National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

The Continuous Adjoint-based Aeroacoustic Shape Optimization by Coupling the Ffowcs-Williams and Hawkings Analogy with the URANS equations

PhD Thesis

Morteza Monfaredi

Supervisor: Kyriakos C. Giannakoglou
Professor NTUA

Athens, 2021

National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

The Continuous Adjoint-based Aeroacoustic Shape Optimization by Coupling the Ffowcs-Williams and Hawkings Analogy with the URANS equations

PhD Thesis

Morteza Monfaredi

Examination Committee:

1. Kyriakos Giannakoglou*(Supervisor), Professor NTUA,
   School of Mechanical Engineering

2. Spyridon Voutsinas*, Professor, NTUA,
   School of Mechanical Engineering

3. Vasilis Riziotis*, Assistant Professor, NTUA,
   School of Mechanical Engineering

4. Christophe Schram, Professor, von Karman Institute, Belgium,
   Environmental and Applied Fluid Dynamics Department

5. John Anagnostopoulos, Professor, NTUA,
   School of Mechanical Engineering

6. Konstantinos Belimbasakis, Professor, NTUA,
   School of Naval Architecture and Marine Engineering

7. Georgios Papadakis, Assistant Professor, NTUA,
   School of Naval Architecture and Marine Engineering

**Member of the Advisory Committee*

Athens, 2021

Abstract

This doctoral thesis is concerned with the mathematical formulation, programming and verification of the continuous adjoint method to a hybrid noise prediction tool in order to perform aeroacoustic shape optimization with an affordable computational cost, even in industrial applications.

Aeroacoustic noise radiated by unsteady flows can be computed using a hybrid acoustic prediction tool, in which the near–field flow results from an unsteady CFD simulation while the acoustic propagation to far–field from the acoustic analogy. In this thesis, the GPU–enabled URANS solver PUMA (Parallel Unstructured Multi–row Adjoint), developed by the Parallel CFD & Optimization Unit of the NTUA, is used as the CFD tool. The compressible flow prediction variant of PUMA is extended with noise prediction capabilities based on the permeable surface Ffowcs Williams and Hawkings (FW-H) equation solver, in the frequency domain, resulting to a hybrid URANS/FW-H solver. At first, the implementation of the FW–H integral is verified through comparisons with the analytical solution of the sound field from a monopole source in uniform flow. The accuracy of the hybrid solver is additionally verified in different test cases by comparing results computed by the FW-H analogy with those of exclusively URANS runs.

The unsteady continuous adjoint solver of PUMA, previously developed for aerodynamic objective functions, is extended to deal with aeroacoustic shape optimization problems. The aeroacoustic objective function is defined in the frequency domain and stands for the total energy contained in the sound pressure spectrum. The method allows the computation of the sensitivity derivatives (SDs) of objective functions with respect to (w.r.t.) the shape controlling parameters (design variables) at a cost independent of their number, enabling the efficient use of Gradient–Based optimization methods.

The SDs are computed based on the so–called Field Integral (FI) adjoint formulation, resulting to the sum of surface and volume integrals in the final expression. Throughout the adjoint development, variations in the eddy viscosity and distance from the nearest wall due to shape changes are taken into account by differentiating the Spalart-Allmaras turbulence model and the Eikonal equation, i.e. a PDE computing distances from the walls, respectively. Use of a permeable FW–H surface located outside the grid displacement area, offers some simplicities regarding the mathematical development and includes the contribution of the acoustic analogy to the adjoint mean–flow and turbulence equations solely as source terms. The unsteady problems are treated in different ways depending on whether the period is constant or may change during the optimization. For the former, flow fields over a single period of the phenomenon are stored only; in contrast, the latter requires the flow fields over the whole solution time window, resulting to increased solution time and storage requirements.

A verification of the part of the code that differentiates the FW–H integral is performed by comparison of its results with closed–form derivatives' expressions. For cone of the constrained optimization problems, a method to handle equality constraints is developed based on a gradient projection with a new deferred correction scheme.

The developed aeroacoustic shape optimization tool is applied to a series of problems. Since a URANS based flow solution is performed, cases are selected from those with strong tonal behavior in their acoustic footprint. In all applications, a good agreement between predicted noise from the hybrid solver and pure URANS is achieved. For a number of pitching and plunging 2D airfoils at different flow conditions, adjoint–based computed SDs are verified w.r.t. those computed by finite differences. Aeroacoustic shape optimization is performed for these airfoils, achieving omni–directional noise reduction. Among them, a lift–constrained noise minimization is also performed which shows to be able to successfully retain the mean lift at its baseline level while still reducing noise. For a plunging airfoil in transonic flow, an evolutionary algorithm is also used to perform shape optimization for a multi–objective function (noise and lift) and results are compared with those of the adjoint–based optimization. Two cases with varying period during the optimization are considered, namely a 2D vortex shedding cylinder in laminar flow and the rod–airfoil benchmark. For the latter, aerodynamic and aeroacoustic results are extensively compared and validated w.r.t. available data in the literature. In addition, an early termination of the unsteady adjoint solution in the rod–airfoil case is shown to be able to considerably reduce the solution time and storage requirement while still computing acceptable SDs. In the same case, it is shown that the developed optimization tool supports objective functions defined in specific frequency ranges.

Regarding 3D applications, the flow around a sphere is solved and acoustic results are verified by comparison to URANS and then a shape optimization is performed. The industrial application of the developed software is conducted within the MADELEINE project funded by the European Union, by optimizing the geometry of an aero–engine intake. To save computational cost, periodic boundary conditions are used to reduce the solution domain size together with the use of a moving reference frame which leads to steady flow and adjoint runs. In order to assure a periodic adjoint solution, a continuous circumferential distribution of receivers at given radius and axial position is used for the computation of the objective function. The unsteady flow fields required for computing the SDs are achieved by properly rotating the steady flow fields.

The first 3 years of this PhD was conducted within the SmartAnswer project funded by the European Union.

Keywords: Aerodynamic, Computational Fluid Dynamics, Computational Aeroacoustic, Shape Optimization, Continuous Adjoint Method, FW-H Analogy.

Acknowledgments

Throughout the completion of this thesis I have received a great deal of support and assistance. My deepest gratitude and respect go to my supervisor, Kyriakos Giannakoglou, Professor NTUA, for trusting me and for giving me the opportunity to work in his amazing research group. Without his enthusiasm, encouragement, support and continuous optimism this thesis would not have been completed.

I am thankful to Spyridon Voutsinas, Professor NTUA, and Vasilis Riziotis, Assistant Professor NTUA, for their critical evaluations and comments on the thesis presentation and their participation in the advisory committee of my thesis. Besides, I would like to thank the rest of my thesis committee, Christophe Schram, Professor von Karman Institute, John Anagnostopoulos, Professor NTUA, Konstantinos Belimbasakis, Professor NTUA and Georgios Papadakis, Assistant Professor NTUA, for their time and insightful comments.

There are no proper words to convey my deep gratitude and respect to Dr. Xenofon Trompoukis and Dr. Konstantinos Tsiakas for the invaluable help they provided to me in the fields of GPU programming and CFD and for so much time they dedicated to the development of the PUMA software. Also, I owe special thanks to Dr. Varvara Asouti for setting-up and maintenance of the GPU cluster of the PCOpt/NTUA. Moreover, her helps in the field of EA-based optimization methods played an important role in this thesis.

I was lucky to be surrounded by people in the PCOpt/NTUA family who made me feel at home during my PhD. For this reason, I am deeply grateful to Ioannis Vryonis, Konstantinos Samouchos, Konstantinos Gkaragkounis, Dr. Evangelos-Papoutsis Kiachagias, Themistoklis Skamagkis, Marina Kontou, Erfan Farhikhteh, Andreas Margetis, Christos Karliampas, Bhargav Krishna Chitneedi, Nikolaos Galanos, Dimitrios Kapsoulis, Flavio Gagliardi and James Koch. We have shared a lot of good moments together; I will always remember this period of my life.

The industrial application of the present work was performed as part of the MADELEINE project [1] which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769025. I would like to thank Rolls-Royce plc. (Dr. S. Shahpar) and the research group of the Institute of Sound and Vibration Research of the University of Southampton (Dr. L. Wu) for providing the initial intake geometry and the boundary conditions of the fan inlet for the industrial case.

I would like to thank European Union and all my fellows in the SmartAnswer project [2] as the first 3 years of this PhD received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie SmartAnswer project (grant agreement No 722401).

I would like to thank my friends for their great companion and moral supports.

Last but not least, I am immensely grateful to my mother, father and sister for their unconditional support that made me feel I have them next to me and backing me during this period abroad.

Acronyms

ABC adjoint boundary conditions

API Application Programming Interface

CAA Computational Aeroacoustics

CFD Computational Fluid Dynamics

CPs Control Points

CUDA Compute Unified Device Architecture

DES Detached Eddy Simulations

DNS Direct Numerical Simulation

DPA Double Precision Arithmetics

E–SI Enhanced–Surface Integral

EA Evolutionary Algorithm

EASY Evolutional Algorithms SYstem

FAE field adjoint equations

FD finite differences

FFD Free–Form Deformation

FI Field Integral

FW-H Ffowcs Williams and Hawkings

G-B Gradient–Based

GPGPU General Purpose Graphic Processing Unit

GPU Graphic Processing Unit

LBM Lattice Boltzmann Metod

LCPE Low–Cost Pre–Evaluation

LES Large Eddy Simulation

MAEA Metamodel–Assisted Evolutionary Algorithm

MOO Multi–Objective Optimization

MPA Mixed Precision Arithmetics

OASPL overall sound pressure level

PCA Principal Component Analysis

PCOpt/NTUA Parallel CFD & Optimization Unit of the National Technical University of Athens

PDE Partial Differential Equations

PSM Problem Specific Model

PTM Potential–Theoretic Method

PUMA Parallel Unstructured Adjoint Multi-row

r.m.s. root mean squared

SA Spalart-Allmaras

SDs sensitivity derivatives

SI Surface Integral

SPA Single Precision Arithmetics

URANS Unsteady Reynolds-Averaged Navier-Stokes

Contents

Chapter 1

Introduction

Effects of noise pollution on human health are increasingly receiving attention. Noise exposure is not only injurious to the auditory system, but also linked with sleep disturbances, impaired cognitive performance and cardiovascular disease [92]. In [49], high levels of aircraft noise were associated with increased risks of stroke, coronary heart disease, and cardiovascular disease for both hospital admissions and mortality in areas near Heathrow airport in London. In response, governments and aviation regulatory bodies around the world are imposing tight regulations regarding noise pollution, particularly during the last decades. For example, based on the Flightpath 2050 report of the European Commission [60], it is mandated to reduce the perceived noise level by 65% from the level in 2000 by the year 2050. This means designers must investigate innovative methods to further improve the process of designing quieter and more efficient systems.

Besides, the sensitivity of the human ear to the noise is based on the logarithm of the acoustic power which adds an extra challenge for noise reduction. For instance, in order to reduce the perceived sound loudness by half, it is required to reduce the noise by almost 10 dB, corresponding to a significant drop by 90% in acoustic power.

Nowadays, flow–induced noise is a major concern in a wide range of industrial applications. Noise from fans in cooling system [145] and side–view mirrors [36, 51] are recognized as important challenge in automotive industry. The noise radiation is also a key factor in wind turbine applications [128] and design of heating, ventilating and air conditioning systems [62]. Noise in aeronautical applications, depending on its source, is divided to engine and air–frame noise. The latter includes the noise emitted by the landing gear [125, 138], wings and the high–lift devices [120, 59]. In high by–pass ratio turbofans, the blade tips of the large diameter fan constitute an important source of noise generation. In addition, the design of the intake is of paramount importance since it is responsible for providing a uniform air supply to the engine while, at the same time, regulating the fan noise propagation towards far–field. During the last decades, the design trend for civil aircraft engines has been towards high and ultra high by-pass ratio turbofans, resulting to a significant reduction in engine noise. Consequently, for many modern aircraft, the air-frame noise is as important as engine noise.

It is therefore essential for the aeroacoustic community to seek for efficient noise reduction strategies for both air-frame and engine noise. In its first step, this requires accurate (and fast) flow simulation tools to identify different noise sources resulting from air-frame and engine together with a noise propagation tool to realize the acoustic behavior in far-field. Of utmost importance is the second step which calls for the incorporation of the acoustic simulation tool within an efficient optimization algorithm. This thesis addresses these two steps; firstly, by making use of a Graphic Processing Unit (GPU)-enabled flow simulation software and extending it with an aeroacoustic noise prediction tool; secondly, by developing a continuous adjoint-based shape optimization method for the acoustic tool. Evolutionary algorithm is also selectively used in this thesis.

1.1 Use of GPUs for Flow Solution

With major progresses made in the field of Computational Fluid Dynamics (CFD), during the last decade, use of GPUs, that were initially designed for graphics rendering, have been increasing as accelerators for high performance computing. Major drawbacks of GPUs were their programming languages and their Application Programming Interface (API)s that were either extremely low-level or oriented exclusively towards graphics processing (OpenGL, Microsoft DirectX e.t.c.). With a goal to fix this shortcoming, Brook programming model was released. This offered one of the first development platforms for General Purpose Graphic Processing Unit (GPGPU) programming [17]. Afterwards, a more advanced programming model, namely the Compute Unified Device Architecture (CUDA), was released by NVIDIA for its own GPUs [21]. Since then GPUs are being extensively used for computationally demanding simulation methods such as Unsteady Reynolds-Averaged Navier-Stokes (URANS), Large Eddy Simulation (LES), Direct Numerical Simulation (DNS) [114] and Lattice Boltzmann Metod (LBM) [76]. The latter is well suited for parallel processing due to its highly localized computations.

In [12], a GPU capable Euler solver is developed using both CUDA and Brook, and it is shown that the CUDA implementation of the 3D solver outperforms the former by almost $5\times$. Most of the GPU programming applications in CFD problems were restricted to structured grids [129, 13]. In [129], incompressible flow Navier-Stokes solver is implemented on a multi-GPUs platform with one CPU assigned to each GPU, for cavity flow simulation. The single GPU implementation in [12], was extended in [13] to account for multiple GPUs by the same group of authors. It is shown that use of the limited size shared memory of GPUs which is characterized by its low latency can effectively increase the memory band-width.

However, an efficient GPU implementation of unstructured CFD codes remained a cumbersome task, especially for unstructured codes with vertex storage. This is due to the lack of a priori knowledge of the neighbors of a vertex and the number of neighbors in such a case. This problem was first addressed in [64] for Parallel Unstructured Adjoint Multi-row (PUMA) by the Parallel CFD & Optimization Unit of the National Technical University of Athens (PCOpt/NTUA). By optimizing the use of available GPU cache memory together with a delicate memory access patterns, a speed-up of approximately $25\times$ and $17.5\times$ was achieved for Single and Double Precision Arithmetics (SPA and DPA), respectively. This work also introduced a Mixed Precision Arithmetics (MPA) technique which improves the memory band width and reaches $22.5\times$ speed-up for 3D inviscid flows, while respecting DPA accuracy. In [7], several flux computation schemes are studied and it is shown that the appropriate flux computation scheme for a certain application requires compromising between parallel speed-up and GPU memory consumption.

1.2 Acoustic Analogies

Acoustic analogies [79, 140] together with the Kirchhoff method [85] are analytically based approaches that propagate near field sound sources to far-field. In such hybrid approaches, the near field data which result from an unsteady flow solver using a reasonable mesh that does not extend too far from the source area. Lighthill's formulation [79] of aeroacoustic wave equation is the backbone of the modern aeroacoustics, which is a relatively young research field. Thanks to the source terms in the Lighthill's equation, it became possible to physically understand the sound produced by free turbulence for the first time. This formulation was later extended by Ffowcs Williams and Hawkings (FW-H), giving rise to the FW-H analogy. Although other analogies are in use nowadays [101, 61], the FW-H analogy is known as the most complete version of acoustic analogies with considerable popularity in aeroacoustic community.

Compared to the Kirchhoff integral method, a major advantage of the FW-H analogy for aeroacoustics problems is its validity even in the non-linear flow regime as it is based upon the conservation laws of fluid mechanics rather than the wave equation as in the Kirchhoff approach. In [50], a Potential-Theoretic Method (PTM) is introduced as an alternative to the classical Kirchhoff formulations in the frequency domain; this overcomes the computation of normal derivatives over the control surface and makes use of an arbitrary Kirchhoff surface without the need of performing interpolation for computing the pressure on this surface. This method together with the FW-H analogy are used in [116] to compute the far-field sound resulted from the NACA12 airfoil at a Reynolds number equal to 1.4×10^5 and zero angle of attack. The near-field results from implicit LES. It is shown that the FW-H analogy provides more robust results and is not limited in the control surface selection process, compared to PTM.

Among different formulations of the FW-H analogy in the literature, two commonly used formulations are Farassat's 1 and 1A [30], for application in time domain, because of their relative low computation cost and their robustness, and Lockard's formulation [83] for frequency domain applications which takes into account the presence of a mean flow (wind tunnel applications). The latter can also be used in 2D problems [152, 151, 90, 42]. In [95], formulation 1C is introduced as a possible extension of Farassat's formulations 1 and 1A based on the convective form of the FW-H equation, and its implementation is demonstrated for some canonical test cases with analytical solution such as a monopole and a dipole radiating in a moving medium as well as a rotating monopole. Other formulations are also proposed for moving medium applications [11, 40].

A widely made assumption in the aeroacoustic community is neglecting the quadrupole term in the FW-H analogy. This, however, might be prone to spurious sound, particularly for cases in which strong vortical gusts cross the permeable FW-H surface. In the literature, different methods are suggested to address this issue. Averaging the results based on different integration surfaces is proposed in [124, 88] while an open integration surface is recommended for a subsonic jet noise simulations in [34]. An extrapolation method [148] and use of a correction term based on the frozen gust assumption [82] are among other possible options to fix the spurious sound issue. However, these methods lack robustness, require user intervention and are case-dependent. In addition, the error caused by the wake passing through the FW-H surface is small in comparison to the main noise signal, making thus safe to neglect the quadrupole terms for a wide range of cases [82, 128, 45, 125].

