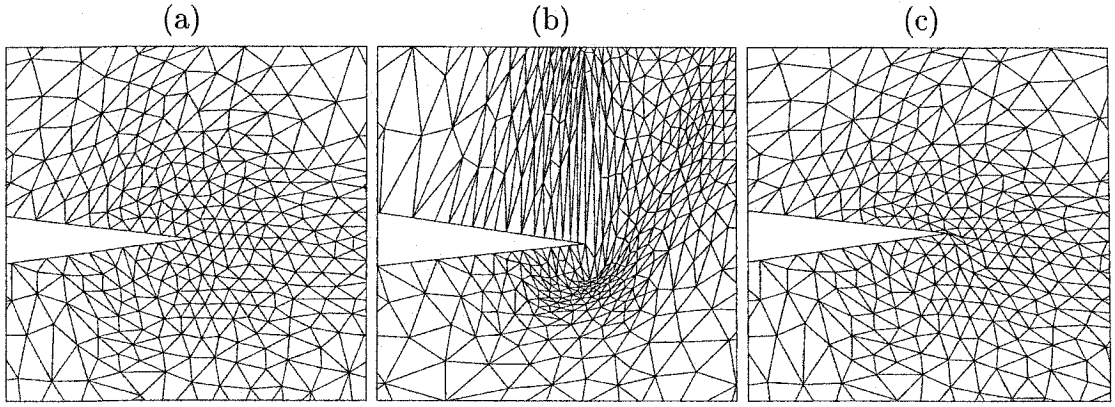


Figure 2.4: One-Zone Mesh Smoothing: Zoom of the Trailing Edge of an Airfoil: (a) Original Grid; (b) At Angle of Incidence of about 4°; (c) After 4 Cycles of Oscillations.

The need for a more robust node movement scheme has clearly been identified in the literature. For example, Mittal and Tezduyar [56] surrounded a simple moving body by a small region of elements moving rigidly with the body, the *inner zone*, and then applied regular smoothing outside this buffer zone, the *outer zone*. The motion was, however, restricted to simple rotations and translations, in two dimensions, described by known algebraic mappings.

A novel generalization [57] of the two-zone smoothing approach is proposed in the hope to prevent the premature breakdown of the flow solver due to mesh degeneracies, when working with complex three-dimensional configurations. In an inner zone immediate to the body, defined by all nodes within a certain distance from the body, see figure 2.5, Laplace's equation is solved with the displacements specified on the body, as before, but with Neumann boundary conditions $\frac{\partial u}{\partial n} = 0$ on the external boundary of the inner zone (dashed line in figure 2.5). This is approximately a "no-shear" boundary condition whose effect is to essentially move the grid points rigidly with the body, but with minimal shear. The Neumann boundary condition is trivially incorporated into the variational approach by neglecting a contour integral in the weak-Galerkin formulation. In the outer zone, constructed from layers of elements growing around the inner zone, as determined by the connectivity of the grid (structured or unstructured), grid smoothing is performed as in the single-zone approach, with Dirichlet boundary conditions on all boundaries. The displacements are specified at the interface between the inner and the outer zones from the solution obtained in the inner zone and with zero displacements imposed in the far-field.

The extent of the inner and outer zones may not span the entire grid. In fact, for efficiency purposes, it is not desired to move the grid in the entire fluid domain, but to restrict the smoothing operations to as small a domain as possible around the moving structure. (In the implementation, the size of each zone is specified by the user.) In

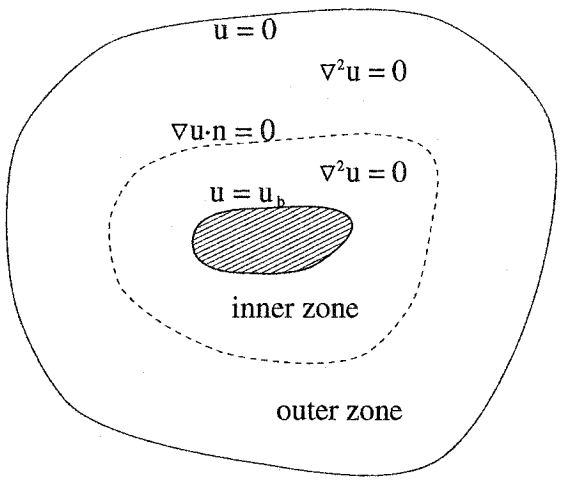


Figure 2.5: Two-Zone Mesh Movement Strategy.

the proposed two-zone approach, two problems of smaller size are solved as opposed to one problem of larger size in the single-zone approach. This makes the two-zone approach slightly more efficient computationally than the one-zone approach, since the work required grows as $\mathcal{O}(N^3)$, where N is the number of moving nodes. Overall, the cost of mesh movement is of the order of one or two percent of the cost of one Newton iteration of the flow solver (for Laplacian smoothing only).

As a variation of the two-zone approach, the Navier equations (2.21) are solved in the inner zone with true no-shear boundary conditions $\boldsymbol{\tau} \cdot \mathbf{n} = 0$ on the interface between the inner and the outer zones. The choice of Poisson's ratio for the grid is arbitrary, and a value of $\nu = 0.30$ is used by default. (The success of this approach depends strongly on the new boundary conditions and mildly on the value of ν.) This mesh movement strategy proved to successfully smooth the displacements in all attempted cases, but at a much higher cost than the decoupled approach using Laplace's equation, since the three components of the displacements must now be solved in a coupled manner. The resulting matrix arising from the discretization is still symmetric positive definite, but unlike the Laplacian solver, an ILU preconditioner

(specifically, Cholesky, without fill-ins) is more cost-effective to solve the new system than a diagonal preconditioner.

Another alteration of the two-zone approach consists of solving the equation

$$\nabla \cdot (k \nabla u) = 0 \tag{2.22}$$

in the outer zone, with

$$k(x,\, y,\, z) = 1 + \frac{1}{d_{tol} + d(x,\, y,\, z)} \tag{2.23}$$

representing a grid stiffness, where d is the minimum distance from a point $(x,\, y,\, z)$ to the nearest wall and d_{tol} is a small tolerance. The exact choice of equation (2.23) is not important, as long as it is a monotonically decaying function. Near the body, the stiffness is large and the motion is mostly rigid. Far from the body, the stiffness is nearly unity and the displacements decay smoothly to zero. That is, the large relative displacements are effected away from the body where the cells are large, while the high stiffness near the body helps to preserve the shape and quality of the small cells.

The enhanced robustness of the two-zone grid-moving procedure is attested by imposing a large amplitude motion to an oscillating airfoil. The inner zone contains all nodes within a distance $0.2c$ from the surface of the airfoil, of chord c; the outer zone is composed of 14 element connectivity levels (see figure 2.6). The initial grid and the grid at the maximum amplitude are plotted together in figure 2.7. Although the details of the grid are not discernible on this scale, the displacements are indeed smooth and no mesh distortion occurs. Needless to say, the one-zone approach fails radically in this case as it cannot sustain deflections of such magnitude.

Equal success has been obtained in three-dimensions for a flexible wing in bending and in torsion. The deflection of a straight wing with a parabolic airfoil section is shown in figure 2.8a for the first torsion mode (second natural mode of vibration).

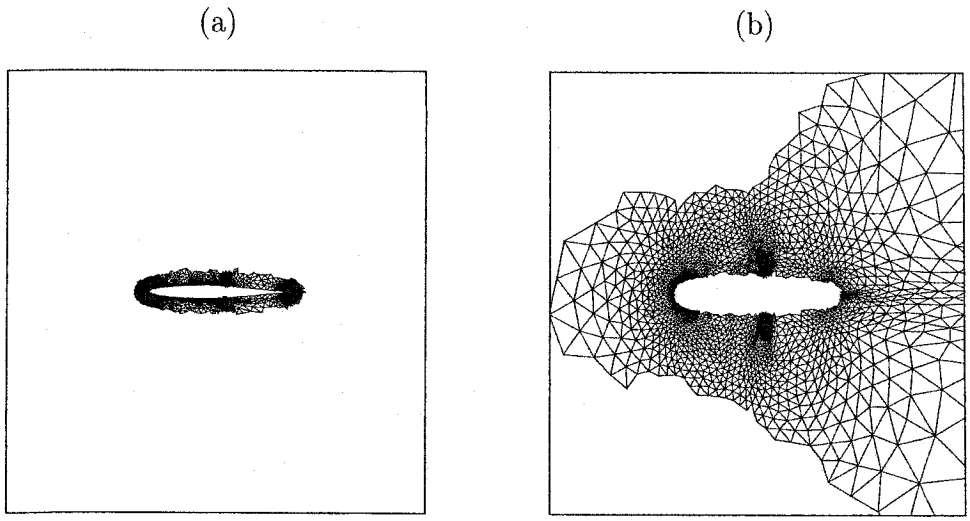


Figure 2.6: Two-Zone Grid Movement Around an Oscillating Airfoil: (a) Inner zone; (b) Outer zone.

The deflection is amplified to illustrate the extent of the movement supported by the two-zone grid movement technology. Cuts of the grid are shown in figure 2.8b, where it can be seen that the grid has moved smoothly following the motion of the wing, although the stretched elements near the sharp leading and trailing edges of the wing indicate that any larger motion would soon exceed the limitations of the method (Laplacian smoothing in the inner zone). The use of the Navier equations in the inner zone eliminates all mesh distortions near the wing tip. In practice, however, the structural deflections are much smaller and the viability of the mesh movement algorithm would not be jeopardized. Further, the use of a finer grid near the surface of the wing would increase the smoothness of the grid motion.

From the conducted numerical experiments, the use of the grid stiffness damping proved to be indispensable in three dimensions where cell distortion can occur in all three directions. Its usage prevents, more precisely, the collapse of cells in the outer zone at the border with the inner zone. In two dimensions, such a treatment was not

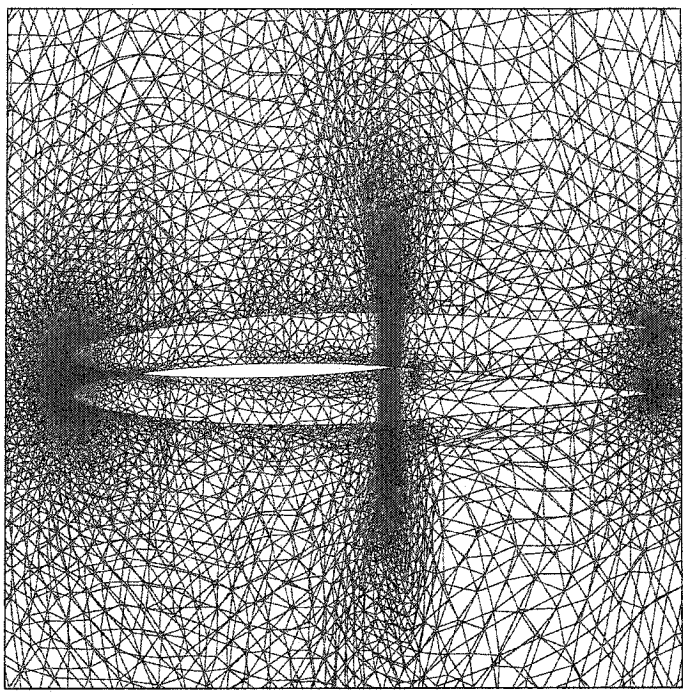


Figure 2.7: Two-Zone Grid Movement Around an Oscillating Airfoil: Initial and Final Grids Superposed ($k = 1$).

required for the grid motions considered. Moreover, with this type of mesh movement in the outer zone, it no longer becomes necessary to solve the more expensive Navier system in the inner zone to obtain smooth displacements. In fact, by a judicious choice of the stiffness function k, it should be possible to eliminate the need for the two-zone approach and use the simpler one-zone approach in most cases.

As a last comment about the advantages of the two-zone Laplace smoothing algorithm, the future considerations to turbulence modeling using wall functions cannot be neglected. At this point, it is believed that the two-zone mesh movement approach is favorable in maintaining a suitable y^+ value at the wall since the mesh is moved mostly rigidly in the inner zone. Thus, any initial suitable y^+ distribution should not be destroyed.

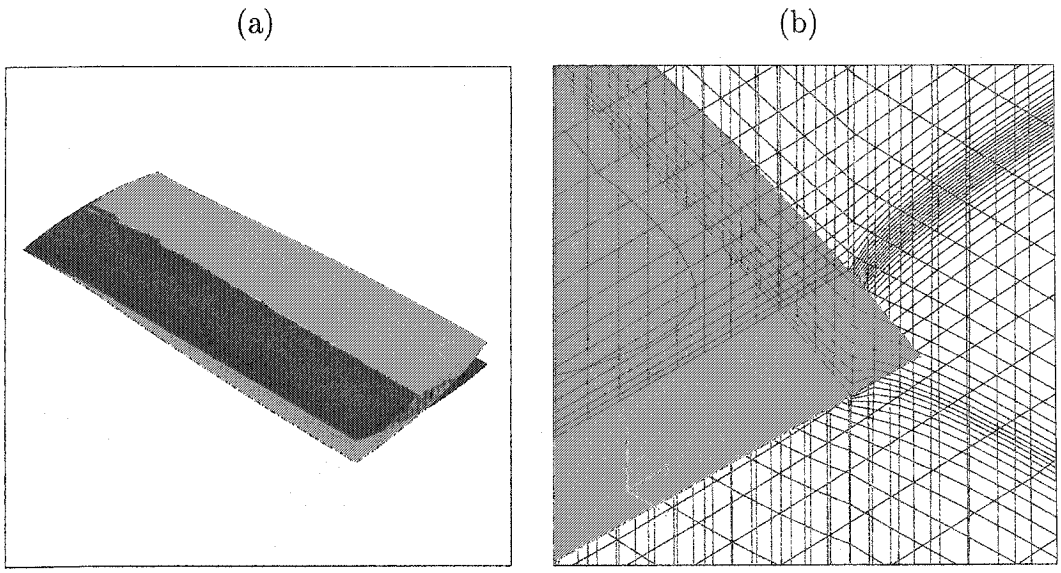


Figure 2.8: Two-Zone Grid Movement Around a Wing in Torsion: (a) Initial and Final Positions of the Wing; (b) Cuts of the Grid Near the Tip of the Twisted Wing.

Finally, the grid velocity field $\frac{d\mathbf{x}}{dt}$ must be obtained from the displacement field, as needed in the governing equations (2.2). As the ALE acronym suggests, the grid movement should be arbitrary. However, in a series of papers, Farhat *et al.* suggest that the discretization of the governing equations may restrict the way the mesh velocity is computed [58, 32]. More precisely, the fundament of the Geometric Conservation Law (GCL) states that a constant uniform flow field must satisfy the discretized equations. If this condition appears trivial, it is not always so at the discrete level, in particular in the context of the finite volume method when unsteady fluxes must be in equilibrium on the moving control volumes (see [58, 32]). The GCL thus restricts the way the grid can be moved and, in addition, failing to satisfy the GCL can lead to small high-frequency fluctuations/oscillations about the mean unsteady flow. To satisfy the GCL, one computes a consistent grid velocity by dividing the volume

swept by a control volume by the size of the time step, while the motion (position) of the grid itself is still arbitrary. However, satisfying the GCL not only restricts the way the grid speed is calculated, but also imposes restrictions on the time-stepping scheme, thus inhibiting (or complicating) the use of common multi-step high-order time-integration schemes.

In the present finite element formulation, a quick investigation of the ALE derivative in equation (2.10) confirms that the GCL is always satisfied for a constant flow ($\nabla f = 0$) and thus the grid velocity $\frac{d\mathbf{x}}{dt}$ and the time-integration scheme can indeed be chosen arbitrarily. Whence, backward multi-step formulas in time are used to compute the grid velocity field, even on deforming boundaries where $\frac{d\mathbf{x}}{dt}$ could be extracted from the stress solution instead. High-order formulas can be employed in a straightforward fashion if copies of the grid are stored over a few time levels. While a first-order approximation for $\frac{d\mathbf{x}}{dt}$ could suffice and would be more economical, since the grid would be stored at only one time level, a formula consistent with the time-integration scheme (same order) is used by default.

2.5 Calculation of the Aerodynamic Loads

The aerodynamic loads $\{\mathbf{t}\} = [\tau]\mathbf{n}$ are calculated over the wall boundary faces from the definition of the stress tensor

$$\tau_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \delta_{ij}\frac{2}{3}\frac{\partial u_k}{\partial x_k}\right). \tag{2.24}$$

The stress tensor is evaluated inside the flow solver, since the velocity field is needed on the surface of the body and inside the fluid domain to compute the derivatives of the velocity in equation (2.24). If computed in a post-processing step, outside the flow solver, the entire grid and solution must be processed, as opposed to processing

only the data on the surface of the body. The in-house development of the flow solver, in this work, easily allows for the calculation of the full stress tensor at minimal cost.

The calculation of the stress tensor from equation (2.24) gives rise to certain complications. The use of linear shape functions in the finite element solution for the pressure and the velocity yields continuous nodal pressures, but discontinuous nodal derivatives for the velocity and the stress components, since the derivatives $\frac{\partial u_j}{\partial x_i}$ are discontinuous at element boundaries. To circumvent this problem, nodal values of the stress tensor can be obtained by averaging the one-sided values in the elements surrounding a node, weighted by the area of these elements. However, integrating the averaged stresses, linearly interpolated over the faces, will not yield the same loads as if the stress tensor were integrated over the faces.

To overcome this difficulty, the components of the resultant loads $t_i = \tau_{ij} n_j$ are integrated over each face of the body, giving a consistent face-based piece-wise constant load over each face. The components t_x, t_y, and t_z of the loads (per unit area) on a face f are defined as

$$\begin{aligned} t_x^{(f)} &= \frac{1}{\mathcal{S}^{(f)}} \int_{\mathcal{S}^{(f)}} (\tau_{xx} n_x + \tau_{xy} n_y + \tau_{xz} n_z)\, d\mathcal{S} \\ t_y^{(f)} &= \frac{1}{\mathcal{S}^{(f)}} \int_{\mathcal{S}^{(f)}} (\tau_{yx} n_x + \tau_{yy} n_y + \tau_{yz} n_z)\, d\mathcal{S} \\ t_z^{(f)} &= \frac{1}{\mathcal{S}^{(f)}} \int_{\mathcal{S}^{(f)}} (\tau_{zx} n_x + \tau_{zy} n_y + \tau_{zz} n_z)\, d\mathcal{S} \end{aligned} \tag{2.25}$$

where (n_x, n_y, n_z) is the unit normal vector pointing outward from the surface of the structure (inward the fluid domain). The evaluation of the loads in this way makes it possible to transfer the aerodynamic loads in a conservative fashion from the flow solver to the stress solver.

The representation of the aerodynamic loads by a constant function over a face raises a question of accuracy. Ideally, it would be desirable to assign a piece-wise continuous linear distribution of the loads, for both the pressure and the shear stresses,

on the surface of the structure, but such that the forces are conserved globally. The pressure is already defined in a continuous fashion by the use of linear shape functions and causes no difficulty. A consistent linear reconstruction of the shear stresses is proposed in the work of Gresho [59], consisting of evaluating the shear stresses from solving an auxiliary system based on the surface integrals of the weak-Galerkin problem, but these ideas will not be elaborated in this work, since the only viscous test case presented in the thesis does not require such a reconstruction. As it will be seen in chapter 5, for the oscillating cylinder test case, only the total lift on the cylinder is required as a "generalized force" acting on the cylinder. The lift, being calculated inside the flow solver, is integrated over all faces based on the computed solution for the linearly distributed pressure and velocity. A reconstruction of the loads on a face by face basis is thus not needed in this case since the loads are applied globally rather than locally. One can however object to such a statement and argue that the use of "consistent" shear stresses, computed à la Gresho, would improve the accuracy of the calculation of the total lift in the flow solver. Experience gained in the calculation of the heat fluxes $Q = -\kappa \nabla T \cdot \mathbf{n}$ using Gresho's "consistent" reconstruction, as opposed to using the definition of Q, proves the contrary for low Reynolds number flows. For such "mild" flows, the temperature profile is linear in the boundary layer and the gradient ∇T is evaluated just as accurately using either method. Since the velocity profile is also linear in the boundary layer, as is the temperature, there is no indication that Gresho's approach would be more accurate in the case of laminar flow, provided that the flow is well-resolved on a sufficiently fine grid. The simplicity of using equation (2.24) directly prevails, at the moment. However, for high Reynolds number turbulent flows, characterized by strong velocity and temperature gradients in the boundary layer, Gresho's approach could prove to be more accurate. In particular, when using wall elements in the turbulence model, the evaluation of the first derivatives in equation (2.24) requires the differentiation of

the specialized logarithmic shape functions in the wall element.

The face-based representation of the loads as used in this work is not totally optimal for Euler flows when only the pressure intervenes. Why use piece-wise face-based loads for the pressure when it is linearly interpolated? A node-based transfer of the loads using a linear distribution should prove to be more accurate than a piece-wise constant distribution of the pressure, even though the total loads on the structure remain the same. The extension of the node-based representation of the aerodynamic loads remains for the future, as a special case for inviscid flow.

One main objective of this thesis — to provide the mechanisms to study the effects of the viscous stress tensor for Navier-Stokes calculations — has thus been fulfilled. A closer look at the viscous effects will be given in chapter 5 for the case of an oscillating cylinder at $Re = 200$. Based on physical intuition, one is led to postulate that the viscous shear stresses would aid to damp the motion of the structure.

Chapter 3

Description of the Stress Solver

The Navier equations of linear elasticity for an isotropic material are used as the stress model, based on the assumption that the structural deformations of the flexible structure are small. Thermal effects in the structure are not modeled at the present. For the external flow of a wing in flutter, or for a fan or a compressor blade, the thermal effects may be rightfully neglected, but in a turbine, in which the blades are cooled, thermal effects must be incorporated in the analysis.

The Navier equations of linear elasticity are discretized using the Rayleigh-Ritz finite element formulation. For the class of problems studied, a modal analysis is adopted, integrating the equations in time using the β−Newmark scheme. The description of the discretization and of the solution algorithm using the modal analysis are given in section 3.1. Additionally, a 2-D thin plate model using specialized elements is presented for the structural analysis of the wing flutter case in chapter 5. The details of this model are presented in section 3.2.

Particular attention is given to the evaluation of the aerodynamic loads during the coupling with the flow solver.

3.1 Isotropic 3-D Linear Stress Model

The finite element formulation of the stress code is based on the energy formulation. The total energy is

$$I = \frac{1}{2}\int_{\mathcal{V}} \rho\{\dot{\mathbf{u}}\}^T\{\dot{\mathbf{u}}\}d\mathcal{V} + \frac{1}{2}\int_{\mathcal{V}} \{\epsilon\}^T[C]\{\epsilon\}\, d\mathcal{V} + \int_{\mathcal{V}} \{\mathbf{F_B}\}^T\{\mathbf{u}\}d\mathcal{V} - \int_{\mathcal{S}} \{\mathbf{t}\}^T\{\mathbf{u}\}d\mathcal{S} \quad (3.1)$$

where, for an isotropic material,

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{Bmatrix} \qquad [C] = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (3.2)$$

and $\lambda = \dfrac{E\nu}{(1+\nu)(1-2\nu)}$, $\mu = \dfrac{E}{2(1+\nu)}$, E is the modulus of elasticity, and ν is Poisson's ratio. Simplifications give

$$\{\epsilon\}^T[C]\{\epsilon\} = \lambda\left(\epsilon_{xx}+\epsilon_{yy}+\epsilon_{zz}\right)^2 + 2\mu\left(\epsilon_{xx}^2+\epsilon_{yy}^2+\epsilon_{zz}^2\right) + \mu\left(\gamma_{yz}^2+\gamma_{zx}^2+\gamma_{xy}^2\right), \quad (3.3)$$

where the components of strain are:

$$\begin{aligned} \epsilon_{xx} &= \frac{\partial u}{\partial x}, & \epsilon_{yy} &= \frac{\partial v}{\partial y}, & \epsilon_{zz} &= \frac{\partial w}{\partial z}, \\ \gamma_{yz} &= \frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}, & \gamma_{zx} &= \frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}, & \gamma_{xy} &= \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}, \end{aligned} \quad (3.4)$$

in Cartesian coordinates. The third term in the expression for the total energy (3.1) represents the work done by the body forces $\{\mathbf{F_B}\}$, with gravity and centrifugal forces due to rotation being examples. Body forces are ignored in this work. The surface integral in I represents the work done on the structure by the external forces, in

occurrence the aerodynamic forces, where $t_i = \tau_{ij} n_j$ are the components of the stress tensor $\mathbf{t}$ acting along each of the coordinate axes.

The system is discretized using the Rayleigh-Ritz finite element method, by taking the variation of the energy functional, yielding a linear system of the form

$$[M]\{\ddot{\mathbf{u}}\} + [D]\{\dot{\mathbf{u}}\} + [K]\{\mathbf{u}\} = \{S\}, \tag{3.5}$$

where $[M]$ is the mass matrix, $[D]$ is the damping matrix, $[K]$ is the stiffness matrix, and $\{S\}$ is the vector of external forces. The damping matrix is zero in this system, but dynamical and mechanical damping, of the form $[D] = \alpha_M[M] + \alpha_K[K]$, can be added to the system.

At this point, it is essential to clearly state the form of the external forces vector $\{S\}$, since it is at the basis of the mechanism by which the coupling with the fluid solver is effected. Labeling the work done on the structure by the external forces as $I_E = -\int_{\mathcal{S}} \{\mathbf{t}\}^T \{\mathbf{u}\} d\mathcal{S}$, and writing the displacements as

$$u(\mathbf{x},t) = \sum_j N_j(\mathbf{x}) u_j(t), \quad v(\mathbf{x},t) = \sum_j N_j(\mathbf{x}) v_j(t), \quad w(\mathbf{x},t) = \sum_j N_j(\mathbf{x}) w_j(t), \tag{3.6}$$

the local vector of forces for one element with boundary face $\mathcal{S}^{(s)}$ is

$$\{S^{(s)}\} = \begin{Bmatrix} \dfrac{\partial I_E}{\partial u_j} \\ \dfrac{\partial I_E}{\partial v_j} \\ \dfrac{\partial I_E}{\partial w_j} \end{Bmatrix} = \begin{Bmatrix} \int_{\mathcal{S}^{(s)}} t_x N_j d\mathcal{S} \\ \int_{\mathcal{S}^{(s)}} t_y N_j d\mathcal{S} \\ \int_{\mathcal{S}^{(s)}} t_z N_j d\mathcal{S} \end{Bmatrix}. \tag{3.7}$$

The global vector $\{S\}$ is constructed by summing the contributions $\{S^{(s)}\}$ over all stress faces $s = 1, \ldots, n_s$. The shape functions N_j are defined on the surface of the structure and correspond to the shape functions used in the Rayleigh-Ritz discretization of the stress model.

The components t_x, t_y, and t_z of the loads (per unit area) are defined as $\{\mathbf{t}\} = [\tau]\mathbf{n}$, where $\mathbf{n}$ is the unit normal vector pointing outward from the surface of the structure (inward the fluid domain) and

$$\tau_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \delta_{ij}\frac{2}{3}\frac{\partial u_k}{\partial x_k}\right) \tag{3.8}$$

is the aerodynamic stress tensor, as described in section 2.5. However, here, the loads which are defined on a fluid face $\mathcal{S}^{(f)}$ must be integrated over a stress face $\mathcal{S}^{(s)}$. Complete details about the transfer of the loads in a conservative manner from the fluid grid to the stress grid will be given in chapter 4.

The system of ordinary differential equations (3.5) are integrated in time using the second-order accurate $\beta-$Newmark method [60]. This implicit multi-step scheme is derived by using a central finite difference formula for $\ddot{\mathbf{u}}$ and $\dot{\mathbf{u}}$ about t^n and averaging $\mathbf{u}$ and $\mathbf{S}$ over the three time steps t^{n-1}, t^n, t^{n+1} :

$$\begin{aligned}
\ddot{\mathbf{u}}^n &= \frac{\mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1}}{\Delta t^2}, \\
\dot{\mathbf{u}}^n &= \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n-1}}{2\Delta t}, \\
\mathbf{u}^n &= \frac{1}{3}\left(\mathbf{u}^{n+1} + \mathbf{u}^n + \mathbf{u}^{n-1}\right), \\
\mathbf{S}^n &= \frac{1}{3}\left(\mathbf{S}^{n+1} + \mathbf{S}^n + \mathbf{S}^{n-1}\right).
\end{aligned} \tag{3.9}$$

The resultant scheme is:

$$\begin{aligned}
\left(\frac{[M]}{\Delta t^2} + \frac{[D]}{2\Delta t} + \frac{[K]}{3}\right)\mathbf{u}^{n+1} &= \left(\frac{2[M]}{\Delta t^2} - \frac{[K]}{3}\right)\mathbf{u}^n + \left(-\frac{[M]}{\Delta t^2} + \frac{[D]}{2\Delta t} - \frac{[K]}{3}\right)\mathbf{u}^{n-1} \\
&\quad + \frac{1}{3}\left(\mathbf{S}^{n+1} + \mathbf{S}^n + \mathbf{S}^{n-1}\right).
\end{aligned} \tag{3.10}$$

The first time step is advanced using a first-order scheme, since the $\beta-$Newmark method is not self-starting. With

$$\ddot{\mathbf{u}}^0 = \frac{2(\mathbf{u}^1 - \mathbf{u}^0 - \dot{\mathbf{u}}^0\Delta t)}{\Delta t^2}, \tag{3.11}$$

the startup scheme becomes

$$\left(2\frac{[M]}{\Delta t^2}\right)\mathbf{u}^1 = \left(\frac{2[M]}{\Delta t^2} - [K]\right)\mathbf{u}^0 + \left(2\frac{[M]}{\Delta t} - [D]\right)\dot{\mathbf{u}}^0 + \mathbf{S}^0, \tag{3.12}$$

where $\dot{\mathbf{u}}^0$ is the initial velocity of the body (assumed to be zero in this context). The direct solution to equation (3.10) requires the inversion of a large sparse system of linear equations at each time step. An iterative solver of the conjugate-gradient family provides an effective solution approach.

Since the equations (3.5) are linear, a *modal analysis* can be used in which the solution is decomposed into modes of the form

$$\begin{aligned} u(\mathbf{x},t) &= U(\mathbf{x})e^{i\beta t}, \\ v(\mathbf{x},t) &= V(\mathbf{x})e^{i\beta t}, \\ w(\mathbf{x},t) &= W(\mathbf{x})e^{i\beta t}. \end{aligned} \tag{3.13}$$

The natural frequencies β are obtained by solving the generalized eigenvalue problem

$$-\beta^2[M] + [K] = 0 \tag{3.14}$$

corresponding to the homogeneous, undamped system (3.5). By the nature of the mass matrix $[M]$ and the stiffness matrix $[K]$, the eigenvalues of the system are all real and positive.

Introducing the generalized coordinates $\{\boldsymbol{\xi}\}$, where $\{\mathbf{u}\} = [P]\{\boldsymbol{\xi}\}$, and multiplying equation (3.5) on the left by the matrix $[P]^T$ of the normalized eigenvectors of equation (3.14), the system reduces to N decoupled ordinary differential equations:

$$\begin{gathered} [P]^T\left([M]\{\ddot{\mathbf{u}}\} + [D]\{\dot{\mathbf{u}}\} + [K]\{\mathbf{u}\}\right) = [P]^T\{\mathbf{S}\} \\ [I]\{\ddot{\boldsymbol{\xi}}\} + \left(\alpha_M[I] + \alpha_K[\Lambda]\right)\{\dot{\boldsymbol{\xi}}\} + [\Lambda]\{\boldsymbol{\xi}\} = [P]^T\{\mathbf{S}\} \end{gathered} \tag{3.15}$$

for the N chosen modes, where the normalized eigenvectors satisfy the orthogonality properties $[P]^T[M][P] = [I]$ and $[P]^T[K][P] = [\Lambda]$ ($[\Lambda]$ is the diagonal matrix

of eigenvalues β^2). These decoupled ordinary differential equations are integrated in time using the β–Newmark scheme (3.10), which, in the modal analysis case, simplifies to:

$$\begin{aligned}\left(\frac{[I]}{\Delta t^2}+\frac{\alpha_M[I]+\alpha_K[\Lambda]}{2\Delta t}+\frac{[\Lambda]}{3}\right)\boldsymbol{\xi}^{n+1} &= \left(\frac{2[I]}{\Delta t^2}-\frac{[\Lambda]}{3}\right)\boldsymbol{\xi}^{n} \\ &+ \left(-\frac{[I]}{\Delta t^2}+\frac{\alpha_M[I]+\alpha_K[\Lambda]}{2\Delta t}-\frac{[\Lambda]}{3}\right)\boldsymbol{\xi}^{n-1} \\ &+ \frac{1}{3}[P]^T\left(\mathbf{S^{n+1}}+\mathbf{S^{n}}+\mathbf{S^{n-1}}\right).\end{aligned} \tag{3.16}$$

Since the matrices $[I]$ and $[\Lambda]$ are diagonal, the ordinary differential equations (3.16) are decoupled and can be solved without requiring the inversion of a large sparse matrix.

In practice, only a few modes — the resonant modes at a particular regime — may be studied at one time. (For a wing flutter analysis, flutter normally occurs when the first bending mode coalesces with the first torsional mode.) This makes the modal analysis a very attractive and effective method since no matrix inversion is required. In a direct transient analysis, the solution to the system of equations (3.10) requires the inversion of a large sparse symmetric matrix. Such a linear system can be best solved using an iterative solver of the conjugate gradient family. The direct transient analysis is thus much more expensive than the modal analysis for the time-integration of the solution. However, the modal analysis requires the determination of the generalized eigenvalues and eigenvectors of the system (3.14). In practice, only the first few eigenvalues, the smallest ones, need to be determined since higher-order modes are unlikely to be excited by flow-induced forces. Further, to capture these high-frequency modes, a very small time-step would have to be used — several orders of magnitude smaller than the time-step used for the first natural mode. In such a case, the benefits from using an implicit scheme are lost.

The generalized eigenvalue solver [61] used in this work has been developed jointly by Mark Jones (mtj@vt.edu), at Virginia Tech, and by Merrell Patrick, at Duke University. The package, written in FORTRAN, was retrieved from the Web site http://www.netlib.org/lanz. This solver was found particularly attractive for its usage of sparse matrix technology, based on a variant of the sparse matrix Compressed Sparse Row (CSR) storage format for symmetric matrices, especially for large three-dimensional configurations where the costs of the matrix storage can be prohibitive. The use of a node reordering scheme, for example the reversed Cuthill-McKee algorithm, proved essential in minimizing the bandwidth of the matrices in order to reduce the workspace required by the eigenvalue solver. The algorithm for the solution of the large sparse symmetric generalized eigenproblem $Kx = \lambda Mx$ is based on Lanczos' method and is capable of finding all eigenvalues in a given range or near some specified value. These options are convenient to find a number of eigenvalues near zero, which represent the low-frequency modes of interest for aeroelastic analyses. Eigenvectors are also computed as part of the solution.

The eigenmodes can also be determined by an external code, for example using NASTRAN [60]. The eigenvalues and eigenvectors are fed into the equation (3.16) and the integration is performed with the supplied external aerodynamic forces. This approach offers a complete control on the time-integration loop to establish an efficient tight coupling loop with the flow solver (see chapter 4), which would otherwise not be possible with NASTRAN alone. This procedure also provides a means by which to validate the current work using NASTRAN.

3.1.1 Natural Frequencies of a Cantilever Beam

The natural frequencies of a cantilever beam (see figure 3.1) are computed as a basic validation of the stress solver developed in this thesis. The test case is taken from

Petyt [62], section 5.13.

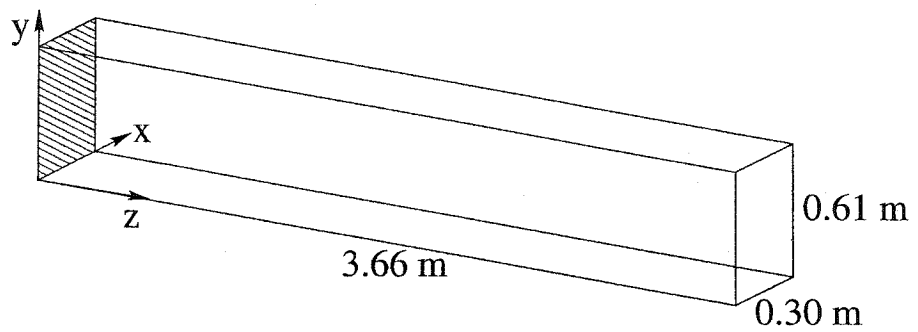


Figure 3.1: Geometry of a Cantilever Beam.

The material properties of the beam are $E = 2.068 \times 10^{11}\,\mathrm{N/m^2}$, $\nu = 0.30$, and $\rho = 8058\,\mathrm{kg/m^3}$. The calculated frequencies are tabulated in table 3.1 for the first three modes of vibration (mode 1: first bending in the $x-$direction; mode 2: first bending in the $y-$direction; mode 3: second bending in the $x-$direction). The table confirms the potential of the stress solver to reproduce the results obtained by Petyt on the coarse $3 \times 6 \times 12$ grid using linear hexahedral elements with uniform mesh spacing. It can also be observed from the results that the frequencies approach the *exact* solution as the grid is refined. (The *exact* solution, referenced in Petyt [62], has been obtained using slender beam theory.)