The FW-H analogy is also widely used for far-field propagation in intake fan noise prediction [109, 111] as well as propeller noise [48]. In [48], a URANS simulation, combined with the FW-H analogy, is used to simulate the noise generated by a high speed 8-bladed propeller, resulting to a good prediction of propeller harmonics compared to experimental data.

The accuracy of the results provided by the FW-H analogy is directly affected by the accuracy of the near-field flow data. Hence, in a wide range of aeroacoustic simulations, this analogy is used with a high fidelity flow solver such as DNS, LES and DES [45, 39] which are characterized with high computational cost, which might be unaffordable for an optimization process. It is known that URANS solvers tend to over-predict the peak levels of the power spectrum [55]; however, they are used in the literature as they are able to predict the overall trend and provide results in qualitative and quantitative agreement with high fidelity solvers, particularly for cases with strong tonal acoustic footprint. In [94], the far-field noise of a pitching airfoil is computed using the FW-H analogy fed by both LES and URANS. It is shown that the URANS based results are in good agreement with LES, though some high-frequency components are absent. In [74], the FW-H analogy combined with URANS is used for acoustic analysis of slat free shear layer.

Use of the URANS equations for acoustic simulations allows the investigation of noise mitigation techniques, with an affordable computational cost. The application of porous coatings as a passive flow control method for reducing the aerodynamic sound from tandem cylinders is studied in [81] using the hybrid URANS/FW-H noise prediction. A popular benchmark case for the numerical simulation of sound generation in air–frame applications is the rod–airfoil configuration in a turbulent flow. In [55], it was shown that a 3D URANS simulation of the rod-airfoil case practically matches the spanwise repeated 2D one and results to minor improvement compared to the 2D solution. Using the URANS/FW-H noise prediction tool for the same benchmark case, in [121], the role of a rotating cylinder is investigated as a noise reduction technique and, in [96], an adjoint–based shape optimization was used to reduce the far–field noise.

1.3 Methods for Aeroacoustic Optimization

Aeroacoustic and/or aerodynamic shape optimization problems can be solved using either gradient–based or global search methods (evolutionary algorithms, simulated annealing, or other meta–heuristics, otherwise referred as stochastic optimization methods). In what follows, a literature survey for both methods are presented.

1.3.1 Evolutionary Algorithms

Evolutionary Algorithms (EAs) are, by far, the most widely used representative of population–based, gradient–free, stochastic optimization methods. An indisputable advantage of EAs is that they may accommodate any ready–to–use evaluation software, to be referred to as the Problem Specific Model (PSM), without having access to its source code. Access to such a tool, even as a black–box solver, is all EAs need. As the search algorithm evolves from generation to generation, the use of appropriate evolution operators (crossover, mutation, elitism) to form the new generation of candidate solutions guarantees that, after an adequate number of generations, the global optimal solution can be found, without being entrapped into local minima. Compared to other optimization methods, EAs are also advantageous as they can easily treat constraints and be extended for Multi–Objective Optimization (MOO) problems to compute Pareto fronts of non–dominated solutions [142]. However, to reach the global optimum without the add–ons to be discussed below, a standard EA usually requires a great number of evaluations and a high computational cost, particularly if the PSM is computationally expensive.

The most common practice for enhancing the EAs efficiency is the use of surrogate evaluation models or metamodels, which gives rise to the so-called Metamodel–Assisted EAs (MAEAs). Surrogate models approximate the response of the PSM, using a set of already evaluated individuals, at much lower computational cost and, of course, with reduced accuracy. The MAEAs use either on–line or off–line trained metamodels, based on whether the metamodel(s) is/are trained before or during the evolution. Also, metamodels might be valid over the entire design space or just in the neighborhood of each individual to be evaluated (usually referred to as global or local metamodels, respectively) [15, 27].

Off–line trained metamodels usually implement a single (global) metamodel and a Design of Experiments (DoE) technique, [91] which selects the initial training patterns. The trained metamodel is exclusively used to evaluate the population members within the EA run. The optimal solution(s) obtained by by such an EA-based run must be re–evaluated on the PSM and, depending on the deviations between performances evaluated on the metamodel and the PSM, the algorithm terminates or goes on with a new search after training a new metamodel.

MAEAs with on–line trained metamodels rely upon the combined use of the PSM and the metamodel. Both tools are employed on the entire EA population, either periodically or by switching from metamodels to the PSM depending on several criteria. A MAEA based on the Low–Cost Pre–Evaluation (LCPE) is proposed by the PCOpt/NTUA in [69], where metamodels are used to pre–evaluate all individuals in a generation and only a few promising population members are re–evaluated on the PSM.

Another way to reduce the computational cost of EAs is the use of distributed and hierarchical optimization schemes [70]. In distributed EAs (DEAs or DMAEAs) [69], a number of sub–populations evolve in semi–isolation and exchange information regularly. According to [63], Hierarchical EAs (HEAs or HMAEAs) are multilevel schemes that each level can be associated with a different evaluation tool, search techniques and/or problem parameterization. In multilevel evaluation, a different evaluation tool/software is assigned to each level. In multilevel search, each level is associated with a different search technique. Any combination of stochastic/heuristic and gradient–based methods can be used. In multilevel parameterization, where different parameterizations are used and each level corresponds to a design space of possibly different dimensionality [147]. Again, populations in each level evolve separately and exchange members or genetic material regularly. The communication among the different levels may be one-way or bidirectional. All of the above variations are available in the general purpose optimization platform Evolutional Algorithms SYstem (EASY) [3] developed by PCOpt/NTUA.

Even if MAEAs are used, the large number of design variables causes a high computational cost, and worsens its prediction capability (this is the so-called "curse of dimensionality" [10]). As a remedy, unsupervised learning techniques that selectively reduce the number of design variables can be used. Principal

Component Analysis (PCA) is used in this context, introduced into the EASY platform in [77] and extended later in [67].

The above mentioned techniques can be used to perform aeroacoustic optimization, given that a noise prediction tool is available. In [126], a GPGPU based aeroacoustic shape optimization is performed for a contra–rotating fan. More than one objectives are used concerning the aerodynamic efficiency and acoustic performance. The sound pressure is evaluated directly using a LBM solver and a MAEAs is utilized for optimization process. Use of metamodels is also reported in [19] where aerodynamic and aeroacoustic shape optimization is performed for a high–speed propeller. The far–field sound is computed by the FW-H integral. A similar strategy is employed in [42] to perform shape optimization, considering noise and lift as objectives, using the PCA–driven MAEA for a plunging airfoil in transonic flow.

1.3.2 Gradient–Based Methods

The second optimization method, which is extensively used in this thesis, is the Gradient–Based (G-B) optimization which is by far the dominant one in the class of deterministic methods. There are plenty of G-B optimization methods [33, 144] requiring computation of the objective function gradient, also referred to as sensitivity derivatives (SDs), or approximation of higher–order SDs, in order to use them in efficient Newton or quasi–Newton methods. Compared to stochastic optimization methods, the main advantage of deterministic, and in particular G-B ones is their computational efficiency, as they guarantee a better solution (closer to the local or global optimum) at each optimization cycle [35]. However, they may easily be trapped into local optima. Also, the choice of the step–size, especially for line search methods, can greatly affect the convergence rate [97].

The most important and demanding constituent of a G-B optimization is the computation of SDs. There are several methods that can compute accurate gradients in problems governed by Partial Differential Equations (PDEs) such as finite differences (FDs), complex variable method and the direct differentiation of the governing PDEs. The simpler is the FD method where each design variable $b_i, i = 1, \ldots, N$ is perturbed by positive and negative value of a small quantity ϵ, followed by a call to the PSM (i.e. a CFD or CFD–CAA solver for aerodynamic and aeroacoustic optimization, respectively) to compute the objective function value J for each perturbed configuration. $\frac{\delta J}{\delta b_i}$ are, then, computed by

$$\frac{\frac{\delta J}{\delta b_i} = J(b_1, b_2, \ldots, b_i + \epsilon, \ldots, b_N) - J(b_1, b_2, \ldots, b_i - \epsilon, \ldots, b_N)}{2\epsilon}$$

that approximates the gradient with second–order accuracy w.r.t. to the selected ϵ. Obviously, such a method requires $2N$ (or N for first–order accuracy) calls to the PSM to compute all gradient components. In addition, the choice of the ϵ value can strongly affect the accuracy as large values deteriorate the accuracy of the predicted SDs, while very small values may lead to numerical errors due to the denominator approaching zero. Similarly, using other methods, (i.e. complex variable method [87, 4] and direct differentiation [26]) have a computational cost proportional to N.

In industrial problems with many design variables, a key issue is the cost for computing the N derivatives of J. This is the reason for which, during the last two or three decades, there is a great interest in the development of adjoint methods [107, 56]. In these methods, the optimization problem is treated as a constrained one, where J is minimized (or maximized) subject to the constraint of satisfying the state equations denoted as $R_n = 0$. Introducing the adjoint variable Ψ into the Lagrangian function $J_{\text{aug}} = J + \Psi_n R_n$, one can formulate a dual problem. Differentiation of J_{aug} w.r.t. b_i and elimination of the derivatives of state quantities w.r.t. b_i yields a set of adjoint equations. The cost of solving these equations is about the cost of a single run of the PSM, hence, making the cost of computing the objective function gradient independent of N. The origin of the adjoint method can be traced back to Lions work in 1971 [80]. The first application of this method in aerodynamic was in 1984 by Pironneau for potential flow problems [108]. It was later extended to flows governed by the compressible Euler equations by Jameson [57, 58].

In the literature, the adjoint method appears in two variants namely, discrete and continuous adjoint. In the former, the governing or primal equations are first discretized and then differentiated and, thus, the adjoint equations are derived directly in discrete form. The discretized residuals of the flow equations can be differentiated "by hand" or by using code transformation tools and/or operator overloading. The latter approach is widely known as Algorithmic Differentiation (AD) [47, 137]. Oppositely, in continuous adjoint, the governing PDEs are first differentiated and then a mathematical development gives rise to the adjoint PDEs, which are discretized and solved [5, 98].

This thesis does not implement discrete adjoint; therefore, any further description and literature review of related research is skipped. However, a few general comments on the two approaches must be provided. Since the continuous adjoint leads to adjoint PDEs, this gives insight into the adjoint equations and boundary conditions. In general, the developer of continuous adjoint method is flexible in choosing the discretization scheme for the adjoint PDEs, which may differ from that used for the primal PDEs, but this should be made carefully in order to compute accurate the gradients of J. On the other hand, discrete adjoint provides the exact gradient of the discrete objective function, with a considerably higher memory footprint though. One should keep in mind that, with a very fine grid and for smooth solutions, both approaches tend to become consistent, computing the exact gradients, [43, 93].

The final expression for computing $\frac{\delta J}{\delta b_i}$ in the continuous adjoint can be derived in different ways which affects the accuracy of the gradients. One option is to obtain an expression that depends only on the quantities computed along the CFD domain boundaries, neglecting the boundary terms containing the residuals of the flow equations (referred to as the severed Surface Integral, severed-SI, approach) [99], the other option leads to expressions that involve also field integrals on the CFD domain (Field Integral, FI, approach) [104]. A comparative study between these two approaches for aerodynamic applications can be found in [73], where it is shown that the FI approach provides more accurate SDs; however, by more complex expressions for the objective function gradient. A third approach known as Enhanced-Surface Integral (E-SI) is introduced by incorporating the adjoint grid displacement equations [71], resulting in boundary integrals only, while matching the accuracy of the FI approach. In this thesis, the FI approach will be employed.

Moreover, in turbulent flows, the differentiation of the turbulence model has been shown to be a key player regarding the accuracy of the computed objective function gradient. Variations in the eddy viscosity were neglected in continuous adjoint development under the commonly used "frozen turbulence" assumption [100]. However, this assumption was questioned by PCOpt/NTUA in [158] (for the first time) and [159] by differentiating the Low-Reynolds Spalart-Allmaras and the High-Reynolds $k-\epsilon$ turbulence models (introducing the notion of adjoint wall functions), respectively. Both these developments improved the accuracy of the computed gradient. This was followed by a series of works on the continuous adjoint to different turbulence models by PCOpt/NTUA [72, 106], and other groups [18]. In [18], the Spalart-Allmaras model for compressible flows is differentiated and variations of the distance from the wall is also taken into account by incorporating the Eikonal equation in the adjoint formulation.

A major obstacle when developing adjoint for unsteady flows is that the adjoint equations march backwards in time, demanding either an excessive amount of data storage for the flow field instants or, an excessive amount of flow field re-computations. Although different techniques such as flow field reconstruction, data compression [143] and temporal/spatial coarsening [96] are proposed to overcoming this issue, unsteady adjoint still remains a quite taxing job. This serves as a reason why adjoint methods have a much stronger background in aerodynamic shape optimization [56, 158, 73] (to name a few of them), than in aeroacoustic ones [115, 153, 154, 29, 66, 89], as aeroacoustic problems are unsteady by nature.

One of the first works involving the adjoint method for aeroacoustic shape optimization is presented in [115] where a discrete adjoint to the 2D URANS equations' solver is developed for inverse shape design and blunt trailing edge noise reduction in turbulent flow. In that work, noise prediction is based on the FW-H analogy. It is shown that drag and noise improvements lead to approximately similar results but they do not yield the same optimized shapes. In [153], a URANS/FW-H based aeroacoustic optimization framework is developed by applying algorithmic differentiation to an open-source solver [127], and is used to reduce the far–field noise from a pitching airfoil in transonic inviscid flow. The same optimization tool is used in [154] to reduce the emitted noise of a 2D jet–flap interaction in turbulent flow by optimal positioning of the wing and flap with respect to the jet nozzle as well as the flap deflection angle. It is also discussed there the effects of using open and closed FW-H integration surface. In addition, use of the same tool can be found in [150, 152] for aeroacoustic shape optimization of a 2D wing–flap configuration and a vortex shedding cylinder. A stochastic noise generation model is coupled with a RANS solver in [155] to be used for broadband noise reduction. In order to reduce the cost of unsteady adjoint, in [96], temporal and spatial coarsening techniques are presented for aeroacoustic shape optimization. The effectiveness of the method is shown for a 2D rod–airfoil configuration. Other acoustic applications of the discrete adjoint can be found in the literature for shape optimization of helicopter rotors for noise minimization [29, 28], aeroacoustic shape optimization for Propeller and rotor-craft [53] and identifying the acoustically optimal distribution of porous material of a flat plate with a porous trailing edge in subsonic flow [156]. In all the above cited papers, discrete adjoint supported by automatic differentiation was used.

As regards continuous adjoint, early papers focused on the optimal control of jet or mixing layer noise. In [122], the sensitivity of aeroacoustic noise radiated by a 2D mixing layer with respect to (w.r.t.) local perturbations of mass and energy is studied. A continuous adjoint–based optimization technique for an LES model, to control the jet noise using active flow actuation, is developed in [38]. In [86], a similar control technique for jet noise shows that, in order to compute reliable gradient directions, the length of the optimization time needs to be restricted. Use of the continuous adjoint for aeroacoustic shape optimization is rare in the literature, even though it offers a lower memory footprint compared to the discrete approach. In [25], the permeable FW-H formula is solved using a finite element method, leading to the necessary continuous adjoint conditions at the interface between the CFD and CAA domains. In [105], a steady continuous adjoint method for the reduction of the side mirror noise of a car using a turbulence-based surrogate objective function is presented. With this model, the omission of the adjoint to the turbulence model equations would merely lead to zero sensitivities, since the objective depends exclusively on turbulence. In regard to engine–intakes, the continuous adjoint to the linearized Euler equations to optimize the shape of a turbofan inlet duct can be found in [109, 110]; however, in these two works, the FW-H analogy in not differentiated. The unsteady continuous adjoint for a hybrid solver including a high fidelity incompressible flow solver and the Kirchhoff integral, for automotive applications, can be found in [66, 65] where the flow-induced sound of the side mirror of a vehicle is reduced. The continuous adjoint method to compressible flows based on the FW-H analogy first appeared by the author of the present thesis in [89] for minimizing the far–field noise from a pitching airfoil. However, [89] was exclusively dealing with an inviscid flow model.

1.4 Thesis Outline

This PhD thesis emphasizes on aeroacoustic shape optimization. In the chapters of this thesis, a hybrid noise prediction tool is developed, the continuous adjoint method for the hybrid solver is formulated and different test cases are presented.

In chapter 2, the governing flow equations are presented together with some details regarding the discretization and numerical solution of the flow equations used by GPU–enabled flow solver PUMA of PCOpt/NTUA which this PhD is exclusively using to predict flow fields. The GPU implementation of PUMA is also briefly outlined in chapter 2.

Chapter 3 is concerned with the development of the noise prediction module for PUMA, resulting to the hybrid URANS/FW-H solver. Details are given regarding the solution of the FW-H solution in both 2D and 3D cases, and then, the coupling of the acoustic module in the frequency domain with the existing URANS solver (PUMA) is outlined.

In chapter 4, the continuous adjoint method for the hybrid solver is presented. The adjoint is formulated for compressible flows using the FI approach with the differentiation of the SA turbulence model included. The adjoint to the FW-H analogy is also presented. The derivation of the adjoint boundary conditions on CFD boundaries is also examined in this chapter.

Chapter 5 is concerned with exclusively verification of the developed acoustic tools. Results of the hybrid solver are compared with the analytical solution of a monopole sound source in a flow–stream, for both 2D and 3D implementations. Additionally, part of the code that differentiates the FW-H integral is verified by comparison of its results with analytical differentiation.

In chapter 6, applications of the adjoint–based optimization by means of the continuous adjoint method developed in chapter 4 are presented. As the flow solution is based on the URANS equations, applications that are characterized with strong tonal behavior in noise generation are selected. In most of the applications, results of the hybrid solver are compared with outcome of pure URANS simulations, also, adjoint–based computed SDs are compared with FDs. A pitching airfoil in inviscid flow is optimized separately for maximum lift and minimum noise. Then, a pitching airfoil in inviscid flow undergoes lift–constrained aeroacoustic shape optimization. Next, a gradient verification is performed for a pitching airfoil in turbulent flow. For a plunging airfoil in transonic flow shape optimization is performed using both adjoint and EAs. The first case with varying period during the optimization is a vortex shedding cylinder in laminar flow. This is followed by the aeroacoustic shape optimization and result validation for the rod–airfoil benchmark case. Finally, shape optimization is performed for an industrial case of an aero-engine intake, separately for minimum total pressure loss and minimum noise.

The conclusions alongside with some recommendations for future work are drawn in chapter 7.

Chapter 2

The Navier–Stokes Equations and Their Numerical Solution

The main focus of this thesis is to develop the aeroacoustic optimization method using the gradient–based technique assisted by the continuous adjoint method. Before proceeding to the aeroacoustic noise prediction tool (Chapter 3) and the development of the corresponding continuous adjoint method (Chapter 4), the flow model used to provide the input data for the acoustic analogy is presented. The flow model consists of the URANS equations for compressible flows and the SA turbulence model (to effect closure in turbulent flows). Since, the industrial case to be examined in the last part of this thesis is concerned with a turbofan application (rotating), the governing flow equations are written in a relative frame of reference rotating with a constant angular velocity. The CFD solver together with its adjoint counterpart made part of the in–house software PUMA. The development of PUMA started about two decades ago in the framework of a number of PhD theses carried out in the PCOpt/NTUA ([146],[6],[131], [132]) and, during the last years, it has been transferred to GPUs and enriched with new features and capabilities. This PhD thesis relies exclusively upon the GPU-enabled variant of PUMA.

In this chapter, the governing flow equations are presented followed by their boundary conditions.

2.1 The URANS Equations for Compressible Flows

Let $O(x_1\ x_2\ x_3)$ be a coordinate system rotating at a constant rotation speed $\omega_m\,(m=1,2,3)$. In this system, the Navier–Stokes equations for compressible flows are expressed as

$$R_n^{\mathrm{MF}} = \underbrace{\frac{\partial U_n}{\partial t}}_{\mathrm{MF}^t} + \underbrace{\frac{\partial f_{nk}^{\mathrm{inv}}}{\partial x_k}}_{\mathrm{MF}^{inv}} - \underbrace{\frac{\partial f_{nk}^{\mathrm{vis}}}{\partial x_k}}_{\mathrm{MF}^{vis}} + \underbrace{S_n}_{\mathrm{MF}^s} = 0 \tag{2.1.1}$$

In Eq. 2.1.1, U_n stands for the conservative flow variables namely $U_n = \begin{bmatrix}\rho & \rho v_1^A & \rho v_2^A & \rho v_3^A & \rho E\end{bmatrix}$, with ρ being the fluid density, $v_m^A\,(m = 1, 2, 3)$ the velocity components w.r.t. the absolute/inertial frame of reference and E the total energy per unit mass. Inviscid fluxes f_{nk}^{inv}, viscous fluxes f_{nk}^{vis} and the source terms S_n (corresponding to the Coriolis acceleration) are defined as

$$f_{nk}^{\mathrm{inv}} = \begin{bmatrix} \rho v_k^R \\ \rho v_1^A v_k^R + p\delta_{1k} \\ \rho v_2^A v_k^R + p\delta_{2k} \\ \rho v_3^A v_k^R + p\delta_{3k} \\ \rho h_t v_k^R + v_k^F p \end{bmatrix} \quad f_{nk}^{\mathrm{vis}} = \begin{bmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ v_\ell^A \tau_{\ell k} + q_k \end{bmatrix} \quad S_n = \begin{bmatrix} 0 \\ \rho \varepsilon_{1\ell k}\omega_\ell v_k^A \\ \rho \varepsilon_{2\ell k}\omega_\ell v_k^A \\ \rho \varepsilon_{3\ell k}\omega_\ell v_k^A \\ 0 \end{bmatrix} \tag{2.1.2}$$

where p stands for the static pressure. In Eqs. 2.1.1-2.1.2 and the rest of the thesis (unless stated otherwise), the Einstein convention is employed according to which summation is indicated by repeated indices in the same term. The relative velocity components v_m^R are linked to the absolute ones v_m^A through the equation $v_m^A = v_m^R + v_m^F$, with $v_m^F = \varepsilon_{m\ell k}\omega_k \left(x_k - x_k^C\right)$ being the rotating/non-inertial frame velocity and x_k^C the position vector of the center of rotation. In Eqs. 2.1.2, τ_{km} is the viscous stress tensor for a Newtonian fluid defined by

$$\tau_{km} = (\mu + \mu_t)\left(\frac{\partial v_k^A}{\partial x_m} + \frac{\partial v_m^A}{\partial x_k} - \frac{2}{3}\delta_{km}\frac{\partial v_\ell^A}{\partial x_\ell}\right) \tag{2.1.3}$$

q_k is the k-th component of the heat flux

$$q_k = \frac{\mathcal{C}_p}{\mathrm{Re}_0}\left(\frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr_t}}\right)\frac{\partial T}{\partial x_k} \tag{2.1.4}$$

where $\mathcal{C}_p$ is the specific heat under constant pressure. T is the static temperature; by assuming a perfect gas, T is related to pressure and density through the equation of state $p = \rho R_g T$ with R_g being the specific gas constant. Pr and $\mathrm{Pr_t}$ are the Prandtl and turbulent Prandtl numbers, respectively. $\gamma = \frac{\mathcal{C}_p}{\mathcal{C}_v}$ is the specific heat ratio with $\mathcal{C}_v$ being the specific heat under constant volume. $h_t = E + \frac{p}{\rho}$ is the total enthalpy which for a perfect gas is linked to pressure (p), density (ρ) and velocity ($v_\ell^A, \ell = 1, \dots, 3$) through

$$h_t = \frac{\gamma p}{\rho\,(\gamma - 1)} + \frac{1}{2} v_\ell^A v_\ell^A \tag{2.1.5}$$

The absolute Mach number is defined by $M = \frac{\sqrt{v_\ell^A v_\ell^A}}{c}$ where c is the local speed of sound which, for perfect gases, is given as $c = \sqrt{\gamma R_g T}$. The total temperature, T_t, is given by

$$T_t = T + \frac{v_\ell^A v_\ell^A}{2\mathcal{C}_p} \tag{2.1.6}$$

and the total pressure for perfect gases in a compressible flow is defined by

$$p_t = p \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma - 1}} \tag{2.1.7}$$

The Reynolds number is defined by $\mathrm{Re} = \frac{\rho v l}{\mu}$ where ρ and v take on their far–field values. l is the characteristic length of the problem, such as the chord length for an airfoil. μ is the dynamic viscosity which is linked to the kinematic viscosity ν through $\mu = \rho\nu$.

Turbulent viscosity μ_t is computed by the one-equation Spalart–Allmaras turbulence model [123]. According to this model, an additional PDE is solved for the turbulence field $\tilde{\nu}$, namely

$$\begin{aligned} R^{SA} = {} & \underbrace{\frac{\partial\left(\rho\tilde{\nu}\right)}{\partial t}}_{\mathrm{SA}^t} + \underbrace{\frac{\partial\left(\rho\tilde{\nu}v_k^R\right)}{\partial x_k}}_{\mathrm{SA}^c} - \underbrace{\frac{\rho}{\mathrm{Re}_0\,\sigma}\left\{\frac{\partial}{\partial x_k}\left[\left(\nu + \tilde{\nu}\right)\frac{\partial\tilde{\nu}}{\partial x_k}\right] + c_{b_2}\frac{\partial\tilde{\nu}}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\right\}}_{\mathrm{SA}^d} \\ & \underbrace{- \rho c_{b_1}\left(1 - f_{t_2}\right)\tilde{S}\tilde{\nu} + \frac{\rho}{\mathrm{Re}_0}\left(c_{w_1} f_w - \frac{c_{b_1}}{\kappa^2} f_{t_2}\right)\left(\frac{\tilde{\nu}}{\triangle}\right)^2}_{\mathrm{SA}^s} \end{aligned} \tag{2.1.8}$$

where $\triangle$ stands for the distance of each point in the flow domain from the closest wall boundary. Solving Eq. 2.1.8, μ_t is computed from $\tilde{\nu}$ by $\mu_t = \rho\tilde{\nu} f_{v_1}$. Eq. 2.1.8 is supplemented by the following relations and constants [123]:

$$
\begin{aligned}
&\chi = \frac{\tilde{\nu}}{\nu}, \quad f_{v_1} = \frac{\chi^3}{\chi^3 + c_{v_1}^3}, \quad f_{v_2} = 1 - \frac{\chi}{1 + \chi f_{v_1}}, \quad S = \sqrt{\varepsilon_{k\ell m}\varepsilon_{kqr}\frac{\partial v_m^A}{\partial x_\ell}\frac{\partial v_r^A}{\partial x_q}}, \\
&\tilde{S} = S + \frac{\tilde{\nu} f_{v_2}}{\mathrm{Re}_0\, \kappa^2\, \Delta^2}, \quad f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}}, \quad g = r + c_{w_2}\left(r^6 - r\right), \\
&r = \min\left(10, \frac{\tilde{\nu}}{\mathrm{Re}_0\, \tilde{S}\kappa^2\, \Delta^2}\right), \; \tilde{\mu} = \rho\tilde{\nu}, \quad f_{t_2} = c_{t_3} e^{-c_{t_4}\chi^2}, \qquad (2.1.9) \\
&c_{v_1} = 7.1, \quad c_{b_1} = 0.1355, \quad c_{b_2} = 0.622, \quad c_{w_1} = \frac{c_{b_1}}{\kappa^2} + \frac{1 + c_{b_2}}{\sigma}, \\
&c_{w_2} = 0.3, \quad c_{w_3} = 2.0, \quad \sigma = \frac{2}{3}, \quad \kappa = 0.41, \quad c_{t_3} = 1.2, \quad c_{t_4} = 0.5
\end{aligned}
$$

2.2 Boundary Conditions

To fully define the flow problem, Eqs. 2.1.1 and 2.1.8 must be associated with a set of appropriate boundary conditions. In this section, some of the available boundary conditions within the PUMA code, presented in the PhD thesis of K. Tsiakas [132], are summarized in below items. These are imposed in a strongly or weakly manner depending on the boundary condition type.

- For no-slip wall boundaries, the absolute velocity is set equal to the wall velocity $v_k^A = v_k^W$. In addition, the turbulence variable $\tilde{\nu}$ is set to zero in case the turbulent boundary layer is resolved down to the wall (Low-Reynolds approach) $\tilde{\nu} = 0$.

- For slip walls, the no-penetration condition applies, namely the normal component of the relative to the wall velocity is set to zero. Let the velocity of the wall boundary be denoted by $v_k^W, (k = 1, 2, 3)$ (i.e. $v_k^W = 0$ for stationary walls and non-zero otherwise). Then the no-penetration condition is expressed as

$$v_k^A \mathsf{n}_k = v_k^W \mathsf{n}_k \qquad (2.2.1)$$

- On the symmetry boundary, symmetry conditions are imposed, i.e. $\frac{\partial U_m}{\partial x_k}\hat{\mathsf{n}}_k^{\mathrm{sym}} = 0, (m = 1, 2, 3, 4, 5)$ and $\hat{\mathsf{n}}_k^{\mathrm{sym}}, (m = 1, 2, 3)$ are the components of the unit normal to the symmetry plane. The no-penetration condition is also imposed, similar to the slip walls (see previous item).

- Along pairs of periodic boundaries, periodic conditions are imposed. In case of peripheral blade rows, two points are periodically paired if their projections on the meridional plane coincide and their circumferential position differs by the blade row pitch. Two paired points share all scalar quantities while vector and tensor quantities (e.g. velocities or their spatial derivatives) should firstly be rotated by the row pitch and then apply similar qualities.

- Depending on the case, the thermal conditions for the wall boundaries can be **a)** adiabatic, **b)** constant temperature (T^W) or **c)** constant heat flux (q^W), which can respectively be written as

$$q_k \mathsf{n}_k = 0, \quad T = T^W \text{ and } \quad q_k \mathsf{n}_k = q^W \tag{2.2.2}$$

- Far–field boundaries are treated by appropriately combining the inlet and outlet boundary conditions, depending on the local velocity field. If flow enters the domain the boundary is locally treated as inlet, otherwise as outlet.

- For subsonic inlet boundaries, the total pressure (p_t^{IN}), total temperature (T_t^{IN}), and inlet absolute velocity direction are specified. The inlet velocity direction is given in terms of two angles, namely θ_1^{IN} and θ_2^{IN}. Thus, for the subsonic inlet boundaries, four quantities are specified and a fifth one are extrapolated from the flow domain. The usual options are to extrapolate **a)** the static pressure, **b)** the absolute velocity magnitude or **c)** the local Mach number. Then, Eqs. 2.1.6-2.1.7 together with the Mach number expression are used to compute all the necessary flow quantities. For supersonic inlet conditions, all five necessary flow quantities are set. Concerning turbulence, the inlet turbulence level is prescribed by setting the values of either the inlet turbulence variable $\tilde{\nu}^{IN}$ or the viscosity ratio $\left(\frac{\nu_t}{\nu}\right)^{IN}$.

- At the subsonic outlet boundary, a single flow quantity is specified which can be **a)** the outlet static pressure distribution, **b)** the outlet mean static pressure or **c)** the outlet mass flow rate. For the last two options, since only the value of an integral quantity over the whole outlet boundary is given, values for pressure and normal to the outlet velocity, respectively, are computed iteratively by uniformly correcting the ones extrapolated from the fluid domain. The remaining four flow quantities are extrapolated from the fluid domain. For the turbulence model, a zero Neumann boundary condition is applied for $\tilde{\nu}$ in the outlet boundaries.

2.3 Discretization of the Governing Equations

Equations 2.1.1 and 2.1.8 are spatially discretized using the vertex–centered variant of the finite volume technique on unstructured meshes, consisting of tetrahedra, pyramids, prisms and hexahedra. At each real–time step, a finite volume is formed around each mesh node by connecting the edges midpoints, face centers and element barycenters of the edges, faces and elements attached to this node. An example of the finite volume formed around a boundary node P is shown in Fig. 2.1, for a 2D case. Note that $\mathsf{n}_k^{PQ_j} = \mathsf{n}_k^{PQ_j^L} + \mathsf{n}_k^{PQ_j^R}$.

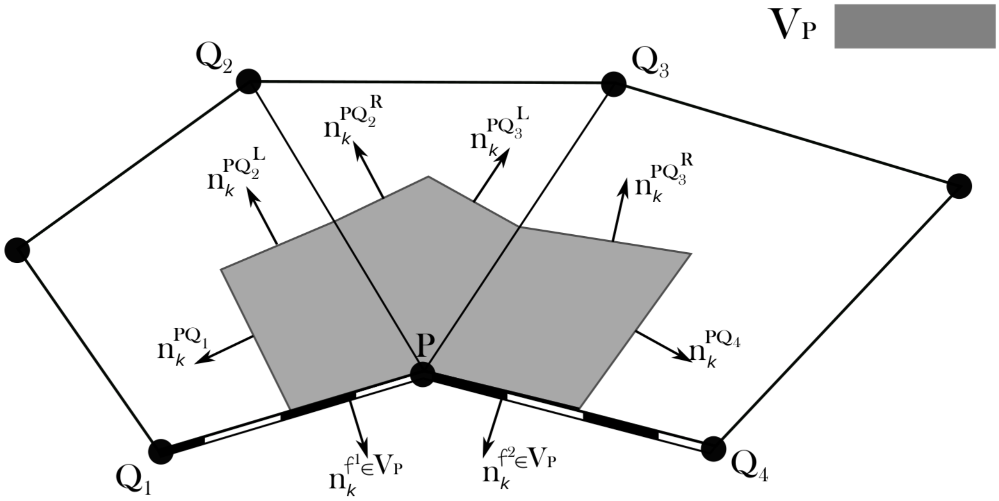


Figure 2.1: A vertex-centered finite volume formed around boundary node P. The normal vectors (n_k) on the finite volume interface are also shown.