Although the frequencies can be estimated rather accurately using a large number of linear elements, the cost of using excessively large grids becomes prohibitive when solving a generalized eigenvalue problem. The cost-effectiveness of quadratic elements is demonstrated in table 3.2, which gives a comparison of the accuracy achieved using 8-node linear versus 20-node quadratic hexahedral elements: a $12 \times 24 \times 48$ grid using linear elements (15925 degrees of freedom) is required to yield the accuracy achieved using a $2 \times 3 \times 6$ grid using quadratic elements (275 degrees of freedom). For any

	Mesh densities $(nx \times ny \times nz)$				
Mode	$3\times6\times12$	$6\times12\times24$	$12\times24\times48$	Petyt $(3\times6\times12)$	Exact
1	21.94	19.35	18.62	22.0	18.6
2	38.77	37.25	36.82	38.3	37.3
3	134.68	117.93	113.29	135.3	116.8

Table 3.1: Calculated Natural Frequencies (Hz) of a Cantilever Beam for Different Meshes Using Linear Hexahedral Elements.

practical application, the large bandwidth of the mass and stiffness matrices quickly exceeds the capabilities of current computers if linear interpolation is used. Quadratic interpolation is a must for accuracy and efficiency.

	Mesh densities $(nx \times ny \times nz)$				
	Quadratic $2\times3\times6$ 275 d.o.f.		Linear $3\times6\times12$ 364 d.o.f.		
Mode	this work	Petyt	this work	Petyt	Exact
1	18.56	18.6	21.94	22.0	18.6
2	36.98	36.5	38.77	38.3	37.3
3	113.64	114.3	134.68	135.3	116.8

Table 3.2: Comparison of Calculated Natural Frequencies (Hz) of a Cantilever Beam Using Linear and Quadratic Hexahedral Elements.

3.2 Orthotropic 2-D Thin Plate Model

The thin plate model used to analyze the AGARD wing 445.6 (see chapter 5) is described in this section. The basic assumptions of the thin plate theory, stated in

Petyt [62], are that $\sigma_z = 0$ (stress in transverse direction is zero) and that the normal to the median plane remains normal to the median plane during the deformation:

$$\begin{aligned} u(x,y,z) &= -z\frac{\partial w}{\partial x}, \\ v(x,y,z) &= -z\frac{\partial w}{\partial y}, \end{aligned} \tag{3.17}$$

where $w(x,y)$ is the displacement relative to the median plane in the $z-$direction. The plate is subject to loads in the direction normal to the median plane.

With the above assumptions, the components of strain become:

$$\begin{aligned} \epsilon_{xx} = -z\frac{\partial^2 w}{\partial x^2}, \qquad \epsilon_{yy} = -z\frac{\partial^2 w}{\partial y^2}, \qquad \epsilon_{zz} = \frac{\partial w}{\partial z}, \\ \gamma_{yz} = 0, \qquad \gamma_{zx} = 0, \qquad \gamma_{xy} = 0 \end{aligned} \tag{3.18}$$

and the strain energy simplifies to

$$U = \frac{1}{2}\int_{\mathcal{V}} \{\chi\}^T [D]\{\chi\} d\mathcal{V}, \tag{3.19}$$

where

$$\{\chi\} = -z \begin{Bmatrix} w_{xx} \\ w_{yy} \\ 2w_{xy} \end{Bmatrix} \tag{3.20}$$

and

$$[D] = \begin{bmatrix} E_x & \nu_{xy}E_x & 0 \\ \nu_{xy}E_x & E_y & 0 \\ 0 & 0 & G_{xy} \end{bmatrix} \tag{3.21}$$

for an orthotropic material. The kinetic energy of the plate element is

$$T = \frac{1}{2}\int_{\mathcal{V}} \rho\dot{w}^2 + \rho z^2 \left(\dot{w}_x^2 + \dot{w}_y^2\right) d\mathcal{V}. \tag{3.22}$$

The work done by the external forces is

$$I_E = \int_{\mathcal{S}} t_z w - z\left(t_x w_x + t_y w_y\right) d\mathcal{S}. \tag{3.23}$$

The presence of second derivatives for w in the strain energy U requires high-order interpolation functions in the finite element discretization. A 4-node rectangular plate element is defined, with w, w_x, and w_y defined at each node, using Hermite interpolation functions of the form

$$\begin{aligned} N_i &= c_0^i + c_1^i x + c_2^i y + c_3^i x^2 + c_4^i xy + c_5^i y^2 + \\ &\quad c_6^i x^3 + c_7^i x^2 y + c_8^i xy^2 + c_9^i y^3 + c_{10}^i x^3 y + c_{11}^i xy^3 \end{aligned} \tag{3.24}$$

for $i = 1, \ldots, 12$. The 12 coefficients c_k^i in each of the 12 shape functions are obtained by building and solving 12×12 linear systems arising from evaluating N_i, $\frac{\partial N_i}{\partial x}$, and $\frac{\partial N_i}{\partial y}$, for $i = 1, \ldots, 12$, at the nodes (x_j, y_j), $j = 1, \ldots, 4$. These linear systems are inverted numerically, working directly in terms of global coordinates (x, y) to conveniently compute second derivatives $\frac{\partial^2 N_i}{\partial x^2}$, $\frac{\partial^2 N_i}{\partial x \partial y}$, and $\frac{\partial^2 N_i}{\partial y^2}$ without transformation from local to global coordinates. This is the *non-conforming* plate element, labeled as such since w_x and w_y may not be continuous across the sides of the element, although they are continuous at the nodes. A *conforming* element can be built by introducing w_{xy} as a fourth nodal degree of freedom, to ensure inter-element continuity, but there is no significant gain in accuracy [63].

The vertical displacement is represented as

$$w = \sum_{i=1}^{12} N_i \alpha_i \tag{3.25}$$

where the degrees of freedom are

$$\{\boldsymbol{\alpha}\} = \{w_1, w_{1,x}, w_{1,y}, \ldots w_4, w_{4,x}, w_{4,y}\}. \tag{3.26}$$

The components of the mass matrix and the stiffness matrix are, respectively:

$$\begin{aligned} M_{ij} &= \int_{\mathcal{V}} \rho N_i N_j + \rho z^2 \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) d\mathcal{V} \\ K_{ij} &= \int_{\mathcal{V}} z^2 \left\{ D_{11}\frac{\partial^2 N_i}{\partial x^2}\frac{\partial^2 N_j}{\partial x^2} + D_{12}\left(\frac{\partial^2 N_i}{\partial x^2}\frac{\partial^2 N_j}{\partial y^2} + \frac{\partial^2 N_i}{\partial y^2}\frac{\partial^2 N_j}{\partial x^2} \right) + \right. \\ &\quad \left. D_{22}\frac{\partial^2 N_i}{\partial y^2}\frac{\partial^2 N_j}{\partial y^2} + D_{33}\frac{\partial^2 N_i}{\partial x \partial y}\frac{\partial^2 N_j}{\partial x \partial y} \right\} d\mathcal{V} \end{aligned} \tag{3.27}$$

for $i, j = 1, \ldots, 12$. The source term vector of aerodynamic forces is:

$$S_i = \int_{\mathcal{S}} \left\{ t_z N_i - z \left(t_x \frac{\partial N_i}{\partial x} + t_y \frac{\partial N_i}{\partial y} \right) \right\} d\mathcal{S}. \tag{3.28}$$

In usual implementations of the model, the plate element is assumed to have a constant thickness h and integration in the $z-$direction is performed analytically beforehand. However, in this work, in view of the 3-D coupling with the CFD code, nodal values of the thickness are considered, giving a continuous structure (see figure 3.2). In this case, numerical integration is carried out in the vertical $z-$direction.

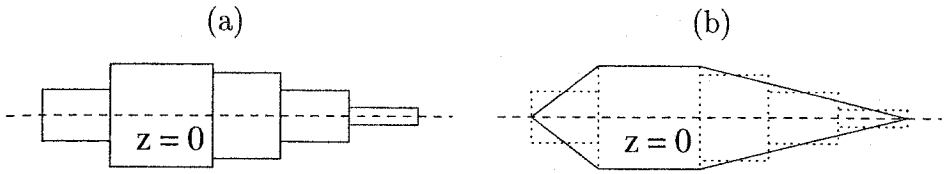


Figure 3.2: Idealization of the Plate Element: (a) h Constant by Element; (b) Nodal Values of h.

In the analysis, the degrees of freedom are defined on the plane $z = 0$ but the aerodynamic forces act on both sides of the plate (on pressure and suction surfaces for a wing). The external loads are conveniently decomposed as:

$$\{\mathrm{t}\} = \{\mathrm{t}_u\} + \{\mathrm{t}_l\} \tag{3.29}$$

and

$$\begin{aligned} S_i \;=\; & \int_{\mathcal{S}} \left\{ t_{u,z} N_i - z \left(t_{u,x} \frac{\partial N_i}{\partial x} + t_{u,y} \frac{\partial N_i}{\partial y} \right) \right\} d\mathcal{S} + \\ & \int_{\mathcal{S}} \left\{ t_{l,z} N_i - z \left(t_{l,x} \frac{\partial N_i}{\partial x} + t_{l,y} \frac{\partial N_i}{\partial y} \right) \right\} d\mathcal{S}. \end{aligned} \tag{3.30}$$

In the modal analysis,

$$\begin{aligned} [I]\{\ddot{\xi}\} + (\alpha_M[I] + \alpha_K[\Lambda])\{\dot{\xi}\} + [\Lambda]\{\xi\} \;&=\; [P]^T\{S\} \\ &=\; [P]^T\{S_u\} + [P]^T\{S_l\}. \end{aligned} \tag{3.31}$$

It is thus possible to have a full 3-D representation of the aerodynamic loads within the framework of the "2-D" plate model. (Note: A 3-D 20-node hexahedral element is introduced to reproduce the geometry in the evaluation of the external loads.)

3.2.1 Vibrations of a Thin Plate

The natural frequencies and deformation of a simply supported thin square plate (see figure 3.3) are computed as a basic validation of the thin plate model implemented in this section. The test case is taken from Reddy [63], example 12.1.

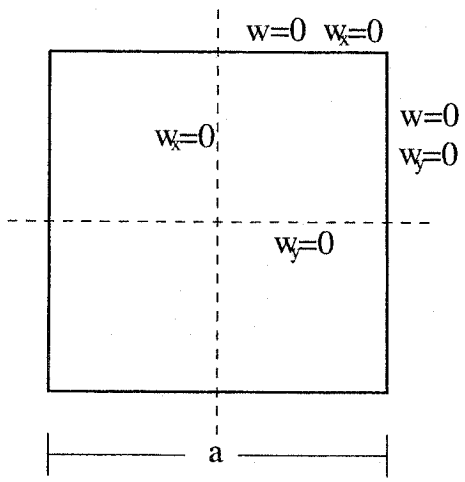


Figure 3.3: Boundary Conditions for a Simply Supported Thin Square Plate.

The material properties of the isotropic plate are $E = 1.0\times10^6\,\mathrm{N/m^2}$, $\nu = 0.25$, and $\rho = 1.0\times10^4\,\mathrm{kg/m^3}$. The width of the plate is $a = 1.0\,\mathrm{m}$ and its thickness is $h = 0.1\,\mathrm{m}$. The calculated frequencies $\tilde{\omega} = \omega\sqrt{\rho/E}a^2/h$ are tabulated in table 3.3 for the first three modes of vibration for a 4×4 grid on the quarter plate. It can also be observed from the results that the frequencies approach the reference computed and exact solutions reported in Reddy [63].

Mode	this work	Reddy	Exact
1	0.9218	0.9215	0.9295
2	4.4489	4.4436	4.5025
3	7.4252	7.4246	7.8670

Table 3.3: Calculated Natural Frequencies (Hz) of a Thin Plate.

The center deflection $\tilde{w} = wEh^3/f_\circ a^4$ of the plate is computed for a uniform transverse load $f_\circ = 100.0\,\mathrm{N/m^2}$. The equilibrium position of the plate is quickly obtained by marching in time using the modal analysis, with a large value of dynamical damping, α_M. The result is presented in table 3.4 for the 4×4 grid. The result agrees well with that of Reddy [63] based on classical plate theory using the non-conforming element (CPT(N)).

	this work	Reddy CPT(N)	Exact
$\tilde{w}(0,0)$	0.04639	0.04643	0.04570

Table 3.4: Calculated Center Deflection of a Thin Plate.

3.2.2 Modal Analysis of the LANN Wing

The thin plate model is applied to the LANN wing [64, 65]. In the experiment, the wing is made out of Armco Nitronic-40 stainless steel, but, for the purpose of this exercise, the material properties of common stainless steel are used: $E = 1.93\times10^{11}\,\mathrm{N/m^2}$, $G = 7.50\times10^{10}\,\mathrm{N/m^2}$, $\nu = 0.287$, and $\rho = 7860.0\,\mathrm{kg/m^3}$. The cross-sectional and planform views of the wing are given in figure 3.4. The wing span is 1.0 m and its root chord is 0.361 m. The sweep angle is 25° at the quarter-chord line.

Certain assumptions are made in view of a few modeling uncertainties. First,

the thickness of the wing in the finite element model is reduced to obtain a more representative total mass of the wing, since the core of the wing is hollow to permit the placement of the instrumentation devices during the wind-tunnel experiments. Second, the thin plate model in the current analysis ignores the camber and the angle of attack of the wing. The model indeed sees a symmetric wing section with the thickness defined about the median plane $z = 0$ and not about its true mean camber line. Third, details are missing on the way the wing apparatus is mounted. In the wind-tunnel experiment, a tongue extends perpendicularly from the wing at the root and is attached to a vibrating device. These discrepancies are also observed in the measured data at NLR [65] and Lockheed [64] (see table 3.5). In this analysis, the wing is assumed to be fully clamped at the root.

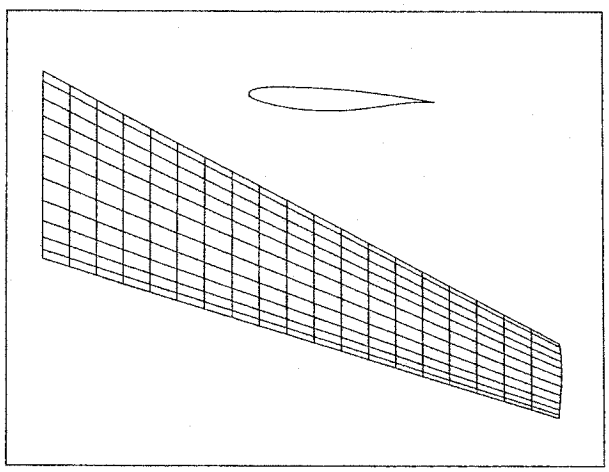


Figure 3.4: Cross-Sectional and Planform Views of the LANN Wing.

The first four computed natural frequencies are tabulated in table 3.5 and compared with the experimental data. A 20×12 grid made of quadrilateral plate elements is used. The various bending and torsion modes of vibration are shown in figure 3.5. A notable difference is pointed out. Although there appears to be a good agreement in the natural frequencies, the mode shapes are not the same for the third and fourth

modes. In the experiment [64, 65], the third mode is a predominantly bending mode while the fourth mode is a predominantly torsion mode. In the calculations, these two modes are inverted, even though their natural frequencies deceitfully match. Despite that the model does not appear to be properly calibrated, a very good agreement is nonetheless observed between the results obtained using the plate model developed in this thesis and those obtained using MSC/NASTRAN (based on the same modeling assumptions).

	Natural Frequency f (Hz)				
Mode #	Lockheed (measured)	Lockheed (computed)	NLR (measured)	This Work	MSC/ NASTRAN
1	31.93 (*bending*)	32.01 (*bending*)	30.56 (*bending*)	31.79 (*bending*)	31.90 (*bending*)
2	115.75 (*bending*)	117.60 (*bending*)	104.46 (*bending*)	120.60 (*bending*)	121.67 (*bending*)
3	249.07 (*bending*)	271.60 (*bending*)	229.39 (*bending*)	260.52 (*torsion*)	254.69 (*torsion*)
4	292.12 (*torsion*)	291.70 (*torsion*)	292.95 (*torsion*)	288.70 (*bending*)	293.17 (*bending*)

Table 3.5: Natural Modes of Vibration for the LANN Wing ([64, 65]).

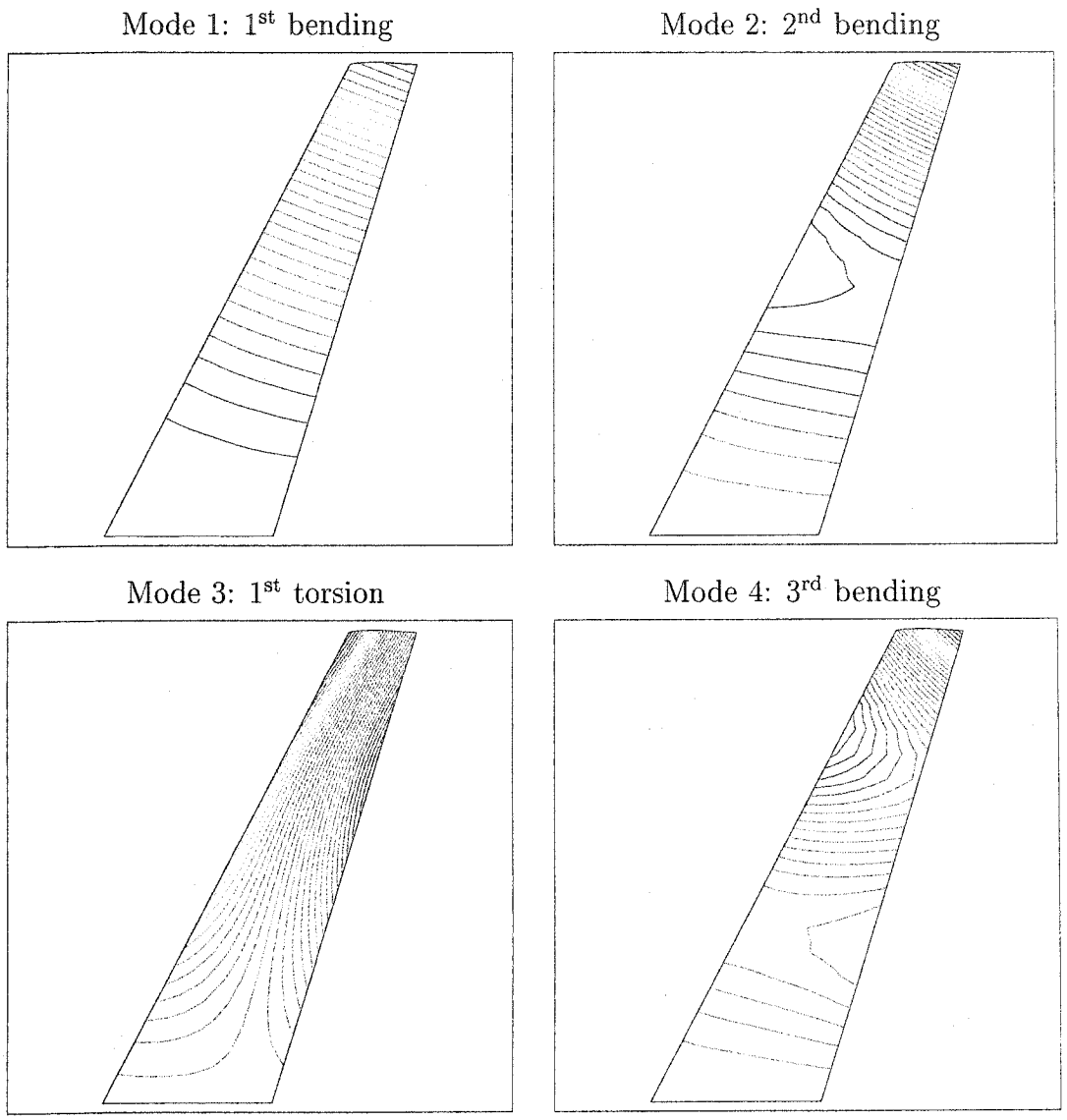


Figure 3.5: Mode Shapes for the LANN Wing for the First Four Natural Modes of Vibration.

Chapter 4

Aeroelastic Calculations

The response of a flexible structure to the action of aerodynamic forces is usually manifested by small-amplitude low-frequency oscillations. These flow-induced oscillations may decay in time, but sustained or undamped oscillations can lead to high-cycle-fatigue and to structural failure. Although only aeroelastic problems in the aeronautical field are included in this thesis, for aircraft and their components, it is recognized that other systems such as transmission line cables, tall buildings, and suspension bridges are equally affected by the impact of wind-induced vibrations. Perhaps one of the most famous witnessed example of the destruction power of resonant wind-driven forces is the collapse of the Tacoma Narrows Bridge in 1940. The deformations of the bridge are pictured in figure 4.1 moments before the catastrophic collapse [66]. The bridge has since been redesigned and rebuilt.

The goal of any aeroelastic analysis is to determine if a system in flight is structurally stable under the action of the aerodynamic loads. *Divergence*, the first phenomenon studied in classical texts in aeroelasticity [67, 68, 69], is a *static* instability which considers only the effects of the elastic forces of the structure under the action of steady aerodynamic loads, neglecting the effects of the inertial forces (dynamics)

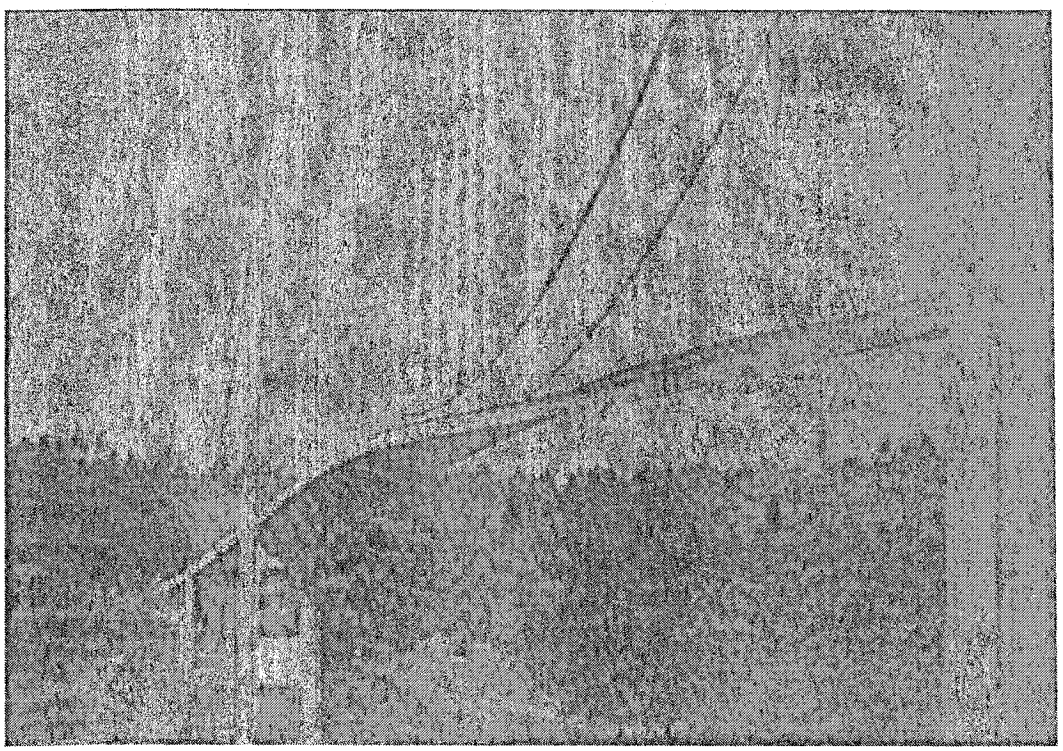


Figure 4.1: Wind-Induced Vibrations of the Tacoma Narrows Bridge [66].

of the structure. Basically, divergence is the point of failure of an elastic structure to static aerodynamic charges.

The study of divergence is of little interest for practical aircraft configurations, since dynamic instabilities are observed much before the onset of divergence. In *dynamic* aeroelasticity, the inertial forces of the structure are studied and the response of the structure due to the unsteady aerodynamic loads leads to oscillations. *Flutter* occurs when sustained oscillations at constant amplitude are observed, signifying a balance of energy exchanged between the fluid and the structure over a period of oscillation. The onset of flutter is important since it defines the limits of dynamic instability beyond which the structural system, extracting energy from the fluid,

becomes unstable. Flutter must be avoided since it can lead to losses in efficiency of the design and to losses in the maneuverability and the stability of the flight vehicle. Furthermore, flutter can reduce the life expectancy of aircraft components. *Stall flutter*, which occurs as a consequence of flow separation, is usually observed at high angles of attack during maneuvers of fighter aircraft, over ice contaminated wings, and in high speed turbomachinery blades. In other unstable motions, *beats* can also be observed.

To assess the dynamic stability of a flight vehicle, one must first understand the mechanisms leading to its instability. In brief, the system will become unstable and subject to resonance when the aerodynamic forces become in-phase with the motion of the structure, or, in other words, when the frequency of the unsteady flow matches a natural frequency of oscillation of the structure. In this case, resonance occurs, leading to unbounded oscillatory motion. For wing flutter, as studied in this thesis, flutter occurs when the first bending mode coalesces with the first torsion mode of the wing, upon excitation from the aerodynamic loads.

A brief overview of classical methods to determine the aeroelastic limits of fluid-structure systems is presented in section 4.1. These methods solve the aeroelastic problem in the frequency domain using an eigenvalue analysis. When the structural and/or aerodynamic operators are nonlinear and implemented in a segregated manner, an analysis in the time domain is most appropriate and often is the only recourse. The methodology in the time domain is described in section 4.2. Finally, in section 4.3, details are expanded about the implementation of a fluid-structure coupling algorithm which interfaces state-of-the-art flow and stress solvers in the aim of performing aeroelastic computations in the time domain.

4.1 Classical Methods in Aeroelasticity

In the previously cited classical textbooks on aeroelasticity, the calculation of the dynamic flutter boundary is studied in the frequency domain for *linear* structural and aerodynamic operators. For the stress model, the discretized equations take the form

$$[M]\{\ddot{\mathbf{u}}\} + [D]\{\dot{\mathbf{u}}\} + [K]\{\mathbf{u}\} = \{F\}, \tag{4.1}$$

where $[M]$ is the mass matrix, $[D]$ is the damping matrix, $[K]$ is the stiffness matrix, and $\{F\}$ is, in occurrence, the vector of external aerodynamic forces. This is indeed the general form obtained for the discretized Navier equations, as seen in equation (3.5). In general, the external forces depend on the position, the velocity, and the acceleration of the structure and, thus, are implicit functions of the displacements $\mathbf{u}$. (The vector $\mathbf{u}$ may alternatively be thought of as a vector of generalized coordinates.) The matrices $[M]$, $[D]$, and $[K]$ have constant coefficients, independent of $\mathbf{u}$.

In the special case where the aerodynamic forces are *linear* in the displacements, one can write

$$\{F\} = [M_a]\{\ddot{\mathbf{u}}\} + [D_a]\{\dot{\mathbf{u}}\} + [K_a]\{\mathbf{u}\}, \tag{4.2}$$

where $[M_a]$, $[D_a]$, and $[K_a]$ are, respectively, the mass, the damping, and the stiffness matrices for the aerodynamic model (with constant coefficients). Assuming a Fourier representation for $\mathbf{u}$, of the form

$$\mathbf{u} = \mathbf{U}(\mathbf{x})e^{i\lambda t}, \tag{4.3}$$

the coupled undamped system formed by (4.1) and (4.2) leads to the generalized eigenvalue problem

$$[A] - \lambda^2[B] = 0. \tag{4.4}$$

The eigenvalues λ give the resonant frequencies of the coupled system for the given flow characteristics, for example at a given free stream velocity U_∞ and dynamic pressure q_∞. (The matrices $[A] = [M - M_a]$ and $[B] = [K - K_a]$ depend on U_∞ and q_∞ via $[M_a]$ and $[K_a]$.) The eigenvalue problem is then solved repeatedly over a range of dynamic pressures for fixed free stream velocities, from which the flutter boundary curve is determined (recall illustration in figure 1.1).

Classical methods for the determination of the flutter boundary in the frequency domain are the *V-g* and the *p-k* methods [69]. These methods are generally employed with closed form solutions obtained using the boundary element method (Fourier transforms or Green's functions) for the aerodynamic operator, following the linearization of the flow equations. The success of the coupled eigenvalue analysis has been demonstrated for linear flow operators, such as potential flow theory for inviscid, incompressible, irrotational flows. However, the determination of the flutter boundary throughout the entire flight regime of an aircraft requires treating flow nonlinearities: compressibility effects become important, fluid viscosity is no longer negligible at high speeds, leading to turbulent boundary layers and flow separation, and finally shock waves appear in the transonic regime. The desire and the need to use the Euler or the Navier-Stokes equations to model the fluid violates the linearity of the flow operator and leads to the breakdown of the coupled eigenvalue analysis. One must then revert to an analysis in the time domain. Despite the difficulties to treat nonlinearities, analyses in the frequency domain are nonetheless particularly attractive, given their low computational costs in comparison with analyses in the time domain. Such linearized analyses can thus be of great value for an initial assessment of the aeroelastic stability of a system, but detailed analyses are recommended for a final evaluation of a design.

4.2 Time-Domain Aeroelastic Analysis

The determination of the flutter boundary in the time domain can be viewed as a divide and conquer process. To start, the stability of a given configuration with known structural properties is investigated under different aerodynamic loads. In flight, the pilot controls the speed U_∞ and the altitude of the aircraft, the latter parameter defining the thermodynamic properties of air: temperature T_∞, pressure p_∞, and density ρ_∞. The two systems of equations are solved in the time domain for various values of the pressure, and damped and undamped solutions are observed. The solution of interest is the neutrally stable one where oscillations at constant amplitude are measured – flutter.

In comparison with the eigenvalue approach, the iterative process involved in the calculation in the time domain of the flutter point at a given speed U_∞ is quite costly, requiring the solution of the unsteady Euler or Navier-Stokes equations over several periods of oscillations, typically over 10, to reach a periodic state, with 30 to 50 time steps per period of the highest mode modeled. The process is repeated over a range of free stream velocities to determine the overall flutter boundary curve. The time-domain aeroelastic analysis should thus be viewed as an analysis tool rather than a design tool, given its high computational requirements. It is best employed for a final assessment of an aeroelastic configuration, once preliminary flutter results have been obtained using simplified aerodynamic and structural models that are much less expensive, albeit less accurate.

The following procedure has been established for a complete fluid-structure coupling for aeroelastic analyses in the time domain:

1. Compute the natural modes of vibration of the structure by solving the generalized eigenvalue problem (3.14). Retain the low-frequency modes likely to

enter in resonance with the flow for the given characteristics of the problem (free stream pressure for external flows, rotational speed of the blades for turbomachinery flows).

2. Compute the steady-state aerodynamic solution, given U_∞ and p_∞. This gives the initial external steady loads on the structure and provides the initial solution for the unsteady simulation.

3. For the modes under analysis, use the modal transient response analysis to march in time until a fully-developed unsteady response is observed for the coupled fluid-structure problem, following an imposed initial small perturbation to trigger flow unsteadiness. The coupling with the aerodynamic solver is activated, with full two-way feedback between the fluid and the structure.

4. At the fixed free stream velocity U_∞, repeat Step 3 by varying the free stream pressure p_∞, based on the response of the structure at other imposed values of the pressure, in order to determine the pressure at flutter at this free stream velocity.

5. Repeat Steps 2-4 for different values of U_∞, once again iterating on the pressure, giving the pressure at flutter for a range of free stream velocities. This curve defines the flutter boundary.

The use of the modal analysis, with selected modes, can yield very good results for a large class of aeroelastic problems encountered in the field of aeronautical sciences. The choice of the modal analysis, as opposed to the use of a complete transient analysis, is mostly justified by its low computational costs. Since the high-frequency structural modes are not likely to be excited by the low-frequency fluid response, only the low-frequency modes may be included in the analysis, without compromise

on the quality of the solution. On one hand, one recalls that the modal analysis is inexpensive to use since it gives rise to a system of decoupled linear ordinary differential equations, of size equal to the number of modes selected. On the other hand, the direct integration of the full structural equations includes all vibrational modes, and a sparse linear system must be solved at each time step, when using an implicit formulation. The use of excessively small time steps, needed to capture the high-frequency modes, negates the advantages of the implicit flow solver, in which case an explicit flow solver would be more appropriate. The cost of repeatedly solving these linear systems outweighs overwhelmingly the overhead cost of solving a generalized eigenvalue problem, which is done only once. (An efficient generalized eigenvalue solver for large systems is however much more difficult to implement than a matrix solver and is rarely available in standard scientific libraries.)

The various implementation aspects of the fluid-structure coupling driver are now elaborated in the remainder of this chapter.

4.3 Fluid-Structure Coupling Algorithm

In view of the inherently high, unavoidable, computational costs for the Euler and the Navier-Stokes equations, a coupling driver is developed for time-accurate aeroelastic calculations in the time domain, while trying to minimize the additional costs due to the coupling with the stress solver. The approach consists of a time-marching algorithm by which the structural equations and the flow equations are integrated in time in a simultaneous fashion at each time step, even if the two solvers exist as separate entities. As advocated in the introduction, the procedure features a two-way coupling between the fluid and the structure, with the aerodynamic loads affecting the motion of the structure, and *vice-versa*.

The following considerations are addressed in the development of the coupling driver:

1. *Unification of the time-integration schemes used in the fluid and the stress solvers:* Are the time steps the same? Should they be?

2. *Transfer of the structural displacements and of the aerodynamic loads across the solvers:* How can the displacements and the loads be interpolated such that the work done by the fluid on the structure be conserved? How can the geometric difficulties of non-matching grids be overcome?

3. *Execution of the CFD and CSD external modules:* How can the codes communicate among themselves in a synchronized and efficient manner? How tight must the coupling be?

The fluid-structure coupling algorithm is implemented in a stand-alone program which synchronizes the execution of the fluid (CFD) and the stress (CSD) solvers and monitors the two-way exchanges of data, the displacements and the aerodynamic loads, across the two solvers. In this way, the CFD and the CSD solvers do not communicate directly with one another, but do so through the intermediary of the coupling driver (see figure 4.2). This desired autonomy of the fluid and the stress solvers allows to develop and upgrade the solvers independently, and, in addition, permits to compute the solutions concurrently over a network of workstations to reduce the overall computational time. This modular approach allows to easily substitute a solver for a different one, provided that a model-specific interface exists to process the input and the output files for the grid and the solution.

In addition to the major changes to the CFD code to support aeroelastic calculations (extension of the governing equations based on the ALE formulation and

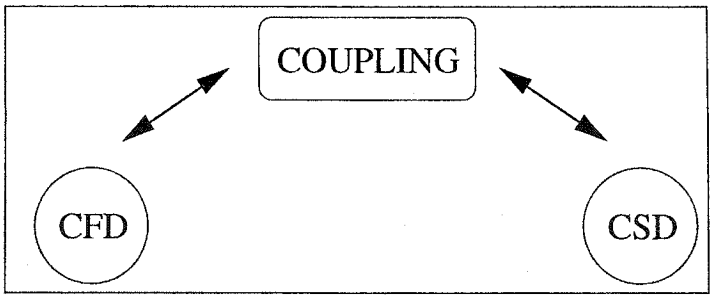


Figure 4.2: Schematic of the Fluid-Structure Coupling Driver.

implementation of the grid movement algorithm), minor changes are implemented to enable inter-process communications between the solvers and the coupling driver, see subsection 4.3.3, in order to avoid the overhead costs associated with restarting the codes. Such costs include saving and loading the new grid and the solution at each time step as well as re-initializing the data structures for the preconditioned linear matrix solver upon a restart.

The fluid-structure coupling driver extracts, on one hand, the aerodynamic loads from the flow solution and feeds them as external forces to the stress solver and, on the other hand, extracts the displacements from the stress solution and feeds them as transient boundary conditions to the fluid solver, through files, as described in the appendices A and B. The interpolation of the displacements and of the aerodynamic forces across the two grids on the surface of the structure is performed by the coupling driver, outside the CFD and the CSD solvers, using the concept of a *virtual grid.*

The virtual grid may be thought of as a third grid which represents the surface of the structure. The virtual grid is visible only by the coupling driver and it is used solely for the purpose of interpolating the solution across the CFD and the CSD grids. In the interpolation process, the displacements are interpolated from the CSD grid to the virtual grid, then from the virtual grid to the CFD grid, and conversely for the aerodynamic loads. The role of the virtual grid gains merit and value when

interpolating the solution across non-matching grids. The complete details about the construction of the virtual grid and its usage by the interpolation module are given in subsection 4.3.2 and in appendix C. At this point, the above ideas concerning the virtual grid should be sufficient to aid to the comprehension of the interpolation process required in the coupling strategy.

The details of the coupling driver are examined in the next three subsections. In subsection 4.3.1, the control loop for the time-integration is introduced. This is the core of the coupling driver. In subsection 4.3.2, the interpolation module across general non-matching grids is defined. Finally, in subsection 4.3.3, the use of stream sockets as a tool for inter-process communication across the various codes is described.

4.3.1 Implicit Coupling Time Loop

The coupling is performed in an implicit fashion, where, at each time step, the fluid and the stress solvers are invoked in an iterative fashion, possibly several instances per time step, until the aerodynamic loads and the structural displacements are in simultaneous equilibrium. The coupling time step is fixed and the same time step is used in both the CFD and the CSD solvers, which are assumed to exist as independent external modules.

The decision to invoke the CSD solver first and the CFD solver second, or *vice-versa*, is guided by physical intuition. Given the larger inertia of the structure, relative to the air, the structure in motion (large momentum) should be less sensitive to small perturbations of the surrounding air, while the air (small momentum) should respond more quickly to small deflections of the body. At a new time step, the error introduced by freezing the flow and advancing the position of the structure first should be less than the one introduced if the solution sequence were inverted. Although a few

iterations can be performed to reach equilibrium, and wash out these errors, extra iterations can result if the CFD solution is first advanced from a frozen CSD state. Numerical evidence indeed confirms the superiority of this approach. Nonetheless, the ultimate choice on the sequence of integration can be overridden by the user.

The issues pertaining to the tightness of the coupling between the two solvers are now discussed. In general, in a *loose coupling*, the CFD time step is chosen to be smaller than the CSD time step, say to be $\Delta T/N$, and the CFD solution is integrated over these N time steps, at once, without feedback from the stress solver at the intermediate CFD time steps. The response of the structure is updated only at the coupling time step ΔT. The rationale of the loose coupling is to speed-up the calculations by reducing the number of data exchanges. With the use of a small time step, typical of an explicit CFD solver for stability purposes, a loose coupling can yield potentially accurate solutions, since the discretization error is proportional to ΔT. Despite using a small time step, the structural displacements cannot be said to be in simultaneous equilibrium with the aerodynamic loads at all instants in time, hence the name "loose coupling".

In a *tight coupling*, several CSD-CFD sub-iterations are performed within the same implicit time step. By the nature of the nonlinear implicit flow solver, several Newton iterations are required at a time step in order to converge the flow solution. Instead of fully converging the Newton process at each update of the displacements, the idea consists of updating the displacements from within the Newton loop of the flow solver. That is, as the solution evolves as part of the nonlinear procedure, the corrections to the flow variables will affect the position of the structure, which will change to reach a new equilibrium with the evolving flow solution. There is no incentive to wait till the Newton iterations have fully converged before updating the displacements, since then the Newton loop has to be repeated to reach the new fluid-structure equilibrium.

For explicit solvers, the above predictor-corrector strategy is not possible, since the solution is advanced at once, without iteration, at each time step, and the tight coupling simplifies to one coupling instance per time step.

The justification to adopt a specific coupling strategy is guided by the desire to achieve the best quality solution at the minimal cost. It should be clear, at this point, that a tight coupling is a superior approach, from the accuracy point of view, but that the extra coupling instances required in a tight coupling will add to the total cost of the simulation.

A tight coupling is advocated, in this work, with the use of implicit solvers, which are unconditionally stable and which allow large time steps. It is recalled that the cost per time step for an implicit CFD solver can be several times more than that for an explicit solver. The strategy to use a small time step is, thus, not optimal for implicit solvers, since the gain in using large time steps is negated. Given the relatively small costs to compute the stress solution (using a modal analysis) and to perform the coupling (interpolation and mesh movement), less than a few percent of the total cost of the flow solver, there is no reason not to favor a tight coupling which guarantees simultaneous convergence of the two solvers. Furthermore, this procedure allows to use the same time step for both implicit solvers, without the need for a cumbersome staggered time-integration scheme to ensure stability. Full flexibility exists, nonetheless, to allow the use of different time steps for the two solvers and to vary the number of fluid-structure coupling iterations to perform a loose coupling.

The advocated coupling algorithm is illustrated by the flow chart in figure 4.3. Following the selected strategy, invoking the CSD module first, the current loads are written to a file in a suitable format, as inputs to the stress solver, and the new structural positions are computed by the stress solver. Next, the new structural displacements are extracted from the stress solution file and loaded into the virtual

grid by the coupling driver. In the second step, when the CFD module is invoked, the nodal displacements over the fluid grid are computed from the nodal displacements on the virtual grid and written in a suitable format for the fluid solver. The interior of the grid for the fluid is moved from within the flow solver, where the entire fluid grid is active in memory, prior to computing the new flow solution. The flow solution is then advanced in time and the aerodynamic loads are transferred to the faces of the virtual grid from the latest fluid solution, thus completing one CSD-CFD sub-cycle, with all the information contained within the virtual grid object. The above process can be repeated a few times within a time step, until the fluid and the structure are in equilibrium. The loop is then advanced to the next coupling time step. A similar implicit coupling procedure has also been outlined in other work [70].

How tight must the coupling be? From the numerical experiments conducted, a legitimate strategy consists of up to 3 to 4 coupling instances per time step. An equal number of Newton iterations are performed between coupling instances, such that convergence at the end of the time step is satisfied. In all practical test cases, this meant about 2 to 3 Newton iterations per coupling instance. There is no real need to update the structural position more often. This simultaneous convergence of the two codes eliminates repeating calculations within a time step and the costs of the computations are thus effectively reduced to the costs of an unsteady simulation.

The absolute change in work done by the aerodynamic forces on the structure, defined as

$$\Delta W_{(k+1)}^{n+1} = \left| W_{(k+1)}^{n+1} - W_{(k)}^{n+1} \right|, \tag{4.5}$$

$$W_{(k)}^{n+1} = \int_{\mathcal{S}} \mathbf{F}_{(k)}^{n+1} \cdot \mathbf{u}_{(k)}^{n+1} d\mathcal{S}, \tag{4.6}$$

from one coupling sub-iteration k to another $k+1$, but at the same time step $n+1$, is used as the global convergence criterion. This criterion, which combines both the aerodynamic forces and the structural displacements, constitutes a good indicator

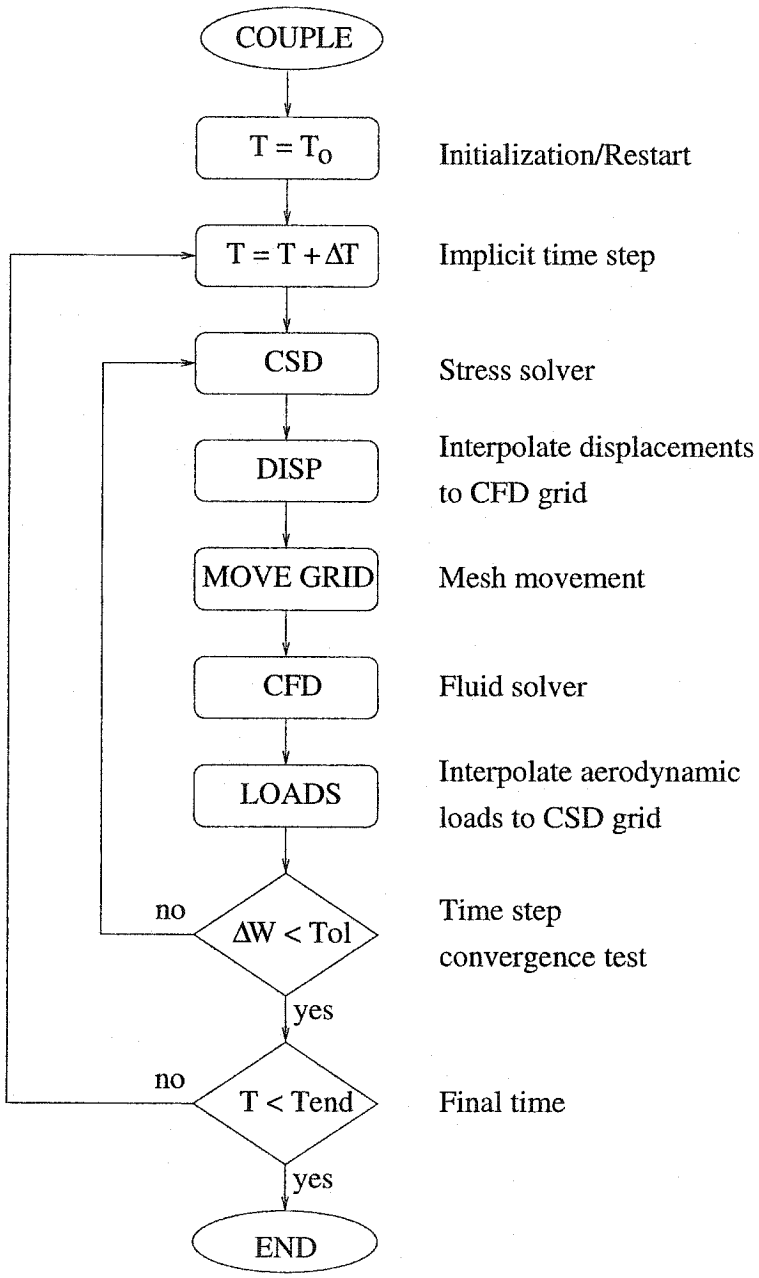


Figure 4.3: Flow Chart for the Coupling Driver.

of the global convergence. The relative error $\left|\Delta W^{n+1}_{(k+1)}/W^{n+1}_{(k+1)}\right|$ may equivalently be employed in the convergence test. It is also understood that global convergence requires the local convergence of the nonlinear flow equations. Local convergence is not an issue for the stress solver, since the Navier equations are linear.

The advantages of keeping the two solvers as separate entities are manifold. It is not practical, from the development and implementation point of view, to embed the governing equations for the fluid and the structure into the same code. Such a code would be difficult to maintain and too restrictive in its range of applications. Instead, it is much more convenient to treat the fluid and the stress solvers as black boxes, whether such codes are commercial codes or in-house codes, and to provide an exterior coupling via an interface, as established in this work. Furthermore, it should be clear by now that a tight coupling with simultaneous convergence of the two solvers can be achieved, even in a segregated solution approach.

Finally, calculations tend to suggest that a restriction exists on the size of the coupling time step if the motion of the structure is too large, even if the CFD and the CSD implicit solvers are unconditionally stable. Such limitations on the size of the coupling time step are not obvious to quantify and have seemingly not been reported in the open literature.

Complete restart facilities are provided to save the state of the simulation in case of an interruption, since it is seldom possible to perform the whole simulation at once, due to constraints in the availability of the computing resources. A user-friendly graphical interface written in TCL/TK has been developed to ease the management of the numerous solution files for the CFD and the CSD codes as well as for the coupling driver itself. A look at the graphical interface is displayed in figure 4.4. Although not essential in performing aeroelastic calculations, this integrated graphical aid is very useful to monitor the execution of the three codes in an effective manner.

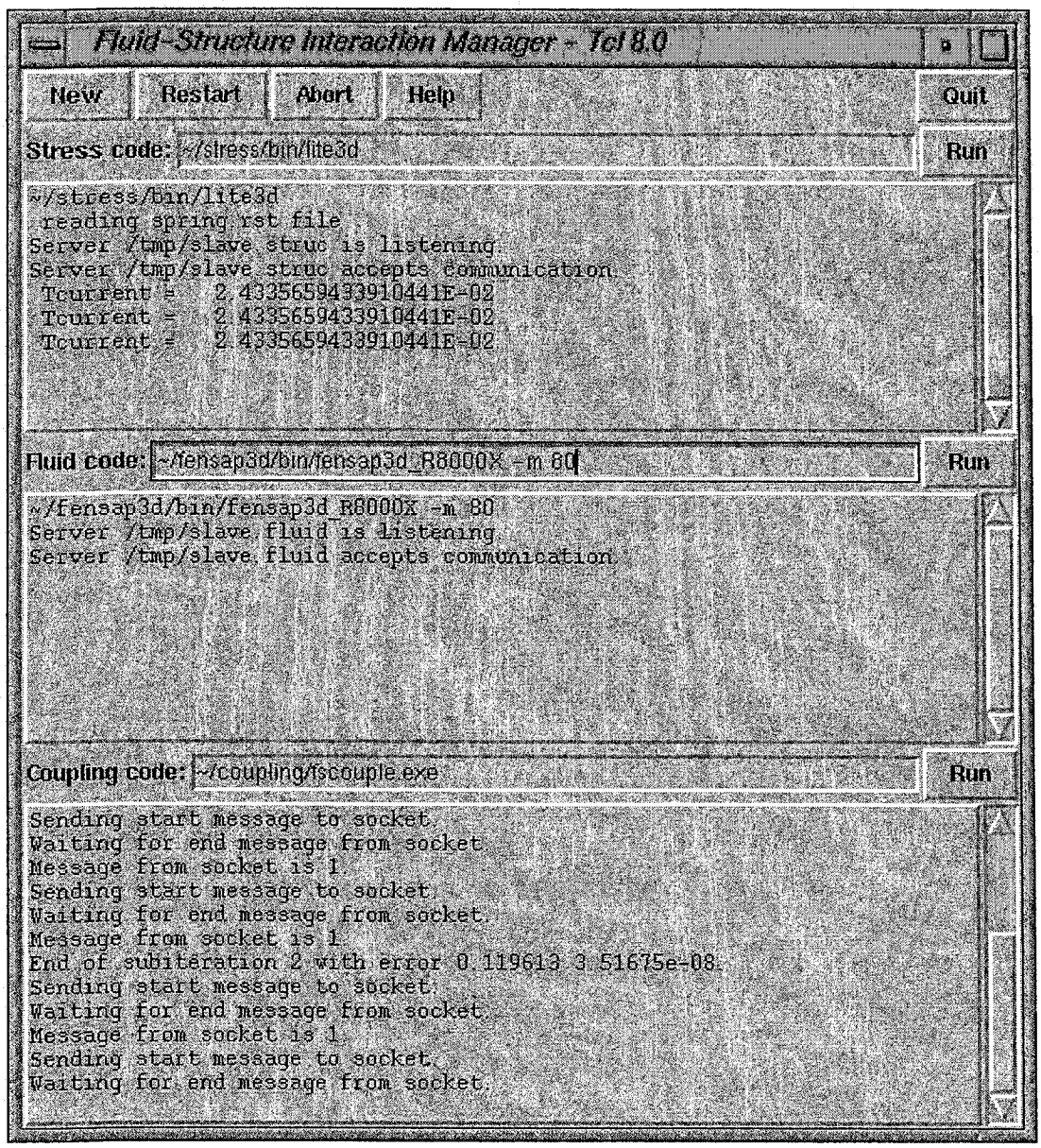


Figure 4.4: Graphical Interface for the Fluid-Structure Coupling Driver.

4.3.2 Surface Interpolation Using the Virtual Grid

Without minimizing the importance of the time control loop, the focal point of the coupling driver is without doubt the interpolation of the structural displacements and of the aerodynamic loads across the CFD and the CSD grids. In the special case where the grids for the CFD and the CSD surfaces match identically, that is the nodes and the faces on both grids coincide, no interpolation is needed. However, it is generally the case that the CFD and the CSD surfaces are made of a different number of non-coincident nodes and elements (see figure 4.5). On one hand, it is common to use a much finer grid for the fluid than for the solid, especially near the walls for Navier-Stokes calculations. On the other hand, for the stress solver, quadratic hexahedral elements are most often used, as opposed to linear tetrahedral elements for the flow solver. Thus, not only are the wall faces of different types, but the degree of the interpolation may be different too.

Another difficulty can arise if the discrete geometries described by the two grids are not identical. For example, a coarse grid cannot describe a curved surface as precisely as a fine grid, as shown in figure 4.6 for a fine and a coarse discretization of an airfoil. In this case, it is customary to expect that some nodes on the fine mesh will not lie on the surface of the structure described by the coarse mesh. These nodes are then projected onto the surface of the fluid grid, since the latter is likely to better represent the fine details of the geometry.

It would appear somehow ironic to devote, on one part, great efforts to construct a sophisticated CFD code satisfying conservation of mass, momentum, and energy, yet to transfer the aerodynamic loads in a non-conservative fashion. In view of the mentioned geometric difficulties, the interpolation in a conservative manner of the displacements and of the loads across non-matching grids promises to be a challenging task. Such generalization to non-matching grids is, however, needed, because

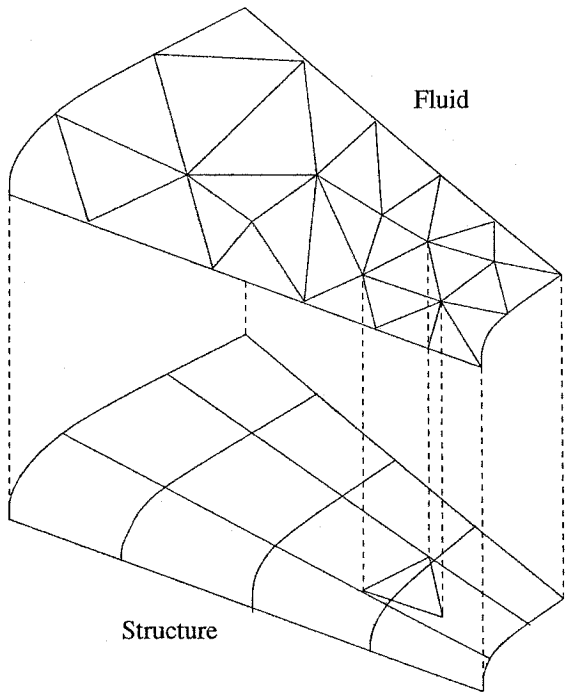


Figure 4.5: Non-Matching CFD and CSD Grids for a Wing Surface.

the constraint of using matching grids is too restrictive for use with automatic grid generation and mesh adaptation.

The difficulties associated with the use of non-matching structured and unstructured grids have been acknowledged in the literature. In their work, Farhat and his co-workers [71] integrate the aerodynamic loads over a face of the stress grid using Gauss-Legendre quadrature and evaluate the loads (in paper [71] only the pressure for Euler calculations) at the Gauss integration points on the fluid surface. However, some fluid faces may not contain any integration points and their contribution will not be accounted for, thus preventing strict conservation of the loads. "Accuracy" can be increased by increasing the number of integration points, but in all but fortuitous cases, there is still no guarantee that the desired conservation properties sought in this work will be satisfied.

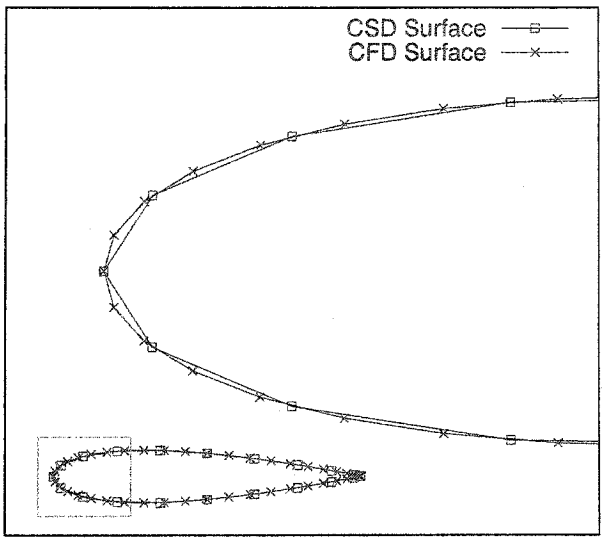


Figure 4.6: Non-Conforming CFD and CSD Surfaces for a NACA0012 Airfoil.

On the other hand, Löhner and his team have recognized the importance of conserving the aerodynamic loads and the work done by the fluid on the structure [72, 37]. The loads $F_s(\mathbf{x})$ and $F_f(\mathbf{x})$ are set equal in the weighted integral sense:

$$\int_\Gamma W F_s \, d\Gamma = \int_\Gamma W F_f \, d\Gamma \tag{4.7}$$

for a set of weighting functions $\{W\}$. Using a Galerkin finite element approach, the integral on the left yields a consistent mass matrix to be inverted to obtain nodal values of the stress loads F_s. However, the difficulty lies in integrating the right-hand-side vector with the weighting function W defined over a face on the stress grid and the loads F_f being defined over a non-matching face on the fluid grid. Similarly to Farhat's work, an adaptive Gaussian quadrature method is used to evaluate the right-hand-side vector of forces. By looping over the fluid faces, in this case, the forces can be integrated in a conservative fashion, since all fluid faces are visited. The accuracy of the evaluation depends on the coarseness of the stress grid and the interpolation error of the stress weight functions at the integration points of the fluid

faces. The above strategy has been used effectively with an Euler solver for which only the (node-based) pressures had to be transferred.

The concept of using an intermediate grid, labeled *virtual surface*, for the representation of the surface at the fluid-interface was proposed by Guruswamy [73]. Although its documented usage is vague, a structured $ij-$grid was used as the virtual surface, for the case of a wing, to allow the use of surface splines for the direct interpolation of the loads and the displacements. In this thesis, the concept of a *virtual grid* is adopted for the representation of the moving surface of the structure in contact with the fluid, but its implementation and usage are completely different. The interpolation methodology based on the virtual grid, defined herein, first permits to transfer the aerodynamic loads in a truly conservative manner, and, second, extends naturally to include both the pressures and the viscous shear stresses for Navier-Stokes calculations [74] by permitting to define face-based loads on the fluid grid. Strict conservation is guaranteed locally, over each face of the stress grid, and globally, over the entire fluid and stress surfaces. Consistent node-based loads are obtained at the grid points on the stress grid. The interpolation module is part of the coupling driver, hence it is independent of the CFD and CSD solvers used.

As briefly introduced earlier in section 4.3, the main idea is that the virtual grid is a third grid describing the surface of the structure, as drawn in figure 4.7. The virtual grid is an entity belonging to the coupling driver only, with both the displacements and the loads being defined on it. The displacements are interpolated from the structure grid to the virtual grid in a first step, then from the virtual grid to the fluid grid in a second step. A more elaborate yet analogous strategy is adopted for the transfer of the face-based aerodynamic loads.

These steps may appear cumbersome at first sight: Why not interpolate the displacements from the CSD grid directly to the CFD grid (and *vice-versa* for the loads)

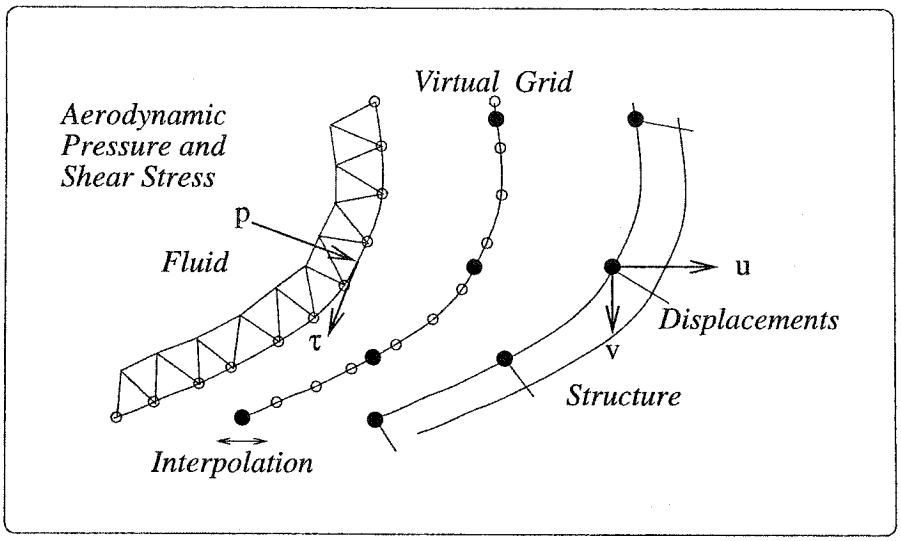


Figure 4.7: Surface Interpolation of the Displacements and Loads and Representation of the Virtual Grid.

without the intermediary of the virtual grid? The answer is that geometric difficulties arise when the CFD and the CSD grids do not match, as it is generally the case. In some abstract sense, the virtual grid may be viewed as a correspondence table with pre-defined node-to-node, node-to-face, and face-to-face mappings which facilitates the interpolation process. Since no calculations are actually performed on the virtual grid, no errors are introduced in the solvers and it then becomes possible, having imposed certain conditions on the construction of the virtual grid, to transfer the aerodynamic loads in a conservative fashion. In brief, there is no direct connection between the CFD and the CSD grids other than through the virtual grid.

In this work, the virtual grid is constructed from the union of the CFD grid and the CSD grid at their surface of intersection. The construction process of the virtual grid is such that the CFD and the CSD surface grids are subsets of the virtual grid. More precisely, all nodes that belong to the CFD and the CSD grids on the surface of the structure are present in the virtual grid. Further, all wall faces associated with

the CFD and the CSD grids are made up of an integral number of faces of the virtual grid. Specifically, any face of the virtual grid must lie entirely inside a CFD or a CSD grid face. Such a face is not allowed to cut any face of the CFD and the CSD grids. This construction facilitates the interpolation of the displacements, defined at the nodes, and allows for the exact integration of the aerodynamic loads over the surface of the structure in order to achieve conservation of the work done by the two systems.

It is noted that the virtual grid is constructed only once as part of the pre-processing steps prior to the start of the simulation. Even if the grids move, they move together in the same fashion, along with the virtual grid, and it is not necessary to reconstruct the virtual grid at any time step.

An edge-oriented algorithm is adopted for the construction of the virtual grid, using triangular faces, while preserving the integrity of the background CSD and CFD surfaces. More precisely, the edges of the CSD grid are swept and new triangular faces are introduced as a result of cutting CFD edges by CSD edges (see algorithm to follow). This edge-based approach makes it possible for the CFD and the CSD surfaces to be made of triangular or quadrangular faces. Triangular faces, arising from unstructured tetrahedral meshes, are the most common face type for CFD grids. Linear tetrahedral elements are rarely used in stress analysis codes; quadratic hexahedral elements, with quadrangular faces, are preferred for accuracy purposes.

In the implementation of the algorithm which constructs the virtual grid, it is assumed that the CFD grid is finer than the CSD grid. This assumption is irrelevant to the outcome of the process, but, when the CFD grid is indeed finer than the CSD grid, which in general it is, the decomposition process will execute faster since the number of node searches will be reduced. The speed of execution, proportional to the number of nodes on the surfaces of the two grids, is not a major concern since

the virtual grid is constructed once at the beginning. However, for later use with unsteady mesh adaptation, some optimization (ideally parallelization) will be desired when the topology of the CFD grid changes frequently.

The efficiency of the algorithm rapidly deteriorates when the two surfaces (CSD and CFD) are not conforming. Such a case is always present when the geometry possesses curved surfaces. The CSD surface is then projected onto the CFD surface. A loss in efficiency is incurred during the searches over the active faces of the virtual grid when determining the face closest to the node to add. When the two surfaces are "close", the searches will end whenever the geometric distance between a point and the surface is less than a prescribed threshold. In all other cases, an exhaustive search over all faces must be conducted to safely identify the closest face, in particular when the geometry contains small features. The implementation of an octree search algorithm would greatly reduce the search operations and speed-up the algorithm.

The algorithm is summarized in the following 4 steps.

1. Use the CFD grid as the initial virtual grid, subdivided into triangular faces.

2. Add the nodes of the CSD grid to the virtual grid. If a node already exists, the request is ignored. Otherwise, the virtual grid face containing this node is subdivided into three new triangular faces by connecting the new node to the three nodes of the face. Special care is taken when a node collapses onto a vertex or an edge of an element to avoid creating new elements with zero area. On non-conforming segmented geometries, the CSD nodes are projected onto the fluid surface, since the latter is likely to be finer and more suitable to represent the curved geometry of the body. At this point, all nodes of the CFD and the CSD grids are contained in the virtual grid.

3. Scan the edges of the CSD grid to determine their intersection with the edges of

the background virtual grid. As the elements of the virtual grid are cut, they are traversed along each edge of the CSD faces. The splitting process is illustrated in figure 4.8. Once the intersection point P of the two edges AB and ab has been computed, split the virtual grid element in two triangles DAP and DPB by joining the intersection point P to the third node D of the face. In a dual step, create two more triangles PAC and PCB in the opposite element of the virtual grid (the opposite element is the neighboring element sharing the cut edge AB). These two "mirror" elements are used as a starting point for the next cutting operation (thus avoiding costly searches over the grid). That is, the edge ab must cut either PAC or PCB.

4. In a cleanup phase, construct the node-to-face and face-to-face pointers between the virtual grid and the CFD and the CSD surfaces. While all nodes and all faces of the virtual grid have been associated with the faces of the CFD grid at some point during the construction of the virtual grid, the mappings to the CSD grid are incomplete. For instance, a search is required to find which CSD face contains a given virtual face whenever the latter was originally contained inside a CSD face (not crossing its edges). A frontal search algorithm is used to scan the faces.

A diagram illustrating the construction of the virtual grid is shown in figure 4.9. It can be seen that the nodes of the virtual grid consist of all nodes of the CFD and the CSD grids, plus the added nodes resulting from the intersection points of the CFD and the CSD edges. The faces are reconnected after the introduction of the new points.

The node and the face mappings, from the CFD grid to the virtual grid and from the CSD grid to the virtual grid, are constructed once prior to the simulation and they are used to avoid repeated searches during the interpolation process. (In three

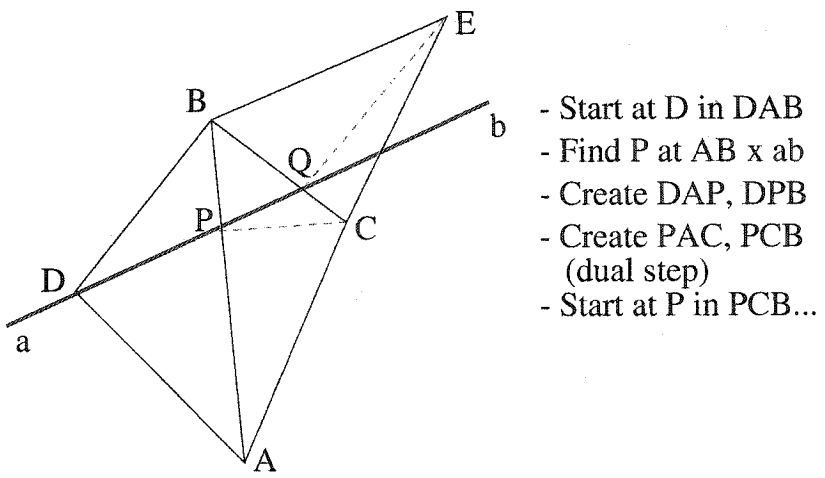


Figure 4.8: Splitting of an Edge in the Construction of the Virtual Grid.

dimensions, the cost of these searches can be significant.) The mappings are simple to create since there exists a one-to-one correspondence among the nodes and the faces do not intersect. The grids may be structured or unstructured, with quadrilateral or triangular faces on the surface of the structure.

The algorithm to construct the virtual grid is demonstrated for a square domain. Figure 4.10a shows two grids for the square domain, one grid hypothetically representing the fluid and the other representing the structure. The virtual grid resulting from the union of the two grids is displayed in figure 4.10b. The coarseness of the grids is intended for visualization purposes. In practice, the mesh for the fluid is much finer than the mesh for the structure. The geometric properties of the virtual grid are respected: all nodes and all edges belonging to the CFD and the CSD grids are reproduced in the virtual grid. That is, each of the two grids, CFD or CSD, is a subset of the virtual grid. The "poor" appearance of the virtual grid, caused by the thin skewed elements, is purely of cosmetic concern. All elements are used as part of the interpolation process only and not during the solution process of the CFD and the CSD codes.

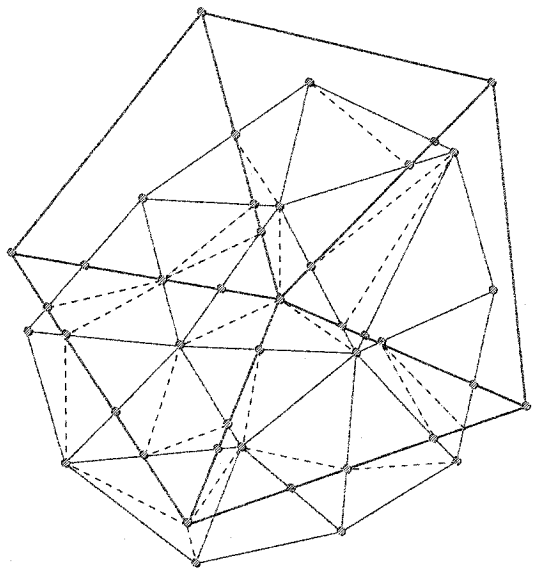


Figure 4.9: Construction of the Virtual Grid (Thick Red Lines: CSD Grid; Thin Blue Lines: CFD Grid; Dashed Lines: Added Edges to the Virtual Grid).

The construction process of the virtual grid provides the basis to guarantee the conservation of the loads and work done in the case where the CFD and the CSD geometric surfaces coincide exactly. When the surface discretizations are not identical, their areas will be different and conservation cannot be strictly achieved. For example, figure 4.6 shows slightly different surfaces for the geometry of an airfoil, near its nose where the curvature is pronounced. However, a scaling of the area of the projected faces is imposed, face by face, to enforce conservation. This scaling is applied to the shape functions of the loads during their interpolation.

The structural displacements, defined at the nodes of the CSD surface, are transferred to the matching nodes on the virtual grid. The displacements at the other nodes of the virtual grid are obtained using linear interpolation over the triangular faces. This simple interpolation process can introduce discrepancies if the order of

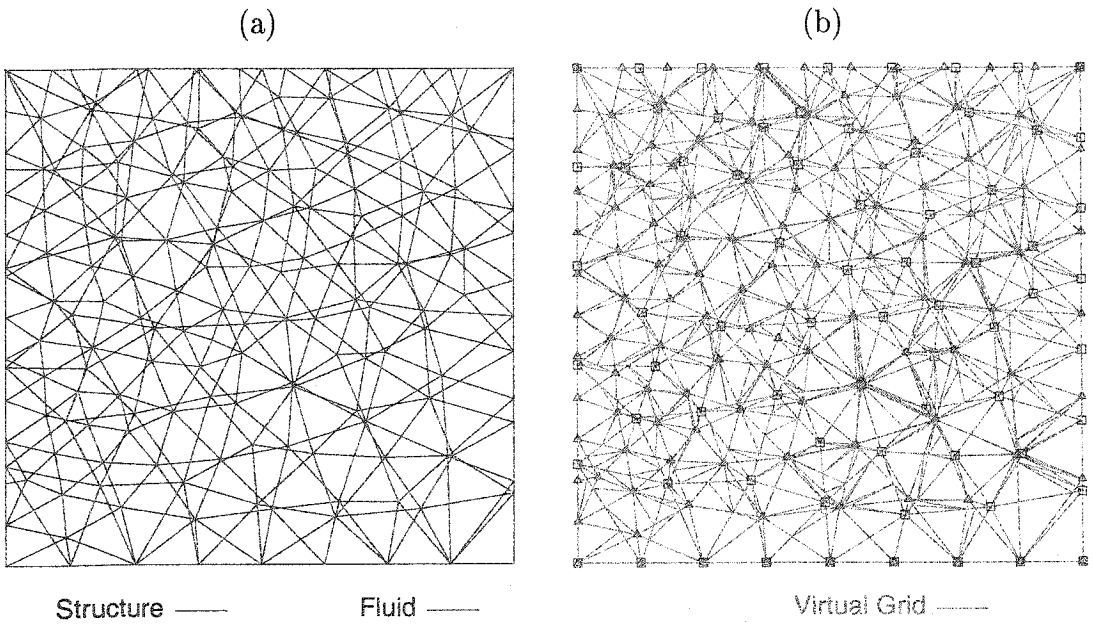


Figure 4.10: Construction of the Virtual Grid for a Square Surface: (a) the CFD and CSD Surfaces; (b) the Resulting Virtual Grid.

the interpolant is different from the shape functions used in the stress solver (for example, quadratic shape functions on curved surfaces). Such a modeling error is small and can easily be overcome if a knowledge of the shape functions is passed to the interpolation module of the coupling driver. Finally, once the displacements have been defined at all the nodes of the virtual grid, they are transferred directly to the nodes of the CFD grid.

If the node-based displacements are easily interpolated, on the other hand, the face-based aerodynamic loads require special attention to transfer them in a conservative fashion. First, the face-based loads $\mathbf{t}$ defined on the CFD surface are transferred to the faces of the virtual grid. From equations (2.24) and (2.25), these loads include both the pressure and the shear stresses, hence the present formulation lends itself to viscous flow calculations without modifications.

The somehow tedious reconstruction of the loads over the faces of the grid for the structure is guided by the principle of conservation of work done. That is, within the discretization error, the work done on the fluid surface must equal the work done on the surface of the structure:

$$W_f = \iint_{A_f} \{\mathbf{t}\}^T \{\mathbf{u}\} dA = \iint_{A_s} \{\mathbf{t}\}^T \{\mathbf{u}\} dA = W_s. \tag{4.8}$$

Also, since the shape functions sum up to unity, conservation of work done also leads to the conservation of the total forces on both grids.

The enforcement of conservation of work done is trivial when the discretizations of the two grids match at the surface of contact. Delicate care must be taken when the fluid and the structure faces are not identical, but, as a consequence of the way the virtual grid is constructed, the integration of the loads is greatly simplified. A difficulty however persists: for a virtual grid face $A^{(v)}$ contained in a stress face $A^{(s)}$, the displacements u_i defined over the stress face are based on the shape functions for $A^{(s)}$ while the surface of integration is $A^{(v)}$.

Recalling equation (3.7), obtained in the derivation of the Rayleigh-Ritz formulation, the surface integrals

$$\int_{A^{(s)}} t_x \bar{N}_j dA, \quad \int_{A^{(s)}} t_y \bar{N}_j dA, \quad \int_{A^{(s)}} t_z \bar{N}_j dA \tag{4.9}$$

must be evaluated over each boundary face of the CSD grid. The difficulty arises in that the shape functions $\bar{N}_j$ are defined over the structure element, while the loads t_x, t_y, and t_z are defined over the fluid element. In the evaluation of the integral, say for t_x, the integral is broken as

$$\int_{A^{(s)}} t_x \bar{N}_j dA = \sum_{v \subset s} \int_{A^{(v)}} t_x^{(v)} \bar{N}_j dA. \tag{4.10}$$

As illustrated in figure 4.11, the integral over a stress face $A^{(s)}$ is the direct sum of the integrals over the virtual grid faces $A^{(v)}$ making up the stress face. Since the

stress solver has no knowledge of the faces $A^{(v)}$ on the virtual grid, these integrals are pre-evaluated in the coupling driver, although this requires some knowledge about the stress solver on the coupling driver's side. It is important to state that the true type of the face, quadrilateral or triangular, is used for the integration of the loads on the CSD grid, with the appropriate shape functions. Triangular faces are only introduced for the purpose of constructing the virtual grid.