2.3.1 Discretization of the Inviscid Terms

Applying the Green-Gauss theorem to the integral of the inviscid terms for the finite volume of node P and discretizing the result, one obtains

$$\int\limits_{V_P^{t_{i+1}}} \frac{\partial f_{nk}^{\text{inv}}}{\partial x_k} \mathrm{d}V = \int\limits_{\partial V_P^{t_{i+1}}} f_{nk}^{\text{inv}} \hat{\mathsf{n}}_k \mathrm{d}\left(\partial V\right) \simeq \sum_{\forall Q \in \mathcal{N}(P)} \Phi_n^{\text{inv},PQ} + \sum_{\forall f \in \mathcal{B}(P)} \Phi_n^{\text{inv},f \in V_P} \qquad (2.3.1.1)$$

where $\partial V_P^{t_{i+1}}$ is the boundary of the finite volume formed around node P at real-time step t_{i+1} and $\hat{\mathsf{n}}_k\left(t\right), \left(k = 1, 2, 3\right)$ the corresponding unit normal (outward) vector components. $Q \in \mathcal{N}\left(P\right)$ is a node neighbouring P, $f \in \mathcal{B}\left(P\right)$ is a boundary face (if any) emanating from node P. In case of mesh elements other than tetrahedra, only the nodes connected to P through a mesh edge are assumed as neighbours. $\Phi_n^{\text{inv},PQ}$ is computed using the Roe's approximate Riemann solver [112] as

$$\Phi_n^{\text{inv},PQ} = \frac{1}{2}\left(f_{nk}^{\text{inv},P} + f_{nk}^{\text{inv},Q}\right)\mathsf{n}_k - \frac{1}{2}\left|\tilde{A}_{nmk}^{PQ}\mathsf{n}_k\right|\left(U_m^R - U_m^L\right) \qquad (2.3.1.2)$$

where $A_{nmk} = \frac{\partial f^{\text{inv}}_{nk}}{\partial U_m}$ is the flux Jacobian, and $\mathsf{n}_k = \mathsf{n}_k^{PQ,t_{i+1}}$ is the normal vector to the finite volume interface between nodes P and Q at time t_{i+1}, with magnitude equal with the area of the interface, Fig. 2.1. $\left|\tilde{A}^{PQ}_{nmk}\mathsf{n}_k\right| = P_{n\ell}\left|\Lambda_{\ell r}\right| P^{-1}_{rm}$ where $|\Lambda|$ is the diagonal matrix containing the absolute eigenvalues of $A_{nmk}\mathsf{n}_k$ computed using the Roe–averaged quantities between the left (L) and right (R) states. The flow variables at L and R are extrapolated from P and Q using spatial gradients and appropriate limiting functions [134, 9, 135]. In the first term of Eq. 2.3.1.2 , the nodal values are used instead of L and R states. According to [5], this maintains the second-order accuracy of the scheme and facilitates the use of a similar scheme in the numerical solution of the continuous adjoint equations.

The last term in Eq. 2.3.1.1 is computed differently for the wall and inlet/outlet boundaries. For the wall boundaries, $f \in \mathcal{B}^{\mathcal{W}}(P)$,

$$\Phi_n^{\text{inv}, f\in V_P} = f_{nk}^{\text{inv}, f\in V_P}\mathsf{n}_k \tag{2.3.1.3}$$

where $f_{nk}^{\text{inv}, f\in V_P}$ is computed by Eq. 2.1.2 using the flow quantities at node P and taking into account the slip wall conditions, as described in section 2.2. In Eq. 2.3.1.3, n_k are the components of the normal vector on the boundary face f, with magnitude equal to face corresponding to the finite volume of node P.

For the inlet/outlet boundaries, $\Phi_n^{\text{inv}, f\in V_P}$ is computed using Eq. 2.3.1.2, where n_k is replaced by $\mathsf{n}_k^{f\in V_P^{t_{i+1}}}$ and node Q represents a halo (fake) node whose flow variables are set using the inlet and outlet boundary conditions as described in section 2.2.

Both Eqs. 2.3.1.2 and 2.3.1.3, include terms of the form $v_k^R\mathsf{n}_k$ along the finite volume boundaries which can be computed as $v_k^R\mathsf{n}_k = v_k^A\mathsf{n}_k - v_k^F\mathsf{n}_k$. Computation of $v_k^F\mathsf{n}_k$ must satisfy the Geometric Conservation Law (GCL) [78] as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{V_P^{t_{i+1}}} \mathrm{d}V = \int\limits_{\partial V_P^{t_{i+1}}} v_k^G \hat{\mathsf{n}}_k \mathrm{d}\left(\partial V\right) \tag{2.3.1.4}$$

where v_k^G is the velocity of the grid. In case of mesh movement (flow around of inside moving boundaries), the finite volume $V_P^{t_{i+1}}$ is computed using a second-order backward difference formula (BDF2). When solving the flow equations in a relative frame of reference on a non-deformable grid, one may substitute v_k^F for v_k^G into Eq. 2.3.1.4. Taking into account that, in such a case, the volume of a cell remains constant, Eq. 2.3.1.4 becomes

$$\int\limits_{\partial V_P^{t_{i+1}}} v_k^F \hat{\mathsf{n}}_k \mathrm{d}\left(\partial V\right) \simeq \sum_{\forall Q\in\mathcal{N}(P)}\left(v_k^{F,PQ}\mathsf{n}_k^{PQ}\right)^{t_{i+1}} + \sum_{\forall f\in\mathcal{B}(P)}\left(v_k^{F,f\in V_P}\mathsf{n}_k^{f\in V_P}\right)^{t_{i+1}} = 0 \tag{2.3.1.5}$$

2.3.2 Discretization of the Viscous Terms

Applying the Green–Gauss theorem to the integral of the viscous terms and discretizing the result one obtains

$$\int\limits_{V_P^{t_{i+1}}} \frac{\partial f_{nk}^{\text{vis}}}{\partial x_k} \mathrm{d}V = \int\limits_{\partial V_P^{t_{i+1}}} f_{nk}^{\text{vis}} \hat{\mathsf{n}}_k \mathrm{d}\left(\partial V\right) \simeq \sum_{\forall Q \in \mathcal{N}(P)} \Phi_n^{\text{vis},PQ} + \sum_{\forall f \in \mathcal{B}(P)} \Phi_n^{\text{vis},f\in V_P} \tag{2.3.2.1}$$

where $\Phi_{m+1}^{\text{vis},PQ} = \tau_{mk}^{PQ} \mathsf{n}_k^{PQ,t_{i+1}}$, $(m = 1, 2, 3)$ are the momentum viscous fluxes crossing the finite volume interface between nodes P and Q, and $\Phi_5^{\text{vis},PQ} = v_\ell^A \left(\tau_{\ell k}^{PQ} + q_k^{PQ}\right) \mathsf{n}_k^{PQ,t_{i+1}}$ the corresponding energy viscous fluxes with τ_{mk}^{PQ} and q_k^{PQ} being the stress tensor and the components of the heat flux, respectively. In order to compute the velocity and temperature spatial derivatives at the finite volume interface between P and Q which is a prerequisite for computing τ_{mk}^{PQ} and q_k^{PQ}, the scheme proposed in [139] is used. For any quantity ϕ, $\left.\frac{\partial \phi}{\partial x_m}\right|_{PQ}$ is computed using

$$\left.\frac{\partial \phi}{\partial x_m}\right|_{PQ} = \left(\overline{\frac{\partial \phi}{\partial x_m}}\right)_{PQ} - \left[\left(\overline{\frac{\partial \phi}{\partial x_\ell}}\right)_{PQ} \mathsf{t}_\ell - \frac{\phi^Q - \phi^P}{\sqrt{\left(x_\ell^Q - x_\ell^P\right)\left(x_\ell^Q - x_\ell^P\right)}}\right] \mathsf{t}_m \tag{2.3.2.2}$$

where

$$\mathsf{t}_m = \frac{x_m^Q - x_m^P}{\sqrt{\left(x_\ell^Q - x_\ell^P\right)\left(x_\ell^Q - x_\ell^P\right)}} \tag{2.3.2.3}$$

and

$$\left(\overline{\frac{\partial \phi}{\partial x_m}}\right)_{PQ} = \frac{1}{2}\left[\left(\frac{\partial \phi}{\partial x_m}\right)_P + \left(\frac{\partial \phi}{\partial x_m}\right)_Q\right]$$

Along the boundaries, the viscous fluxes for the momentum and energy equations are computed as $\Phi_{m+1}^{vis,f\in V_P} = \tau_{mk}^P \mathsf{n}_k^{f\in V_P^{t_{i+1}}}$, $(m = 1, 2, 3)$ and $\Phi_5^{vis,f\in V_P} = \left[\left(\tau_{\ell k} v_\ell^A\right)^{f\in V_P} + q_k^{f\in V_P}\right] \mathsf{n}_k^{f\in V_P^{t_{i+1}}}$, respectively, by also taking the appropriate boundary conditions defined in section 2.2 into account.

2.3.3 Discretization of the Temporal Term

The integral of the temporal term of Eq. 2.1.1 can be expanded using the Reynolds transport theorem as

$$\int\limits_{V_P^{t_{i+1}}} \frac{\partial U_n}{\partial t} \mathrm{d}V = \frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{V_P^{t_{i+1}}} U_n \mathrm{d}V - \int\limits_{\partial V_P^{t_{i+1}}} U_n v_k^G \hat{\mathsf{n}}_k \mathrm{d}\left(\partial V\right) \tag{2.3.3.1}$$

Discretizing the first integral on the right-hand-side (r.h.s.) using a second-order accurate backward difference formula (BDF2), for constant time-step Δt, together with the discretization of the second integral, Eq. 2.3.3.1 reads

$$\begin{aligned}\int\limits_{V_P^{t_{i+1}}} \frac{\partial U_n}{\partial t} \mathrm{d}V \simeq & \frac{1}{2\Delta t}\left(3U_n^{P,t_{i+1}} V_P^{t_{i+1}} - 4U_n^{P,t_i} V_P^{t_i} + U_n^{P,t_{i-1}} V_P^{t_{i-1}}\right) \\ & - \sum_{\forall Q \in \mathcal{N}(P)} \Phi_n^{grid,PQ} - \sum_{\forall f \in \mathcal{B}(P)} \Phi_n^{grid,f \in V_P}\end{aligned} \tag{2.3.3.2}$$

with $\Phi_n^{grid,PQ} = U_n^{PQ}\left(v_k^G \mathsf{n}_k\right)^{PQ,t_{i+1}}$ and $\Phi_n^{grid,f \in V_P} = U_n^P\left(v_k^G \mathsf{n}_k\right)^{f \in V_P,t_{i+1}}$.

Grid velocities v_k^G are computed so as to satisfy the GCL, Eq. 2.3.1.4. Employing a BDF2 scheme, this leads to

$$\sum_{\forall Q \in \mathcal{N}(P)} \left[v_k^G \mathsf{n}_k\right]^{PQ,t_{i+1}} + \sum_{\forall f \in \mathcal{B}(P)} \left[v_k^G \mathsf{n}_k\right]^{f \in V_P,t_{i+1}} = \frac{1}{2\Delta t}\left(3V_P^{t_{i+1}} - 4V_P^{t_i} + V_P^{t_{i-1}}\right) \tag{2.3.3.3}$$

which must be satisfied by the discretization scheme. Employing the same scheme for the time-integration of the flow quantities as for computing the grid velocities fulfills the GCL.

2.3.4 Discretization of the Source Terms

The source values are assumed to remain constant within each finite volume and are, thus, discretized as

$$\int\limits_{V_P^{t_{i+1}}} S_n \mathrm{d}V \simeq S_n^P V_P^{t_{i+1}} \tag{2.3.4.1}$$

2.3.5 Discretization of the Turbulence Model Terms

The temporal, diffusion and source terms of the Spalart–Allmaras equation are discretized with the same schemes as for the mean flow equations. The discretization of the convection term is different. Applying the Green–Gauss theorem to this term leads to the following form

$$\int\limits_{V_P^{t_{i+1}}} \frac{\partial\left(\rho\tilde{\nu}v_k^R\right)}{\partial x_k}\mathrm{d}V = \int\limits_{\partial V_P^{t_{i+1}}} \tilde{\mu}v_k^R\hat{\mathsf{n}}_k\mathrm{d}\left(\partial V\right) = \sum_{\forall Q\in\mathcal{N}(P)} \tilde{\mu}^{PQ}\left(v_k^R\mathsf{n}_k\right)^{PQ,t_{i+1}} + \sum_{\forall f\in\mathcal{B}(P)}\left(\tilde{\mu}v_k^R\right)^P\mathsf{n}_k^{f\in V_P^{t_{i+1}}} \tag{2.3.5.1}$$

where an upwind scheme is used to compute $\left(\tilde{\mu}\right)^{PQ}$ expressed as

$$\tilde{\mu}^{PQ} = \begin{cases} \tilde{\mu}^P\ , & \text{for } \left(v_k^R\mathsf{n}_k\right)^{PQ,t_{i+1}} > 0 \\ \tilde{\mu}^Q\ , & \text{for } \left(v_k^R\mathsf{n}_k\right)^{PQ,t_{i+1}} < 0 \end{cases} \tag{2.3.5.2}$$

2.4 Numerical Solution of the Discretized Equations

The unsteady residuals of the discretized equations can be expressed as

$$\begin{aligned} \mathcal{R}_n^{\mathrm{GE},P} = & \sum_{\forall Q\in\mathcal{N}(P)}\left(\Phi_n^{\mathrm{GE,conv}} - \Phi_n^{\mathrm{GE,diff}}\right)^{PQ,t_{i+1}} \\ & + \sum_{\forall f\in\mathcal{B}(P)}\left(\Phi_n^{\mathrm{GE,conv}} - \Phi_n^{\mathrm{GE,diff}}\right)^{f\in V_P,t_{i+1}} \\ & + S_n^P V_P^{t_{i+1}} + \frac{1}{2\Delta t}\left(3\mathcal{U}_n^{P,t_{i+1}}V_P^{t_{i+1}} - 4\mathcal{U}_n^{P,t_i}V_P^{t_i} + \mathcal{U}_n^{P,t_{i-1}}V_P^{t_{i-1}}\right) \end{aligned} \tag{2.4.1}$$

where $\mathcal{U}$ stands for either U_n ($n = 1,\ldots,5$) or $\tilde{\mu}$ and GE (Governing Equation) for either MF or SA. Equations 2.4.1 are solved using a dual time-stepping technique that updates of the flow variables as follows

$$\frac{V_P}{\Delta\tau_P}\Delta\mathcal{U}_n^P = -\mathcal{R}_n^{\mathrm{GE},P} \tag{2.4.2}$$

where τ is the pseudo–time. Using a point–implicit scheme and denoting the real-time iteration by i and the pseudo–time one by j, the system of equations reads

$$\left[\frac{V_P}{\Delta\tau_P}\delta_{nm} + \left(\frac{\partial\mathcal{R}_n^{\text{GE},P}}{\partial\mathcal{U}_m}\right)_{i+1,j}\right]\left(\Delta\mathcal{U}_m^P\right)_{i+1,j+1} = -\left(\mathcal{R}_n^{\text{GE},P}\right)_{i+1,j} \tag{2.4.3}$$

By splitting the left-hand-side (l.h.s.) term of these equations in diagonal and off-diagonal terms, these are written as

$$\left(\mathcal{D}_{nm}^{\text{GE},P}\right)_{i+1,j}\left(\Delta\mathcal{U}_m^P\right)_{i+1,j+1} + \sum_{\forall Q\in\mathcal{N}(P)}\left(\mathcal{Z}_{nm}^{\text{GE},PQ}\right)_{i+1,j}\left(\Delta\mathcal{U}_m^Q\right)_{i+1,j+1} = -\left(\mathcal{R}_n^{\text{GE},P}\right)_{i+1,j} \tag{2.4.4}$$

where

$$\left(\mathcal{D}_{nm}^{\text{GE},P}\right)_{i+1,j} = \frac{\partial\left(\mathcal{R}_n^{\text{GE},P}\right)_{i+1,j}}{\partial\mathcal{U}_m^P}, \quad \left(\mathcal{Z}_{nm}^{\text{GE},PQ}\right)_{i+1,j} = \frac{\partial\left(\mathcal{R}_n^{\text{GE},P}\right)_{i+1,j}}{\partial\mathcal{U}_m^Q}$$

The system of Eq. 2.4.4 is solved using the Jacobi method, as no synchronizations are required in each iteration when the solution of the system is parallelized on the GPU.

For more details regarding the solution of the discretized equations and the computation of the pseudo-time step, one should refer to [132].

2.5 GPU Implementation of the Flow Solver

GPUs and CPUs are different in terms of architecture and hardware capabilities. GPUs are shared memory processors, meaning that all GPU threads which are executed in parallel, access the same RAM. This may lead to thread race conditions particularly in scatter-add algorithms, employed in the computation of the numerical fluxes and the corresponding numerical flux Jacobians, making the numerical solution process unpredictable. In addition, GPUs (even the latest ones) have limited cache memory compared to CPUs, demanding a different memory handling, so as to minimize the overall memory latency. For these reasons, an efficient GPU-based flow solver calls for different implementation of some numerical algorithms and memory access compared to a CPU-based one.

In this section, some of the GPU-specific techniques, initially developed in the PhD Thesis of X. Trompoukis [131] are summarized. These techniques result in a GPU variant of the flow solver which can be up to 45 times faster compared to the CPU variant of the same flow solver. At this point, it must be noted that the speed-up figures may vary, depending on the actual GPU and CPU device used for the comparison.

2.5.1 Computation of Numerical Fluxes and Flux Jacobians

Use of an edge–based algorithm is the most efficient method for computing the numerical fluxes and their Jacobians in a vertex–centered CPU–based flow solver. In this approach, a sweep over all mesh edges is performed, the numerical fluxes and Jacobians are computed at the finite volume interface associated with each edge and, then, these are added to each of the edge's end-nodes, forming r.h.s. (residuals) and l.h.s. (Jacobian) terms. The GPU equivalent of such an approach is to parallelize this procedure by associating each GPU thread with a mesh edge. However, this may result to a race condition among threads executed in parallel as the GPU memory is shared.

Atomic operations can be used to avoid racing condition. However, this implies an implicit synchronization among the parallel threads if racing condition happens. To overcome this problem, three different techniques have been developed in [131], namely the edge coloring method, the one-kernel and the two-kernel scheme. These techniques are summarized herein.

Computation using Edge Coloring

Historically, this was the first method developed to avoid using atomic operations. In this technique, mesh edges are grouped in a way that threads belonging to the same warp do not access the same memory. The kernel that computes the right-hand-side (r.h.s.) and left-hand-side (l.h.s.) terms is called consecutively for each color group. The parallel efficiency is deteriorated by this method particularly for unstructured meshes that results to a large number of groups. For this reason, the edge coloring technique is not used in the scope of this thesis.

Computation Using a One-Kernel Scheme

In this technique, each mesh node is associated with a GPU thread that computes the numerical fluxes and Jacobians of all mesh edges emanating from the node and add them to the memory position corresponding to this node. This solves the problem of race condition at the expense of extra computations as the numerical fluxes and their corresponding Jacobian matrices are computed twice for all internal mesh edges. Despite the excessive computation, this method outperforms the edge coloring technique as it offers a higher GPU occupancy by the kernel thanks to a more efficient memory access. In addition, this technique offers a low memory requirement as no intermediate data are stored (similarly to the case of the two-kernel technique to be presented right next). Since it offers the best compromise between execution time and memory consumption, it is used for most scatter-add operations in the flow solver.

Computation Using a Two-Kernel Scheme

This technique was proposed in [7] to eliminate the extra computations associated with the one-kernel technique. The scatter-add operation is performed in two steps using two GPU kernels. In the first step, a kernel is launched assigning each mesh edge to a GPU thread to compute the numerical fluxes and Jacobian matrices at the finite volume interface corresponding to every edge and store them using intermediate memory positions. Then, a second kernel associates every GPU thread with a mesh node and sweeps all edges emanating from the node and adds flux and Jacobian contributions to the r.h.s. and l.h.s. Computing the fluxes and Jacobian matrices once for each mesh edge, this technique is computationally efficient and provides the highest speed-up. However, the higher memory requirements of the two-kernel technique restricts its use in the flow solver especially when large 3D meshes are involved, due to the well-known limitations in GPU memory.

2.5.2 GPU Memory Handling

Cache memory in GPUs are limited compared to CPUs rendering the importance of the pattern of storing and accessing data in memory on the efficiency of a GPU code. In this section, the patterns used for storing l.h.s. coefficients are described, since accessing them represents the majority of the memory accesses performed during the numerical solution of the flow equations.

In the CPU implementation of the flow solver, the diagonal terms ($\mathcal{D}$) on the l.h.s. are stored as a list of 5×5 (for compressible flow) matrices. This is far from optimal for a GPU code as the amount of cached memory is small and, in order to minimize cache miss memory accesses, each memory segment transferred through the bus to each multi-processor must contain as much useful data for the threads of the current warp as possible. Therefore, a different pattern is followed. The $(0, 0)$ element of the matrix $\mathcal{D}$ for the node associated with thread 0 is stored, followed by the $(0, 0)$ matrix element of the node associated with thread 1 and so on, up to thread 31, completing the first thread warp (collection of 32 threads which are executed simultaneously). Then, the second element of the matrix $\mathcal{D}$ for node of thread 0 follows, and so on. After the whole matrix is stored for all the nodes associated with the first warp, storage continues similarly for the next warps. In case the number of nodes on the mesh is not a multiple of 32, a small amount of extra memory is used so as to ensure that even the last warp will follow the same memory access pattern. This method ensures that memory accesses performed by the threads of a warp are as few as possible and, as a result, the memory bandwidth achieved is close to the hardware nominal value.

The storage in memory of the off-diagonal terms $\mathcal{Z}$ is even more challenging. The number of $\mathcal{Z}$ matrices that need to be stored is equal to two times the number of mesh edges. In a CPU code, matrices would be stored similar to $\mathcal{D}$, one after the other, but following the ordering of mesh edges. Using such an approach on a GPU code would not guarantee a minimum memory accesses required by the threads of a warp. To overcome this, the $\mathcal{Z}$ matrices for the first edges emanating from the nodes associated with warp 0 are first stored, followed by the $\mathcal{Z}$ matrices for the second edges and so on. The storage of each matrix follows the same pattern used for storing matrices $\mathcal{D}$. However, since the number of neighbors per node is not fixed, threads of the same warp may need to access different number of $\mathcal{Z}$ matrices. To eliminate redundant memory accesses, the nodes of the mesh are ordered w.r.t. the number of their neighbors. By doing so, we ensure that the threads of the same warp need to access approximately the same number of $\mathcal{Z}$ matrices. To keep the same memory pattern for all the threads of a warp, even if the number of neighbors differs slightly among them, the memory space allocated for each node is equal to the one needed from the warp thread with the greater number of neighbors.

2.5.3 Mixed Precision Arithmetics

Apart from improving the memory access pattern (described in the previous subsection), reducing the amount of data to be accessed can also reduce the time spent on accessing memory. This can be achieved by using MPA [64]. In MPA, different arithmetics are used to store the l.h.s. and r.h.s. terms. The r.h.s. terms, contain the residuals of the flow equations represent the physics of the system of equations, while l.h.s. terms ($\mathcal{D}$ and $\mathcal{Z}$ matrices) are linked to the convergence properties of the numerical solution. This means that using inexact values for l.h.s. terms will slightly affect the convergence history of the numerical solution without jeopardizing the solution accuracy. Hence, in MPA approach, r.h.s. terms are computed and stored using DPA, while l.h.s. terms are computed using DPA but stored using SPA. Using MPA not only reduces the number of memory accesses and cache-miss operations needed for accessing the l.h.s. terms, but also reduces memory consumption for storing the demanding l.h.s. terms, which can be crucial in applications involving meshes with a large number of nodes.

Chapter 3

The Noise Prediction Method

Computational Aeroacoustics (CAA) is related to methods and numerical computations which predict the sound emitted by aerodynamic bodies. In a broad sense, these can be classified as direct and hybrid methods, which are shortly discussed in what follows.

Direct methods are based on pure CFD approaches for sound prediction. In fact, the sound field together with the flow field are computed by solving the compressible flow equations. Flow solvers based on DNS [37] or LES [136], capable of resolving principal noise–generating features of the turbulent flow, are the most suitable models for direct noise prediction. Detached Eddy Simulations (DES) can also provide good results if only large flow structures are considered. URANS computations can be used to predict the trend of tonal component in problems with a strong periodic behavior. In contrast to hybrid methods (to be presented next), direct ones can take the effect of sound on flow and inhomogeneity in acoustic medium into account. However, the use of direct methods in industrial applications is rather limited. This is due to the high computational cost of these methods, particularly if the acoustic noise at a location far from the source area must be computed as a fine grid should be extended far away from the near–field, up to the receiver location.

In hybrid methods, the computation of the flow is decoupled from that of sound [20, 118]. The main advantage of such an approach is that it may perform the acoustic propagation in a much cheaper way. In fact, the sound prediction is performed into two steps, namely source computation and noise propagation. The first step performs the computation of the noise sources in an area close to the real source of the noise. This step requires an unsteady flow simulation. In the next step, the sound signal is propagated to the receivers location. The main assumption of any hybrid method is the one–way coupling of flow and sound, as the effect of the sound on the flow field is neglected. The fact that the flow field should become available only in a restricted area around the source of noise, further reduces the computational cost of hybrid methods. Classical CFD tools based on DNS, LES, DES or URANS can be used to obtain the unsteady flow field in the source area. The acoustic propagation part can be performed by methods based on either computational or analytical methods.

The computational approach may involve the numerical solution of the linearized Euler equations (LEE) [52] or the linearized Navier–Stokes equations [75]. In contrast to analytical methods, computational ones are not limited by the free field propagation requirement and are able to handle refractional effects and reflection at solid boundaries. However, these approaches are usually prone to dispersion and diffusion errors; specific treatment of boundary conditions and discretization techniques must be used to avoid them. They also need their own computational grid, thus the cost increases whenever the receiver is located quite far from the source area.

The analytical propagation methods are based on either the Kirchhoff integral [14, 31] or acoustic analogies [140, 23, 79]. These methods propagate the acoustic noise by means of the analytical solution of the wave equation in a free field. Therefore, they are valid only for free field propagation as they cannot take interactions with other surfaces within the propagation field into account.

The Kirchhoff integral can be used only in regions where the linear wave equation accurately describes the flow. This means that the input acoustic pressure and its time and normal derivatives on the integration surface must be within the linear flow region. In [8], this technique was used to compute the radiated noise from an airfoil encountering a vortical gust. In [119], results from two different Kirchhoff formulations were compared to the results from LEE and acoustic analogy for supersonic jet problems. It was shown that all methods including the Kirchhoff formulations provides solutions which are consistent with the outcome of a LES simulation.

Acoustic analogies can also be used to propagate noise from the source region. These are derived by rearranging the flow equations. Equivalent noise sources are distributed over a data surface to simulate the noise generation mechanism inside/on that data surface. The origin of the acoustic analogies can be traced back to Lighthill [79] who first addressed the problem of turbulence and jet noise. This was later extended by Curle [23] to account for the presence of stationary solid surface and, finally, generalized by Ffowcs Williams and Hawkings [140] to include moving surfaces. The FW-H analogy is considered as the most complete acoustic analogy. In contrast to the Kirchhoff method, the FW-H analogy is able to use non–linear input data. A comparison of the FW–H analogy and the Kirchhoff method, both using permeable surfaces for integration, is performed in [14]. Herein, the permeable version of FW-H analogy is used. Details regarding the derivation of the FW-H equation, its analytical solution and its implementation into the PUMA software are given in following.

3.1 The Ffowcs Williams and Hawkins Analogy

For the cases demonstrated in this thesis, the permeable version of the FW-H analogy is used to propagate the noise to receivers. As shown in Fig. 3.1, the data surface from which the noise sources are propagated is mathematically represented by $f = 0$, enclosing the near-field CFD domain. This surface will be referred to as the FW-H surface and f is a signed distance from this surface, with negative and positive values of f corresponding to its interior and exterior, respectively. $H(f)$ is the Heaviside function which is zero for $f < 0$ and unity for $f > 0$. Since solid boundaries are displaced due to either a body motion and/or during a shape optimization loop and the grid or part of it is adapted to the displaced boundaries, it was decided to limit such a grid displacement within the area encapsulated by the FW-H surface. In other words, care is taken to define the FW–H surface in the part of the grid that is not displaced either by the moving body or during the optimization.

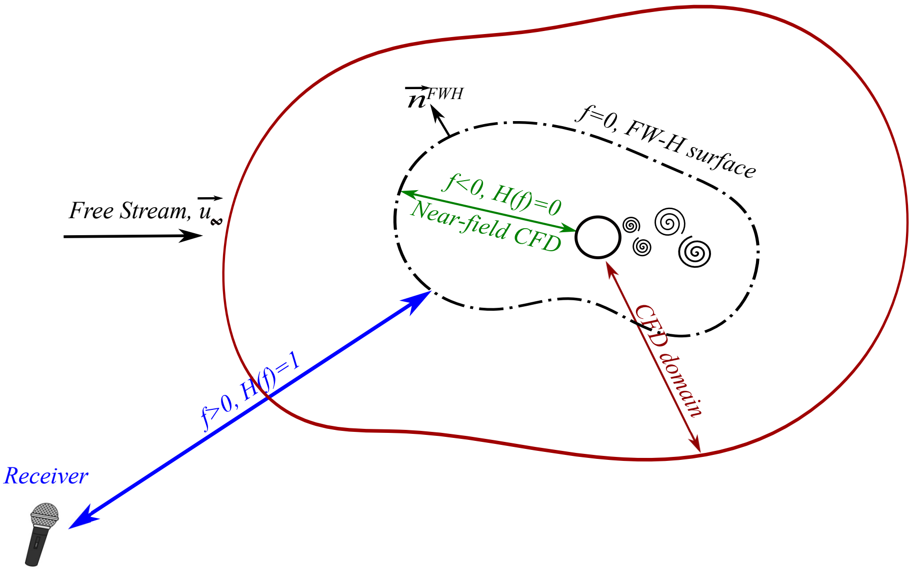


Figure 3.1: Schematic of a permeable FW-H surface, along which $f = 0$. $f < 0$ corresponds to the near-field CFD domain while $f > 0$ refers to the area outside of the FW–H surface. The CFD domain boundaries (red line) delineates the area (2D) or volume (3D) filled with the CFD grid.

The FW-H equation is a rearrangement of the continuity and momentum equations. In this section, v^A is replaced by v. In order to derive the permeable version of the FW-H equation in differential form, it is assumed that there is no mean–flow and the FW-H surface is assumed to have the velocity v_i^{FWH}. As already mentioned, the location of the FW-H surface is defined by $f(\vec{y}, t) = 0$ which satisfies

$$\frac{\partial f}{\partial t} + v_i^{\text{FWH}} \frac{\partial f}{\partial y_i} = 0 \tag{3.1.1}$$

Let us rewrite the continuity and momentum equations in the following form

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_i}{\partial y_i} = 0 \tag{3.1.2}$$

$$\frac{\partial \rho v_i}{\partial t} + \frac{\partial \rho v_i v_j}{\partial y_j} = -\frac{\partial p}{\partial y_i} + \frac{\partial \tau_{ij}}{\partial y_j} \tag{3.1.3}$$

A reference state without the presence of a mean-flow can be defined as

$$\begin{aligned} \rho' &= \rho - \rho_\infty \\ p' &= p - p_\infty \\ v' &= v \end{aligned} \tag{3.1.4}$$

where the subscript ∞ indicates free-stream quantities. Using Eq. 3.1.4 in Eqs. 3.1.2 results to

$$\frac{\partial \rho'}{\partial t} + \frac{\partial \rho v_i}{\partial y_i} = 0 \tag{3.1.5}$$

Multiplying Eq. 3.1.5 with $H(f)$ results to

$$\frac{\partial (H(f)\rho')}{\partial t} + \frac{\partial (H(f)\rho v_i)}{\partial y_i} = \rho' \frac{\partial H(f)}{\partial t} + \rho v_i \frac{\partial H(f)}{\partial y_i} \tag{3.1.6}$$

The Heaviside function has the following properties

$$\begin{aligned} \frac{\partial H(f)}{\partial t} &= \frac{\partial H(f)}{\partial f}\frac{\partial f}{\partial t} = -\delta(f) v_i^{\text{FWH}} \frac{\partial f}{\partial y_i} \\ \frac{\partial H(f)}{\partial y_i} &= \delta(f)\frac{\partial f}{\partial y_i} \end{aligned} \tag{3.1.7}$$

where $\delta(f)$ is the Dirac delta function. Substituting Eq. 3.1.7 into Eq. 3.1.6 and rearranging terms

$$\frac{\partial (H(f)\rho')}{\partial t} + \frac{\partial (H(f)\rho v_i)}{\partial y_i} = \mathcal{Q}\delta(f) \tag{3.1.8}$$

where

$$\mathcal{Q} = \left(\rho v_i - \rho' v_i^{\text{FWH}}\right) n_i^{\text{FWH}} \tag{3.1.9}$$

and it has been used that $\frac{\partial f}{\partial y_i} = n_i^{\text{FWH}} |\nabla f|$ where $\vec{n}^{\text{FWH}}$ is the unit outward normal vector to the FW-H surface. For the sake of simplicity, without loss of generality, f is defined in such a way that $|\nabla f|= 1$.

The same steps can be performed on Eq. 3.1.3 which leads to

$$\frac{\partial(H(f)\rho v_i)}{\partial t}+\frac{\partial(H(f)\rho v_i v_j)}{\partial y_j}+\frac{\partial(H(f)p)}{\partial y_i}-\frac{\partial(H(f)\tau_{ij})}{\partial y_j}=\mathcal{F}_i\delta(f) \tag{3.1.10}$$

where

$$\mathcal{F}_i=\left(\rho v_i\left(v_j-v_j^{\text{FWH}}\right)+p\delta_{ij}-\tau_{ij}\right)n_j^{\text{FWH}} \tag{3.1.11}$$

The next step is to subtract the divergence of Eq. 3.1.10 from the time derivative of Eq. 3.1.8 . Doing so results to

$$\frac{\partial^2(H(f)\rho')}{\partial t^2}=\frac{\partial}{\partial t}(\mathcal{Q}\delta(f))-\frac{\partial}{\partial y_i}(\mathcal{F}_i\delta(f))+\frac{\partial^2}{\partial y_i\partial y_j}(H(f)(\rho v_i v_j+p\delta_{ij}-\tau_{ij})) \tag{3.1.12}$$

Subtracting $c_\infty^2\frac{\partial^2(H(f)\rho')}{\partial y_i^2}$ from both sides of Eq. 3.1.12 gives the FW-H equation as

$$\left(\frac{\partial^2}{\partial t^2}-c_\infty^2\frac{\partial^2}{\partial y_i\partial y_i}\right)(H(f)\rho')=\frac{\partial}{\partial t}(\mathcal{Q}\delta(f))-\frac{\partial}{\partial y_i}(\mathcal{F}_i\delta(f))+\frac{\partial^2}{\partial y_i\partial y_j}(H(f)\mathcal{T}_{ij}) \tag{3.1.13}$$

where

$$\mathcal{T}_{ij}(\vec{y},t)=\rho v_i v_j+\left(p-c_\infty^2\rho'\right)\delta_{ij}-\tau_{ij} \tag{3.1.14}$$

is the Lighthill's tensor. The l.h.s. of Eq. 3.1.13 contains a wave operator and the r.h.s. different source terms contributing to noise generation. The contribution of $\mathcal{T}_{ij}$ is known as the quadrupole term. $\mathcal{F}_i$ is known as the dipole term, related to unsteady force, and $\mathcal{Q}$ can be thought of as an unsteady mass addition also known the monopole contribution.

Using an appropriate Green function, Eq. 3.1.13 can be solved analytically, given that the r.h.s. terms are available. Different formulation are available in the literature for 3D problems [95, 40, 24]. Although most of the significant acoustic phenomena are 3D, the computational cost of performing a 3D simulation to gather the required acoustic sources in the near–field might be limiting, particularly if to be used within an optimization framework. Besides, in cases where the flow mechanism generating noise are pseudo–2D, a 2D simulation predicts well the overall features of the radiated sound with over–predicted amplitudes. Therefore, 2D results can be used to determine the trends even though they do not capture all relevant physics. The usefulness of such 2D simulations is shown in [120] through comparison of 2D and 3D solutions for slat noise and in [96, 89] where 2D acoustic simulations are used for shape optimization with affordable cost. For this reasons, and since herein both 2D and 3D cases are tested, this thesis makes use of both 2D and 3D solutions of the FW-H equation which are presented below.