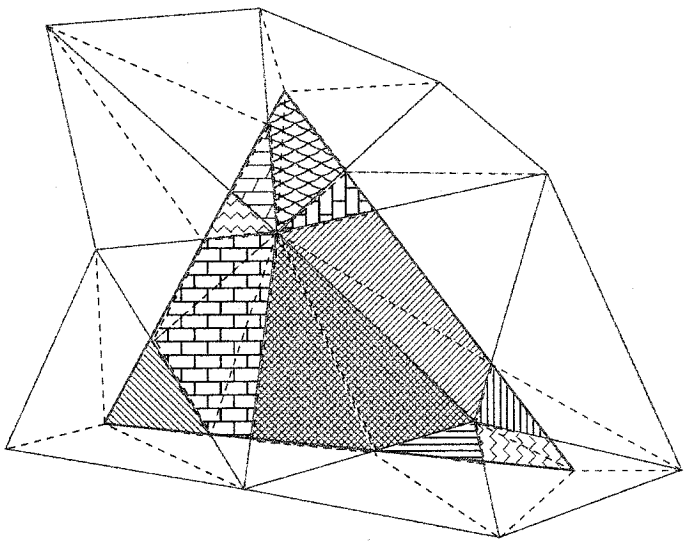


Figure 4.11: Evaluation of a Surface Integral on the Virtual Grid: Break-up of the Virtual Faces Over a Structure Face.

The integrals are evaluated using Gauss-Legendre integration, such as for t_x,

$$\int_{A^{(v)}} t_x^{(v)} \bar{N}_j dA = \sum_{k=1}^{ngaus} t_x^{(v)}(\xi_k, \eta_k)\, \bar{N}_j(\bar{\xi}_k, \bar{\eta}_k)\, |J_k|\, w_k \tag{4.11}$$

where

$$\begin{aligned} \bar{\xi}_k &= \sum_i N_i(\xi_k, \eta_k)\bar{\xi}_i, \\ \bar{\eta}_k &= \sum_i N_i(\xi_k, \eta_k)\bar{\eta}_i, \end{aligned} \tag{4.12}$$

with $(\bar{\xi}_i, \bar{\eta}_i)$ being the local coordinates of the nodes of the face $A^{(v)}$ in terms of the local coordinates system of the face $A^{(s)}$ and N_i and $\bar{N}_i$ the shape functions

for the faces $A^{(v)}$ and $A^{(s)}$, respectively. (See figure 4.12.) $|J_k|$ is the determinant of the Jacobian of the transformation (from global to local coordinates) and w_k is the Gaussian quadrature weight. In this work, using the solver FENSAP, the loads are reported as piece-wise constant over each fluid face; hence, at least three Gauss points are necessary to calculate exactly the weighted contribution of the loads on the triangular face $A^{(v)}$ to each node of the face $A^{(s)}$ since the shape functions $\bar{N}_i$ are bi-linear. In this fashion, the conservative integration of the loads is reduced to a simple sum over the virtual faces of the stress faces. The loads are conserved over each stress face and, consequently, globally.

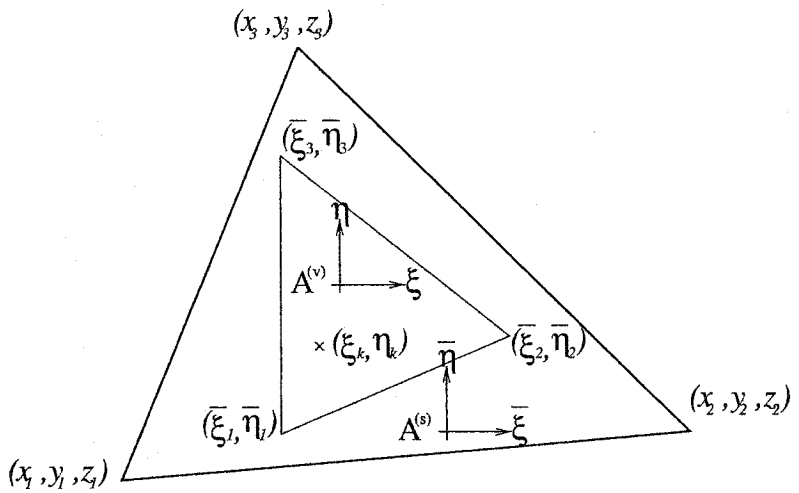


Figure 4.12: Local Coordinates Systems for $A^{(v)}$ and $A^{(s)}$.

4.3.3 Inter-Process Communication

In the past, restarting the solvers at each coupling instance has limited the exchanges of information between the fluid and the stress solvers to once every few time steps, resulting in a very loose coupling and the need for the use of very small time steps to limit the losses in accuracy. A restart at every coupling instance can greatly penalize

the overall performance of the code due to the overhead costs associated with writing and reading the new grid and the restart solution and with initializing the data structures required by the solvers. Further, these operations, performed sequentially, can create a bottleneck in parallel computations.

With the objective to provide the tightest level of coupling in order to guarantee simultaneous convergence of the two solvers at each time step at the lowest cost possible, stream sockets [75] are used to enable inter-process communications between the solvers and the coupling driver. Alternatively, the Message Passing Interface (MPI) standard has also been adopted in aeroelastic codes [12], as opposed to using stream sockets. The MPI library offers more flexibility than the use of stream sockets; in particular, MPI allows for communication across computers and provides tools for parallelization, which the current use of stream sockets does not support. Although the advantages of the MPI library are apparent, stream sockets are still used, perhaps since they are part of the standard libraries of the C programming language.

Inter-process communication requires slight modifications to the CFD and the CSD codes, in order to implant instructions to accept socket connections to listen and talk to the coupling driver. This restriction is not viewed, however, as a setback of the implementation. In fact, one common denominator of all aeroelastic calculations reported in the literature is the use of in-house developed CFD codes. The use of commercial stress solvers (primarily MSC/NASTRAN) is more common. In view of this observation, it is reasonable to assume that possibilities exist to implement tools allowing inter-process communication, at least across the CFD solver and the coupling driver.

At execution, the fluid and the stress solvers (the clients) are started in the background until they halt at suitable breakpoints, awaiting instructions from the COUPLING module (the server). When the COUPLING module is started, recall the

flow chart in figure 4.3, messages are sent to the clients, via the stream sockets, one socket per client, with specific directives to the clients. For instance, the COUPLING module sends a message to the fluid solver to start a time step, then awaits the response from the socket to signal the completion of the task (iteration or time step). The stress solver is invoked in an identical fashion. It is emphasized that there is no direct communication between the stress and the fluid solvers.

In addition of eliminating the need for a restart at every coupling instance, the use of sockets further improves the overall efficiency of the implicit flow solver since the CFD code remains active in memory across coupling instances. A quasi-Newton method can be used over several coupling instances and across time steps. Freezing the Jacobian matrix eliminates the costs of assembling the global matrix and of factoring it to obtain the preconditioning matrix at every Newton iteration. Finally, although the memory requirements are cumulated for the two solvers, when running on the same machine with the two solvers resident in memory, such a constraint is soon alleviated if the solvers are run concurrently on different machines.

The unavailability of inter-process communication does not prevent a tight coupling between the solvers, say if a commercial code is used and its source code is not available. In this case, the code is restarted at every coupling instance despite the high overhead costs, the penalty being as high as doubling the overall execution costs, or worse. However, since the flow solver already requires modifications to extend the governing equations using the ALE formulation to account for the presence of moving bodies, without which fluid-structure interactions lose their significance, it may be assumed that access to the source code of the flow solver is possible, at least in a limited fashion.

For now, only instructions to control the execution of the clients are passed by sockets. The transfer of the loads and the displacements between the server and the

clients using sockets instead of files is a possibility, although it is not expected that such a strategy would contribute to a significant reduction in the total execution time, since the time spent during the coupling is a fraction of the time spent in the flow solver. Overall, the cost of an aeroelastic calculation is about the same as the cost of the corresponding unsteady flow calculation.

Chapter 5

Numerical Results

Test cases are presented to demonstrate the scope and the validity of the aeroelasticity methodologies developed in this thesis. In chapter 2, the efficiency and the robustness of the grid moving algorithm were demonstrated. In this chapter, the test cases are chosen to demonstrate the accuracy of the treatment of the ALE formulation for moving and deforming bodies, at first for an imposed motion, without coupling to a stress solver, and secondly, with two-way feedback and full coupling with the stress solver for validation of standard aeroelastic configurations. In addition, several experiments are conducted to investigate different solution strategies. Is a tight coupling really necessary? What are the effects of neglecting the shear stresses from the aerodynamic forces? Answers to these questions may help justify, or discourage, the use of simplified methods and shortcuts to reduce the overall computing times of aeroelastic calculations and flutter analyses.

The main difficulty encountered during the validation of this work was the limited availability of well-documented aeroelastic benchmarks, in particular in three-dimensions. Moreover, the scope of these benchmarks had to lie within the modeling capabilities of the stress and the flow solvers developed in this work, and the simula-

tions had to run in a reasonable amount of time on the available computing resources. One three-dimensional aeroelastic calculation of a wing in flutter is presented, for the AGARD 445.6 wing. The other problems considered are two-dimensional in nature, mostly for ease of visualization, grid generation, and for the reduced computing resources required. It is noted that although the technology has been developed in the three-dimensional framework, its application is identical in two and three dimensions.

5.1 Burgers' Equation on a Moving Grid

The first result, although trivial in appearance, shows the consistency of the ALE moving-grid technology for the 1-D Burgers viscous equation

$$\frac{\partial u}{\partial t} + \left(u - \frac{dx}{dt}\right)\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \tag{5.1}$$

based on the ALE formulation (in non-dimensional form). The finite element discretization mimics the one in FENSAP (see section 2.3). The computational grid is moved in time, see the figure 5.1a, with the prescribed grid speed

$$\frac{dx}{dt} = \sin(2t)\cos(x/2). \tag{5.2}$$

The initial flow velocity distribution at $t = 0$ is the step function

$$u(x,0) = \begin{cases} 1, & \text{if } x < 0 \\ 0, & \text{if } x > 0 \end{cases} \tag{5.3}$$

and the viscosity parameter is $\nu = 0.10$. The time step is $\Delta t = 0.05$. The evolution of the solution is shown in figure 5.1b at $t = 0.5,\ 1.5,\ 2.5,\ 3.5$ on the moving grid and on a stationary grid. The numerical results also coincide with the exact solution, even when the grid moves. Equally accurate results are obtained for different grid motions, verifying that the node movement can indeed be chosen arbitrarily in the interior of the domain. This fact is also verified for the other problems studied.

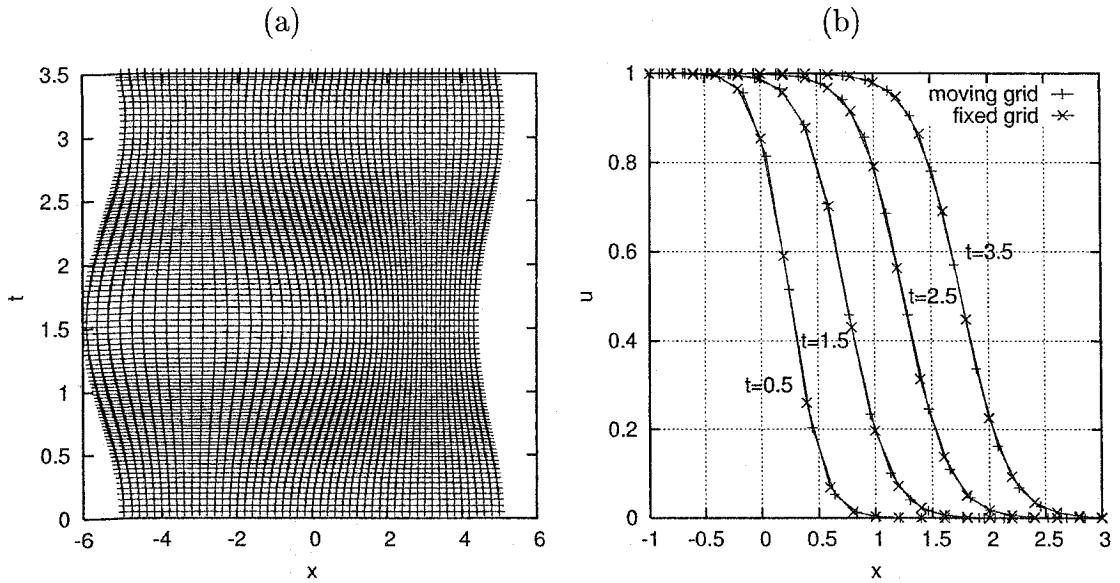


Figure 5.1: Solution to Burgers' Viscous Equation on a Moving Grid: (a) Evolution of the Moving Mesh in Time; (b) Velocity Profile at Times $t = 0.5, 1.5, 2.5, 3.5$.

5.2 Pitching Airfoil

The problem studied is that of a transonic flow past a pitching NACA64A010 airfoil section based on the NASA Ames model [76]. The NASA Ames model is 10.6% thick and has a pointed trailing edge, as opposed to the analytical NACA64A010 geometry which is 10% thick and has a square trailing edge.

In this problem, a rigid body pitching motion is prescribed and the unsteady flow response is calculated. Such a problem demonstrates the capabilities of the flow solver to handle transient moving mesh boundaries, including mesh movement in the interior of the domain. However, there is no feedback from the flow field to the motion of the body since the body motion is imposed.

The pitch angle varies as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t), \tag{5.4}$$

or

$$\alpha(\tau) = \alpha_m + \alpha_0 \sin(2k_c\tau) \tag{5.5}$$

in terms of the non-dimensional time $\tau = \frac{tU_\infty}{c}$. The physical parameters (AGARD Aeroelastic CT6, CI55 from [77]) describing the motion of the airfoil are:

α_m	=	$-0.21°$	mean angle of attack,
α_0	=	$1.01°$	amplitude of pitch,
ω	=	$2\pi f$	angular frequency,
f	=	$34.4\,\mathrm{Hz}$	frequency,
c	=	$0.500\,\mathrm{m}$	chord length,
b	=	$0.250\,\mathrm{m}$	semi-chord length,
k_c	=	$\frac{\omega c}{2U_\infty} = 0.202,$	reduced frequency,
p_∞	=	$133912\,\mathrm{N/m^2}$	free stream pressure,
M_∞	=	0.796	free stream Mach number,
U_∞	=	$\frac{\omega c}{2k_c} = 267.50\,\mathrm{m/s}$	free stream velocity.

The airfoil is rotated about the pivot point $x = 0.248c$. Since the motion of the airfoil is prescribed, there is no feedback from the fluid to the airfoil and a one-way coupling is used every five fluid time steps. That is, the new position of the airfoil is determined every $5\Delta t_{fluid}$, with quadratic interpolation of the airfoil's position at the intermediate time steps. This strategy reduces the number of times the mesh movement solver is invoked.

The accuracy of the calculation strongly depends on the resolution of the shock waves present in the flow. For enhanced accuracy, anisotropic mesh adaptation is

used on an unstructured grid with triangular elements for the calculation of the initial steady solution at a fixed zero angle of attack [51]. This adapted grid is used later for the unsteady simulation, although it is not optimal since the shocks move along the surface of the airfoil. Ideally, unsteady mesh adaptation (not available) should be used in order to follow the flow features and to provide the most accurate numerical solution. The final adapted mesh at $t = 0\,\mathrm{s}$, $\alpha = 0°$ is shown in figure 5.2.

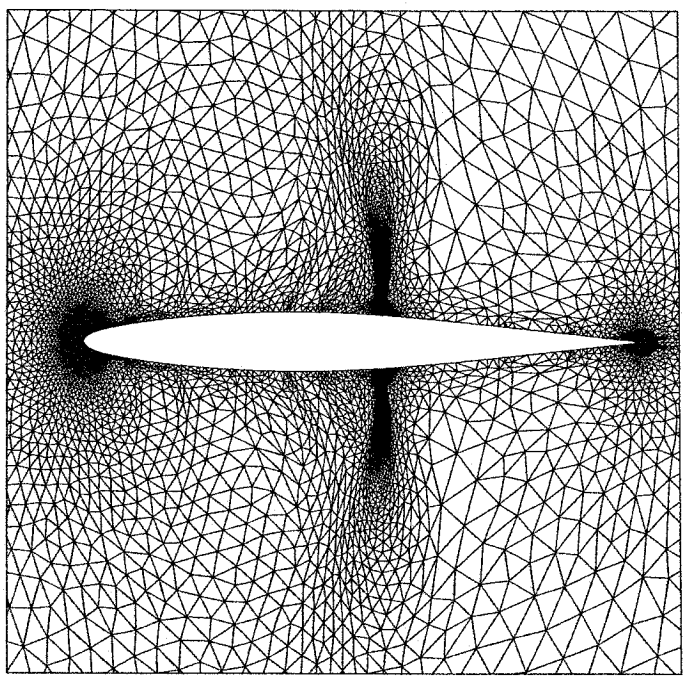


Figure 5.2: Zoom of the Adapted Mesh for the NACA64A010 Airfoil at $t = 0\,\mathrm{s}$, $\alpha = 0°$.

Care was taken when adapting the mesh for the steady-state solution. Small asymmetries in the adapted anisotropic unstructured mesh can lead to a slight asymmetry in the solution due to the great sensitivity of the position of the shocks on the pressure and the suction sides of the airfoil. In practice, such asymmetries are transitory since the process of mesh adaptation is iterated until both the grid and

the solution have converged, yielding, in this case, a symmetric optimal grid and solution at convergence. However, since it is not feasible to adapt the mesh during the unsteady simulation, the process of mesh adaptation is purposely incompletely converged. Why? If the adapted elements are clustered in too thin a band around the shock, the moving shock will leave this region of highly stretched cells and the accuracy of the unsteady solution will be compromised. Although not perfect, it is better, on average, to use a broader region of elements around the shock position so as to better represent the moving shock in time.

To circumvent the risk of introducing undesired mesh-dependent asymmetries in the steady-state solution, the steady solution is first computed on the half-domain, for $y \geq 0$, using anisotropic mesh adaptation to improve the quality of the grid, and then the half grid is mirrored to obtain the full mesh for the unsteady problem. This operation guarantees symmetry in the steady solution used as the initial condition for the unsteady simulation and eliminates the possibility of introducing a small phase error in the lift in the initial solution. In practice, these errors are small and such a manipulation is not required. Here, however, for the purpose of validation, it was decided to suppress this known source of error.

Figure 5.3a shows the hysteresis curve c_L vs α for an inviscid calculation at $M = 0.796$, but with $\alpha_m = 0°$, with comparison to experimental data [76, 77] (without wind-tunnel corrections). The solution is also in agreement with the numerical calculations by Alonso and Jameson [78]. The non-dimensional time step is $\Delta t = 0.081$, corresponding to 192 steps per cycle. The periodic unsteady state is observed after about four cycles of forced oscillations, as can be observed in figure 5.3 for the evolution of c_L and c_D in time. It is interesting to note the effects of the unsteadiness in the lift diagram when compared to the theoretical result $c_L = 2\pi\alpha$ for a steady calculation (which is still quite accurate for an amplitude of $1°$ in the transonic regime).

That is, when the oscillating airfoil returns to zero angle of incidence, the lift value is no longer zero, since the shock waves are no longer aligned, due to the phase difference between the instantaneous angle of attack and the lift. This asynchrony in the position of the shocks is at the source of the presence of drag for this unsteady inviscid flow. (There is no viscous drag; only shock-induced drag is present.) The drag coefficient, c_D, is plotted in figures 5.3c-d, although no experimental nor numerical data are available for comparison. It is noted that the frequency of the drag coefficient is twice the frequency of the lift coefficient. This observation is correct since the drag repeats its cycle during each upward and downward pitching motion of the airfoil. Mach contours of the solution are displayed in figure 5.4 to show the evolution of the solution at four instants in time, once the periodic state has been achieved. Mach number and C_p profiles on the surface of the airfoil are plotted in figure 5.5, at the same instances in time. The small oscillations in the curves are due to noise in the solution at the lowest coefficient of artificial dissipation on the unstructured grid.

The shape (width and inclination) of the hysteresis curve (ellipse) is very sensitive to phase errors between the motion of the airfoil and the lift response. Figure 5.6 shows the hysteresis curves for $c_L(t + \delta t)$ versus $\alpha(t)$ for small phase changes $\delta t = 0, \pm\Delta t, \pm 2\Delta t$ (the time step is $\Delta t = 0.081$). It can be observed that a small phase lag in the lift response (or the position of the airfoil) changes the width of the hysteresis curve. Phase errors can be easily introduced at the start of the unsteady calculation from the steady-state solution. Further, such phase errors seem to persist throughout the unsteady calculation.

Beyond the asymmetries in the grid, another source of error in the phase comes from the initial perturbation of the airfoil at $t = 0$. The airfoil, initially at rest at $\alpha = 0°$, is suddenly set into forced sinusoidal motion, resulting in a discontinuity in the angular velocity $\frac{d\alpha}{dt}$ at $t = 0$. Perhaps the motion should be set in a smoother

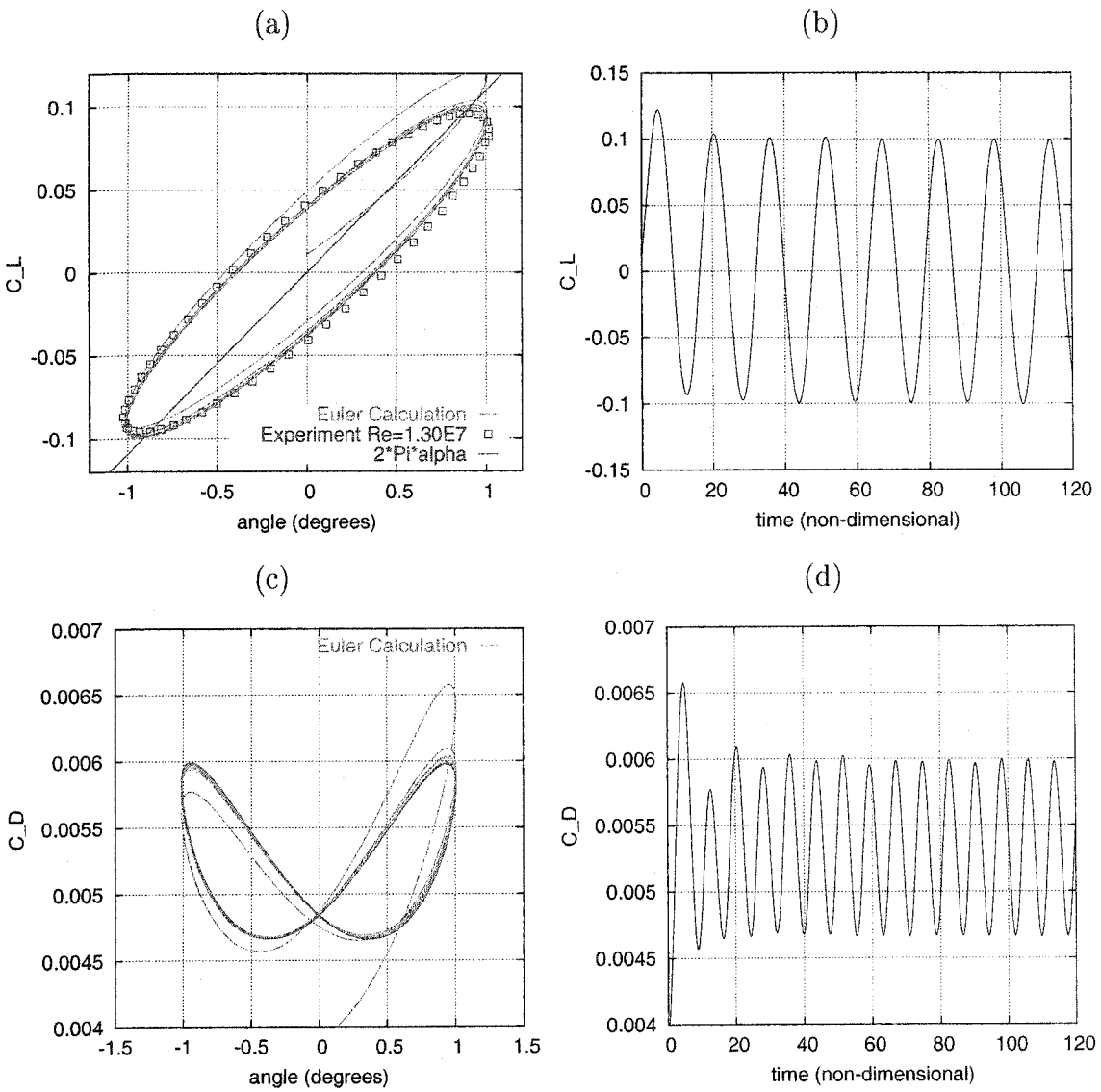


Figure 5.3: Oscillating NACA64A010 Airfoil: (a) Hysteresis Curve c_L vs α; (b) Evolution of c_L in Time; (c) Hysteresis Curve c_D vs α; (d) Evolution of c_D in Time.

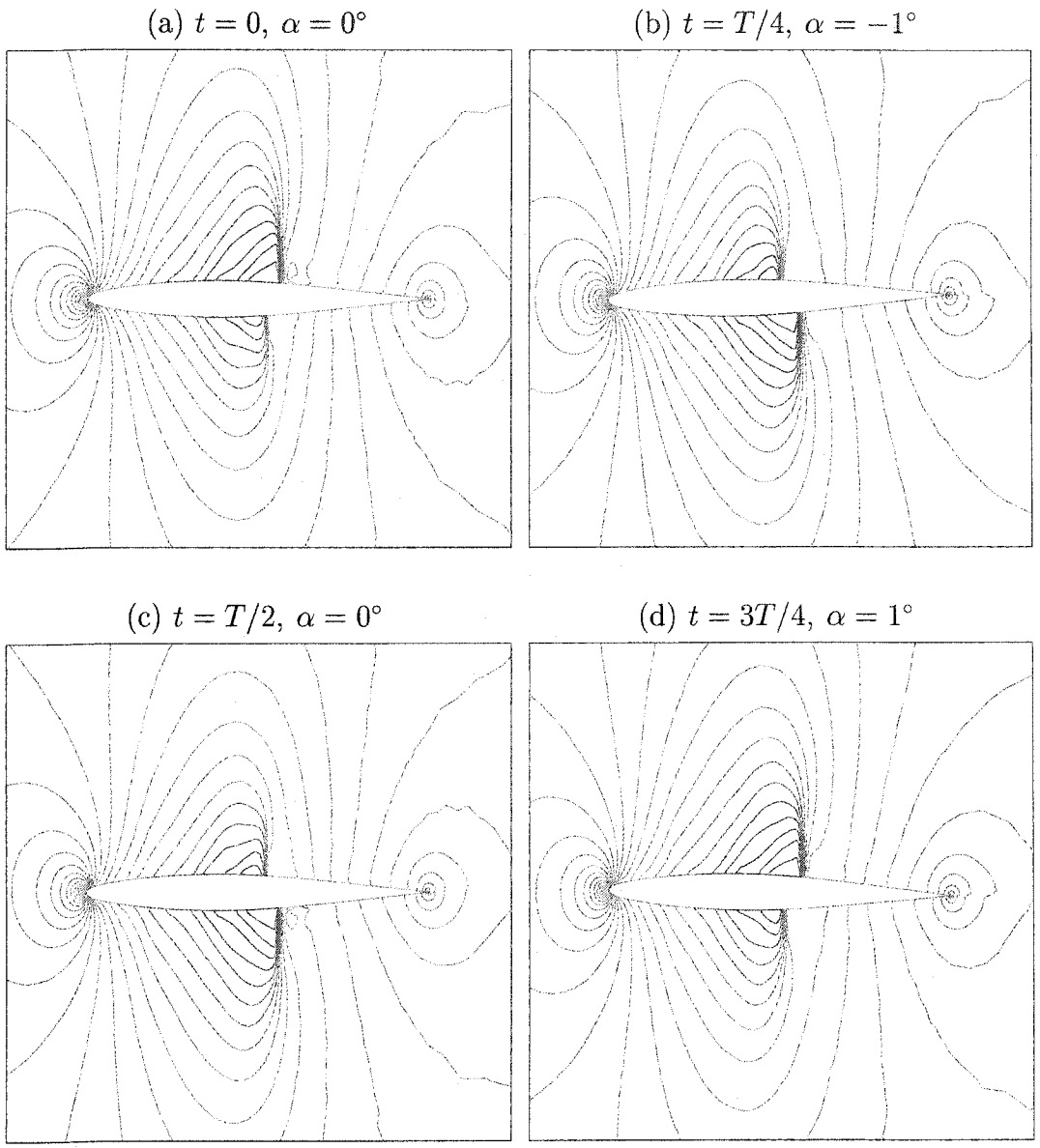
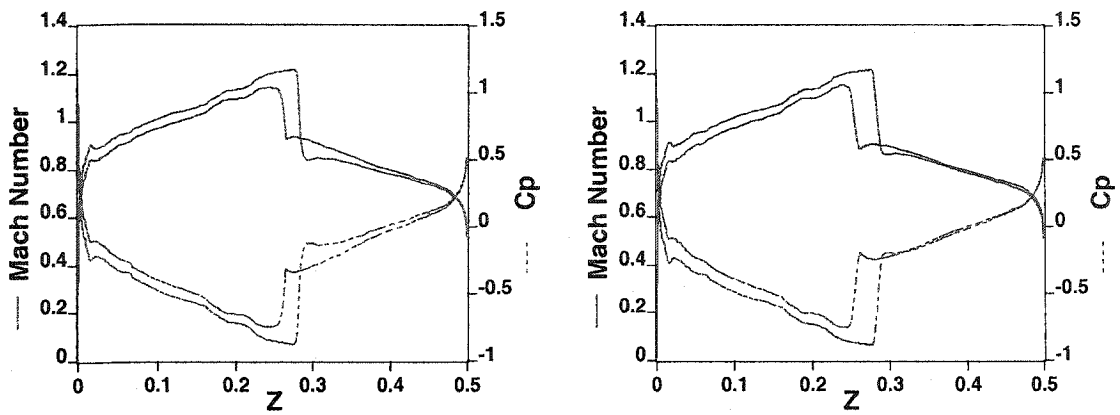


Figure 5.4: Mach Contours for the Oscillating NACA64A010 Airfoil at Four Different Times During One Cycle of Oscillation.

(a) $t = 0$, $\alpha = 0°$ (b) $t = T/4$, $\alpha = -1°$



(c) $t = T/2$, $\alpha = 0°$ (d) $t = 3T/4$, $\alpha = 1°$

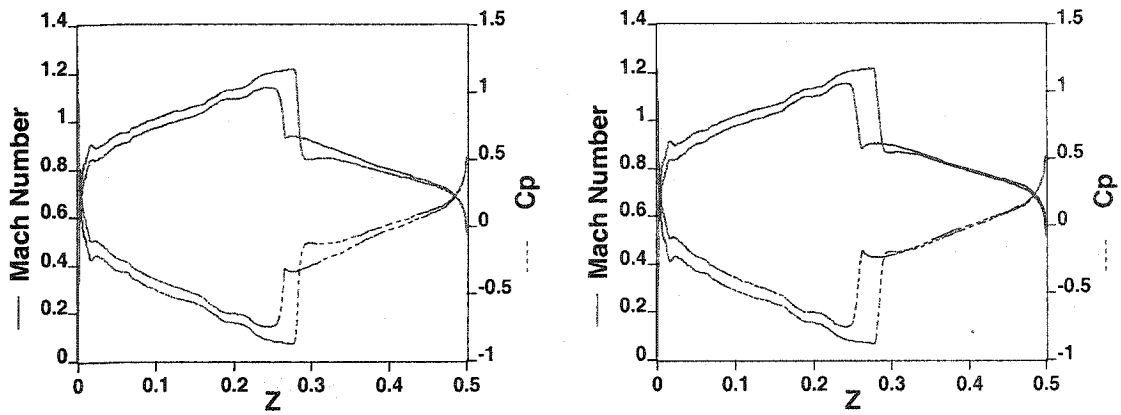


Figure 5.5: Mach Number and C_p Profiles for the Oscillating NACA64A010 Airfoil at Four Different Times During One Cycle of Oscillation.

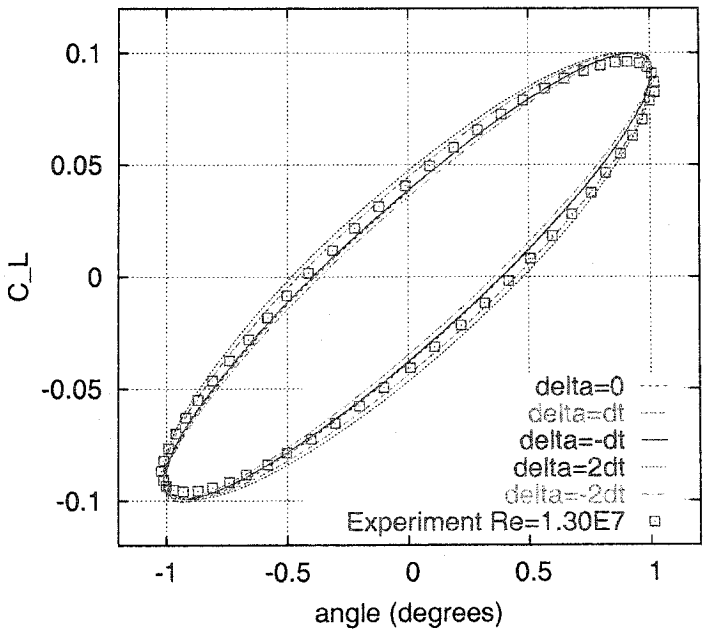


Figure 5.6: Effect of Phase Error on Hysteresis Curve for the Oscillating NACA64A010 Airfoil.

fashion. In the wind-tunnel experiment [77], the airfoil is started from rest at an initial angle $\alpha_m = -0.21°$, but such a configuration should have no effect on the evolution of the unsteady solution.

The application of the numerical time-integration scheme must not be neglected. When a high-order multi-step scheme is used, the solution must be available over two or three time levels. At startup time, the solution is known at only one time level, at $t = 0$, and a first-order scheme is used for the first time step. Thereafter, a second-order scheme may be used since the solution is known over at least two time levels. To improve the accuracy of the solution, the initial time step may be halved so as to reduce the size of the truncation error in the numerical scheme. However, for this problem, the time step is already sufficiently small that different startup strategies do not seem to affect the quality of the solution.

The order of interpolation used in the calculation of the ALE derivative $\frac{d\mathbf{x}}{dt}$ (equation (2.10)) is varied to observe its effects on the solution. The primitive variables are differentiated in time using Gear's 2^{nd} order scheme, while a 1^{st}, a 2^{nd}, or a 3^{rd} order scheme may be used to compute $\frac{d\mathbf{x}}{dt}$. The components $\frac{dy}{dt}$ (transverse to the flow) and $\frac{dz}{dt}$ (along the flow) are plotted vs the node number for the nodes on the moving boundary. There is a notable difference between the 1^{st} order approximation and the 2^{nd} and 3^{rd} order approximations, as observed in figure 5.7. (The 3^{rd} order approximation is nearly the same as the 2^{nd} order approximation and is not plotted in the figure.) However, for the overall solution (hysteresis curves c_L vs α and c_D vs α) no differences are visible on this plotting scale. It is noted that $\frac{d\mathbf{x}}{dt}$ plotted in figure 5.7 is computed a posteriori from the mesh positions over several time levels, hence all three solutions have been computed over the same sequence of grids since the motion is imposed. Only the term $\frac{d\mathbf{x}}{dt}$ has changed in the governing equations, suggesting the minimal effect of the order of its approximation. Even if these effects are small between the 1^{st} and the 2^{nd} order approximations, these terms must however not be neglected from the equations. In summary, a 1^{st} order approximation for the grid derivatives appears accurate (at least for forced motion) and has the advantage of requiring only the storage of the moving grid at two time levels.

The unsteady computation is repeated using the assumption of constant total enthalpy, despite the inherent unsteadiness of the flow due to the motion of the airfoil. This assumption may seem reasonable since the variations in the total enthalpy are about $\pm 1\%$ from the free stream value in the previous calculation. The response of the airfoil, in terms of lift and drag, is shown in figure 5.8 and compared with that obtained when solving the energy equation. The results display some small but noticeable sensitivity to the exchanges of energy between the moving airfoil and the

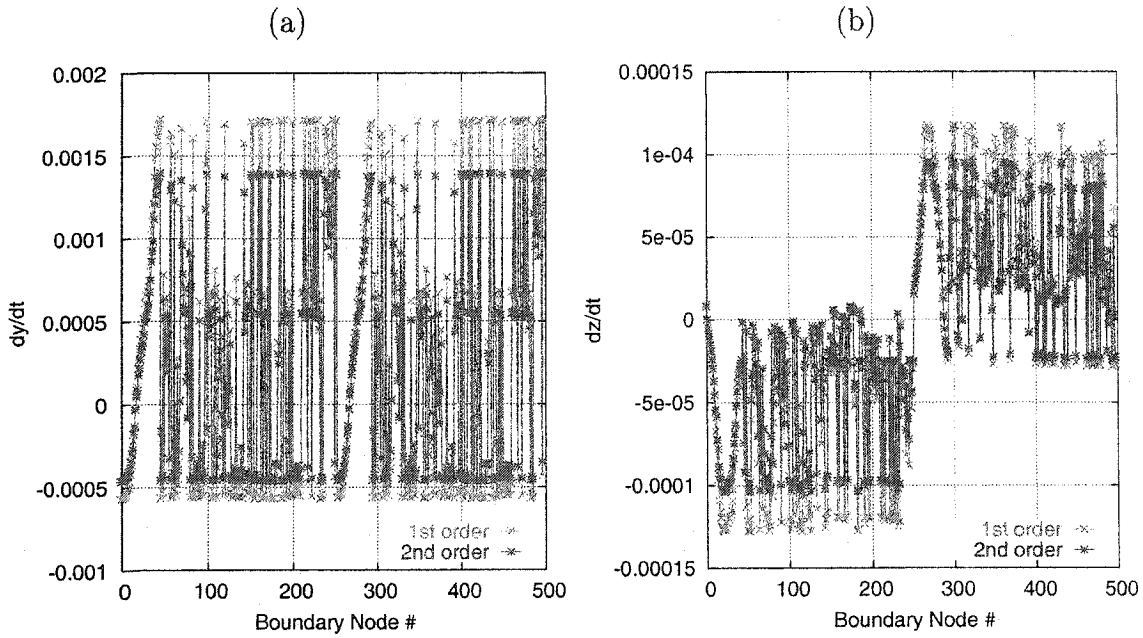


Figure 5.7: Approximation Order for ALE Derivative at One Instant in Time: (a) Vertical Grid Velocity on Moving Boundary; (b) Axial Grid Velocity on Moving Boundary.

fluid, even when these are small, in the form of extra damping when the total enthalpy is assumed constant. Although the option of constant total enthalpy is appealing from the numerical point of view, since it reduces the computational costs, the effects of this strategy must first be studied before adopting it, to prevent adverse effects on the quality of the solution.

5.3 Oscillating Cylinder

In this test case, a circular cylinder is mounted on a spring, as shown in figure 5.9, with the cylinder free to oscillate in the direction transverse to the flow. The goal is to reproduce the *lock-in* phenomenon which occurs when the spring-mounted cylinder

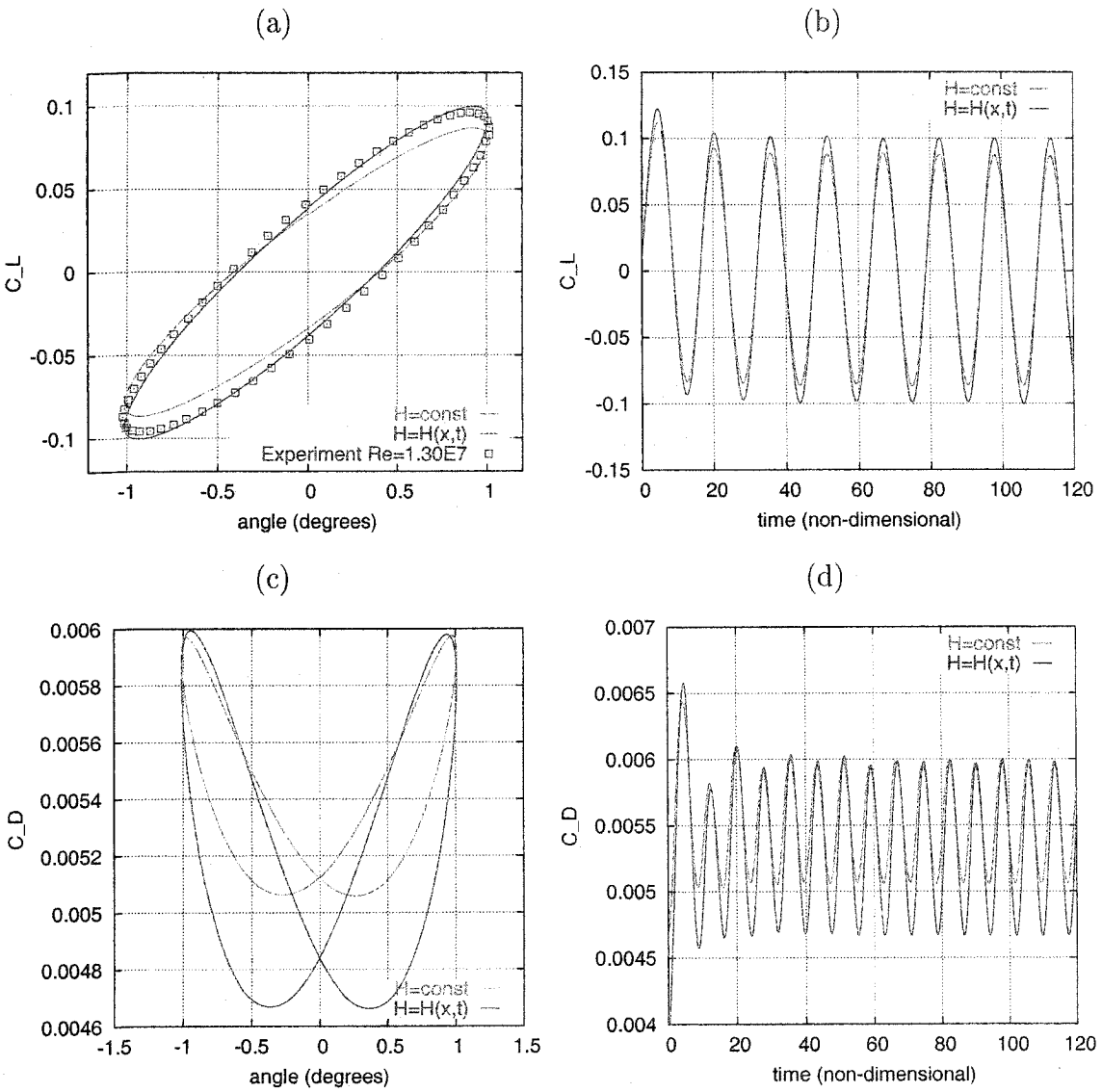


Figure 5.8: Oscillating NACA64A010 Airfoil With Constant Total Enthalpy: (a) Hysteresis Curve c_L vs α; (b) Evolution of c_L in Time; (c) Hysteresis Curve c_D vs α; (d) Evolution of c_D in Time.

exhibits sustained flow-induced oscillations when the frequency of the shedding vortices approaches the natural frequency of oscillation of the spring-cylinder system [79]. The stiffness k of the spring and the mass m of the cylinder are chosen such that the natural frequency $f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$ of the cylinder matches the numerically computed shedding frequency f_v of the vortices. In all the cases studied, there is no mechanical damping ($c = 0$). The only damping present in the system comes from the fluid in terms of the aerodynamic damping.

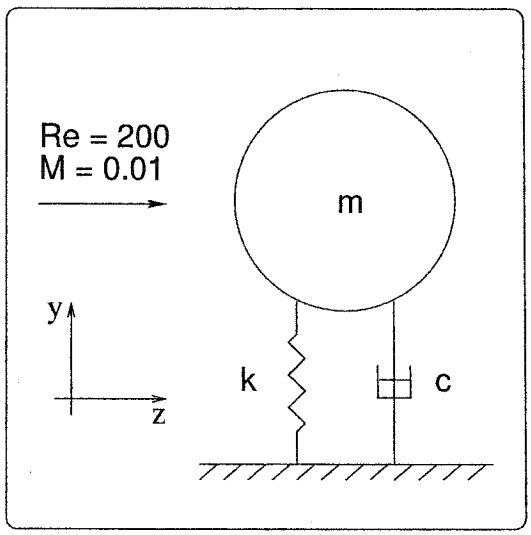


Figure 5.9: Configuration of Oscillating Cylinder.

Another objective of this test case is to demonstrate the capabilities of the fluid-structure coupling technology for viscous flows. It will be confirmed that the shear stresses play an important role in determining the response of the cylinder, as opposed to computing the aerodynamic forces based solely on the pressure. In fact, the coupling poses no difficulty other than the increase in computational time for solving the Navier-Stokes equations. The flow studied here is laminar, but no difficulties are foreseen for turbulent flows once these capabilities become available in the flow solver for unsteady flows.

The role of the time step will also be examined, with its relevance to the accuracy of the solution. A Fourier analysis of the lift response will be performed in order to guide the selection of a cost-effective time step without undermining the accuracy of the solution. Finally, the impact of the level of coupling will be discussed.

The free stream Mach number is $M = 0.01$, which is a reasonable assumption to model an incompressible flow using a compressible flow code. At ambient air temperature $T = 288\,\mathrm{K}$, the fluid velocity is $U_\infty = 34.02\,\mathrm{m/s}$, the density is $\rho_\infty = 1.2096\,\mathrm{kg/m^3}$, and the fluid viscosity is $\mu_\infty = 0.00017887\,\mathrm{Pa \cdot s}$. The diameter of the cylinder which corresponds to a Reynolds number of 200 is $D = 0.008693\,\mathrm{m}$.

Although the flow is essentially incompressible at such a low Mach number, the energy equation is nonetheless solved for, since the flow is unsteady. The Navier-Stokes equations are integrated in time using the second-order Gear scheme until a periodic state is observed, at which time the Strouhal number is measured from the unsteady lift distribution on the cylinder. No artificial dissipation is required in the momentum and the energy equations since the natural viscosity is significant at $Re = 200$. The second-order form of the artificial dissipation is used to stabilize the continuity equation.

At a flow Reynolds number of 200, vortices shed in the wake of a fixed cylinder at a Strouhal number $St = \dfrac{fD}{U_\infty}$ of approximately $St_\circ = 0.20$, where f is the shedding frequency of the vortices and D is the diameter of the cylinder. More precisely, using Roshko's relation [56]:

$$St_\circ = 0.212\left(1 - \frac{12.7}{Re}\right), \tag{5.6}$$

a curve-fit expression based on experimental data for the Strouhal number past a fixed cylinder, one obtains $St_\circ = 0.1985$ at $Re = 200$. Calculations on a fixed cylinder give a measured Strouhal number of $St_\circ = 0.1977$, which is in agreement with experimental and other numerical data at $Re = 200$.

A grid refinement study was performed until the unsteady lift and drag on the fixed cylinder became more or less independent of the mesh. Also, different domain sizes were used, the conclusion being that a sufficiently large domain was needed since far-field boundary conditions are approximated by the constant free stream flow in the code. The computational domain extends from $[-5D, -10D] \times [30D, 10D]$, with the cylinder centered at $(0, 0)$. Ideally, a smaller domain could be used, pending the implementation of non-reflective boundary conditions in the flow solver. The final grid consists of 19619 nodes in the plane and 19272 quadrilateral elements. A zoom of the computational grid near the cylinder is shown in figure 5.10. In the region downstream of the cylinder, the grid clustering ranges from 24 to 16 grid points per wavelength ($\approx 5D$) of the shedding vortices. There are 160 nodes on the surface of the cylinder. The same grid is used in all computations.

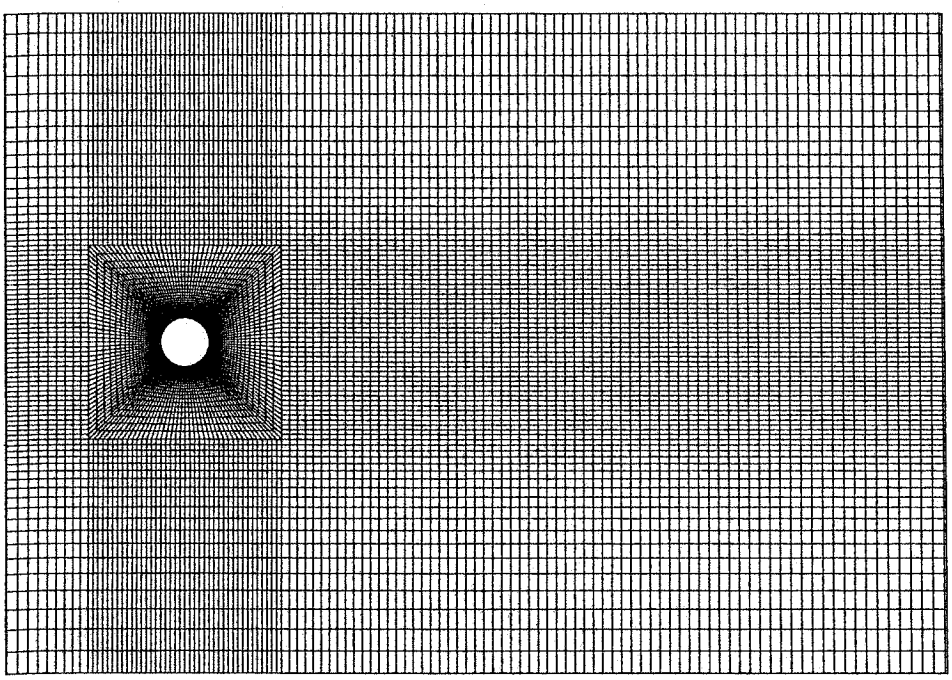


Figure 5.10: Zoom of the Computational Grid for the Oscillating Cylinder Test Case (19619 nodes in the plane, 19272 elements).

In this problem, mesh adaptation is not used since the flow is inherently unsteady and the technology is not available for time-accurate unsteady mesh adaptation. Mesh adaptation could be used in a static fashion every few time steps, but time-accuracy cannot be guaranteed following the interpolation of the solution over several grids. The fine mesh used in this case already yields accurate solutions.

The computations are carried out in two steps: first, the unsteady solution is computed past the fixed cylinder, then this solution is used to trigger the motion of the cylinder. The unsteadiness in the flow field is initiated by perturbing the initial flow field in an asymmetric fashion in order to accelerate the establishment of the vortices. A few cycles, less than 20, suffice to wash out the solution transients resulting from the initial perturbation. The lift over the stationary cylinder is plotted in figure 5.11 as a function of time. The inset in the figure shows the starting lift evolution following the initial perturbation of the flow, while the full-size graph shows that a constant amplitude periodic lift has been obtained at a much further time. Mach number and pressure contours are plotted in figure 5.12 at four instants during one period at $t = 0$, $t = T/4$, $t = T/2$, and at $t = 3T/4$.

The structural model consists of the single ordinary differential equation

$$m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky = F(t), \tag{5.7}$$

where y is the elevation of the cylinder relative to its position at rest, m is the mass of the cylinder, c is the damping coefficient (set to zero), k is the stiffness of the spring, and $F(t)$ is the external aerodynamic force, being simply the lift, acting in the upward direction. The natural angular frequency of the system is $\omega_n = \sqrt{\frac{k}{m}}$. At lock-in, $\omega_n = \frac{2\pi St\, U_\infty}{D}$.

Fully-coupled aeroelastic calculations are conducted at various natural frequencies of the cylinder, starting from the established periodic unsteady solution past

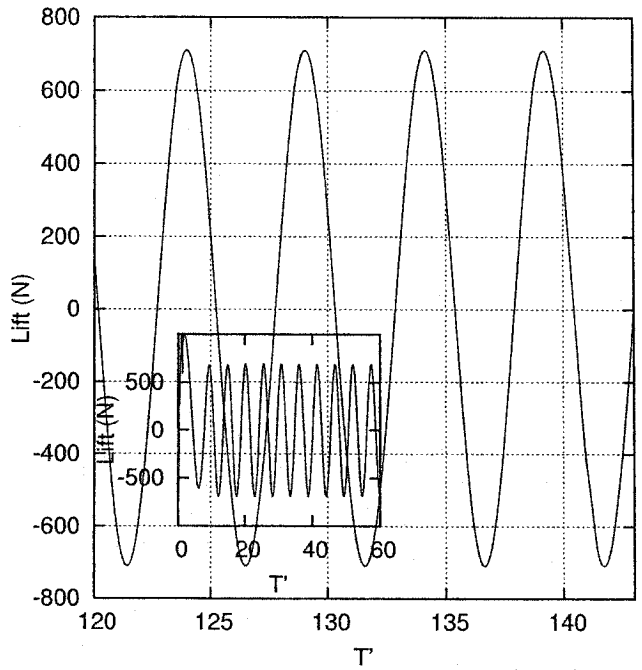


Figure 5.11: Unsteady Lift Distribution on a Stationary Cylinder at $Re = 200$.

the stationary cylinder. These calculations require the simultaneous solution of the Navier-Stokes equations and of the spring model equation (5.7) using the coupling algorithm developed in the chapter 4. A tight coupling is used with multiple (maximum 10) coupling instances per time step. However, in this case, no surface interpolation is required since the total lift, computed from within the flow solver, is passed directly to the structural equation. The time step is selected on the basis of 100 equal time steps per period of shedding of the vortices past the stationary cylinder.

The parameters for the cylinder-spring system are chosen as $m = 0.00001\,\text{kg}$ with k ranging from $140\,\text{N/m}$ to $360\,\text{N/m}$. The authors of [80] suggest "a high value for the mass parameter $\rho_m = \dfrac{m}{\rho D^2}$, so that the forces induced by the fluid become irrelevant with respect to the mechanical forces developed by the spring."[1] A value of

[1]In 3-D, the mass parameter is written as $\rho_m = \dfrac{m}{\rho b D^2}$, where b is the span of the cylinder in the $z-$direction. Here $b = 1\,\text{m}$.

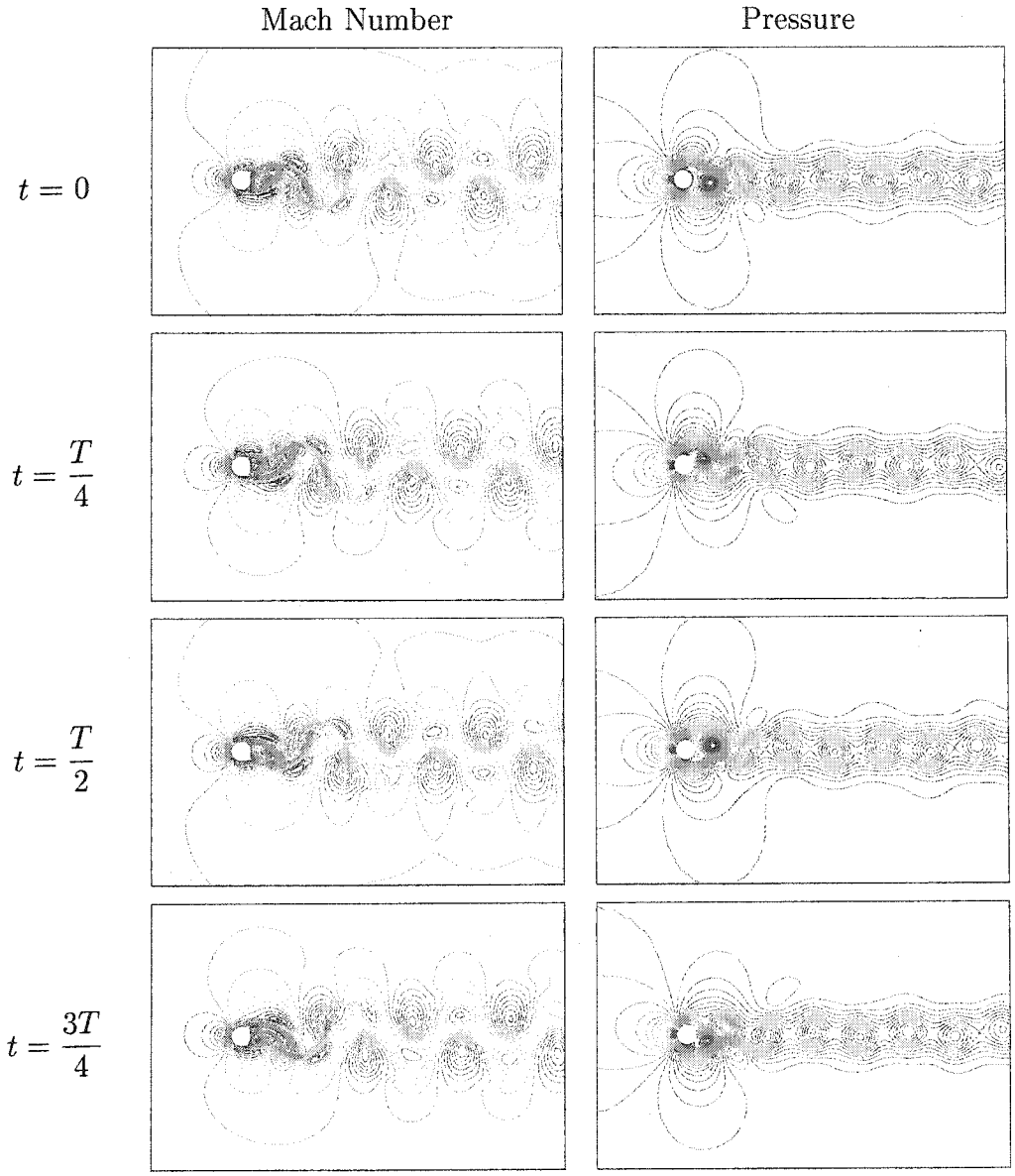


Figure 5.12: Mach Number and Pressure Contours at Various Instants for the Flow Past a Fixed Cylinder at $Re = 200$.

$\rho_m = 1000$ was used in [80], while a low value of $\rho_m = 10.94$ is used in this example. This light mass of the cylinder causes the cylinder to respond rapidly to the flow: a periodic motion is observed after a few cycles of oscillations only. The response time is much longer at a larger mass parameter (stiffer spring and heavier cylinder).

The lift over the cylinder is plotted in figures 5.13 and 5.14 for increasing values of the natural frequency of the cylinder (varying k, holding m). It is interesting to note that the coupled lift response is the superposition of several harmonics. Near $f_v/f_n \approx 1$, or $f_n/f_o \approx 0.92$ (where f_o is the frequency corresponding to the measured $St_o = 0.1977$ for the fixed cylinder), it can be seen that the modes coalesce into nearly one mode of constant amplitude, confirming the observation of the lock-in phenomenon, which is not as easily observed at such a low mass parameter.

The lock-in phenomenon near the resonant frequency can be observed in figure 5.15. At lock-in, the flow vortex frequency f_v approaches the natural frequency f_n of oscillation of the cylinder and $f_v/f_n \approx 1$. This occurs at about $f_n/f_o \approx 0.92$, at a value somewhat less than the resonant frequency for the fixed cylinder. Away from the resonant frequency, the shedding of the vortices is less and less synchronized with the motion of the cylinder. Finally, the amplitude of the motion is about $0.55D$, see figure 5.16, which is in accordance with the other published works cited [56, 80].

5.3.1 Effect of the Viscous Shear Stresses

As a first analysis, the viscous shear stresses are neglected in the calculation of the lift. Only the pressure forces are passed to the structural solver, although a viscous solution is actually computed. The amplitude and the lift are plotted as functions of time in figure 5.17 with and without the contribution of the shear stresses. It is reassuring to observe that the inclusion of the viscous terms dampens the amplitude of the motion,

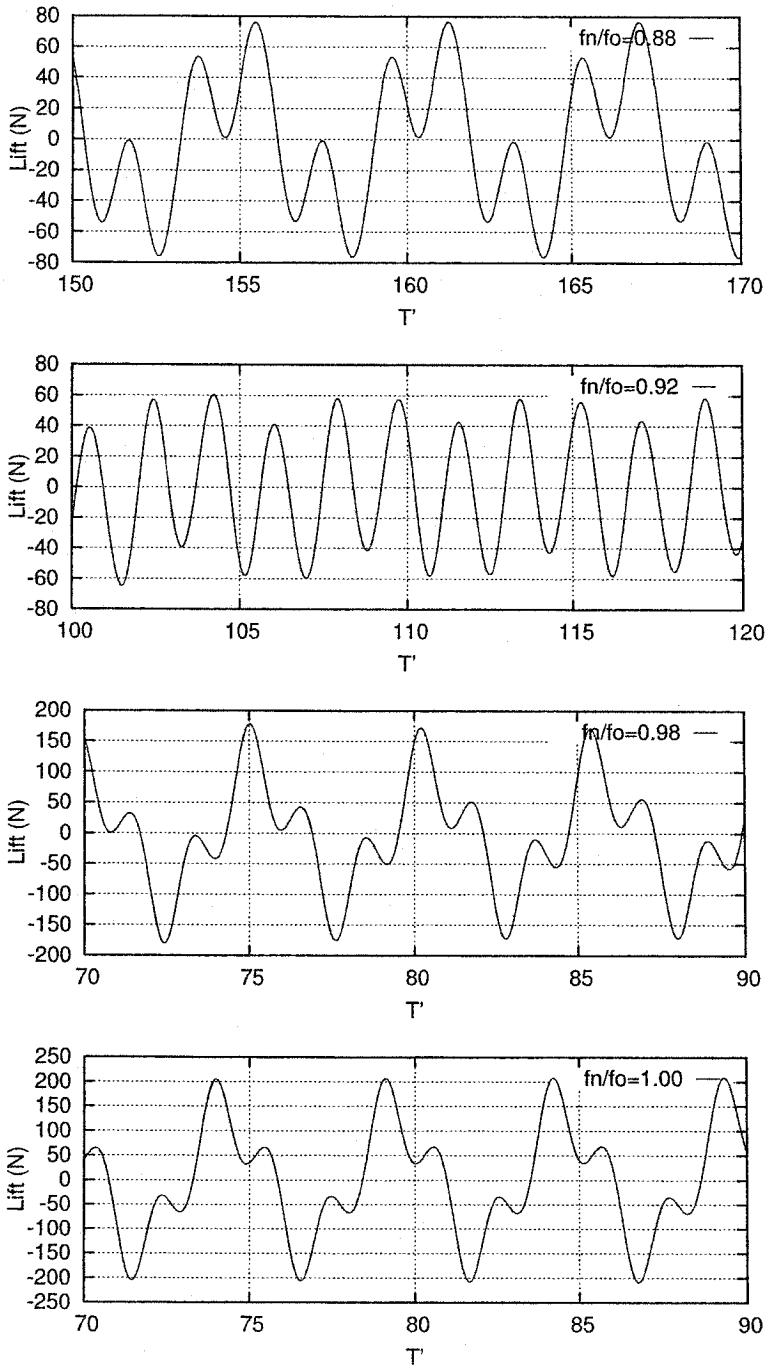


Figure 5.13: Unsteady Lift on the Cylinder at Various Structural Frequencies (Below the Resonant Frequency of the Fixed Cylinder).

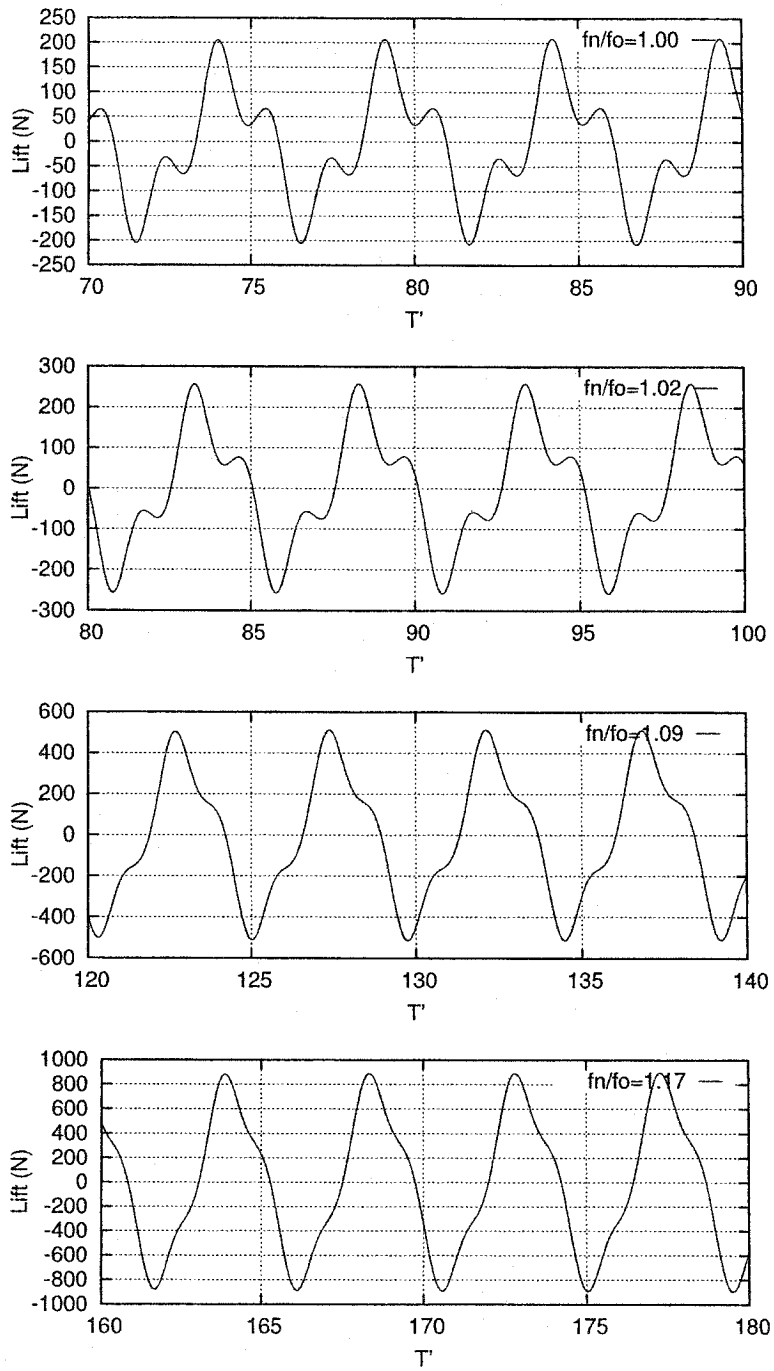


Figure 5.14: Unsteady Lift on the Cylinder at Various Structural Frequencies (Above the Resonant Frequency of the Fixed Cylinder).

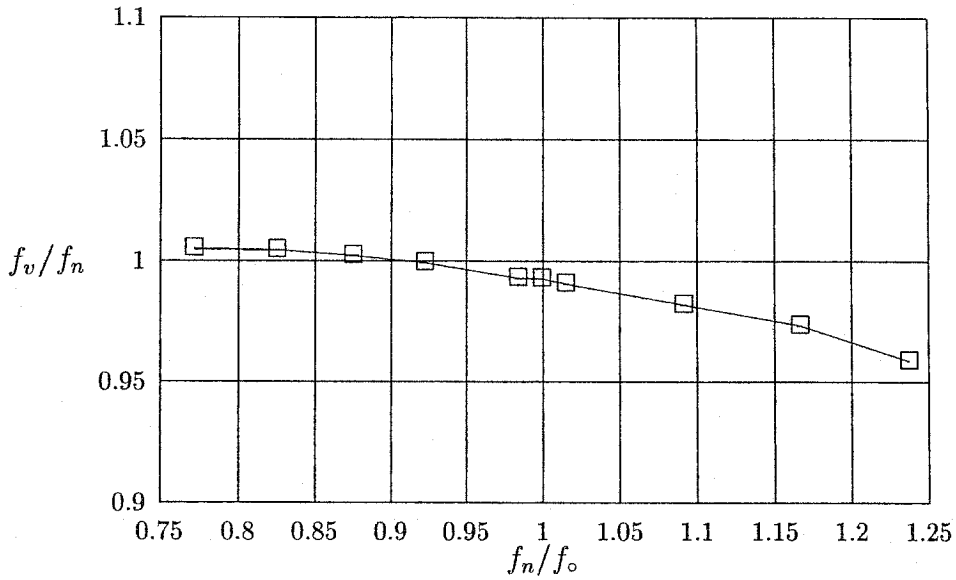


Figure 5.15: Ratio f_v/f_n of the Flow Vortex Frequency to the Cylinder Natural Frequency vs the Cylinder Natural Frequency Normalized by the Resonant Frequency.

as anticipated based on physical intuition for a viscous solution (despite the fact that the magnitude of the lift is larger when the viscous effects are considered). At such a low Reynolds number, the viscous effects have a noticeable impact on the response of the structure. However, it may be justified that at higher Reynolds numbers, inviscid solutions may be rightfully used to give a safe estimate of the aeroelastic stability limits.

5.3.2 Effects of the Time Step on the Accuracy

The accuracy of the solution of the coupled problem is closely related to the size of the time step Δt. In general, the smaller the time step is, the easier it is to achieve simultaneous convergence. As an illustration, a reference time step is chosen in an *ad hoc* fashion such as to use 100 equally spaced time steps per period of oscillation of the

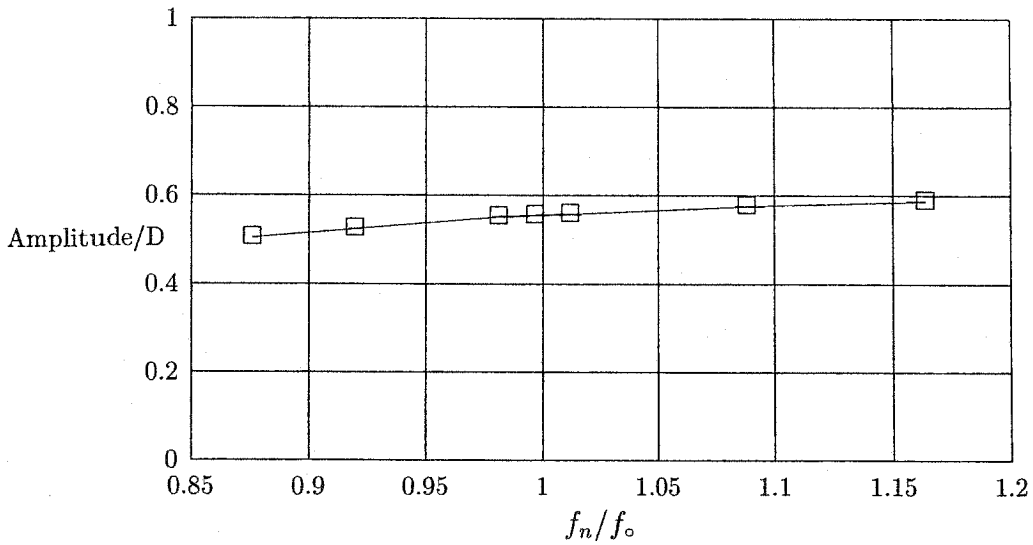


Figure 5.16: Amplitude of Motion of the Cylinder vs the Cylinder Natural Frequency Normalized by the Resonant Frequency.

vortex-shedding frequency for the fixed cylinder. The goal is to select a reasonably large time step that guarantees accuracy, but that is not too small to unnecessarily undermine the benefits of the implicit flow solver and increase the total simulation time.

At a Strouhal number of approximately $St \approx 0.1977$, the non-dimensional period is $T = 1/St \approx 5.06$, giving $\Delta t_{ref} \approx 0.0506$ for the flow solver. This reference time step is used in all the calculations in which the cylinder oscillates. The unsteady lift on the cylinder is shown in figure 5.18 for solutions computed on the oscillating cylinder at the natural frequency $f_n = 0.9844 f_\circ$ for factors 1/4, 1/2, 1, and 2 of the reference time step. These time steps correspond to 400, 200, 100, and 50 time steps per period of oscillation, respectively. A large discrepancy is observed for the lift response at $2\Delta t_{ref}$. Even the unit reference time step appears to be slightly too large.

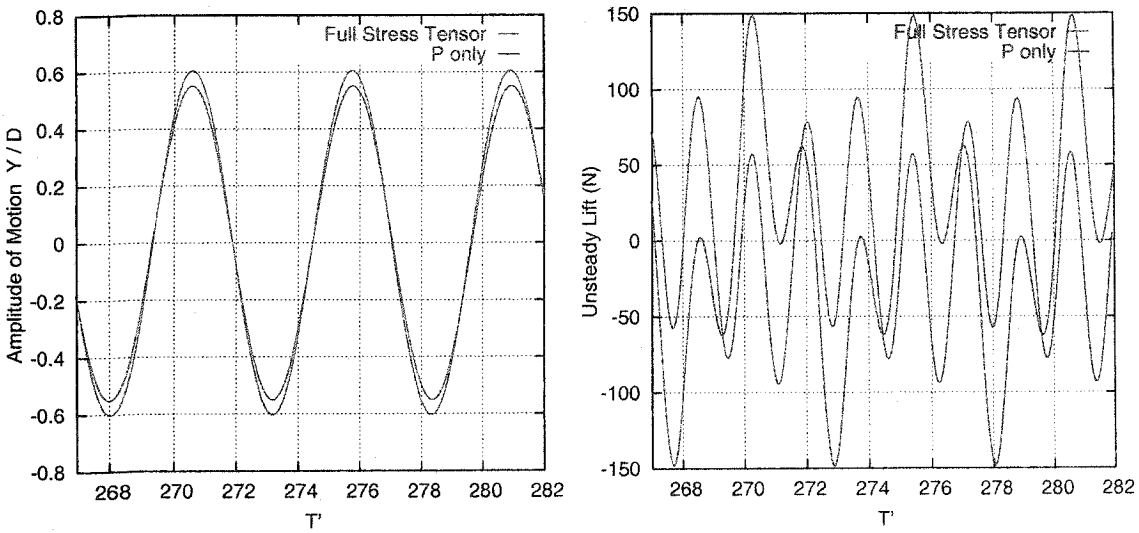


Figure 5.17: Effect of the Viscous Shear Stresses on the Oscillating Cylinder.

A Fourier decomposition is performed to identify the periodic nature of the observed coupled lift response. The result of the decomposition, shown in figure 5.19, indicates that the lift response includes strong first and third harmonic components, of the form

$$L(t) = a_1 \cos(\omega t + \phi_1) + a_3 \cos(3\omega t + \phi_3) . \tag{5.8}$$

The presence of the third harmonic in the lift response is also observed in [56], but is not as strong at a larger value of the mass parameter. At 50 time steps per period of the fundamental, at $2\Delta t_{ref}$, about 17 time steps are used to resolve the third harmonic. This is insufficient and explains the discrepancies in the solutions at $2\Delta t_{ref}$. Ideally, 30 time steps per period of oscillation of the harmonic corresponding to the highest frequency observed (shortest period) are required to accurately capture the response.