3.1.1 FW-H Integral for 2D problems

Herein, for two reasons, the frequency domain solution to the FW-H equation is sought. First reason is that this thesis aims to develop the continuous adjoint for an aeroacoustic objective function defined in the frequency domain (to be presented in chapter 4). The second reason is that a time domain solution for 2D problems faces several difficulties due to the tail effect. This is presented below.

A solution to Eq. 3.1.13 in the time domain can be obtained using the Green function. The Green function satisfies the following equation

$$\left(\frac{\partial^2}{\partial t^2} - c_\infty^2 \frac{\partial^2}{\partial y_i \partial y_i}\right) G_{2D}(\vec{x}_r, t; \vec{x}_s, \tau) = \delta(\vec{x}_r - \vec{x}_s)\delta(t-\tau) \tag{3.1.1.1}$$

where $\vec{x}_r$ and $\vec{x}_s$ are the receiver and any source position vectors, respectively, and τ is the source time. G_{2D} is the 2D Green function given by

$$G_{2D}(\vec{x}_r, t; \vec{x}_s, \tau) = \frac{H(c_\infty(t-\tau) - r)}{2\pi c_\infty \sqrt{c_\infty^2 (t-\tau)^2 - r^2}} \tag{3.1.1.2}$$

where $r = |\vec{x}_r - \vec{x}_s|$. The FW-H equation, Eq. 3.1.13 , can be solved by convolution of the sources on the r.h.s. with the Green function,

$$H(f)\hat{\rho}'(\vec{x}_r, \tau) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} Src(\vec{x}_s, \tau) G_{2D}(\vec{x}_r, t; \vec{x}_s, \tau) dS d\tau \tag{3.1.1.3}$$

where Src includes all the sources on the r.h.s. of Eq. 3.1.13 . One of the advantages of the permeable version of the FW-H analogy is that the FW-H surface can be placed at a distance where the quadrupole term $\mathcal{T}_{ij}$ is insignificant from that distance on. Doing so together with taking the Dirac delta functions in the monopole and the dipole term into account, reduces the surface integral of Eq. 3.1.1.3 to a line integral. However the problem of infinite bounds of the time integral still remains in Eq. 3.1.1.3 . The Heaviside function can be used to cut the upper bound of the time integral whereas the lower bound is always infinite. This is the tail effect that might introduce inaccuracies if a quite large time integration limit is not used [83].

To avoid this problem, Eq. 3.1.13 can be solved in the frequency domain. However, applying Fourier transform directly to this equation results to spatial integrals that are difficult to simplify due to the sifting property of $\delta(f)$. The remedy proposed by Lockard [83], is to apply a Galilean transformation before the Fourier transformation, resulting to a convective wave equation.

In air frame noise applications, the Galilean transformation may transform a given problem of one type to another. For example, a problem with a moving receiver can alternatively be seen as one with a stationary receiver in a moving medium, which is the case of a wind tunnel study. In order to account for the presence of a mean flow with constant speed, the convective wave equation can be solved instead of Eq. 3.1.13 , by applying the Galilean transformation to the latter. In this way, simplifications regarding the math and implementation, particularly in 2D problems are made. Here the mean flow is v_∞; the surface velocities v_i^{FWH} in the expressions of $\mathcal{Q}$, $\mathcal{F}_i$ and $\mathcal{T}_{ij}$ should be replaced by the relative surface velocity $v_i^{\text{FWH}} - v_{\infty i}$, practically $-v_{\infty i}$ since the FW-H surface is not moving in the wind tunnel case ($v_i^{\text{FWH}} = 0$). To drive the convective wave equation, the Galilean transformation

$$
\begin{aligned}
&x_i = y_i + v_{\infty_i} t \\
&\bar{t} = t \\
&\frac{\partial}{\partial y_i} = \frac{\partial}{\partial x_i} \\
&\frac{\partial}{\partial t} = \frac{\partial}{\partial \bar{t}} + v_{\infty_i} \frac{\partial}{\partial x_i}
\end{aligned}
$$

is applied to Eq. 3.1.13 yielding

$$
\begin{aligned}
&\left(\frac{\partial^2}{\partial \bar{t}^2} + v_{\infty i} v_{\infty j} \frac{\partial^2}{\partial x_i \partial x_j} + 2 v_{\infty i} \frac{\partial^2}{\partial x_i \partial t} - c_\infty^2 \frac{\partial^2}{\partial x_i \partial x_i} \right) (H(f) \rho') \\
&= \frac{\partial}{\partial \bar{t}} (Q \delta(f)) - \frac{\partial}{\partial x_i} (F_i \delta(f)) + \frac{\partial^2}{\partial x_i \partial x_j} (T_{ij} H(f))
\end{aligned}
\tag{3.1.1.4}
$$

with

$$
\begin{aligned}
Q(\vec{x}, \bar{t}) &= (\rho u_i - \rho_\infty u_{\infty i}) \, n_i^{\text{FWH}} \\
F_i(\vec{x}, \bar{t}) &= \left(\rho \left(v_i - 2 u_{\infty i} \right) v_j + \rho_\infty v_{\infty i} v_{\infty j} + p \delta_{ij} - \tau_{ij} \right) n_j^{\text{FWH}} \\
T_{ij}(\vec{x}, \bar{t}) &= \rho \left(v_i - v_{\infty i} \right) \left(v_j - v_{\infty j} \right) + \left(p - c_\infty^2 \rho' \right) \delta_{ij}
\end{aligned}
\tag{3.1.1.5}
$$

Eq. 3.1.1.4 is rewritten in the frequency domain, by using the Fourier transformation, as follows

$$\begin{aligned}&\left(\frac{\partial^2}{\partial x_i \partial x_i} + k^2 - 2iM_i k\frac{\partial}{\partial x_i} - M_i M_j \frac{\partial^2}{\partial x_i \partial x_j}\right)(H(f)\hat{p}')\\&= -i\omega\hat{Q}\delta(f) + \frac{\partial}{\partial x_i}\left(\hat{F}_i\delta(f)\right) - \frac{\partial^2}{\partial x_i \partial x_j}\left(\hat{T}_{ij}H(f)\right)\end{aligned} \tag{3.1.1.6}$$

where the hat symbol (^) represents variables expressed in the frequency domain and ω is the frequency. $M_i = v_{\infty i}/c_\infty$ and the wave number is defined by $k = \omega/c_\infty$. The integral solution to Eq. 3.1.1.6 provides the pressure fluctuation, with computations performed in the frequency domain, at the receiver's location as follows

$$\begin{aligned}H(f)\hat{p}'(\vec{x}_r,\omega) = &-\int_{f=0}\hat{\mathcal{F}}_i(\vec{x}_s,\omega)\frac{\partial\hat{G}(\vec{x}_r,\vec{x}_s,\omega)}{\partial x_{s_i}}ds - \int_{f=0} i\omega\hat{\mathcal{Q}}(\vec{x}_s,\omega)\hat{G}(\vec{x}_r,\vec{x}_s,\omega)ds\\&-\int_{f>0}\hat{\mathcal{T}}_{ij}(\vec{x}_s,\omega)\frac{\partial^2\hat{G}(\vec{x}_r,\vec{x}_s,\omega)}{\partial x_{s_i}\partial x_{s_j}}dV\end{aligned} \tag{3.1.1.7}$$

$\hat{G}(\vec{x}_r,\vec{x}_s,\omega)$ is the Green function for 2D problems in the frequency domain which is given by

$$\hat{G}_{2D}(\vec{x}_r,\vec{x}_s,\omega) = \frac{i}{4\beta}\exp\left(iMk\bar{x}_1/\beta^2\right)H_0^{(2)}\left(\frac{k}{\beta^2}\sqrt{\bar{x}_1^2+\beta^2\bar{x}_2^2}\right) \tag{3.1.1.8}$$

where

$$\begin{aligned}\bar{x}_1 &= (x_{r_1} - x_{s_1})\cos\theta + (x_{r_2} - x_{s_2})\sin\theta\\ \bar{x}_2 &= -(x_{r_1} - x_{s_1})\sin\theta + (x_{r_2} - x_{s_2})\cos\theta\end{aligned} \tag{3.1.1.9}$$

where the angle $\theta = arctan(u_{\infty 2}/u_{\infty 1})$ is the far-field flow angle and $\beta = \sqrt{1-M^2}$ is the Prandtl-Glauert factor. $H_0^{(2)}(z) = \mathrm{J}_0(z) - i\mathrm{Y}_0(z)$ stands for the Hankel function of the second-kind of zero-order and $\mathrm{J}_0(z)$ and $\mathrm{Y}_0(z)$ are the Bessel function of first and second kind, respectively.

3.1.2 FW-H Integral for 3D problems

As mentioned before, the purpose of this thesis is to develop the continuous adjoint for an aeroacoustic objective function defined in the frequency domain. For this reason, in 3D problems, similar to 2D ones, the frequency domain solution of the FW-H analogy is employed. To do so, the same FW-H integral as in 2D problems in Eq. 3.1.1.7 is used with the only difference that the Green function is changed to the one for 3D problems which for subsonic flows is given by

$$\hat{G}_{3D}(\vec{x}_r, \vec{x}_s, \omega) = -\frac{\exp(-ikr^+)}{4\pi r^*} \tag{3.1.2.1}$$

where r^+ and r^* are given by

$$\begin{aligned} r^+ &= (-\vec{M} \cdot \vec{r} + r^*) \\ r^* &= \sqrt{(\vec{M} \cdot \vec{r})^2 + |\vec{r}|^2 \beta^2} \\ \vec{r} &= \vec{x}_r - \vec{x}_s \\ \vec{M} &= \vec{v}_\infty / c_\infty \end{aligned} \tag{3.1.2.2}$$

The monopole and dipole sources are computed as in the 2D case using Eq. 3.1.1.5 .

3.1.3 Implementation of the FW-H Integral in PUMA

The contribution of the quadrupole term in the FW-H integral Eq. 3.1.1.7 requires a volume integral resulting to a higher computational cost compared to the monopole and dipole terms that require only a surface integral. The quadrupole term is thus assumed to be negligible in many studies. Luckily, this is a safe assumption in many cases since the monopoles and dipoles are the dominant sources [14, 149]. Besides, errors introduced by neglecting volume sources can be even smaller if a permeable integration surface is used; in this case, the surface should enclose all the noise sources. In fact, in this case the quadrupole contributions are implicitly taken into account. As a result, the far–field sound signal can be computed only based on flow variables on the integration surface. However, in the presence of strong wakes, the passage of vortical structures through the FW-H surface can lead to spurious noise which is canceled only if quadrupole terms are included [141]. To address this problem some methods such as the use of an open FW-H surface [133] or on body integration [44] was proposed.

For the cases included in this thesis, a closed permeable FW-H surface is used and the quadrupole terms are neglected. The same set–up is also used for similar test cases in the literature [113, 153, 152]

The FW-H analogy is hybridized with PUMA according to the following steps.

- The unsteady flow solution is performed and, at the end of each time step, terms $\mathcal{Q}$ and $\mathcal{F}_i$ are computed using Eq. 3.1.1.5 and stored at all grid nodes lying on the FW-H surface.

- At the end of the unsteady flow simulation, the mean values of $\mathcal{Q}$ and $\mathcal{F}_i$ are subtracted from their instantaneous values. This is because mean values correspond to the zero frequency without contributing to noise generation.

- Since it is almost infeasible to achieve perfectly periodic flow data from CFD simulations, a Hanning window $\mathcal{H}(t)$ is applied to $\mathcal{Q}$ and $\mathcal{F}_i$ to eliminate possible discontinuities between the beginning and the end of each signal and avoid spectral leakage [83, 88]. The Hanning window $\mathcal{H}(t)$ used in this work is the one proposed in [83], given by

$$\mathcal{H}(t) = \begin{cases} 0.5(1 - cos(8\pi t/T_s)) & \text{if } 0 \leqslant t < T_s/8 \text{ or } 7T_s/8 \leqslant t < T_s \\ 1 & \text{if } T_s/8 \leqslant t < 7T_s/8 \end{cases} \tag{3.1.3.1}$$

  where T_s is the length of the time window and $\mathcal{H}(t)$ is the scaling factor. One can notice that no scaling is applied in the central $(3/4)$ part of the input signal; thus, this windowing technique preserves the relative amplitude of the signal center. This window is designed to be energy preserving while the amplitude is not necessarily preserved. Note that inaccuracies in the peak amplitude of the signal are introduced if the amount of the unsteady flow data are inadequate.

- Fourier transformation is performed to transform $\mathcal{Q}$ and $\mathcal{F}_i$ into the frequency domain ($\hat{\mathcal{Q}}$ and $\hat{\mathcal{F}}_i$).

- Pressure fluctuations in the frequency domain are computed by numerically integrating Eq. 3.1.1.7 . In practice, this computationally cheap integral (a surface integral after neglecting the quadrupole terms) must be computed for each receiver at each frequency. At the end, an inverse Fourier transformation is applied to recover the signal in the time domain.

Chapter 4

The Continuous Adjoint Method for Aeroacoustic Shape Optimization

This chapter focuses on the development of the continuous adjoint method for aeroacoustic shape optimization problems. In gradient–based optimization methods, applied to problems governed by (systems of) PDEs, such as in aerodynamic or aeroacoustic optimization, a main burden is to compute (sometimes, to approximate) the gradient of an objective function J w.r.t. a set of design variables $b_i,\ i=1,\dots,N$, in each optimization cycle. In order to avoid any confusion with the spatial gradient of flow quantities, from this point on, the gradient $\frac{\delta J}{\delta b_i}$ will be referred to as the SDs.

For the sake of demonstration, let J be an integral quantity defined along some surface boundaries of the flow domain S and/or over the fluid volume Ω.

$$J = \int\limits_{\Omega} J_\Omega \mathrm{d}\Omega + \int\limits_{S} J_S \mathrm{d}S \tag{4.1}$$

Note that S along which the quantity J is defined is generally a subset of the whole boundary of the flow $\partial\Omega$.

Any field quantity Φ can be expressed as a function of the flow variables $\mathcal{U}_n,\ (n=1,\dots,6$ with $\mathcal{U}_m := U_m$ for $m=1,\dots,5$ and $\mathcal{U}_6 := \tilde{\nu})$ and the position in space $x_k,\ (k=1,\dots,3)$. It is important to distinguish between the partial derivative of a field quantity Φ w.r.t. b_i $\left(\dfrac{\partial \Phi}{\partial b_i}\right)$ and the corresponding total derivative $\left(\dfrac{\delta \Phi}{\delta b_i}\right)$. The partial derivative refers to the change in Φ caused only by changes in the flow variables $\mathcal{U}_n$ due to changes in b_i, while the total derivative includes also the change in Φ caused by the change in position x_k of all mesh nodes. Thus, partial and total derivatives are linked through

$$\frac{\delta \Phi}{\delta b_i} = \frac{\partial \Phi}{\partial b_i} + \frac{\partial \Phi}{\partial x_k}\frac{\delta x_k}{\delta b_i} \tag{4.2}$$

Equation 4.1 can be differentiated by taking Eq. 4.2 into account and using also the Leibniz rule, resulting to

$$\begin{aligned}\frac{\delta J}{\delta b_i} = &\int\limits_{\Omega} \frac{\partial J_\Omega}{\partial \mathcal{U}_n}\frac{\partial \mathcal{U}_n}{\partial b_i} d\Omega + \int\limits_{\partial\Omega} J_\Omega \frac{\delta x_k}{\delta b_i} \mathsf{n}_k dS + \int\limits_{S} \frac{\partial \left(J_S dS\right)}{\partial \mathcal{U}_n}\frac{\delta \mathcal{U}_n}{\delta b_i} \\ &+ \int\limits_{S} \frac{\partial \left(J_S dS\right)}{\partial \left(\mathsf{n}_k dS\right)}\frac{\delta \left(\mathsf{n}_k dS\right)}{\delta b_i} + \int\limits_{S} \frac{\partial \left(J_S dS\right)}{\partial x_k}\frac{\delta x_k}{\delta b_i}\end{aligned} \tag{4.3}$$

The second, fourth and fifth integrals of Eq. 4.3 can be computed by taking the parameterization of the surface into account. However, the first and third integrals contain variations in the flow variables w.r.t. b_i which are associated with high computational cost. Note that the simplest method for computing this gradient is by means of FDs with a cost that scales with N; it, thus, becomes very expensive in problems with many design variables (N >>). For example, if central FDs are used, the cost is equal to two equivalent flow solutions (EFS) for each design variable (one around the geometry resulting by perturbing the design variable by a small number ϵ and one around the geometry resulting by altering the design variable by $-\epsilon$). Even if direct differentiation is employed, the cost is equivalent to one EFS per design variable. In this thesis, FDs are occasionally used to compute "accurate" SDs to compare with the outcome of the adjoint method. The adjoint method, however, makes the cost of computing $\frac{\delta J}{\delta b_i}$ independent of the number of design variables.

Between the two variants of the adjoint method (discrete and continuous), the continuous adjoint method for compressible turbulent flows with an earoacoustic objective function will be developed in the following sections. The continuous adjoint method can be formulated in two different ways, namely the surface integral (SI) [99, 157], and FI [103, 84] formulations. Although they are mathematically equivalent but lead to different sensitivity derivative expressions with different accuracy, especially in turbulent flow cases. The development of the continuous adjoint method will be presented based on the FI formulation.

In aerodynamic optimization problems, the objective function (such as total pressure losses, lift or drag) is usually defined along some of the boundaries of the CFD domain. This contributes to the adjoint boundary conditions (ABC) and the SDs. Conversely, in aeroacoustic problems, the objective function is defined at the remote receiver location and does not affect the ABC and SDs, while it contributes to the adjoint equations by source terms applied along the FW-H surface. A proper objective function J for aeroacoustic cases can be expressed in the frequency domain as the total energy contained in the spectrum of the sound pressure at $\vec{x}_r$ as

$$J = \int\limits_{\omega} |\hat{p}'(\vec{x}_r, \omega)| \, \mathrm{d}\omega \tag{4.4}$$

where $\hat{p}'(\vec{x}_r, \omega)$ results from Eq. 3.1.1.7 and $|\hat{p}'| = \sqrt{\hat{p}'^2_{\mathrm{Re}} + \hat{p}'^2_{\mathrm{Im}}}$; subscripts Re and Im refer to the real and imaginary part of a complex variable. The integral may cover all resolved or a predetermined range of frequencies. Throughout this thesis, Eq. 4.4 may be referred to as the aeroacoustic or noise objective function.