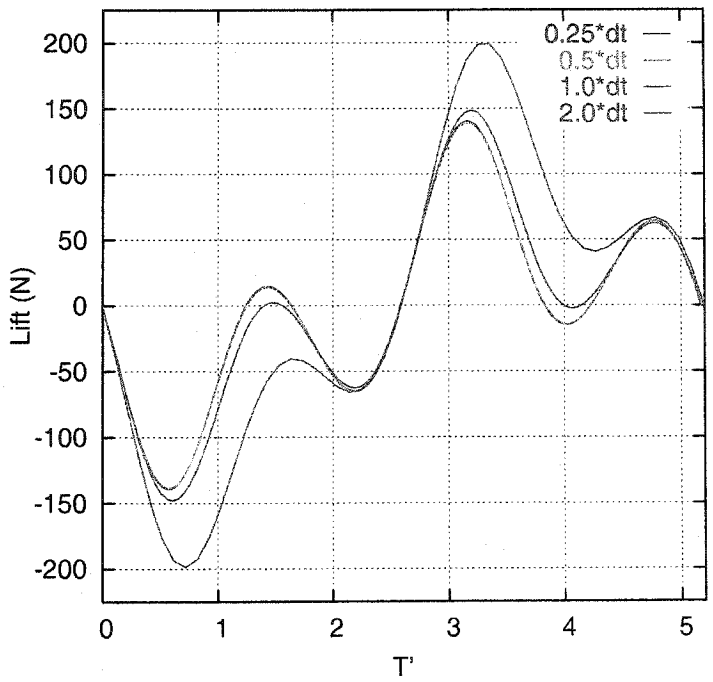


Figure 5.18: Effect of Time Step on Response of Cylinder.

5.3.3 Impact of Tightness of Coupling

It was advocated in the introduction of the thesis that frequent exchanges of the displacements and of the aerodynamic loads, as often as at every Newton iteration of the fluid solver, provide the most cost-effective means for obtaining time-accurate aeroelastic solutions. The degree of tightness of the fluid-structure coupling is now studied by varying the frequency of the solution updates, with special attention to the quality of the solution.

Other than the cost of the flow solver, the only other significant cost in the coupling is due to the mesh movement. The cost of the stress solver (integrating one second-order ordinary differential equation) is insignificant. For an implicit finite element stress solver, which can be quite expensive computationally, guidelines can be established as to the minimal level of coupling necessary to achieve a desired accu-

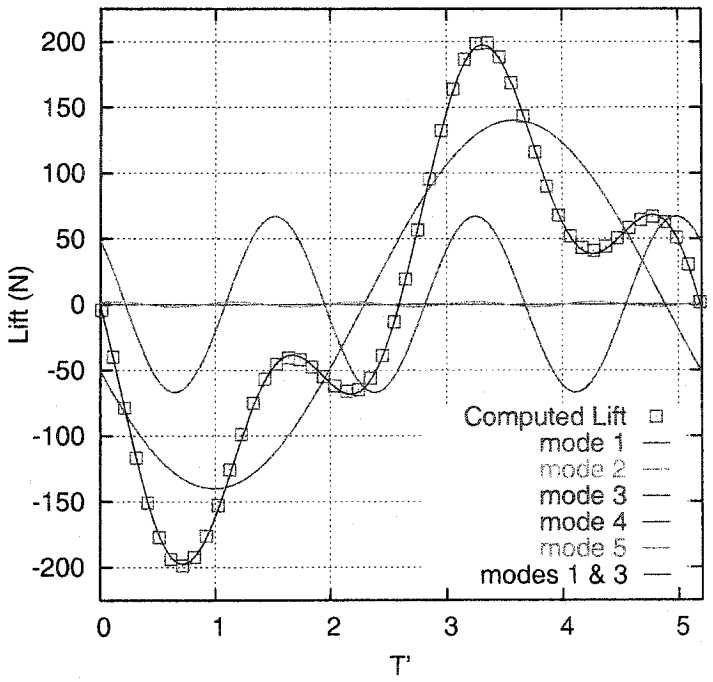


Figure 5.19: Fourier Decomposition of an Unsteady Lift Response at $2\Delta t$.

racy without penalizing the overall performance of the code. Thresholds are used to monitor the convergence of the aeroelastic system.

Comparisons are made in the following cases:

a) 10 coupling instances per Δt (every 2 Newton iterations),

b) 1 coupling instance per Δt (at 1^{st} Newton iteration),

c) 1 coupling instance per $2\Delta t$ ($\Delta t_{struc} = 2\Delta t_{fluid}$),

d) 1 coupling instance per $5\Delta t$ ($\Delta t_{struc} = 5\Delta t_{fluid}$).

As a starting point, the configuration with $f_v/f_n = 0.9954$ is selected, with as reference the solution computed with one structural update per two Newton iterations of the flow solver, case (a), for up to 20 Newton iterations of the flow solver per time-step.

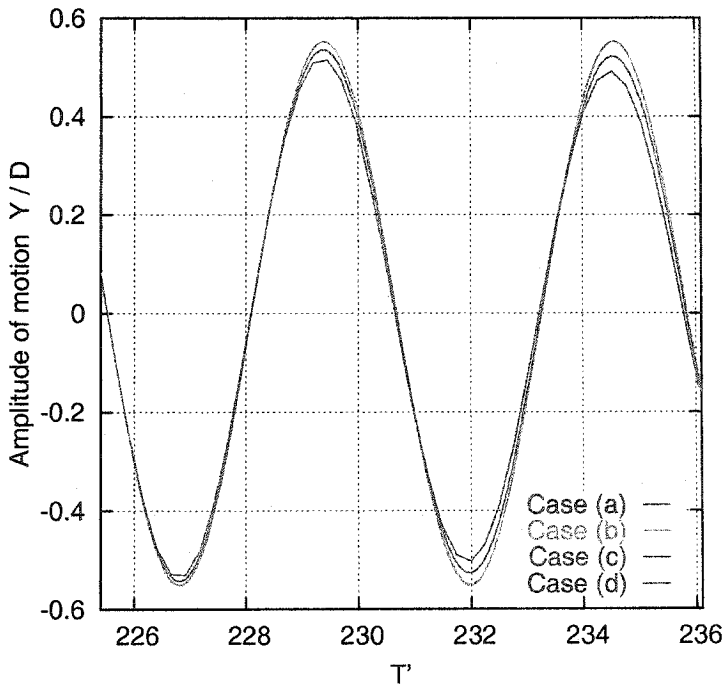


Figure 5.20: Amplitude of Motion for Oscillating Cylinder for Different Coupling Levels.

The solutions for the amplitude of the motion are indistinguishable for cases (a) and (b), as seen in figure 5.20. With the loose couplings, cases (c) and (d), the amplitude of the motion exhibits a response of lower amplitude.

The curves for the lift coefficient are perhaps more puzzling, as plotted in figure 5.21. The lift coefficient is similar for cases (a) and (b), showing that one coupling instance per fluid time step is sufficient for this problem. Subsequent coupling instances within a fluid time step do not contribute, in this case, to an improvement in the solution. (Indeed, the correction to the displacements at the second coupling instance of a time step is 6 orders of magnitude smaller than the initial displacement at that time step.) On the other hand, the lift coefficient curves show "instabilities" in the case of a loose coupling. The spikes in the curve for cases (c) and (d) occur

at every coupling instance with the stress solver and are likely an indication of the accumulated error in the coupled solutions over these time steps. These nonphysical spikes in the lift slightly retard (damp) the motion of the cylinder, but the large inertia of the cylinder is sufficient to maintain a smooth non-oscillatory motion. Such noise in the flow solution is not desired, even if the motion of the structure remains smooth.

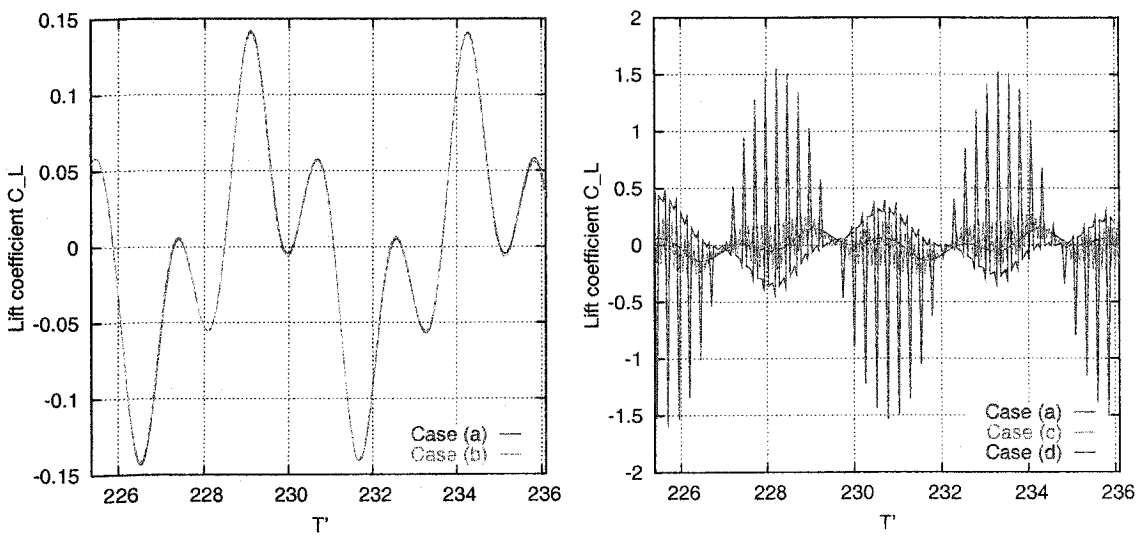


Figure 5.21: Lift Coefficient for Oscillating Cylinder for Different Coupling Levels.

To understand the phenomenon for the instabilities in the lift, the solution for case (c) is compared to a solution using a fluid time step $2\Delta t$, denoted case (e). In this case, there is one coupling instance per time step, as in case (b), but the time step has been doubled. Physically, there is still one coupling instance per time interval $2\Delta t$, but the time steps are the same for both solvers. The lift and the amplitude of the motion of the cylinder are plotted in figure 5.22. The motion is damped, as expected, but the lift response is smooth. It is quite interesting to realize that taking several small time steps to integrate the flow solution over one time step of the structure is unstable, while taking equal (larger) time steps for the fluid and

the structure is stable. This conclusion refutes the idea that sub-cycling for the flow solver is more accurate.

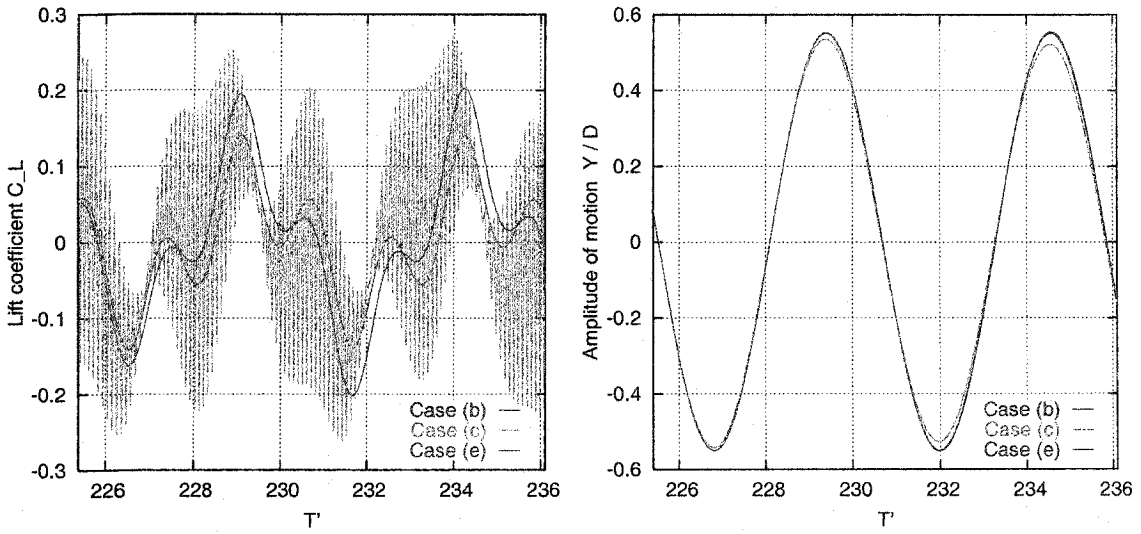


Figure 5.22: Lift Coefficient and Amplitude for Oscillating Cylinder for Coupling Levels With Different Time Steps.

A word of caution warrants the above conclusion. A safer conclusion would state that the stability of the coupling primarily depends on the nature of the discretization schemes used by the two solvers. It makes sense that consistency of the time-integration schemes must exist across the codes and that equal time steps must be used to achieve time-accurate solutions of the coupled problem when using implicit formulations. Although difficult to prove, it appears that lagging the position of the structure, in a loose coupling, causes small overshoots in the lift response, which are countered by small undershoots in the position of the structure, in order for the fluid and the structure to achieve a forced equilibrium position. Eventually, these oscillations lead to instabilities. It thus appears impossible, in this context, to use a loose coupling algorithm using implicit time-integration schemes. A minimum of one coupling to the stress solver per fluid time step is advocated for stability, with extra

coupling instances as necessary for accuracy based on the size of the time step and the regime of the flow.

The importance of the coupling level is difficult to assess for other aeroelastic configurations. Even if a loose coupling has failed for the oscillating cylinder test case, there is no reason to believe that a loose coupling can lead to unstable and erroneous solutions for other configurations, nor that it could be stable. The negative effects of a loose coupling have been observed for the cylinder, but is it because the motion of the cylinder is large? For aeroelastic analyses of a flexible structure, the amplitude of the motion is much smaller and perhaps a loose coupling is adequate. However, since the costs of coupling are so negligible, there is no justification not to provide a tight coupling to ensure simultaneous time-accurate convergence of the fluid and structure systems.

5.4 Flutter Analysis of a Swept-Back Wing

A partial flutter analysis of a swept-back wing is performed to demonstrate the capabilities of the developed aeroelastic methodology in three dimensions. In this class of truly aeroelastic problems, the structure is assumed to be flexible and its motion is determined by a finite element stress solver. In the present study, the structure is assumed to be made of a linear elastic material subject to the action of external aerodynamic loads. As for the previous class of problems, two-way feedback is active.

The test case consists of the first AGARD standard aeroelastic configuration, for which wind-tunnel experiments were conducted at the NASA Langley Research Center [81]. The wing, designated as the Wing 445.6, has a NACA65A004 cross section, a 45° sweep angle along the quarter-chord line, a root chord of 55.88 cm (22 in), a wing semispan of 76.2 cm (30 in), a taper ratio of 0.6576 (TR = tip chord/root chord),

and a panel aspect ratio of 1.6525 (AR_{panel} = (panel span)^2/(panel area)). The cross-sectional and planform views of the wing are shown in figure 5.23.

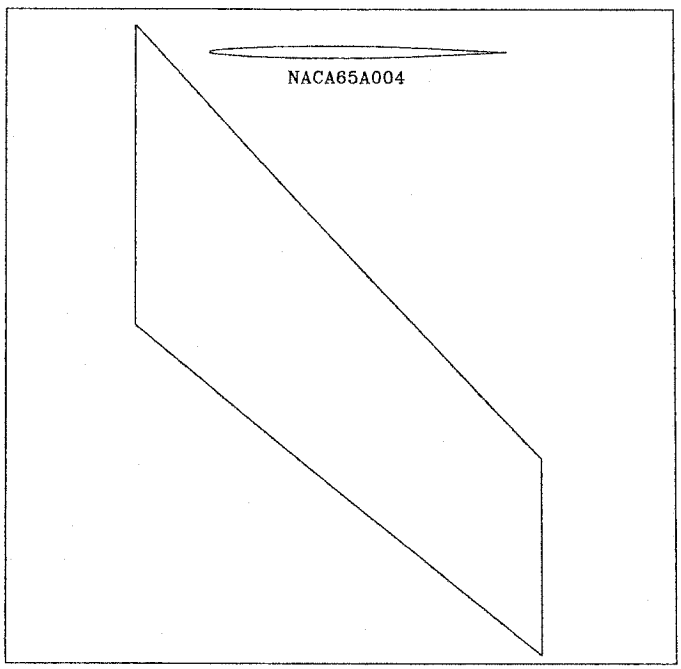


Figure 5.23: Cross-Sectional and Planform Views of the AGARD Wing 445.6.

In this study, an Euler calculation is performed to determine the flutter point at $M = 0.499$ on the flutter curve and the results are compared to the available experimental data (weakened model 3). For the purpose of Euler calculations, the rounded trailing edge of the NACA65A004 airfoil section has been tapered and the wing tip, originally square in the wind-tunnel experiment, has been rounded. Numerical results for this configuration have been published by J.T. Batina's group at NASA Langley Research Center for Euler calculations [8] and subsequently for Navier-Stokes calculations [11], with excellent agreement with the experiment.

There are a number of uncertainties about the physical configuration, the primary one being the material properties of the wing. While its shape is known, its stiffness is not well defined. In the wind-tunnel experiment, the wing was built out of laminated

mahogany with holes drilled and filled with plastic foam to reduce the overall stiffness, such that flutter could be observed within the operating range of the Langley wind-tunnel. For mahogany, Young's modulus in the direction parallel to the grain is $E = 1.04 \times 10^{10}\,\mathrm{N/m^2}$ and the density is about $\rho = 520\,\mathrm{kg/m^3}$ at 12 percent moisture content [82].

The lack of precise data about the structure of the wing makes it difficult to reproduce the same configuration. The report [83] summarizes a finite element analysis using MSC/NASTRAN to compute the natural frequencies of the first four natural modes of the wing, yielding the frequencies 9.60 Hz (first bending), 38.17 Hz (first torsion), 48.35 Hz (second bending), and 91.54 Hz (second torsion), in close agreement with the measured experimental natural frequencies by Yates [81]. In the reported finite element analysis [83], the wing is modeled using a 10×10 grid made of thin plate elements with varying thicknesses. The material properties of the wing are calibrated to best reproduce the experimental data: $\rho = 410.33\,\mathrm{kg/m^3}$, $E_x = 3.4497 \times 10^9\,\mathrm{N/m^2}$, $E_y = 1.03425 \times 10^9\,\mathrm{N/m^2}$, $G = 0.44818 \times 10^9\,\mathrm{N/m^2}$, and $\nu_{xy} = 0.31$. It is noted that the $y-$axis is aligned with the elastic axis (reported at 43.15° in [81] but entered as 41.35° in the NASTRAN model [83]). See figure 5.24. The wing is fixed at the root.

A finite element analysis is performed using the thin plate element following the orthotropic NASTRAN model listed in the NASA report [83]. From the reported mass of the wing in the wind tunnel experiment [81], $m_{wing} = 1.8623\,\mathrm{kg}$, and the volume of the discretized wing, $V_{wing} = 0.0044531\,\mathrm{m^3}$, the density of the wing is obtained as $\rho_{wing} = 418.20\,\mathrm{kg/m^3}$ (which is somehow less than the density of mahogany due to the holes filled with plastic foam, but in accordance with the value of $410.33\,\mathrm{kg/m^3}$ used in the NASTRAN model). In order to match the first natural frequency of the weakened wing model, the other structural parameters are set to: $E_x = 3.18275 \times 10^9\,\mathrm{N/m^2}$, $E_y = 0.95421 \times 10^9\,\mathrm{N/m^2}$, $G = 0.41349 \times 10^9\,\mathrm{N/m^2}$,

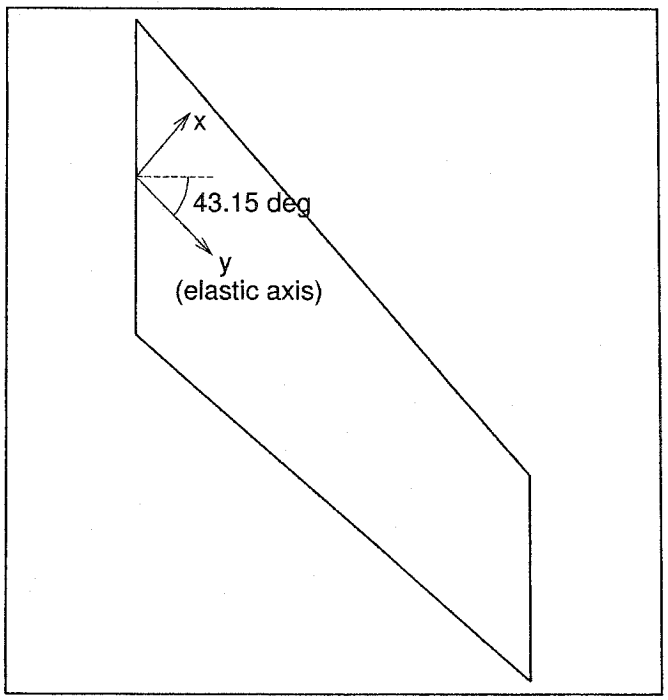


Figure 5.24: Elastic Axis for the AGARD Wing 445.6.

and $\nu_{xy} = 0.31$. The elastic axis is taken at 43.15°. The natural frequencies obtained are tabulated in table 5.1. The agreement with the experimental data is excellent.

In anticipation of constructing the virtual grid in view of the fluid-structure coupling, a volumetric stress grid is built from quadratic hexahedral 20-node bricks. The 2-D plate elements used in the stress analysis are extracted from the 3-D grid by projecting the corner nodes of the hexahedral elements onto the plane $z = 0$. The grid for the structural analysis of the wing is shown in figure 5.25, using 19×21 quadrilateral plates, with the thickness being specified at the nodes (linear variation of the thickness in the element). It can be observed that the nodes have been clustered near the leading edge of the wing. This clustering has no effect on the computed natural frequencies, but such a clustering is necessary to have an accurate definition of the

		Natural Frequency f (Hz)		
Mode #	Shape	Yates [81]	Batina [8]	Current Work
1	first bending	9.60	9.60	9.60
2	first torsion	38.10	38.17	38.96
3	second bending	50.70	48.35	48.08
4	second torsion	98.50	91.54	94.37

Table 5.1: Natural Modes of Vibration for the AGARD Wing 445.6.

curved geometry of the wing when constructing the virtual grid for the interpolation of the aerodynamic loads and of the displacements. A stress grid which is too coarse will lead to large errors in the interpolation and make it impossible to guarantee conservation of total work done. Moreover, the construction process of the virtual grid may fail if the grids are too different in regions of high curvature.

The modes of vibration are shown in figure 5.26, where the various bending and torsion modes are clearly identified by plotting the contour lines for the vertical deflection of the wing. Again, these deflections agree qualitatively well with the results published in [8].

An unstructured tetrahedral mesh is used for the fluid domain, whose extent is shown in figure 5.27, and a grid quality study is performed using 3-D mesh adaptation [52]. A flow solution is first obtained on a coarse initial mesh, then solution-based anisotropic mesh adaptation is applied to concentrate the mesh points in zones of high gradients and to stretch and align the cells with the flow directions. The mesh is also coarsened in zones of uniform flow where fewer points are needed to resolve the flow. A few solution-adaptation cycles are necessary to achieve convergence. The final optimal grid maximizes the accuracy of the flow solution given a desired mesh size.

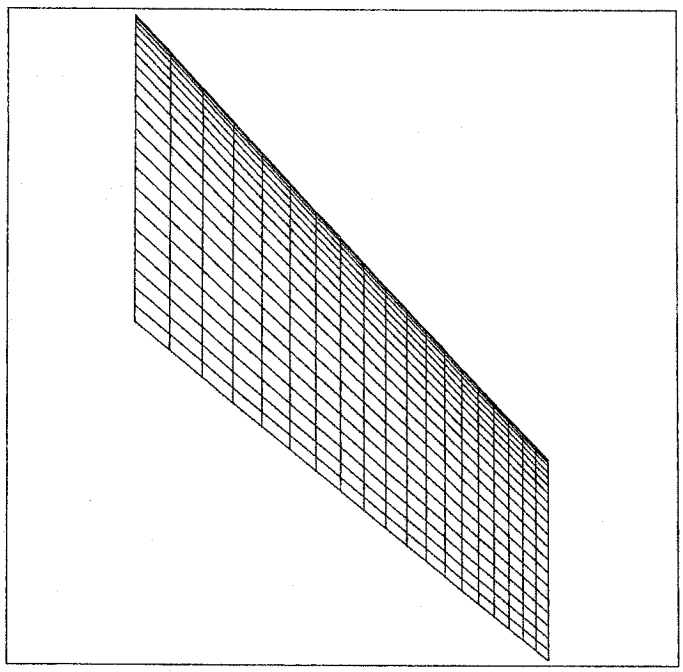


Figure 5.25: Surface Stress Grid for the AGARD Wing 445.6: 19 Elements Along the Span, 21 Elements Along the Chord.

The impact of the adaptation is assessed by plotting the C_p distribution at the span station $z = 0.4\,\mathrm{m}$ on the different grids, as shown in figure 5.28. A gradual improvement in the quality of the solution can be observed, although this improvement is mild. This is understandable since the wing is quite thin and that there are no salient flow features at $M_\infty = 0.499$ for inviscid flow (no boundary layer, no vortices, no shocks).

Before proceeding to the unsteady calculations, the effects of artificial viscosity on the quality of the steady solution are evaluated (on the final adapted grid). The C_p distribution is plotted in figure 5.29 at the station $x = 0.40$ along the wing for decreasing values of the coefficient of artificial viscosity using the $\epsilon_1 h$ formulation (see section 2.3). It is observed that the solution approaches a limiting curve as the $\epsilon_1 \to 0$, which hopefully converges to the correct solution.

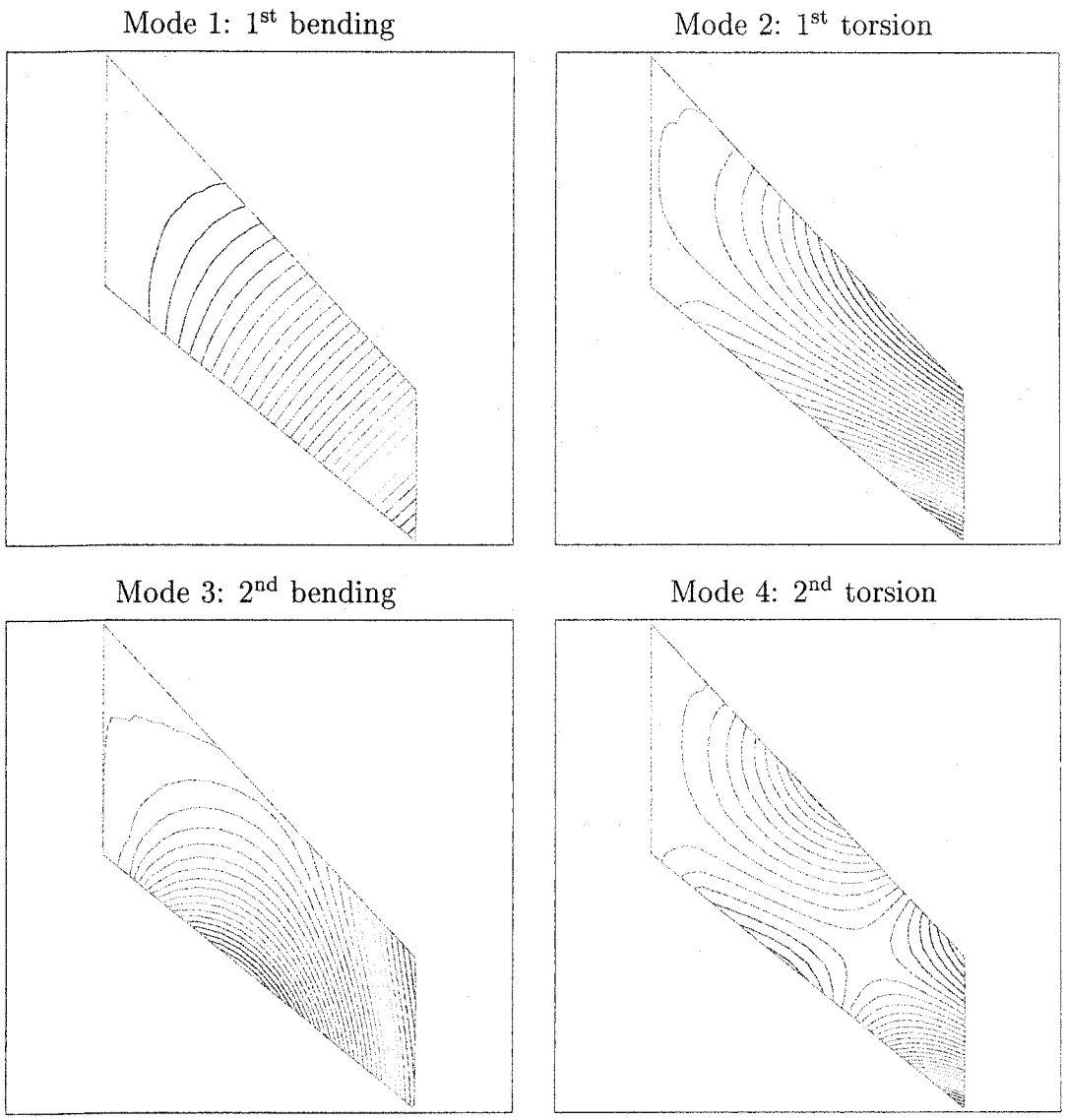


Figure 5.26: Vertical Deflection Contour Lines of AGARD Wing 445.6 for First Four Natural Modes of Vibration.

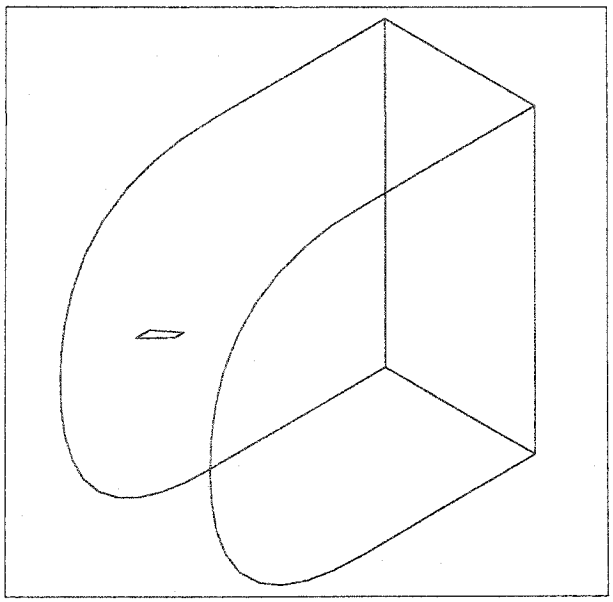


Figure 5.27: Computational Domain for Flow Around the AGARD Wing 445.6.

Based on the above results, the second adapted mesh will be used in the flutter calculations, since it yields results as accurate as those on the third adapted, but at a lesser cost (138620 vs 209556 nodes). A value of $\epsilon_1 = 0.015$ in all equations is retained for the stabilization terms in the unsteady calculations.

A few oscillations exist at the trailing edge of the wing due to the coarseness of the mesh in that region. It is hoped that these small oscillations will not adversely affect the outcome of the flutter calculations.

A justification is now provided in acknowledgement of the small kink in the C_p curve near the leading edge of the wing. (No data were found in the literature regarding the steady C_p profiles on the wing.) At first thought to be a solver or a grid related problem, further tests hint that there is indeed a genuine kink in the C_p near the leading edge. Two-dimensional incompressible results past a NACA 65A008 airfoil are computed using FENSAP (at $M_\infty = 0.01$), an explicit 2-D finite volume code, and a source panel method and they are compared to the theoretical results

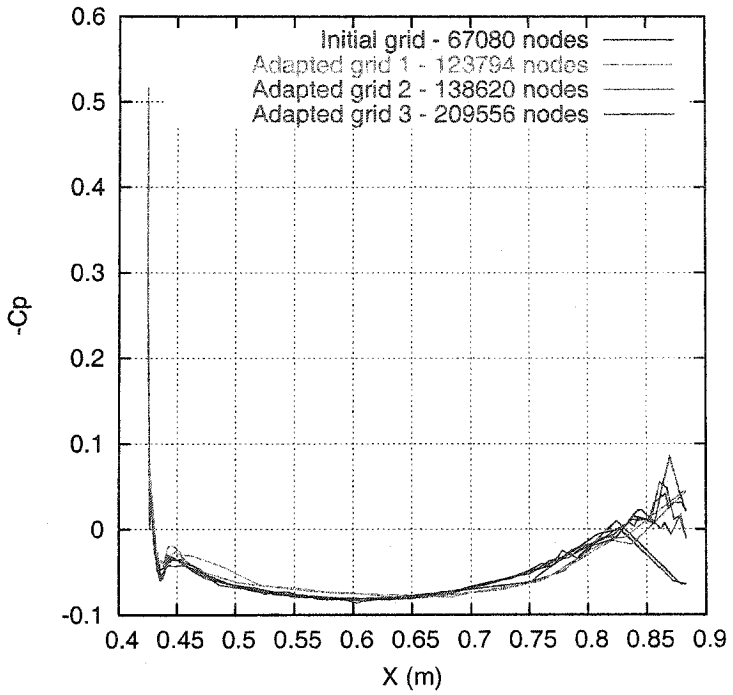


Figure 5.28: Evolution of C_p for AGARD Wing 445.6 on Adapted Grids at $z = 0.4\,\mathrm{m}$ Along the Span.

by Abbott and von Doenhoff [84] in figure 5.30. The kink is not displayed in the results by Abbott and von Doenhoff, but it is however captured by all three solvers, suggesting that this observed behavior must be due to the specific geometry of the NACA 65A airfoil and its curvature near the nose. The trend observed for the wing with the thinner NACA 65A004 airfoil section is thus likely correct.

In view of aeroelastic calculations, the virtual grid is constructed for the representation of the wing surface during the interpolation process of the displacements and of the aerodynamic loads. On the surface of the wing, the volumetric stress grid has 819 quadrangular faces while the fluid grid has 80298 triangular faces. With the use of a volumetric stress grid using quadratic hexahedral 20-node bricks, the segmented geometries described by the two grids are nearly identical and the relative error in

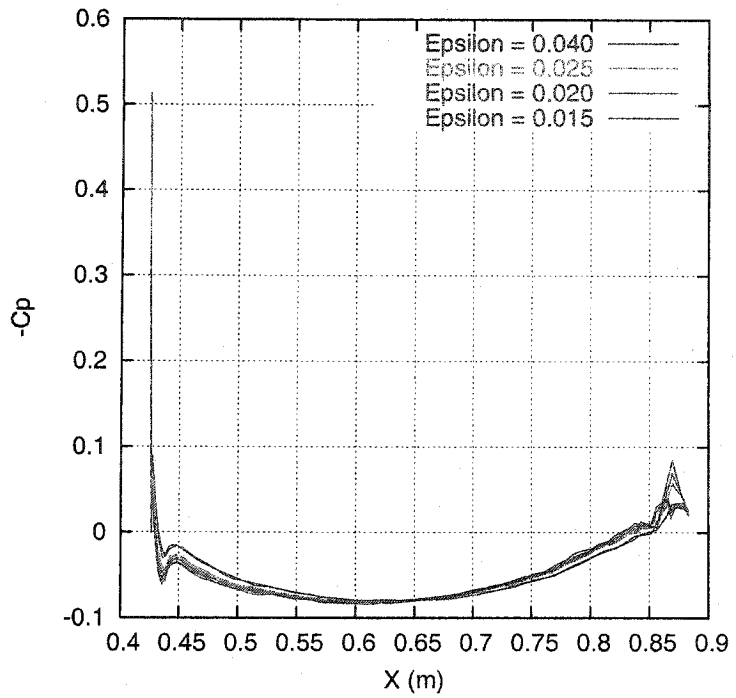


Figure 5.29: Effect of Artificial Dissipation on C_p for the AGARD Wing 445.6 at $z = 0.4\,\mathrm{m}$ Along the Span.

the areas of the two surfaces differs by less than 10^{-5}. The two grids on the surface do not match and the grid for the structure is projected onto the grid for the fluid, since the latter is finer. The virtual grid contains 113134 triangular faces and 56662 nodes. A view of the virtual grid near the wing tip at the leading edge is shown in figure 5.31.

For the purpose of the demonstration, an aeroelastic calculation is performed at $M_\infty = 0.499$ for the weakened 3 model configuration. The starting flow conditions are determined based on the flutter data from the experimental results by Yates [81]. Given $U_f = 172.46\,\mathrm{m/s}$ at $M_\infty = 0.499$, the free stream temperature at flutter is $T_\infty = 297.23\,\mathrm{K}$ (using $\gamma = 1.4$ and $R = 287.05\,\mathrm{J/(kg \cdot K)}$). From the given air density at flutter $\rho_\infty = \rho_{air} = 0.4277\,\mathrm{kg/m^3}$, the pressure at flutter is obtained

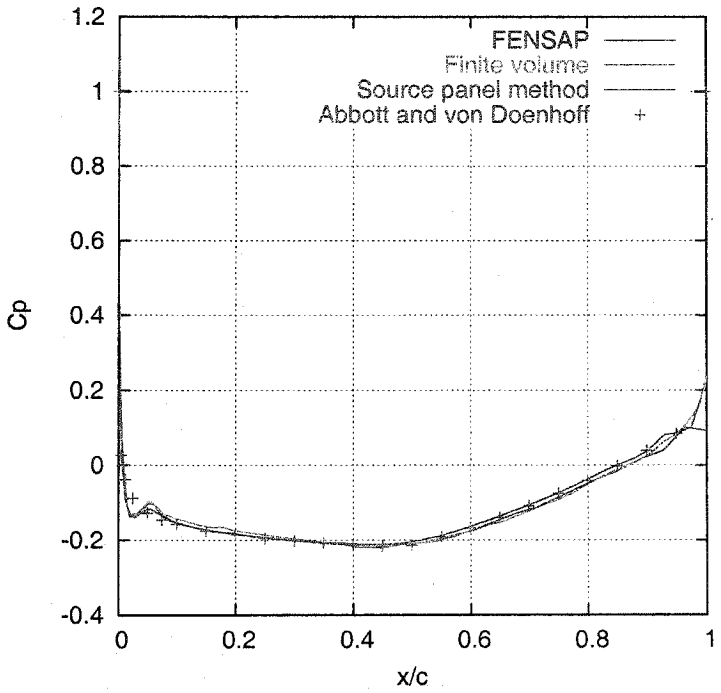


Figure 5.30: Comparison of C_p for the NACA 65A008 Airfoil for an Incompressible Inviscid Fluid.

as $p_\infty = 36485.07\,\text{Pa}$ using the ideal gas law. (The variables M_∞, T_∞, and p_∞ are specified in the FENSAP input parameter file to define the flow.)