The aeroacoustic objective function is linked to the flow data on the FW-H surface through the FW-H integral. Therefore, differentiation of this objective function w.r.t. b_i results to variations in the flow variables w.r.t. b_i. In order to avoid this computationally expensive term, the augmented objective function J_{aug} is introduced, which is defined as

$$J_{\mathrm{aug}} = J + \int\limits_{T_s}\int\limits_{\Omega} \Psi_n R_n \mathrm{d}\Omega \mathrm{d}t + \int\limits_{T_s}\int\limits_{\Omega} \tilde{\nu}_a R_{\tilde{\mu}} \mathrm{d}\Omega \mathrm{d}t, \quad n = 1, \dots, 5 \tag{4.5}$$

In Eq. 4.5, $\Psi_n, (n = 1, \dots, 5)$ are the mean flow adjoint variables and $\tilde{\nu}_a$ the adjoint turbulence model variable. Both Ψ_n and $\tilde{\nu}_a$ act as Lagrange multipliers since they multiply the equality constraints of the flow equations in the problem of minimizing J_{aug}. T_s is the time window in which the flow simulation is performed, practically from time 0 till the end of the unsteady simulation. Upon convergence of the flow equations (i.e. $R_n = 0$ and $R_{\tilde{\mu}} = 0$) $J_{\mathrm{aug}} = F$ and, consequently the SDs can be computed from $\frac{\delta J_{\mathrm{aug}}}{\delta b_i}$.

Next step is to differentiate J_{aug} w.r.t. the design variables b_i. The volume integrals of J_{aug} are differentiated by directly passing the $\frac{\delta}{\delta b_i}$ operator inside the integrals. Since these integrals contain spatial gradients of flow–related quantities (e.g. inviscid and viscous fluxes), a useful relation of treating variations of such quantities is sought [103, 71]. Using Eq. 4.2 and computing the spatial gradient of the total variation of an arbitrary flow related quantity Φ one obtains

$$\frac{\partial}{\partial x_\ell}\left(\frac{\delta \Phi}{\delta b_i}\right) = \frac{\partial}{\partial x_\ell}\left(\frac{\partial \Phi}{\partial b_i}\right) + \frac{\partial^2 \Phi}{\partial x_k \partial x_\ell}\frac{\delta x_k}{\delta b_i} + \frac{\partial \Phi}{\partial x_k}\frac{\partial}{\partial x_\ell}\left(\frac{\delta x_k}{\delta b_i}\right) \tag{4.6}$$

Using Eq. 4.2, one can also get an expression for the total derivative of the spatial gradient of Φ as

$$\frac{\delta}{\delta b_i}\left(\frac{\partial \Phi}{\partial x_\ell}\right) = \frac{\partial}{\partial b_i}\left(\frac{\partial \Phi}{\partial x_\ell}\right) + \frac{\partial^2 \Phi}{\partial x_k \partial x_\ell}\frac{\delta x_k}{\delta b_i} \tag{4.7}$$

Subtracting Eq. 4.6 from Eq. 4.7, the total variation of the gradient of Φ is linked to the gradient of the total variation of Φ through

$$\frac{\delta}{\delta b_i}\left(\frac{\partial \Phi}{\partial x_\ell}\right) = \frac{\partial}{\partial x_\ell}\left(\frac{\delta \Phi}{\delta b_i}\right) - \frac{\partial \Phi}{\partial x_k}\frac{\partial}{\partial x_\ell}\left(\frac{\delta x_k}{\delta b_i}\right) \tag{4.8}$$

Eq. 4.8 is extensively used in the remainder of this section, in which the FI continuous adjoint is presented.

Differentiation of J_{aug} results to

$$\frac{\delta J_{\text{aug}}}{\delta b_i} = \frac{\delta J}{\delta b_i} \underbrace{+\int\limits_{T_s}\int\limits_{\Omega} \Psi_n \frac{\delta R_n}{\delta b_i} \text{d}\Omega \text{d}t}_{\mathcal{I}_{\text{MF}}} \underbrace{+\int\limits_{T_s}\int\limits_{\Omega} \tilde{\nu}_a \frac{\delta R_{\tilde{\mu}}}{\delta b_i} \text{d}\Omega \text{d}t}_{\mathcal{I}_{\text{SA}}} \tag{4.9}$$

During the mathematical development of $\frac{\delta J_{\text{aug}}}{\delta b_i}$, volume and surface integrals containing $\frac{\partial U_n}{\partial b_i}$ arise. These integrals are associated with high computational cost. In order to eliminate the expensive volume integral, these are collected together and the multiplier of $\frac{\delta U_n}{\delta b_i}$ is set to zero. By doing so, a new set of PDEs, the so-called field adjoint equations (FAE) arises. Similar approach is followed for the surface integrals leading to the introduction of ABC. Surface or volume integrals including variation of geometric quantities contribute into the final SDs formula. Volume integrals of such a kind are computed by differentiating the mesh morphing technique that is employed to deform the CFD mesh in each optimization cycle. This differentiation is done either analytically, should an analytical mesh morphing method be used, or through FDs.

In what follows, the three terms on the r.h.s. of Eq. 4.9 are differentiated.

4.1 Differentiation of the Mean Flow Equations

Term $\mathcal{I}_{\text{MF}}$, corresponding to mean flow equations, is expanded as

$$\begin{aligned}\mathcal{I}_{\text{MF}} =& \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \Psi_n \frac{\delta}{\delta b_i}\left(\frac{\partial U_n}{\partial t}\right) \text{d}\Omega \text{d}t}_{\mathcal{I}_{\text{MF\_temp}}} + \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \Psi_n \frac{\delta}{\delta b_i}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right) \text{d}\Omega \text{d}t}_{\mathcal{I}_{\text{MF\_inv}}} \\ & \underbrace{-\int\limits_{T_s}\int\limits_{\Omega} \Psi_n \frac{\delta}{\delta b_i}\left(\frac{\partial f_{nk}^{vis}}{\partial x_k}\right) \text{d}\Omega \text{d}t}_{\mathcal{I}_{\text{MF\_vis}}} + \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \Psi_n \frac{\delta S_n}{\delta b_i} \text{d}\Omega \text{d}t}_{\mathcal{I}_{\text{MF\_src}}}\end{aligned} \tag{4.1.1}$$

Introducing Eq. 4.8 into $\mathcal{I}_{\text{MF\_temp}}$ and since time discretization does not depend on b_i this term becomes

$$\mathcal{I}_{\text{MF\_temp}} = \int\limits_{T_s}\int\limits_{\Omega} \frac{\partial}{\partial t}\left(\Psi_n \frac{\delta U_n}{\delta b_i}\right) \mathrm{d}\Omega \mathrm{d}t - \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \frac{\delta U_n}{\delta b_i}\frac{\partial \Psi_n}{\partial t} \mathrm{d}\Omega \mathrm{d}t}_{\text{FAE}} - \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \Psi_n \frac{\partial U_n}{\partial x_k}\frac{\partial}{\partial t}\left(\frac{\delta x_k}{\delta b_i}\right) \mathrm{d}\Omega \mathrm{d}t}_{\text{SD}} \tag{4.1.2}$$

The second term on the r.h.s. contains variations in the flow variables w.r.t. b_i and, since this is a volume integral, it contributes to the FAE. The last term on the r.h.s. of Eq. 4.1.2 contains only variations in geometric terms, hereby it gives contribution to the SDs expression. The first term of the r.h.s. of Eq. 4.1.2 is expanded as follows

$$\int\limits_{T_s}\int\limits_{\Omega} \frac{\partial}{\partial t}\left(\Psi_n \frac{\delta U_n}{\delta b_i}\right) \mathrm{d}\Omega \mathrm{d}t = \int\limits_{\Omega} \Psi_n \frac{\delta U_n}{\delta b_i} \mathrm{d}\Omega \Bigg|_0^{T_s} \underbrace{- \int\limits_{T_s}\int\limits_{\partial\Omega} \Psi_n \frac{\delta U_n}{\delta b_i} v_m^{grid} n_m \mathrm{d}S \mathrm{d}t}_{\text{S.I\_1}} \tag{4.1.3}$$

where $v_m^{grid} = \frac{\delta x_m}{\delta t}$ is the grid velocity on the boundary. A zero initial Dirichlet condition for the adjoint field at the end of the primal solution time eliminates the first term on the r.h.s. of Eq. 4.1.2 at T_s; this term is also eliminated at $t = 0$ as the initial flow field is fixed. If the exact period of the unsteady problem is known and this remains constant during the optimization, the first term on the r.h.s. of Eq. 4.1.2 , is eliminated due to the periodic adjoint and flow field. The second term on the r.h.s. involves variations in the flow variables w.r.t. b_i and, since this is a surface integral, contributes to the definition of boundary conditions for the adjoint equations.

The inviscid term $\mathcal{I}_{\text{MF\_inv}}$ in Eq. 4.1.1 is expanded as

$$\mathcal{I}_{\text{MF\_inv}} = \int\limits_{T_{sol}}\int\limits_{\Omega} \Psi_n \frac{\partial}{\partial x_k}\left(\frac{\delta f_{nk}^{inv}}{\delta b_i}\right) \mathrm{d}\Omega \mathrm{d}t - \underbrace{\int\limits_{T_{sol}}\int\limits_{\Omega} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_\ell}\frac{\partial}{\partial x_k}\left(\frac{\delta x_\ell}{\delta b_i}\right) \mathrm{d}\Omega \mathrm{d}t}_{\text{SD}} \tag{4.1.4}$$

The last integral on the r.h.s. contributes to the SD expression; the first integral can be expanded as

$$\int_{T_{sol}}\int_{\Omega}\Psi_n \frac{\partial}{\partial x_k}\left(\frac{\delta f^{inv}_{nk}}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t = \underbrace{\int_{T_{sol}}\int_{\partial\Omega}\Psi_n \frac{\delta f^{inv}_{nk}}{\delta b_i} n_k \mathrm{d}S\mathrm{d}t}_{\text{S.I\_2}} - \underbrace{\int_{T_{sol}}\int_{\Omega}\frac{\partial\Psi_n}{\partial x_k} A_{nmk}\frac{\delta U_m}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \tag{4.1.5}$$

The viscous term in Eq. 4.1.1 is expanded as

$$\mathcal{I}_{\text{MF\_vis}} = \underbrace{-\int_{T_s}\int_{\partial\Omega}\Psi_n \frac{\delta f^{vis}_{nk}}{\delta b_i} n_k \mathrm{d}S\mathrm{d}t}_{\text{S.I\_3}} + \underbrace{\int_{T_s}\int_{\Omega}\frac{\partial\Psi_n}{\partial x_k}\frac{\delta f^{vis}_{nk}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{MF\_vis1}}} + \underbrace{\int_{T_s}\int_{\Omega}\Psi_n \frac{\partial f^{vis}_{nk}}{\partial x_e}\frac{\partial}{\partial x_k}\left(\frac{\delta x_e}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}} \tag{4.1.6}$$

In order to expand $\mathcal{I}_{\text{MF\_vis1}}$, two new variables are defined namely, the adjoint stress tensor τ^{adj}_{mk} and heat flux q^{adj}_k, given by

$$\begin{aligned}
\tau^{adj}_{mk} &= (\mu+\mu_t)\left[\frac{\partial\Psi_{m+1}}{\partial x_k} + \frac{\partial\Psi_{k+1}}{\partial x_m} + \frac{\partial\Psi_5}{\partial x_m}v^A_k + \frac{\partial\Psi_5}{\partial x_k}v^A_m - \frac{2}{3}\delta_{mk}\left(\frac{\partial\Psi_{l+1}}{\partial x_l} + \frac{\partial\Psi_5}{\partial x_l}v^A_l\right)\right]\\
q^{adj}_k &= C_p\left(\frac{\mu}{\text{Pr}} + \frac{\mu_t}{\text{Pr}_\text{t}}\right)\frac{\partial\Psi_5}{\partial x_k}
\end{aligned} \tag{4.1.7}$$

Since the turbulence model is also differentiated, variations of some in the functions related to the Spalart-Allmaras model in Eq. 2.1.9, are required too. A new operator $\mathcal{P}(a,c)$ is defined, similar to the one used in [132], denoting the partial derivative of function a w.r.t. quantity c is introduced. The differentiation of each term in Eq. 2.1.9 can be found in [132]. Therefore, $\mathcal{I}_{\text{MF\_vis1}}$ is expanded as follows

$$\begin{aligned}
\mathcal{I}_{\text{MF\_vis1}} = & \underbrace{-\int_{T_s}\int_{\Omega}\left(\frac{\partial\tau^{adj}_{mk}}{\partial x_k} - \frac{\partial\Psi_5}{\partial x_k}\tau_{km}\right)\frac{\partial v^A_m}{\partial U_l}\frac{\delta U_l}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}}\\
& \underbrace{-\int_{T_s}\int_{\Omega}\frac{\partial q^{adj}_k}{\partial x_k}\frac{\partial T}{\partial U_l}\frac{\delta U_l}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \underbrace{-\int_{T_s}\int_{\Omega}\left(\tau^{adj}_{mk}\frac{\partial v^A_m}{\partial x_l} + q^{adj}_k\frac{\partial T}{\partial x_l}\right)\frac{\partial}{\partial x_k}\left(\frac{\delta x_l}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}}\\
& \underbrace{+\int_{T_s}\int_{\partial\Omega}\tau^{adj}_{mk}n_k\frac{\delta v^A_m}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{S.I\_4}} \underbrace{+\int_{T_s}\int_{\partial\Omega}q^{adj}_k n_k\frac{\delta T}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{S.I\_5}}
\end{aligned} \tag{4.1.8}$$

$$\underbrace{+\underset{T_s}{\int}\underset{\Omega}{\int}\frac{\partial \Psi_{m+1}}{\partial x_k}\frac{\tau_{km}}{\mu+\mu_t}\left[\mathcal{P}\left(\mu_t,\tilde{\mu}\right)+\mathcal{P}\left(\mu_t,f_{v_1}\right)\mathcal{P}\left(f_{v_1},\chi\right)\mathcal{P}\left(\chi,\tilde{\mu}\right)\right]\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}}$$
$$\underbrace{+\underset{T_s}{\int}\underset{\Omega}{\int}\frac{\partial \Psi_{5}}{\partial x_k}v_m^A\frac{\tau_{km}}{\mu+\mu_t}\left[\mathcal{P}\left(\mu_t,\tilde{\mu}\right)+\mathcal{P}\left(\mu_t,f_{v_1}\right)\mathcal{P}\left(f_{v_1},\chi\right)\mathcal{P}\left(\chi,\tilde{\mu}\right)\right]\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}}$$
$$\underbrace{+\underset{T_s}{\int}\underset{\Omega}{\int}\frac{\mathcal{C}_p}{\mathrm{Pr_t}}\frac{\partial \Psi_{5}}{\partial x_k}\frac{\partial T}{\partial x_k}\left[\mathcal{P}\left(\mu_t,\tilde{\mu}\right)+\mathcal{P}\left(\mu_t,f_{v_1}\right)\mathcal{P}\left(f_{v_1},\chi\right)\mathcal{P}\left(\chi,\tilde{\mu}\right)\right]\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}} \tag{4.1.9}$$

The last term on the r.h.s. of Eq. 4.1.1 leads to

$$\mathcal{I}_{\text{MF\_src}}=\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\varepsilon_{m\ell k}\Psi_{m+1}\rho\omega_\ell\frac{\partial v_k^A}{\partial U_q}\frac{\delta U_q}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t+\underset{T_s}{\int}\underset{\Omega}{\int}\varepsilon_{m\ell k}\Psi_{m+1}\omega_\ell v_k^A\frac{\partial \rho}{\partial U_q}\frac{\delta U_q}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \tag{4.1.10}$$

4.2 Differentiation of the Spalart-Allmaras Equation

Term $\mathcal{I}_{\text{SA}}$ into Eq. 4.9 is split in four terms arising from the differentiation of the temporal, convection, diffusion and source terms of the SA PDE,

$$\mathcal{I}_{\text{SA}}=\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\tilde{\nu}_a\frac{\delta \text{SA}^t}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{SA\_temp}}}+\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\tilde{\nu}_a\frac{\delta \text{SA}^c}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{SA\_conv}}}+\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\tilde{\nu}_a\frac{\delta \text{SA}^d}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{SA\_diff}}}+\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\tilde{\nu}_a\frac{\delta \text{SA}^s}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{SA\_src}}}$$

Temporal term $\mathcal{I}_{\text{SA\_temp}}$ is developed as

$$\mathcal{I}_{\text{SA\_temp}}=\underset{T_s}{\int}\underset{\Omega}{\int}\frac{\partial}{\partial t}\left(\tilde{\nu}_a\frac{\delta\tilde{\mu}}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t-\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\frac{\delta\tilde{\mu}}{\delta b_i}\frac{\partial\tilde{\nu}_a}{\partial t}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}}-\underbrace{\underset{T_s}{\int}\underset{\Omega}{\int}\tilde{\nu}_a\frac{\partial\tilde{\mu}}{\partial x_k}\frac{\partial}{\partial t}\left(\frac{\delta x_k}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}} \tag{4.2.1}$$

The first term of the r.h.s. of Eq. 4.2.1 is expanded as

$$\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial}{\partial t}\left(\tilde{\nu}_a\frac{\delta\tilde{\mu}}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t=\int\limits_{\Omega}\tilde{\nu}_a\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\Bigg|_0^{T_s}\underbrace{-\int\limits_{T_s}\int\limits_{\partial\Omega}\tilde{\nu}_a\frac{\delta\tilde{\mu}}{\delta b_i}\frac{\delta x_m}{\delta t}n_m\mathrm{d}S\mathrm{d}t}_{\text{S.I\_6}} \tag{4.2.2}$$

Similar to Eq. 4.1.3 , a zero initial condition for the adjoint field at the end of the primal solution time eliminates the first term on the r.h.s. of Eq. 4.1.2 at T_s. This term is also eliminated at $t = 0$ thanks to the fixed initial flow field. The second term on the r.h.s. involves variations in the flow variables w.r.t. b_i and, since is a surface integral, contributes to the boundary conditions of the adjoint SA equation.