The choice of the time step is guided by the natural frequencies of the wing. A reference time step of $\Delta t = 0.001\,\text{s}$ is selected, corresponding to 103 time steps for the first bending mode and to 21 time steps for the third mode (second bending). Smaller time steps were investigated, yielding similarly accurate results, but at an increased cost. A tight coupling is performed with two coupling instances per time step, with up to 3 Newton iterations of the flow solver per time step (or less if convergence is achieved sooner).

Various responses of the wing are obtained by varying the pressure p_∞, holding T_∞ thus altering the density of air ρ_∞ (and the wing-to-air mass ratio). These

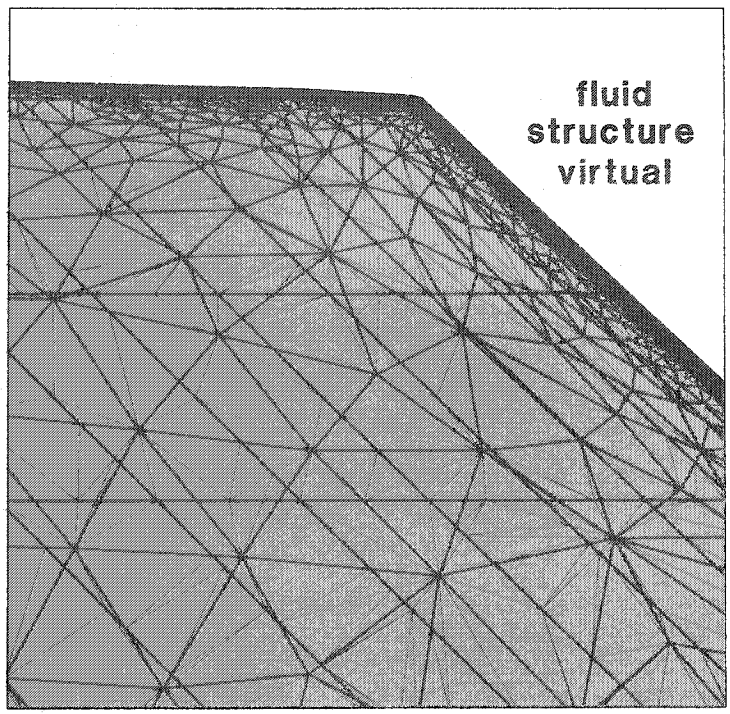


Figure 5.31: Construction of the Virtual Grid for the AGARD Wing 445.6.

responses are shown in figure 5.32 for the first two modes of oscillations: 1st bending (Mode 1) and 1st torsion (Mode 2). Near flutter, the modes coalesce and approach the flutter frequency. The pressure at which flutter is observed is at about 114% of the measured pressure at flutter. The convergence of the flow solver and of the coupling driver is shown is figure 5.33 for the first 40 time steps of the aeroelastic calculation at 110% of the flutter pressure. Other runs exhibit the same convergence pattern.

In the analysis of the results, the flutter speed index is defined as $\frac{U_f}{b\omega_\alpha\sqrt{\mu}}$, where:

U_f = free stream flow speed at flutter = 172.46 m/s

b = root semi-chord = 0.2794 m

ω_α = natural (angular) frequency of the first torsion mode = 244.77 Hz

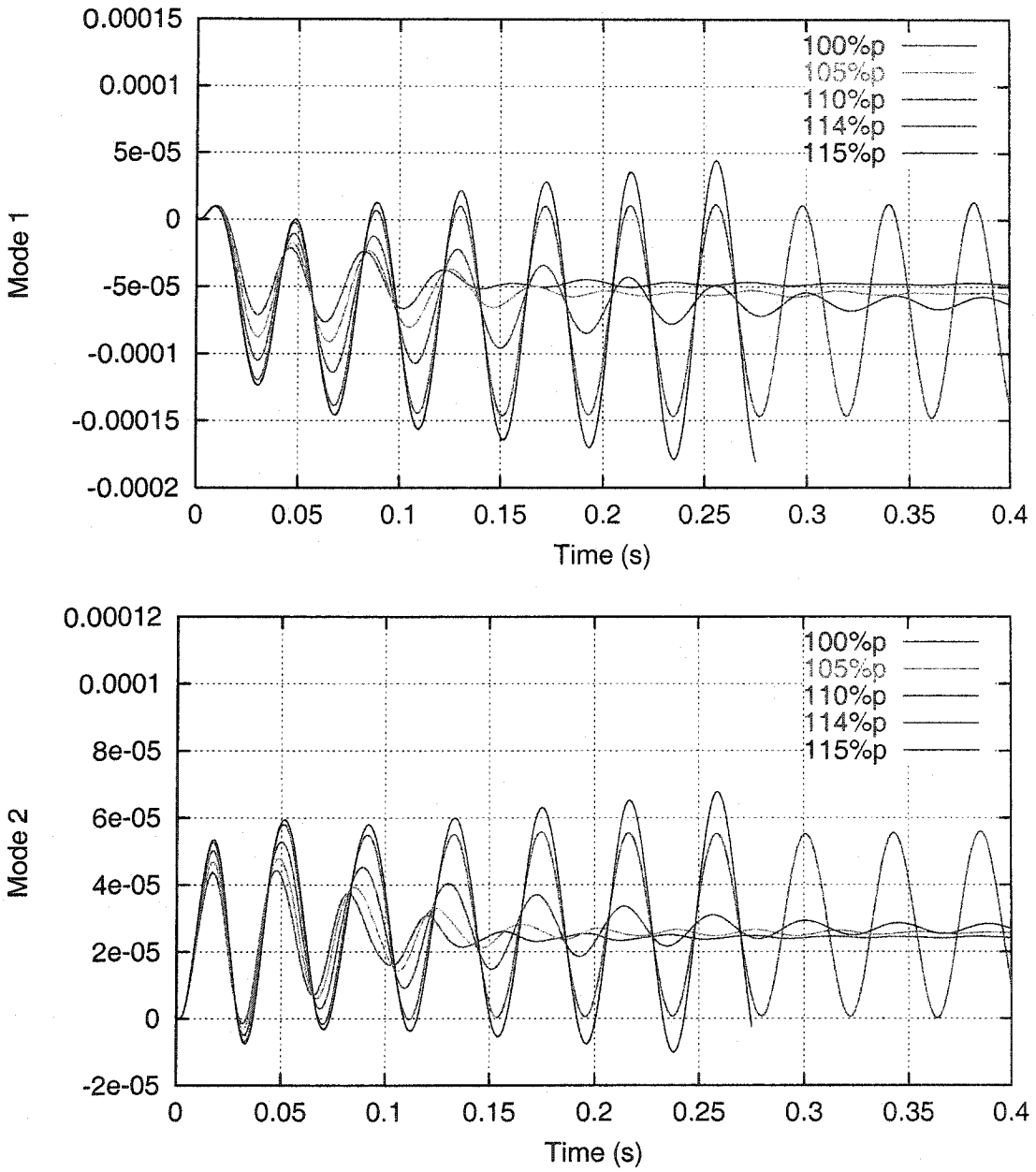


Figure 5.32: Aeroelastic Response of the AGARD Wing 445.6 at Different Pressures: Mode 1 (1^{st} bending); Mode 2 (1^{st} torsion).

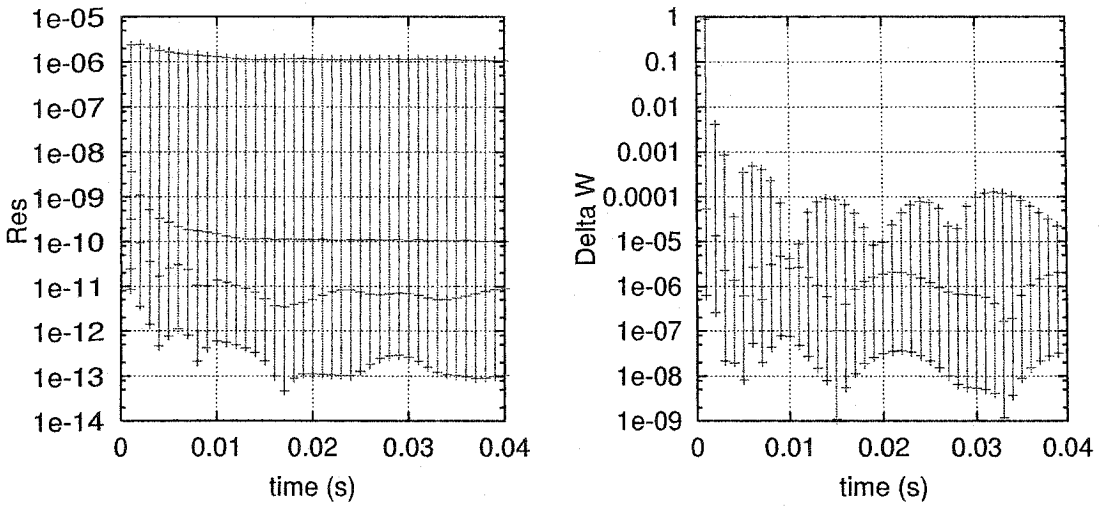


Figure 5.33: Convergence History for the AGARD Wing 445.6: (a) Residual of the Flow Solver; (b) ΔW for the Coupling Driver.

$$\mu = \text{mass ratio} = \rho_{wing} V_{wing} / \rho_{air} V_{cone} = 29.262$$

$$\rho_{wing} = \text{density of wing} = 418.20\,\text{kg/m}^3$$

$$\rho_{air} = \text{density of air} = 0.4875, \text{kg/m}^3 \text{ (at } p_\infty = 41592.98\,\text{Pa)}$$

$$V_{wing} = \text{volume of wing} = 0.0044531\,\text{m}^3$$

$$V_{cone} = \text{volume of a truncated right cone enclosing wing}$$

$$= \frac{\pi h}{12}\left(c_{root}^2 + c_{root}c_{tip} + c_{tip}^2\right) = 0.13054\,\text{m}^3$$

$$c_{root} = \text{wing chord at the root} = 0.5588\,\text{m}$$

$$c_{tip} = \text{wing chord at the tip} = 0.3688\,\text{m}$$

$$h = \text{wing semispan} = 0.762\,\text{m}.$$

The flutter speed index is thus calculated as 0.466. This value over-predicts the experimental value for this configuration, though for inviscid subsonic flow the value should have been under-predicted (in the subsonic range) [8]. See table 5.2. Turbulent

subsonic solutions will yield slightly higher values for the flutter index than the ones predicted for inviscid solutions [11]. Moreover, structural damping has been neglected, where in fact the presence of mechanical damping in the system should have led to an even higher predicted value of the flutter index.

	Flutter Speed Index $\frac{U_f}{b\omega_\alpha\sqrt{\mu}}$		
Mach #	Yates [81]	Batina [8]	This work
0.499	0.4459	0.439	0.466
0.678	0.4174	0.417	—
0.901	0.3700	0.352	—
0.954	0.3059	—	—
0.957	0.3095	—	—
0.960	0.3076	0.275	—
0.990	—	0.310	—
1.072	0.3201	0.466	—
1.141	0.4031	0.660	—

Table 5.2: Flutter Speed Index for AGARD Wing 445.6: Experimental Data; Computed, Inviscid Flow; Current Work.

The main reason for the difference between the computed and the experimental values is believed to be the somehow "softer" structural parameters used in the stress analysis of the wing ($E_x = 3.18275 \times 10^9\,\mathrm{N/m^2}$, vs $E_x = 3.4497 \times 10^9\,\mathrm{N/m^2}$). The way the wing is clamped possibly plays a role in the outcome of the simulation. In this work, the wing is fully clamped at the root, whereas it is only partially clamped in the middle of the chord in the wind-tunnel experiment. Other uncertainties exist with regard to the accuracy of the flow solution. Although a grid analysis study was

performed using mesh adaptation and the effect of artificial dissipation on C_p was examined for the steady-state solution, see figure 5.29, the unsteady solutions could still be too dissipative. The impact of artificial dissipation on the amplitude of the aeroelastic response is clearly demonstrated in figure 5.34: increasing the coefficient of artificial viscosity severely dampens the amplitude of the coupled response and displays the same behavior as added mechanical damping. If the variations in C_p appeared small as the artificial viscosity coefficient was varied, as seen in figure 5.29, the impact on the aeroelastic response is more significant. As a consequence, increasing the coefficient of artificial viscosity delays the onset of flutter and increases the value of the dynamic pressure (or velocity) at which flutter can be observed, giving a much over-estimated value of the flutter index.

The validity of the time step is verified by integrating the system at time steps of $\Delta t/2 = 0.0005\,\mathrm{s}$ and $2\Delta t = 0.002\,\mathrm{s}$ for the case at 110% of the experimental flutter pressure. It can be observed, see figure 5.35, that the reference time step is indeed sufficiently small to yield accurate results. Reducing the time step does not have a significant impact on the quality of the solution for this test case, but, at the same time, the time step should not be increased.

The level of tightness of the fluid-structure coupling is investigated for the case at 110% of the experimental flutter pressure (which here yields a damped response). The solution is let to evolve for a given time using different coupling strategies with one, two, and three coupling instances per time step. A maximum of 6 Newton iterations may be performed per time step. For example, at three coupling instances per time step, each coupling is effected after 2 Newton iterations of the flow solver. Although all three strategies are stable, as advocated, the quality of the solution deteriorates when only one coupling instance per time step is performed, as illustrated in figure 5.36. In fact, the predicted aeroelastic response is undamped, whereas a tighter coupling

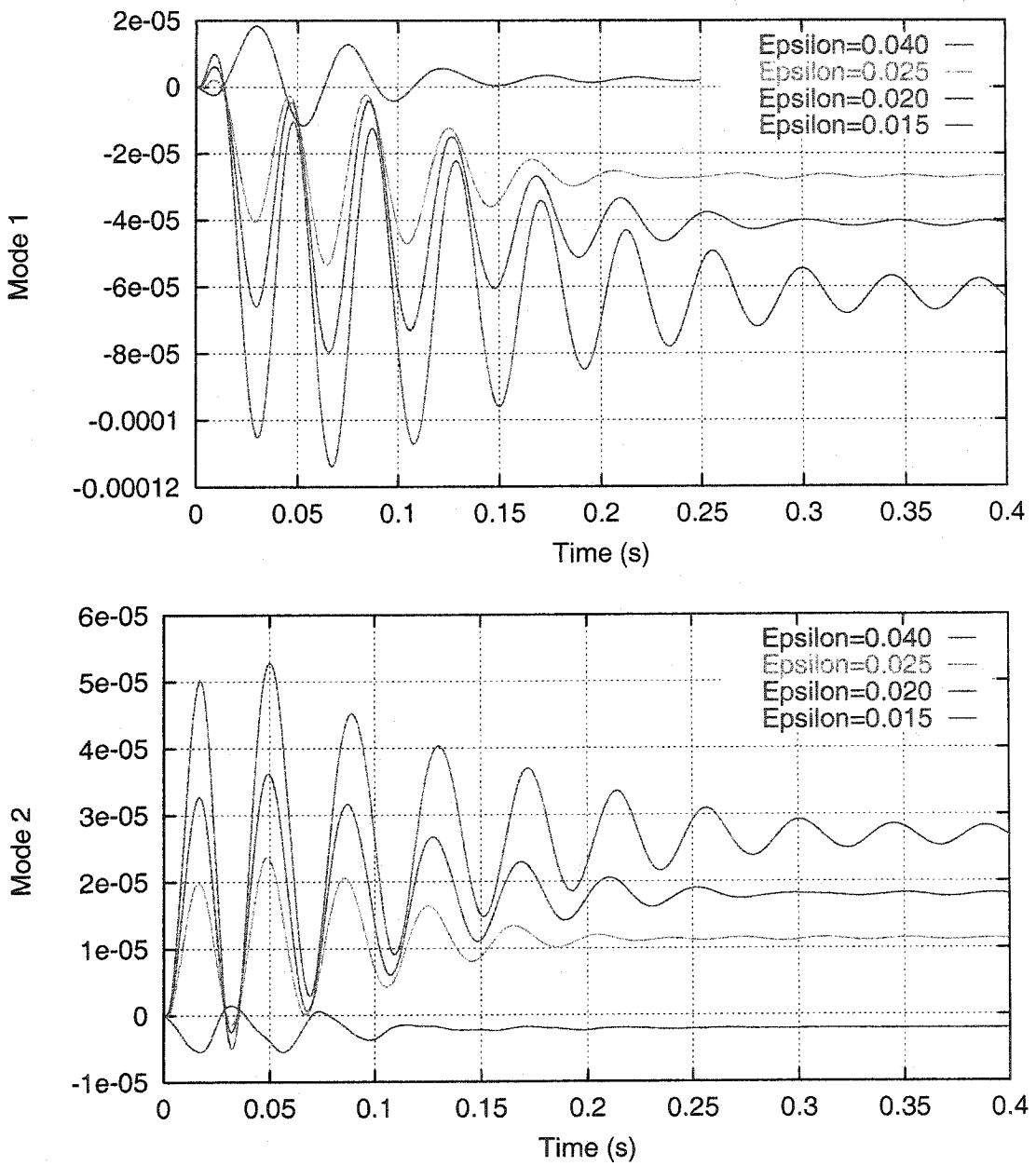


Figure 5.34: Effect of Artificial Dissipation on Aeroelastic Response of AGARD Wing 445.6: Mode 1 (1^{st} bending); Mode 2 (1^{st} torsion).

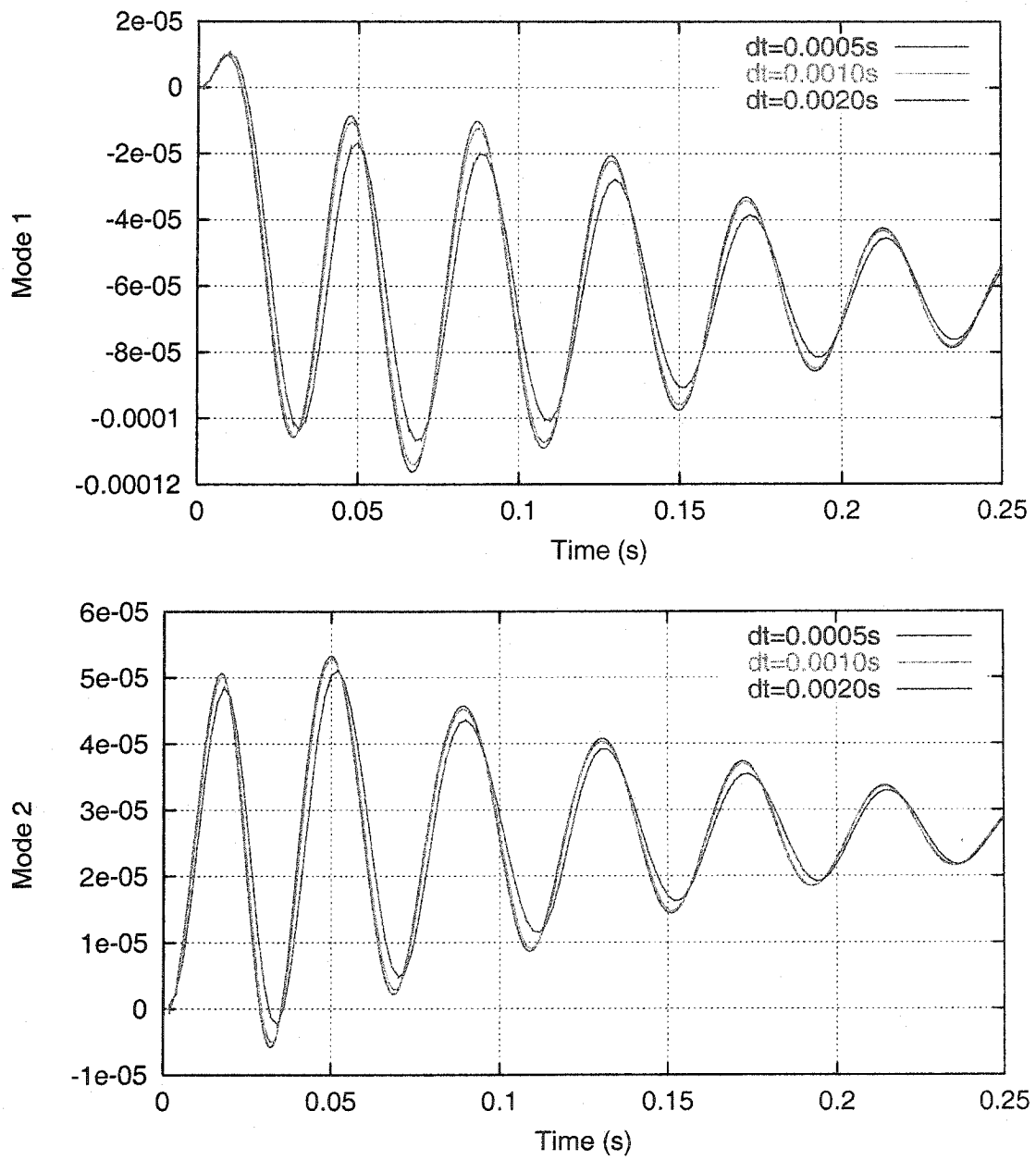


Figure 5.35: Effect of Time Step on Aeroelastic Response of AGARD Wing 445.6: Mode 1 (1^{st} bending); Mode 2 (1^{st} torsion).

clearly demonstrates that it must be damped. It is thus important to closely monitor the convergence history of ΔW for the coupled problem (recall figure 5.33). The great sensitivity of the response of the wing due to incomplete convergence is amazing. There is no marked difference between two and three coupling instances per time step.

Based on this observation, a loose coupling featuring one coupling instance every few time steps is not advised. However, as seen in figure 5.37, the loss in accuracy due to one coupling instance per time step can be alleviated if a much smaller time step is taken. This suggests that accurate solutions can possibly be obtained with one coupling instance per time step, if the time step is drastically reduced, as it is the case for explicit methods. This approach is, of course, not cost-efficient for an implicit flow solver since it negates the main advantage of being able to use a large time step, in comparison to explicit methods. Overall, it is significantly more cost-effective (and accurate) to perform two coupling instances per time step than to halve the time step with one coupling instance per time step.

In summary, the mechanisms of the aeroelastic code have been verified for a three-dimensional configuration. The flutter points at the other Mach numbers were not computed due to the lengthy calculations required. However, from the numerical point of view, these are the most challenging points to compute, since the presence of shock waves introduces stronger nonlinearities in the physical system.

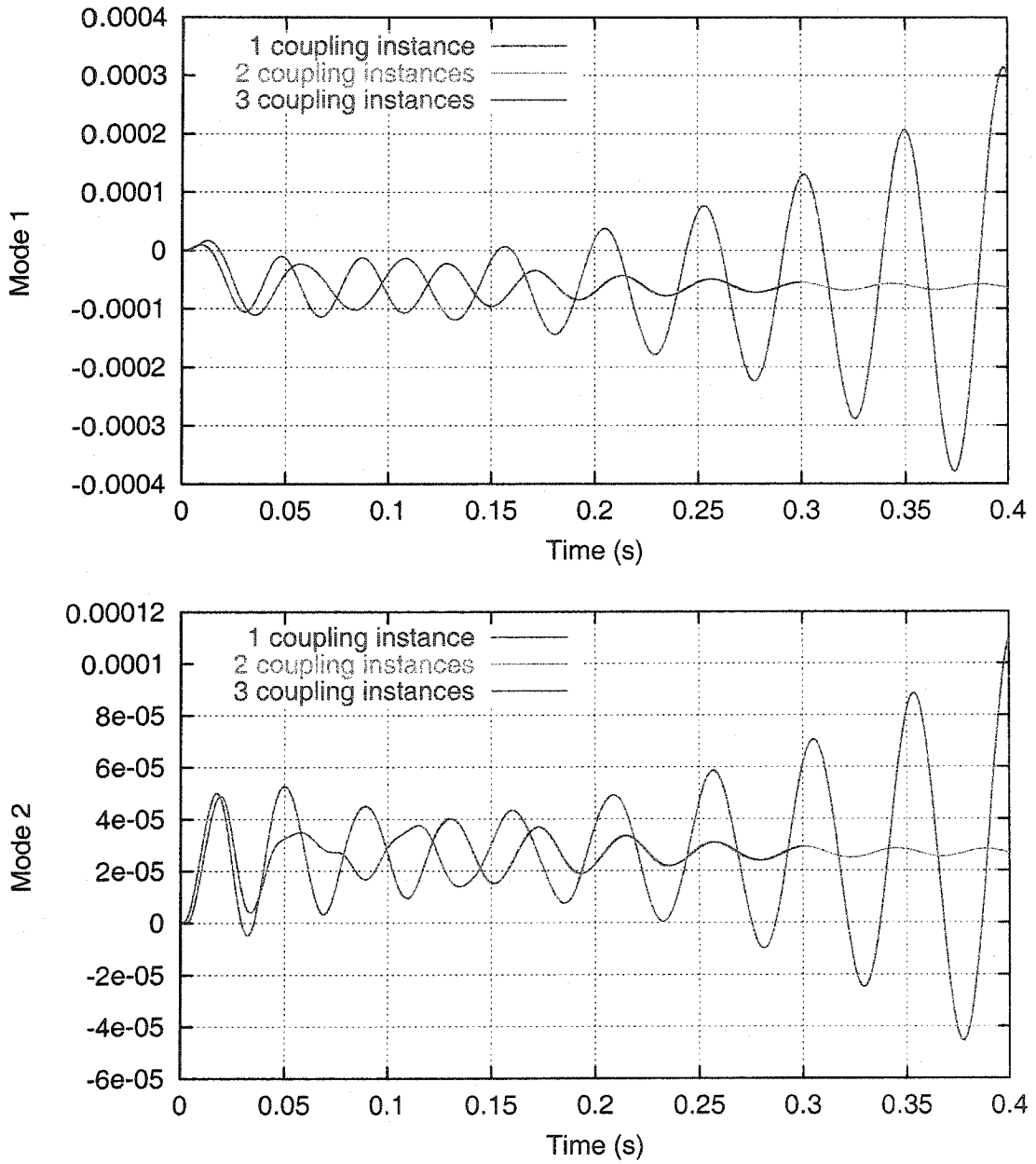


Figure 5.36: Aeroelastic Response of AGARD Wing 445.6 for Different Coupling Strategies: Mode 1 (1^{st} bending); Mode 2 (1^{st} torsion).

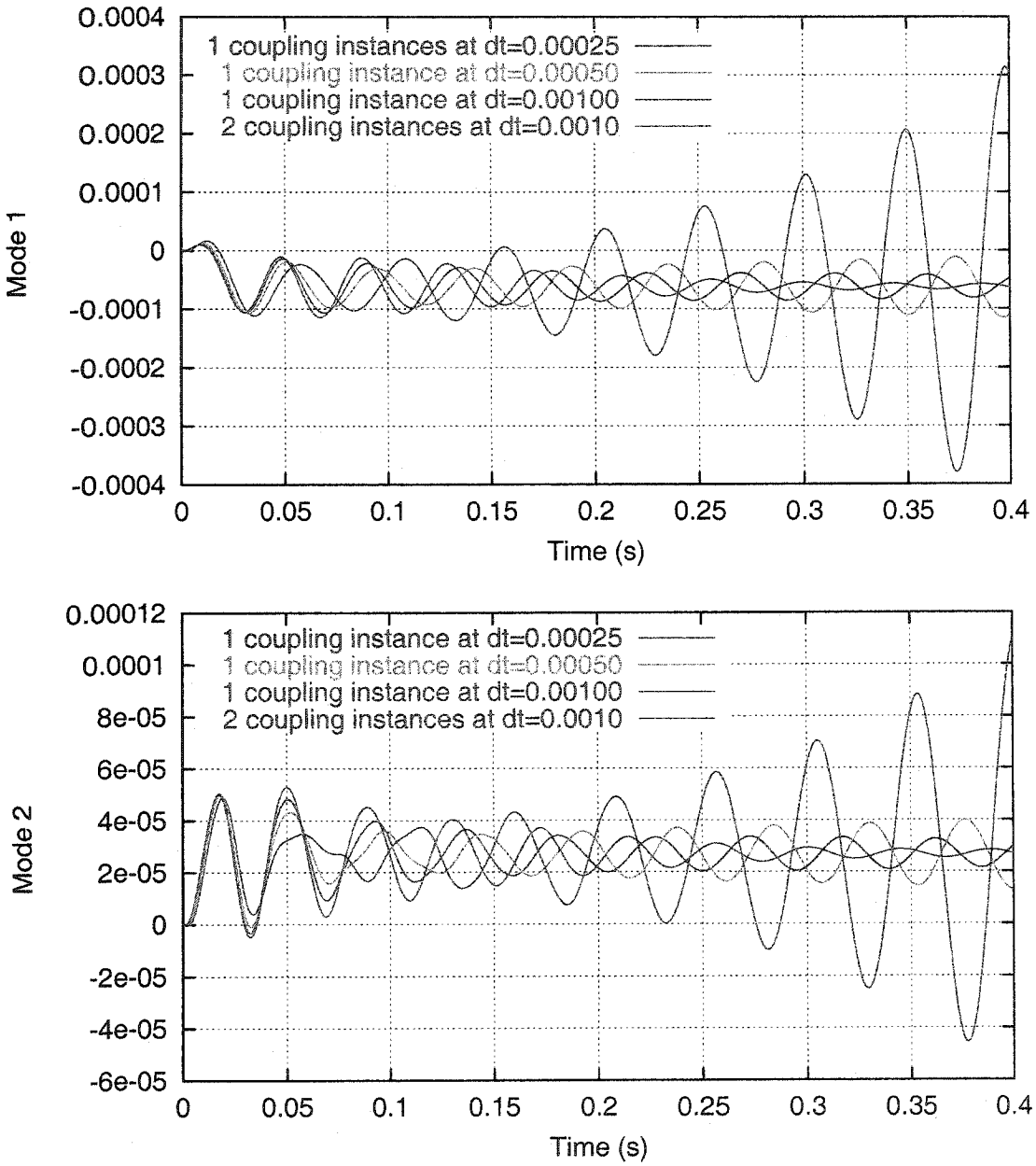


Figure 5.37: Aeroelastic Response of AGARD Wing 445.6 for a Loose Coupling with Varying Time Step: Mode 1 (1^{st} bending); Mode 2 (1^{st} torsion).

Chapter 6

Conclusions

The accomplishments and the contributions achieved in this thesis are summarized. Recommendations for future research paths are presented, based on the various experiences learned from this work.

6.1 Developments and Contributions

A 3-D, parallel, CFD Euler/Navier-Stokes finite element implicit code has been extended using the Arbitrary Lagrangian-Eulerian (ALE) formulation to compute flow fields over moving and deforming structures, with focus on aeroelastic calculations in the time domain.

A two-zone node movement scheme based on Laplacian smoothing, embedded in the CFD code, offers smooth node movement and enhanced robustness at large amplitudes of motion for geometries of arbitrary shape, without the need for global mesh regeneration. The smoothing, applied to the displacements, is such that its action does not destroy concentrations of grid points nor the stretching of cells resulting from

anisotropic mesh adaptation. The current implicit finite element discretization of the flow solver allows for the straightforward application of high-order time-integration schemes, without restrictions on the way the mesh is moved (no GCL-like conditions). Indeed, numerical tests confirm, as it should be, that the solution is independent of the way the mesh is moved in the interior of the domain, demonstrating the correctness of the numerical treatment of the ALE derivative in the governing equations. More specifically, the node movement is truly arbitrary and the ALE derivative in the flow equations adjusts the solution to whatever prescribed motion of the mesh.

Interpolation of the structural displacements and strictly conservative transfer of the integrated aerodynamic loads across the CFD and CSD grids are achieved using the concept of the virtual grid. The virtual grid is constructed automatically from the CFD and CSD surface grids and is applicable to arbitrary complex three-dimensional geometries, including non-matching grids at the fluid-structure interface. Extensive searches are eliminated during the interpolation of the displacements and the integration of the loads can account for both the pressures and the viscous shear stresses for a complete representation of the aerodynamic forces. At each time step, the work done by the fluid on the structure is conserved, both at the face-by-face level and globally. The usage of this interpolation procedure is particularly advantageous in conjunction with mesh adaptation for which great disparities in the distribution of grid points can result at the fluid-structure interface.

The fluid-structure coupling algorithm has been designed and implemented with the philosophy that the flow and the stress solvers exist as separate entities. This modular approach favors the substitution of the solvers, viewed as independent black boxes, by improved models when they become available. The use of implicit solvers allows to seamlessly update the position of the structure and the ensuing response of the fluid several instances per time step such that, at convergence, the structure and

the fluid are in perfect equilibrium. Of utmost importance in simplifying the development effort, equal time steps can be used in the implementation. In this context, no staggered time-integration scheme is necessary for numerical stability. Implementation details on the use of stream sockets are given to eliminate the overhead restart costs incurred at every coupling instance between the two solvers while providing the tightest level of fluid-structure coupling without any loss in efficiency.

The mechanisms of the aeroelastic code have been verified and the developed technology has been applied to 2-D and 3-D aeroelastic configurations for validation purposes.

The level of tightness of the coupling was investigated from a tight coupling to a loose coupling at every few time steps, with a degradation in the quality and the stability of the solution when using a loose coupling. It was concluded that a minimum of one coupling instance per time step was necessary for the stability of the implicit time-integration schemes used in this work, with extra coupling instances required for accuracy depending on the size of the time step and the nature of the aeroelastic configuration.

While the current work provides the framework to perform aeroelastic analyses in the time domain, the lesson to learn from the current results is that the aeroelastic response is sensitive to the modeling choices and to the accuracy of the predicted solutions. Based on the analysis of the results obtained for the AGARD 445.6 wing, it is noted that the artificial viscosity in the flow solver plays the role of an added natural viscosity in the discretized equations (see section 2.3), giving weight to the argument that inviscid solutions predict a lower flutter index than viscous solutions do. Inviscid solutions should thus provide a safe envelope for the flutter boundary, which is not the case for this wing configuration using the current numerical models. It is believed that with accurate modeling of the aerodynamic forces, and a revision

of the structural model, more representative results would be obtained. Nonetheless, regardless of the solvers used, the aerodynamicist must be critically aware of the pitfalls and limitations of the individual solvers in order to assert the quality of the overall solutions. In particular, an unnecessary large value of the artificial viscosity coefficient, needed for the stabilization of the FEM flow solver FENSAP, was shown to damp the structure and to lead to over-predicted values of the flutter boundary.

In retrospect, the aim of this research was to develop a tool to guide the engineer in the analysis process of a design, to use in an industrial context with applications to turbomachinery. The current technology (ALE formulation of the equations and node movement scheme for smoothing the displacements) has been transferred to Pratt & Whitney Canada's in-house flow solver tailored for rotary components. Aeroelastic calculations of fan blades have been performed at P&WC using the developed methodologies [86, 87], adding an industrial perspective to the work of this thesis. These results, involving proprietary geometries, are not presented in the thesis.

The foundations have thus been established to solve three-dimensional complex aeroelastic configurations in the time domain using a Navier-Stokes turbulent solver (upon availability). It may not be surprising, given the costs of these calculations, that cheaper and simpler models (e.g. potential flow) are still preferred today, sometimes since no better technology is available, sometimes by limitations in the computing resources available. Even if the more sophisticated models advocated in this thesis appear perhaps to lie beyond the turnaround analysis time constraint acceptable to the industry, they can still be used for the calibration of the simplistic models, or at least to better understand the limitations of such simpler models.

6.2 Future Work

Several aspects of the implementation of the fluid-structure capabilities developed in this work could be improved and extended, based on the experiences gained during the validation of the benchmarks. Among the observed difficulties, the main issues pertain to the robustness, accuracy, and cost of the flow solver, as well as to the robustness of the node movement scheme.

It would be interesting to measure, as mentioned in chapter 2, the impact of the newly developed grid stiffness damping function applied to the entire domain, as opposed to using the two-zone approach, whose implementation is quite tedious. The return to a one-zone mesh movement approach would facilitate the developments for the support of hybrid meshes (tetrahedral elements with layers of prisms on the wall), preferable for viscous calculations with the implementation of the turbulence model using specialized wall-functions. The extension of the flow solver to model the effects of turbulence, which have been overlooked in this work, constitutes the most substantial development to pursue. However, no difficulties are foreseen from the fluid-structure coupling point of view with regard to the conservative transfer of the aerodynamic loads for viscous flows. In fact, no changes to the coupling procedure and to the stress solver should be required.

The flow solver must also be accelerated in order to reduce the overall costs of aeroelastic calculations. The development of a time-linearized flow solver, based on the Euler or Navier-Stokes equations, would greatly reduce the prohibitive computational costs of direct aeroelastic calculations in the time domain, while likely retain the essential physics of the unsteady phenomena present in the flow.

In the validation test cases presented in chapter 5, it was observed that the coupling with the stress solver did not adversely affect the convergence rate of the Newton

process in the flow solver. One can, however, question if the convergence rate of the coupled problem could be accelerated by considering the gradients in the solutions, using a nonlinear GMRES loop, for example [85]. In view of the inherent nonlinear nature of the flow equations, it appears that regardless of any acceleration scheme for the coupled fluid-structure problem, the nonlinear fluid problem still needs to be iterated. The update of the displacements at every Newton iteration does not seem to stiffen the nonlinear fluid problem and additional iterations are not required to converge the fluid problem. It is thus unlikely that any acceleration scheme would have much of an effect on the coupled problem. However, the idea might be worth exploring when the solid-to-air mass ratio is small (vascular flows, for example).

As stated previously, the long term goal is to compute the aeroelastic response of blades in turbomachines. Applications to turbomachinery bring new challenges from both the modeling point of view and the industrial perspective. The presence of shock waves and of viscous flow phenomena, in particular, passage viscous blockage, vortical flows, and separated and stalled flows, characterize the inherently three-dimensional transonic and turbulent flow regime present in the first stage of an axial compressor. It is also common to make use of cyclic symmetry when studying isolated blades in order to reduce the size of the problem. However, rotor-stator interactions and multistage effects are at the source of the pressure disturbances which initiate flow-induced vibrations of the blades and the assumption of cyclic symmetry breaks down when stators and rotors have different numbers of blades per row. The requisite to model all blades over several stages further amplifies the already intensive computing needs, but efficient numerical methods have been recently developed to reduce these computational requirements [29]. Finally, beyond the complexity of the aeroelastic phenomena occurring in the axial compressor or the fan, the impact of the thermal effects on the blades can be analyzed in the turbine, where hot gases exhausting from

the combustion chamber require the cooling of the blades. Many of these problems have yet to be solved given their high computational costs and the modeling difficulties associated with developing the state-of-the-art numerical codes.

Throughout the course of this work, the use of the moving-grid technology has been envisaged for ice accretion problems as a tool to locally modify the grid without remeshing [88]. As the ice front grows in time, the ice shape being viewed as a deformable surface, the mesh points on the surface of the structure are pushed to conform to the growing ice shape, as shown in figure 6.1 for a representative ice shape on an airfoil. At least, over several time steps of ice accretion, the mesh movement scheme alone suffices to move the grid to follow the growth of the ice, thus avoiding the inconveniences of remeshing the domain. For more significant ice accretion, global remeshing of the domain and mesh adaptation may be unavoidable. As for studying the aeroelastic effects of ice-contaminated wings, such configurations are not likely to attract much interest in the field since ice accretion conditions occur at low altitude, at speeds well below the flutter limits. Any loss of control of an aircraft due to the presence of ice is not due to flutter.

Based on the results obtained for the pitching airfoil and the AGARD 445.6 wing, it was previously stated that the outcome of such aeroelastic calculations strongly depends on the modeling decisions made and by the ability of the numerical models to predict accurate solutions. Although not a remedy by itself, it is believed that *unsteady mesh adaptation* can be a valuable tool in preventing mesh-related uncertainties and in certifying the accuracy of aeroelastic calculations. Its purpose is two-fold: grid enhancement, to prevent the break-down of the flow solver due to excessive mesh distortion following large motions, and solution enhancement at a minimal number of grid points for maximum efficiency and accuracy.

First, mesh adaptation can prevent any grid degeneracies that may arise from the

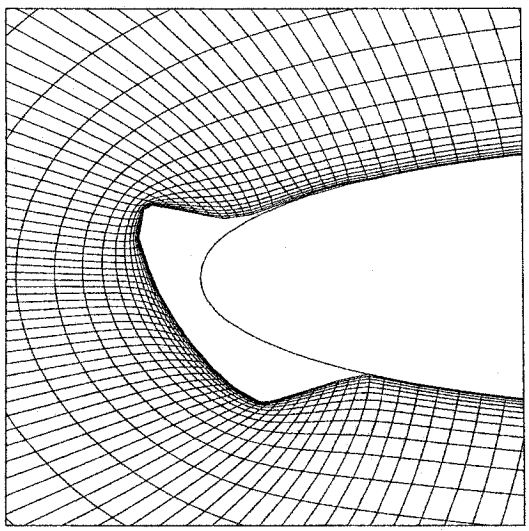


Figure 6.1: Typical Ice Growth on a NACA0012 Airfoil.

movement of the structure, when such movement is large relative to the local grid size. Although the mesh movement methodology developed has proven adequate for small displacements, it does lack the robustness to treat motions of large amplitudes. In this case, mesh adaptation using node movement and edge refinement, coarsening, and swapping [52] can prevent grid degeneracies before they occur by avoiding high aspect ratio elements that may become inverted due to a subsequent large motion of the structure.

Second, mesh adaptation enhances the quality of the solution by clustering nodes in zones of high gradients such as shock waves and boundary layers. Recalling that the primary difficulty in aeroelastic calculations is to determine the transonic dip of the flutter boundary curve, it is thus essential to capture the shock waves accurately since the position and the strength of the shocks are critical in computing the stability limits of the structure. The benefits of static mesh adaptation were observed for the AGARD aeroelastic CT 6 (see section 5.2), but could not be fully exploited for the oscillating airfoil with the moving shock. The use of unsteady mesh adaptation is

inevitable when trying to follow unsteady shock waves. The development of unsteady mesh adaptation will thus improve the quality of the calculations for a given number of grid points and make direct aeroelastic calculations of complex configurations in the time domain more robust, accurate, and reliable.

Appendix A

Flow Solver Coupling Files

This first appendix, adapted from the FENSAP guide [49] written by the author, describes the internal format of the files related to the flow solver that are necessary for the coupling with the stress solver. While the description of the files is utterly technical, it illustrates well the attention to details required to perform the external coupling.

As inputs, the flow solver requires the displacements on the surface of the deforming structure. The nodes interior to the structure are filtered out by the coupling driver to reduce the amount of data transferred to the flow solver. The internal format of the file containing the displacements is described in section A.1.

As outputs, the flow solver saves the aerodynamic loads over each face of the grid on the surface of the structure. For convenience, the loads are calculated inside the flow solver, since the calculation of the shear stresses τ_{ij} requires the gradients of the velocity field to be evaluated from the interior of the domain. The evaluation of the shear stresses in a post-processing step, outside from the flow solver, would require that the entire flow solution be manipulated, a step which is much too costly

in input/output operations. The format of the output file containing the aerodynamic loads is described in section A.2.

Also as an output, the flow solver saves the new grid coordinates such that the calculations may be restarted at a later time. The format of this file is given in section A.3.

These three coupling files may be written and read in binary or ASCII mode by the flow solver FENSAP. Data transfers by files are much faster in binary format (unformatted).

There are numerous other input and output files associated with the flow solver, for example the grid file, the solution file, and the input parameter file. A thorough description of these files is not essential to the understanding of the fluid-structure coupling mechanisms, hence the reader is referred to the FENSAP guide [49] for the details.

A.1 The Time Control File `timebc.dat`

In this file, flow variables can be imposed at Dirichlet nodes and nodal displacements due to the movement of the grid can be imposed at nodes on walls. The flow variables can be specified as initial conditions (at $t = 0$) and/or as transient conditions at given time levels `t1, t2,...`. Nodal displacements may only be specified in time. For static initial displacements, see the section A.3 on the Nodal Displacement File `disp.dat`. The format of the `timebc.dat` file is given below.

```
ntsteps   kcoupl   ktmode
t1  numbc1  nalebc1
  node1   var1    val1
  node2   var2    val2
```

```
    .       .       .
    .       .       .
    .       .       .
t2  numbc2  nalebc2
  node1   var1    val1
  node2   var2    val2
    .       .       .
    .       .       .
    .       .       .
```

The variable `ntsteps` gives the number of time levels `t1, t2,...` listed in the file. If the file `timebc.dat` does not exist, `ntsteps` is set to zero in the code, assuming that no transient Dirichlet boundary conditions will be read. If `t1=0.0`, the conditions are understood to be imposed as initial conditions at the start of the program. This is useful, say, to impose a parabolic velocity profile at the inlet of a duct. This file is read after the solution `restart` file and hence overrides it. The variable `kcoupl` indicates the first iteration of a fluid-structure coupling step (`kcoupl=0`) or a subsequent iteration of a fluid-structure coupling step (`kcoupl`$\geq$`1`). This allows for a tight fluid-structure coupling at the level of the fluid Newton iterations. Thus, this variable indicates the start or the continuation of a fluid Newton iteration. A different initialization is required for both actions. In the case `kcoupl`$\geq$`1`, `val` is the incremental change in the nodal displacement, being the correction to the displacement at the previous sub-iteration. For `kcoupl=0`, `val` is the change in the nodal displacement from one time step to the next.

The variable `ktmode` selects the integration mode. When `ktmode=0`, the integration in time is from `T0` to `Tend`, as inputed in the parameter file `fensap.in`. When `ktmode=1`, the integration in time is advanced to the next time level specified in the `timebc.dat` file and then the program is halted. This allows to obtain the response of the structure during a fluid-structure coupling and subsequently to restart FENSAP.

The time-dependent boundary conditions on the flow variables are imposed at each time level (`numbc`, if any), followed by the nodal displacements at that time level (`nalebc`, if any). It is important that the conditions on the flow variables come before the nodal displacements as these conditions are treated differently in the code. The table below gives the admissible values of `var` for the various fields.

`var`	Flow variable [units]	Nodal displacement [units]
1	pressure [Pa]	x or r displacement [m]
2	u-velocity [1]	y or θ displacement [m (rad)]
3	v-velocity [1]	z displacement [m]
4	w-velocity [1]	
5	static temperature [K]	
6	heat flux [W/m^2]	

All variables are imposed at a node, except for the heat flux which is specified over a face. So, the global face number must be entered instead of the node number (as assigned in the `grid` file).

Although the time-step is governed by `dtmax` entered in the input parameter file, the time step will be locally adjusted such that the time levels `t1, t2,...` are hit exactly. In the case where several time steps are required between each time level specified in the file `timebc.dat`, a loose coupling, linear interpolation is used to find the values of the time-dependent Dirichlet nodes at the intermediate steps. For the displacements, the order of the interpolant is the same as that of the time-integration scheme.

A.2 The Surface Loads File `surface.dat`

This file is used for fluid-structure coupling (aeroelasticity and conjugate heat transfer) and it lists the faces on the walls of solid bodies, followed by the aerodynamic forces and moments and the heat fluxes acting on these faces. The format of this file is:

```
nfaceb1  nfaceb2  nfaceb3
ndperlb   n1    n2    n3   [n4]
ndperlb   n1    n2    n3   [n4]
ndperlb   n1    n2    n3   [n4]
  .       .     .     .
  .       .     .     .
  .       .     .     .
a1     fx1   fy1   fz1   mx1   my1   mz1
a2     fx2   fy2   fz2   mx2   my2   mz2
a3     fx3   fy3   fz3   mx3   my3   mz3
  .      .     .     .     .     .     .
  .      .     .     .     .     .     .
  .      .     .     .     .     .     .
fc1    hf1   ghf1  xc1   yc1   zc1   b1
fc2    hf2   ghf2  xc2   yc2   zc2   b2
fc3    hf3   ghf3  xc3   yc3   zc3   b3
  .      .     .     .     .     .     .
  .      .     .     .     .     .     .
  .      .     .     .     .     .     .
thf20
thf21
  .
  .
thf29
ScaleT     charlen
FX     FY      FZ
MX     MY      MZ
```

On the first line, `nfaceb` is the number of wall faces printed in each of the three sections described below: connectivity, forces, and heat fluxes. A section may be

omitted from being printed if not needed by setting `nfaceb=0`. In the connectivity section (`nfaceb1`), each face is defined by its `ndperlb` nodes `n1, n2, n3,...`. In the forces section (`nfaceb2`), for each face are listed its area `a` and the components `fx, fy, fz` and `mx, my, mz` of the forces and moments acting on the face. Finally, in the heat fluxes section (`nfaceb3`), for each face `fc1, fc2,...` (global face numbering) are listed the heat fluxes `hf1, hf2,...` (calculated from the definition) and `ghf1, ghf2,...` (calculated from Gresho's formulation), followed by the centroids of each face `(xc1,yc1,zc1), (xc2,yc2,zc2),...` of types `b1, b2,...`. The next ten lines give the sum of the heat fluxes (standard definition) over each class of wall types (walls of type 2 are lumped together with walls of type 20). `ScaleT` is the ratio l_∞/U_∞ used for non-dimensionalization of the time t; `charlen` is the characteristic length of the grid. Finally, `FX, FY, FZ` and `MX, MY, MZ` are the total forces and moments acting on the body.

All data in this file are dimensional (in SI units).

A.3 Nodal Displacement File `disp.dat`

This file contains the new positions of the moving nodes in the ALE formulation. The format of the file is:

```
nnode  ntlevl
t1
t2
t3
t4
n1    x1    y1    z1
n2    x2    y2    z2
n3    x3    y3    z3
 .     .     .     .
 .     .     .     .
```

```
 .     .     .     .
x1     y1    z1
x2     y2    z2
x3     y3    z3
 .     .     .
 .     .     .
 .     .     .
```

where `nnode` is the number of moving nodes `n1, n2, n3,...` at the last four times of integration `t1 > t2 > t3 > t4`, followed by the new coordinates (`x,y,z`). For the other `ntlevl-1` time levels, only the grid coordinates are saved (without the global node numbers). The new coordinates are saved only if the grid has been moved. Saving the grid coordinates over four time levels allows the use of high-order schemes to compute $\frac{d\mathbf{x}}{dt}$ following a restart.

The file `disp.dat` is internal to FENSAP and thus the coordinates are saved in non-dimensional form. When used in a restart, the `disp.dat` file is read by FENSAP and the new position of the grid is thus recovered. This approach allows to keep the original grid and, further, storing only the moved nodes is more economical than saving the entire new grids at all the necessary time levels.

Appendix B

Stress Solver Coupling Files

The following appendix describes the internal format of the files belonging to the stress solver for the purpose of the coupling with the flow solver. These formats are specific to the solvers used in this work.

On input, the stress solver requires the aerodynamic loads on the surface of the deforming structure. These loads are read as consistent node-based loads pre-integrated by the coupling driver according to equations (3.7) and (4.10). The format of this file is described in section B.1.

On output, the stress solver writes the nodal displacements over the whole grid. The nodes inside the structure are filtered out by the coupling driver and only the displacements on the boundary of the structure are later passed to the flow solver to minimize the amount of data transferred. The format of this file is described in section B.2.

As for the flow solver, the stress solver can read and write its coupling files in binary or ASCII mode. The binary mode is selected for efficiency.

B.1 The External Loads File `exloads.dat`

In this file, the external time-dependent aerodynamic loads on the surface of the structure are stored in the format:

```
ntsteps   kcoupl   ktmode
t1  nnodeb
  node1   tx1    ty1    tz1
  node2   tx2    ty2    tz2
    .       .      .      .
    .       .      .      .
    .       .      .      .
  fx   fy   fz
  mx   my   mz
```

The variable `ntsteps` gives the number of time levels `t1, t2, ...` listed in the file. If the file `exloads.dat` does not exist, `ntsteps` is set to zero in the code, concluding that no external loads act on the structure. The variables `kcoupl` and `ktmode` carry the same meaning as the ones defined for the flow solver in the file `timebc.dat` (refer to section A.1). During a tight fluid-structure coupling (when the time steps for the flow and the stress solvers are identical), `ktmode=1` and there is usually only one time level specified in the file. The current version of the stress code is in fact limited to this case (`ntsteps=1`).

The nodal stress values `tx`, `ty`, and `tz` at a given time are listed for the `nnodeb` boundary nodes on the surface of the structure, each node being identified by its global node number in the structure grid. The global loads `fx`, `fy`, and `fz` and the moments `mx`, `my`, and `mz` are listed after the nodal stresses.

All data in this file are dimensional (in SI units).

B.2 The Displacements File `displace.dat`

This file contains the structural displacements for the whole structure at the latest time of integration. The format of the file is:

```
nnode
u1      v1      w1
u2      v2      w2
u3      v3      w3
 .       .       .
 .       .       .
 .       .       .
```

where **nnode** is the number of nodes in the CSD grid, followed by the nodal displacements `u, v, w,...`. This file is read by the coupling driver which extracts the displacements on the surface of the structure before feeding them to the flow solver (see section A.3).

Appendix C

Fluid-Structure Coupling Files

The input files for the fluid-structure coupling driver are described in this appendix. The parameter file, which controls the execution of the coupling driver, is described in section C.1. The format of the grid mapping files and of the virtual grid are listed in sections C.2 and C.3, respectively. Finally, a few words are stated in section C.4 about the restart facilities provided by the coupling driver.

There are supplementary output files which are used to monitor the execution of the program. These files are self-explanatory and are not described here.

C.1 The Parameter File

The format of the input parameter file, named `fscouple`, is:

```
timestart  dt  timefinal  tol  ncoupl  mode
nprint
node1
node2
  .
```

.
.

The integration is performed from time `timestart` to time `timefinal` using a constant time step `dt`. All times are in seconds. The integration proceeds to the next time step if the change in the work done on the structure is less than `tol`, up to a maximum of `ncoupl` coupling instances per time step. The variable `mode` indicates if the flow solver is invoked before the stress solver (`mode=1`) or *vice-versa* (`mode=2`). Finally, the coordinates of a number `nprint` of nodes `node1`, `node2`,... of the virtual grid can be printed at each time.

C.2 The Mapping File

There is one mapping file for the fluid grid and one for the structure grid. Each defines the nodes and the faces that are on the surface of the structure and states the correspondence between these surface nodes to the master grid covering the whole domain (fluid or structure). By defaults, these files are named `fluid.map` and `struc.map`. Both files share the same format, given below.

```
nfaceb  nnodeb  ndperlb
  node1   x1   y1   z1
  node2   x2   y2   z2
    .      .      .
    .      .      .
    .      .      .
  n1_1   n2_1   n3_1  [n4_1]
  n1_2   n2_2   n3_2  [n4_2]
    .      .      .      .
    .      .      .      .
    .      .      .      .
```

The variable `nfaceb` gives the number of faces on the structure (fluid or structure). Each face is composed of `ndperlb` nodes (3 or 4) and there are a total of `nnodeb` nodes on the surface. The coordinates are listed first, followed by the connectivity of the faces, using a local numbering system. The fields `node1`, `node2`,... give the global node number of the nodes on the surface in terms of the original grid (fluid or structure).

C.3 The Virtual Grid

The virtual grid defines the surface of the structure and provides correspondence tables to the fluid and the structure grids via the intermediary of the mapping files. The format of the virtual grid file, named `vgrid.map`, is as follows.

```
nfaceb  nnodeb  ndperlb
  index_s1  index_f1   x1   y1   z1
  index_s2  index_f2   x2   y2   z2
     .         .        .    .    .
     .         .        .    .    .
     .         .        .    .    .
  n1_1  n2_1  n3_1   face_s1  face_f1   scale_s1  scale_f1
  n1_2  n2_2  n3_2   face_s2  face_f2   scale_s2  scale_f2
    .     .     .       .         .         .         .
    .     .     .       .         .         .         .
    .     .     .       .         .         .         .
```

The variables `nfaceb`, `nnodeb`, and `ndperlb` have the same meaning as for the mapping file. In this case, the virtual grid is always made of triangular faces, hence `ndperlb=3`. The fields `index_s` and `index_f` indicate the relation of a node of the virtual grid to the structure and the fluid grids (relative to the mapping files). For example, `index_f1` indicates to which node in the fluid mapping file that the first node of the virtual grid corresponds to, if the index is positive, or to which face in the

fluid mapping file it lies in, if the index is negative. The variable `index_s1` carries an analogous meaning.

The pointers `face_s` and `face_f` indicate in which face of the structure and the fluid grids, respectively, a virtual face is contained. For example, the first face of the virtual grid, with nodes `n1_1`, `n2_1`, and `n3_1`, is contained within face `face_s1` of the structure grid and within face `face_f1` of the fluid grid. The fields `scale_s` and `scale_f` are scaling factors for the areas of the faces. In the construction process of the virtual grid, each fluid (structure) face is made up of a number of whole triangular faces. For the structure, `scale_s1` is the ratio of the area of the virtual faces making up the structure face `face_s1` to the area of the original structure face `face_s1`. This ratio is close to unity and any differences are due to the change in the area of a structure face when projected onto a fluid face. By construction, the virtual grid is projected onto the fluid surface, so `scale_f=1` for all virtual faces. This scaling factor is used in the transfer of the aerodynamic loads to allow to conserve the work done on a face-by-face basis when the discretized geometries are not identical.

C.4 The Restart Files

The fluid-structure driver creates two restart files: one for the flow solver and one for the stress solver. By default, these files are named `fluid.sav` and `struc.sav`. They contain the position of the virtual grid on the fluid and the structure surfaces at the time of the interruption. To use in a restart, the extension `.sav` is changed to `.rst`. The format of these internal files is not described since no user intervention is required.

References

[1] T.E. Smith and J.R. Kadambi. "The Effect of Steady Aerodynamic Loading on the Flutter Stability of Turbomachinery Blading", *Journal of Turbomachinery* **115** (1993).

[2] G.P. Guruswamy. "ENSAERO — a Multidisciplinary Program for Fluid/ Structure Interaction Studies of Aerospace Vehicles", *Computing Systems in Engineering* **1**, Nos 2–4 (1990).

[3] D.M. Schuster, D.D. Liu, and L.J. Huttsell. "Computational Aeroelasticity: Success, Progress, Challenge", *Journal of Aircraft* **40**, No. 5 (2003).

[4] S.G. Sheffer et al. "Time-Accurate Simulation of Helicopter Rotor Flows Including Aeroelastic Effects", *AIAA* Paper 97-0399 (1997).

[5] G.P. Guruswamy. "Interaction of Fluids and Structures for Aircraft Applications", *Computers & Structures* **30**, No. 4 (1988).

[6] G.P. Guruswamy and P.M. Goorjian. "Unsteady Transonic Flow Simulation on a Full-Span-Wing-Body Configuration", *AIAA Journal* **26**, No. 12 (1988).

[7] G.P. Guruswamy. "Unsteady Aerodynamic and Aeroelastic Calculations for Wings Using Euler Equations", *AIAA Journal* **28**, No. 3 (1990).

[8] R.D. Rausch, J.T. Batina, and H.T.Y. Yang. "Three-Dimensional Time-Marching Aeroelastic Analyses Using an Unstructured-Grid Euler Method", *AIAA Journal* **31**, No. 9 (1993).

[9] E.M. Lee-Rausch and J.T. Batina. "Wing Flutter Boundary Prediction Using Unsteady Euler Aerodynamic Method", *Journal of Aircraft* **32**, No. 2 (1995).

[10] G.P. Guruswamy. "Vortical Flow Computations on Swept Flexible Wings Using Navier-Stokes Equations", *AIAA Journal* **28**, No. 12 (1990).

[11] E.M. Lee-Rausch and J.T. Batina. "Wing Flutter Computations Using an Aerodynamic Model Based on the Navier-Stokes Equations", *Journal of Aircraft* **33**, No. 6 (1996).

[12] S.A. Goodwin, C. Byun, and M. Farhangnia. "Aeroelastic Computations Using Parallel Computing Systems", *AIAA* Paper 99-0795 (1999).

[13] G.P. Guruswamy and E.L. Tu. "Effects of Modal Symmetry on Transonic Aeroelastic Characteristics of Wing-Body Configurations", *Journal of Aircraft* **26**, No. 3 (1989).

[14] T.H. Fransson and J.M. Verdon. "Panel Discussion on Standard Configurations for Unsteady Flow Through Vibrating Axial-Flow Turbomachine-Cascades", in *Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines, and Propellers (6th Int. Symp. on)*", H.M. Atassi, ed., Springer-Verlag, New York, 1993.

[15] M.A. Bakhle, T.S.R. Reddy, and T.G. Keith Jr. "Time Domain Flutter Analysis of Cascades Using a Full-Potential Solver", *AIAA Journal* **30**, No. 1 (1992).

[16] W. Wu and F. Sisto. "Numerical Simulation for Aeroelasticity in Turbomachines with Vortex Method", *ASME* Paper 94-GT-410, International Gas Turbine and Aeroengine Congress and Exposition, The Hague, Netherlands (1994).

[17] A.H. Boschitsch and T.R. Quackenbush. "Adaptive Grid Computation of Transonic Flows Through Cascades and Investigation of Chordwise Bending Upon Aeroelastic Response", *AIAA* Paper 94-0145 (1994).

[18] J.M. Wolff and S. Fleeter. "Euler Analysis of Oscillating Cascade Unsteady Aerodynamics Using Embedded Composite Grids", *AIAA* Paper 94-0077 (1994).

[19] L. He and J.D. Denton. "Inviscid-Viscous Coupled Solution for Unsteady Flows Through Vibrating Blades: Part 1 — Description of the Method", *Journal of Turbomachinery* **115** (1993).

[20] L. He and J.D. Denton. "Inviscid-Viscous Coupled Solution for Unsteady Flows Through Vibrating Blades: Part 2 — Computational Results", *Journal of Turbomachinery* **115** (1993).

[21] R.S. Abhari and M. Giles. "A Navier-Stokes Analysis of Airfoils in Oscillating Transonic Cascades for the Prediction of Aerodynamic Damping", *ASME* Paper 95-GT-182, International Gas Turbine and Aeroengine Congress and Exposition, Houston, Texas (1995).

[22] T.S.R. Reddy, M.A. Bakhle, and R. Srivastava. "APPLE: An Aeroelastic Analysis System for Turbomachines and Propfans", *AIAA* Paper 92-4712 (1992).

[23] G.A. Gerolymos. "Advances in the Numerical Integration of the Three-Dimensional Euler Equations in Vibrating Cascades", *Journal of Turbomachinery* **115** (1993).

[24] G.A. Gerolymos. "Coupled Three-Dimensional Aeroelastic Stability Analysis of Bladed Disks", *Journal of Turbomachinery* **115** (1993).

[25] M. Vahdati and M. Imregun. "A Non-Linear Aeroelasticity Analysis of a Fan Blade Using Unstructured Dynamic Meshes", *Proceedings of the Institution of Mechanical Engineers. Part C, Journal of Mechanical Engineering Science* **210** (1996).

[26] C. Bréard, M. Imregun, A. Sayma, and M. Vahdati. "Flutter Stability Analysis of a Complete Fan Assembly", *AIAA* Paper 99-0238 (1999).

[27] J.S. Rao. "Turbomachine Blade Vibration", John Wiley & Sons, New York, 1991.

[28] M.B. Giles. "UNSFLOW: A Numerical Method for the Calculation of Unsteady Flow in Turbomachinery", GTL Report No. 205, MIT Gas Turbine Laboratory, 1991.

[29] M. Sleiman. "Simulation of 3-D Viscous Compressible Flow in Multistage Turbomachinery by Finite Element Methods", Ph.D. Thesis, Concordia University, Montréal, Canada, April 1999.

[30] 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**, No. 3 (2003).

[31] B. Koobus and C. Farhat. "On the Implicit Time Integration of Semi-Discrete Viscous Fluxes on Unstructured Dynamic Meshes", *International Journal for Numerical Methods in Fluids* **29** (1999).

[32] H. Guillard and C. Farhat. "On the Significance of the Geometric Conservation Law for Flow Computations on Moving Meshes", *Computer Methods in Applied Mechanics and Engineering* **190** (2000).

[33] 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** (2001).

[34] C. Degand and C. Farhat. "A Three-Dimensional Torsional Spring Analogy Method for Unstructured Dynamic Meshes", *Computers & Structures* **80** (2002).

[35] 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** (2003).

[36] R. Löhner. "Three-Dimensional Fluid-Structure Interaction Using a Finite Element Solver and Adaptive Remeshing", *Computing Systems in Engineering* **1** (1990).

[37] R. Löhner et al. "Fluid-Structure Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids", *AIAA* Paper 95-2259 (1995).

[38] R. Löhner et al. "Applications of Patient-Specific CFD in Medicine and Life Sciences", *International Journal for Numerical Methods in Fluids* **43** (2003).

[39] C.W. Hirt, A.A. Amsden, and J.L. Cook. "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds", *Journal of Computational Physics* **14** (1974).

[40] W.E. Pracht. "Calculating Three-Dimensional Fluid Flows at All Speeds with an Eulerian-Lagrangian Computing Mesh", *Journal of Computational Physics* **17** (1975).

[41] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. "Lagrangian-Eulerian Finite Element Formulation for Incompressible Viscous Flows", *Computer Methods in Applied Mechanics and Engineering* **29** (1981).

[42] T. Nomura and T.J.R. Hughes. "An Arbitrary Lagrangian-Eulerian Finite Element Method for Interaction of Fluid and a Rigid Body", *Computer Methods in Applied Mechanics and Engineering* **95** (1992).

[43] T.E. Tezduyar, J. Liou, and M. Behr. "A New Strategy for Finite Element Computations Involving Moving Boundaries and Interfaces — The Deforming-Spatial-Domain/Space-Time Procedure: I. The Concept and the Preliminary Numerical Tests", *Computer Methods in Applied Mechanics and Engineering* **94** (1992).

[44] T.E. Tezduyar, M. Behr, and S. Mittal. "A New Strategy for Finite Element Computations Involving Moving Boundaries and Interface—The Deforming-Spatial-Domain/Space-Time Procedure: II. Computation of Free-Surface Flows, Two-Liquid Flows, and Flows with Drifting Cylinders", *Computer Methods in Applied Mechanics and Engineering* **94** (1992).

[45] S.K. Aliabadi and T.E. Tezduyar. "Space-Time Finite Element Computation of Compressible Flows Involving Moving Boundaries and Interfaces", *Computer Methods in Applied Mechanics and Engineering* **107** (1993).

[46] C.H. Stephens, A.S. Arena Jr., and K.K. Gupta. "Application of the Transpiration Method for Aeroservoelastic Prediction Using CFD", *AIAA* Paper 98-2071 (1998).

[47] W.Z. Black and J.G. Hartley. "Thermodynamics", 2nd ed., SI version, New York, NY: HarperCollins, 1991.

[48] G.S. Baruzzi, W.G. Habashi, J.G. Guèvremont, and M.M. Hafez. "A Second Order Method for the Finite Element Solution of the Euler and Navier-Stokes Equations", *International Journal for Numerical Methods in Fluids* **20** (1995).

[49] C. Lepage and L.C. Dutto. "A Guide to FENSAP Version 5.0", CFD Lab Internal Technical Report, Concordia University, Montréal, Canada, February 1998.

[50] J. Dompierre et al. "Edge-Based Mesh Adaptation for CFD", *CERCA* Report R95-73, Presented at the Conference on Numerical Methods for the Euler and Navier-Stokes Equations, Montréal, Sept. 1995.

[51] W.G. Habashi et al. "Anisotropic Mesh Adaptation: Towards User-Independent, Mesh-Independent and Solver-Independent CFD Solutions. Part 1: General Principles", *International Journal for Numerical Methods in Fluids* **32**, No. 6 (2000).

[52] "OptiMesh User's Guide – A 3-D Mesh Adaptation Module, Version 4.3.0", Newmerical Technologies International, 2003.

[53] D.G. Holmes and H.A. Chuang. "2D Linearized Harmonic Euler Flow Analysis for Flutter and Forced Response" in *Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines, and Propellers (6th Int. Symp. on)*", H.M. Atassi, ed., Springer-Verlag, New York, 1993.

[54] D.A. Anderson, J.C. Tannehill, and R.H. Pletcher. "Computational Fluid Mechanics and Heat Transfer", Hemisphere Publishing, 1984.

[55] B.A. Robinson, J.T. Batina, and H.T.Y. Yang. "Aeroelastic Analysis of Wings Using the Euler Equations with a Deforming Mesh", *Journal of Aircraft* **28**, No. 5 (1992).

[56] S. Mittal and T.E. Tezduyar. "A Finite Element Study of Incompressible Flows Past Oscillating Cylinders and Airfoils", *International Journal for Numerical Methods in Fluids* **15**, (1992).

[57] C. Lepage and W.G. Habashi. "Fluid-Structure Interactions Using the ALE Formulation", *AIAA* Paper 99-0660 (1999).

[58] M. Lesoinne and C. Farhat. "Geometric Conservation Laws for Flow Problems with Moving Boundaries and Deformable Meshes, and their Impact on Aeroelastic Computations", *Computer Methods in Applied Mechanics and Engineering* **134** (1996).

[59] P.M. Gresho et al. "The Consistent Galerkin FEM for Computing Derived Boundary Quantities in Thermal and/or Fluids Problems", *International Journal for Numerical Methods in Fluids* **7**, (1987).

[60] K. Blakely. "MSC/NASTRAN Basic Dynamic Analysis, User's Guide, Version 68", The MacNeal-Schwendler Corporation, Los Angeles, 1993.

[61] M.T. Jones and M.L. Patrick. "LANZ: Software for Solving the Large Sparse Symmetric Generalized Eigenproblem", Web Technical Report `http://netlib2.cs.utk.edu/lanz/`, 1996.

[62] M. Petyt. "Introduction to Finite Element Vibration Analysis", Cambridge University Press, New York, 1990.

[63] J.N. Reddy. "An Introduction to the Finite Element Method", 2nd edition, McGraw-Hill, Inc., New York, 1993.

[64] J.B. Malone and S.Y. Ruo. "LANN Wing Test Program: Acquisition and Application of Unsteady Transonic Data for Evaluation of Three-Dimensional Computational Methods", Report AFWAL-TR-83-3006, Lockheed Corporation, 1983.

[65] R.G. den Boer. "Revised Results of the Unsteady Transonic Pressure Measurements on the LANN Model", Report NLR TP 89051 U, National Aerospace Laboratory NLR, The Netherlands, 1989.

[66] D. Smith. "A Case Study and Analysis of the Tacoma Narrows Bridge Failure", 99.497 Engineering Project, Department of Mechanical Engineering, Carleton University, Ottawa, Canada, March 29, 1974.

[67] Y.C. Fung. "An Introduction to the Theory of Aeroelasticity", John Wiley & Sons, Inc., N.Y., 1955.

[68] R.L. Bisplinghoff and H. Ashley. "Principles of Aeroelasticity", Dover Publications, Inc., New York, 1962.

[69] E.H. Dowell (ed.), H.C. Curtiss, Jr., R.H. Scanlan, and F. Sisto. "A Modern Course in Aeroelasticity", 2nd edition, Kluwer Academic Publishers, Boston, 1989.

[70] J.R. Cebral and R. Löhner. "Fluid-Structure Coupling: Extensions and Improvements", *AIAA* Paper 97-0858 (1997).

[71] C. Farhat et al. "High Performance Solution of Three-Dimensional Nonlinear Aeroelastic Problems Via Parallel Partitioned Algorithms: Methodology and Preliminary Results", *Advances in Engineering Software* **28** (1997).

[72] J.R. Cebral and R. Löhner. "Conservative Load Projection and Tracking for Fluid-Structure Problems", *AIAA* Paper 96-0797 (1996).

[73] G.P. Guruswamy and C. Byun. "Fluid-Structural Interactions Using Navier-Stokes Flow Equations Coupled with Shell Finite Element Structures", *AIAA* Paper 93-3087 (1993).

[74] C. Lepage and W.G. Habashi. "Conservative Interpolation of Aerodynamic Loads for Aeroelastic Computations", *AIAA* Paper 2000-1449 (2000).

[75] D. A. Curry. "Using C on the UNIX System. A Guide to System Programming", O'Reilly & Associates Inc., 1991.

[76] S.R. Bland. "AGARD Two-Dimensional Aeroelastic Configurations", *AGARD* Advisory Report No. 156, 1979.

[77] S.S. Davis. "NACA 64A010 (NASA Ames Model) Oscillatory Pitching", in Compendium of Unsteady Aerodynamic Measurements, *AGARD* Report No. 702 (1982).

[78] J. Alonso and A. Jameson. "Fully-Implicit Time-Marching Aeroelastic Solutions", *AIAA* Paper 94-0056 (1994).

[79] G.H. Koopmann "The Vortex Wakes of Vibrating Cylinders at Low Reynolds Numbers", *Journal of Fluid Mechanics* **28** (1967).

[80] P.A. Mendes and F.A. Branco. "Analysis of Fluid-Structure Interaction by an Arbitrary Lagrangian-Eulerian Finite Element Formulation", *International Journal for Numerical Methods in Fluids* **30** (1999).

[81] E.C. Yates, Jr., N.S. Land, and J.T. Foughner, Jr. "Measured and Calculated Subsonic and Transonic Flutter Characteristics of a 45° Sweptback Wing Planform in Air and in Freon-12 in the Langley Transonic Dynamics Tunnel", *NASA* TN D-1616, 1963.

[82] "Wood Handbook: Wood as an Engineering Material", U.S. Department of Agriculture Forest Products Laboratory, Agriculture Handbook No. 72, 1974.

[83] C.V. Spain. "Development of NASTRAN Finite Element and Aeroelastic Models of AGARD Standard Configuration I - Wing 445.6", Internal Report, Lockheed Martin Engineering and Sciences, 2000 (draft).

[84] I.H. Abbott and A.E. von Doenhoff. "Theory of Wing Sections", Dover Publications, Inc., New York, 1959.

[85] H.G. Matthies and J. Steindorf. "Fully Coupled Fluid-Structure Interaction Using Weak Coupling", *Proceedings in Applied Mathematics and Mechanics* **1** (2002).

[86] P. Godin. "NS3D's Aeroelasticity Tool Part I - Theory", Internal Memorandum, Pratt & Whitney Canada, Jan. 2000.

[87] P. Godin. "NS3D's Aeroelasticity Tool Part II - Aerodamping Results", Internal Memorandum, Pratt & Whitney Canada, Jan. 2000.

[88] H. Beaugendre, F. Morency, and W.G. Habashi. "ICE3D, FENSAP-ICE's 3D In-Flight Ice Accretion Module", *AIAA Journal of Aircraft* **40**, No. 2 (2003).