Convection term ($\mathcal{I}_{\text{SA\_conv}}$) can be developed as

$$\begin{aligned}\mathcal{I}_{\text{SA\_conv}}=&\underbrace{-\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial\tilde{\nu}_a}{\partial x_k}\tilde{\mu}\frac{\partial v_k^A}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}}\underbrace{-\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial\tilde{\nu}_a}{\partial x_k}v_k^R\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}}\underbrace{+\int\limits_{T_s}\int\limits_{\partial\Omega}\tilde{\nu}_a\mathsf{n}_k\frac{\delta}{\delta b_i}\left(\tilde{\mu}v_k^R\right)\mathrm{d}S\mathrm{d}t}_{\text{S.I\_7}}\\&\underbrace{-\int\limits_{T_s}\int\limits_{\Omega}\tilde{\nu}_a\frac{\partial\left(\tilde{\mu}v_k^R\right)}{\partial x_\ell}\frac{\partial}{\partial x_k}\left(\frac{\delta x_\ell}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t+\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial\tilde{\nu}_a}{\partial x_k}\tilde{\mu}\frac{\partial v_k^F}{\partial x_\ell}\frac{\delta x_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}}\end{aligned} \tag{4.2.3}$$

The diffusion term is treated similarly

$$\begin{aligned}\mathcal{I}_{\text{SA\_diff}}=&\underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\tilde{\nu}_a\left\{\frac{\partial}{\partial x_k}\left[\left(\nu+\tilde{\nu}\right)\frac{\partial\tilde{\nu}}{\partial x_k}\right]+c_{b_2}\frac{\partial\tilde{\nu}}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\right\}\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}}\\&\underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\frac{\delta}{\delta b_i}\left(\frac{\partial}{\partial x_k}\left\{\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\tilde{\nu}}{\partial x_k}\right\}\right)\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{SA\_diff1}}}\underbrace{+\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\frac{\delta}{\delta b_i}\left(\tilde{\nu}\frac{\partial^2\tilde{\nu}}{\partial x_k^2}\right)\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\text{SA\_diff2}}}\end{aligned} \tag{4.2.4}$$

where

$$
\begin{aligned}
\mathcal{I}_{\text{SA\_diff1}} = {} & \underbrace{\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\mathcal{P}\left(\nu,\rho\right)\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \\
& \underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{\tilde{\nu}}{\rho}\left(\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\left(1+c_{b_2}\right)-\frac{\partial}{\partial x_k}\left\{\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\right\}\right)\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \\
& \underbrace{+\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{1}{\rho}\left(\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\left(1+c_{b_2}\right)-\frac{\partial}{\partial x_k}\left\{\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\right\}\right)\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}} \\
& \underbrace{+\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\frac{\partial}{\partial x_\ell}\left\{\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\tilde{\nu}}{\partial x_k}\right\}\frac{\partial}{\partial x_k}\left(\frac{\delta x_\ell}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}} \\
& \underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\tilde{\nu}}{\partial x_\ell}\frac{\partial}{\partial x_k}\left(\frac{\delta x_\ell}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}} \\
& \underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\partial\Omega}\rho\tilde{\nu}_a\mathsf{n}_k\frac{\delta}{\delta b_i}\left\{\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\tilde{\nu}}{\partial x_k}\right\}\mathrm{d}S\mathrm{d}t}_{\text{S.I\_8}} \\
& \underbrace{+\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{\partial\Omega}\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\mathsf{n}_k\frac{\delta\tilde{\nu}}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{S.I\_9}}
\end{aligned}
\tag{4.2.5}
$$

$$
\begin{aligned}
\mathcal{I}_{\text{SA\_diff2}} = {} & \underbrace{\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{1}{\rho}\left\{\rho\tilde{\nu}_a\frac{\partial^2\tilde{\nu}}{\partial x_k^2}+\frac{\partial}{\partial x_k}\left[\frac{\partial\left(\tilde{\nu}_a\tilde{\mu}\right)}{\partial x_k}\right]\right\}\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}} \\
& \underbrace{-\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{\tilde{\nu}}{\rho}\left\{\rho\tilde{\nu}_a\frac{\partial^2\tilde{\nu}}{\partial x_k^2}+\frac{\partial}{\partial x_k}\left[\frac{\partial\left(\tilde{\nu}_a\tilde{\mu}\right)}{\partial x_k}\right]\right\}\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}}
\end{aligned}
$$

$$\underbrace{-\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\tilde{\nu}\frac{\partial}{\partial x_\ell}\left(\frac{\partial\tilde{\nu}}{\partial x_k}\right)\frac{\partial}{\partial x_k}\left(\frac{\delta x_\ell}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t+\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial\left(\rho\tilde{\nu}_a\tilde{\nu}\right)}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_\ell}\frac{\partial}{\partial x_k}\left(\frac{\delta x_\ell}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\mathrm{SD}}$$

$$\underbrace{+\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\partial\Omega}\rho\tilde{\nu}_a\tilde{\nu}\mathsf{n}_k\frac{\delta}{\delta b_i}\left(\frac{\partial\tilde{\nu}}{\partial x_k}\right)\mathrm{d}S\mathrm{d}t}_{\mathrm{S.I\_10}}\underbrace{-\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{\partial\Omega}\frac{\partial\left(\rho\tilde{\nu}_a\tilde{\nu}\right)}{\partial x_k}\mathsf{n}_k\frac{\delta\tilde{\nu}}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\mathrm{S.I\_11}} \qquad (4.2.6)$$

For the differentiation of the source terms, the following definitions are used

$$\begin{aligned}
&\mathcal{C}_1=\mathcal{P}\left(\tilde{S},f_{v_2}\right)\left[\mathcal{P}\left(f_{v_2},\chi\right)+\mathcal{P}\left(f_{v_2},f_{v_1}\right)\mathcal{P}\left(f_{v_1},\chi\right)\right]\\
&\mathcal{C}_2=\begin{cases}0 & \text{, for } r>10\\ \mathcal{P}\left(f_w,g\right)\mathcal{P}\left(g,r\right)\left(\frac{\tilde{\nu}}{\Delta}\right)^2 & \text{, else}\end{cases}\\
&\mathcal{C}_3=\mathcal{C}_2\mathcal{P}\left(r,\tilde{S}\right),\ \ \mathcal{C}_4=\mathcal{C}_2\mathcal{P}\left(r,\tilde{\nu}\right),\ \ \mathcal{C}_5=\mathcal{C}_2\mathcal{P}\left(r,\Delta\right)\\
&\mathcal{C}_6=\left[-c_{b_1}\left(1-f_{t_2}\right)\tilde{\nu}+\tfrac{c_{w1}}{\mathrm{Re}_0}\mathcal{C}_3\right]\\
&\mathcal{C}_7=\left\{c_{b_1}\left[\tilde{S}\tilde{\nu}-\tfrac{1}{\mathrm{Re}_0\,\kappa^2}\left(\tfrac{\tilde{\nu}}{\Delta}\right)^2\right]\right\}\mathcal{P}\left(f_{t_2},\chi\right)
\end{aligned}\qquad (4.2.7)$$

Then, the differentiation of the source terms leads to

$$\mathcal{I}_{\mathrm{SA\_src}}=\underbrace{\int\limits_{T_s}\int\limits_{\Omega}\tilde{\nu}_a\left[-\left(c_{w_1}f_w-\frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\tilde{\nu}}{\Delta}\right)^2\right]\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathrm{FAE}}$$

$$\underbrace{+\int\limits_{T_s}\int\limits_{\Omega}\tilde{\nu}_a\left[-c_{b_1}\left(1-f_{t_2}\right)\tilde{S}+2\left(c_{w_1}f_w-\frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\tilde{\nu}}{\Delta^2}\right)\right]\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathrm{FAE\_SA}}$$

$$\underbrace{-2\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\left(c_{w_1}f_w-\frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\frac{\tilde{\nu}^2}{\Delta^3}\frac{\delta\Delta}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t+c_{w_1}\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\mathcal{C}_5\frac{\delta\Delta}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathrm{FAE\_DISTANCE}}$$

$$\underbrace{+\int\limits_{T_s}\int\limits_{\Omega}\left(c_{w_1}\tilde{\nu}_a\mathcal{C}_4+\tilde{\nu}_a\mathcal{C}_7\mathcal{P}\left(\chi,\tilde{\nu}\right)\right)\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathrm{FAE\_SA}}\underbrace{+\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\mathcal{C}_6\frac{\delta\tilde{S}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathcal{I}_{\tilde{S}}} \qquad (4.2.8)$$

$$\underbrace{-\int\limits_{T_s}\int\limits_{\Omega}\left\{\tilde{\nu}(c_{w_1}\tilde{\nu}_a\mathcal{C}_4+\tilde{\nu}_a\mathcal{C}_7\mathcal{P}\left(\chi,\tilde{\nu}\right))-\rho\tilde{\nu}_a\mathcal{C}_7\mathcal{P}\left(\chi,\nu\right)\mathcal{P}\left(\nu,\rho\right)\right\}\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\mathrm{FAE}}$$

where

$$
\begin{aligned}
\mathcal{I}_{\tilde{S}} = & \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \rho\tilde{\nu}_a\mathcal{C}_6\mathcal{P}\left(\tilde{S},S\right)\frac{\delta S}{\delta b_i}\mathrm{d}\Omega}_{\mathcal{I}_{\text{VORTICITY}}} + \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \tilde{\nu}_a\mathcal{C}_6\left[\mathcal{P}\left(\tilde{S},\tilde{\nu}\right)+\mathcal{C}_1\mathcal{P}\left(\chi,\tilde{\nu}\right)\right]\frac{\delta\tilde{\mu}}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_SA}} \\
& \underbrace{+\int\limits_{T_s}\int\limits_{\Omega} \rho\tilde{\nu}_a\mathcal{C}_6\mathcal{P}\left(\tilde{S},\Delta\right)\frac{\delta\Delta}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE\_DISTANCE}} + \underbrace{\int\limits_{T_s}\int\limits_{\Omega} \rho\tilde{\nu}_a\mathcal{C}_6\mathcal{C}_1\mathcal{P}\left(\chi,\nu\right)\mathcal{P}\left(\nu,\rho\right)\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \\
& \underbrace{-\int\limits_{T_s}\int\limits_{\Omega} \tilde{\nu}\tilde{\nu}_a\mathcal{C}_6\left[\mathcal{P}\left(\tilde{S},\tilde{\nu}\right)+\mathcal{C}_1\mathcal{P}\left(\chi,\tilde{\nu}\right)\right]\frac{\partial\rho}{\partial U_\ell}\frac{\delta U_\ell}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}}
\end{aligned}
\tag{4.2.9}
$$

with

$$
\begin{aligned}
\mathcal{I}_{\text{VORTICITY}} = & \underbrace{-\int\limits_{T_s}\int\limits_{\Omega}\frac{\partial}{\partial x_\ell}\left[\rho\tilde{\nu}_a\mathcal{C}_6\mathcal{P}\left(\tilde{S},S\right)\frac{1}{S}\varepsilon_{k\ell m}\varepsilon_{kqr}\frac{\partial v_r^A}{\partial x_q}\right]\frac{\partial v_m^A}{\partial U_p}\frac{\delta U_p}{\delta b_i}\mathrm{d}\Omega\mathrm{d}t}_{\text{FAE}} \\
& \underbrace{-\int\limits_{T_s}\int\limits_{\Omega}\rho\tilde{\nu}_a\mathcal{C}_6\mathcal{P}\left(\tilde{S},S\right)\frac{1}{S}\varepsilon_{k\ell m}\varepsilon_{kqr}\frac{\partial v_r^A}{\partial x_q}\frac{\partial v_m^A}{\partial x_p}\frac{\partial}{\partial x_\ell}\left(\frac{\delta x_p}{\delta b_i}\right)\mathrm{d}\Omega\mathrm{d}t}_{\text{SD}} \\
& \underbrace{+\int\limits_{T_s}\int\limits_{\partial\Omega}\rho\tilde{\nu}_a\mathcal{C}_6\mathcal{P}\left(\tilde{S},S\right)\frac{1}{S}\varepsilon_{k\ell m}\varepsilon_{kqr}\frac{\partial v_r^A}{\partial x_q}\mathsf{n}_\ell\frac{\delta v_m^A}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{S.I\_12}}
\end{aligned}
\tag{4.2.10}
$$

4.3 Differentiation of the Aeroacoustic Objective Function

To complete the differentiation of Eq. 4.9, the only remaining task is to differentiate the aeroacoustic objective function, Eq. 4.4 . Doing so, $\frac{\delta J}{\delta b_i}$ reads

$$
\frac{\delta J}{\delta b_i} = \int\limits_{\omega}\frac{1}{|\hat{p}'|}\left(\hat{p}'_{\text{Re}}\frac{\delta\hat{p}'_{\text{Re}}}{\delta b_i}+\hat{p}'_{\text{Im}}\frac{\delta\hat{p}'_{\text{Im}}}{\delta b_i}\right)\mathrm{d}\omega
\tag{4.3.1}
$$

For the sake of simplicity, hereafter, $\hat{p}'(\vec{x}_r,\omega)$, $\hat{G}(\vec{x}_r,\vec{x}_s,\omega)$, $\hat{F}_k(\vec{x}_s,\omega)$ and $\hat{Q}(\vec{x}_s,\omega)$ are shorted to $\hat{p}'$, $\hat{G}$, $\hat{F}_k$ and $\hat{Q}$, respectively. The real and imaginary part of $\hat{p}'$ can be found based on Eq. 3.1.1.7 . Since the grid does not change at the FW-H surface location during the optimization, the derivatives of the Green function and its spatial derivatives as well as those of the surface element $\mathrm{d}S$, w.r.t. b_i are zero. So, variations in the real and imaginary part of $\hat{p}'$ w.r.t. b_i read

$$\begin{aligned}
\frac{\delta \hat{p}'_{\text{Re}}}{\delta b_i} =& -\int\limits_{f=0}\left[\left(\frac{\delta \hat{F}_k}{\delta b_i}\right)_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}} - \left(\frac{\delta \hat{F}_k}{\delta b_i}\right)_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}}\right]\mathrm{d}S \\
&+\int\limits_{f=0}\omega\left[\left(\frac{\delta \hat{Q}}{\delta b_i}\right)_{\text{Re}}\hat{G}_{\text{Im}} + \left(\frac{\delta \hat{Q}}{\delta b_i}\right)_{\text{Im}}\hat{G}_{\text{Re}}\right]\mathrm{d}S \\
\frac{\delta \hat{p}'_{\text{Im}}}{\delta b_i} =& -\int\limits_{f=0}\left[\left(\frac{\delta \hat{F}_k}{\delta b_i}\right)_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}} + \left(\frac{\delta \hat{F}_k}{\delta b_i}\right)_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}}\right]\mathrm{d}S \\
&-\int\limits_{f=0}\omega\left[\left(\frac{\delta \hat{Q}}{\delta b_i}\right)_{\text{Re}}\hat{G}_{\text{Re}} - \left(\frac{\delta \hat{Q}}{\delta b_i}\right)_{\text{Im}}\hat{G}_{\text{Im}}\right]\mathrm{d}S
\end{aligned} \tag{4.3.2}$$

Introducing Eq. 4.3.2 to 4.3.1 results to

$$\begin{aligned}
\frac{\delta J}{\delta b_i} =& -\int\limits_{\omega}\int\limits_{f=0}\left[\frac{1}{|\hat{p}'|}\left\{\hat{p}'_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}} + \hat{p}'_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}}\right\}\left(\frac{\delta \hat{F}_k}{\delta b_i}\right)_{\text{Re}}\right]\mathrm{d}S\mathrm{d}\omega \\
&-\int\limits_{\omega}\int\limits_{f=0}\left[\frac{1}{|\hat{p}'|}\left\{\hat{p}'_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}} - \hat{p}'_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}}\right\}\left(\frac{\delta \hat{F}_k}{\delta b_i}\right)_{\text{Im}}\right]\mathrm{d}S\mathrm{d}\omega \\
&-\int\limits_{\omega}\int\limits_{f=0}\omega\frac{1}{|\hat{p}'|}\left[\left(\hat{p}'_{\text{Im}}\hat{G}_{\text{Re}} - \hat{p}'_{\text{Re}}\hat{G}_{\text{Im}}\right)\left(\frac{\delta \hat{Q}}{\delta b_i}\right)_{\text{Re}} + \left(\hat{p}'_{\text{Re}}\hat{G}_{\text{Re}} - \hat{p}'_{\text{Im}}\hat{G}_{\text{Im}}\right)\left(\frac{\delta \hat{Q}}{\delta b_i}\right)_{\text{Im}}\right]\mathrm{d}S\mathrm{d}\omega
\end{aligned} \tag{4.3.3}$$

In Eq. 4.3.3 , $\frac{\delta \hat{F}_k}{\delta b_i}$ and $\frac{\delta \hat{Q}}{\delta b_i}$ include derivatives of the flow variables w.r.t. b_i in the frequency domain. These should be expressed in the time domain to contribute to the adjoint flow equations. To do so, the Fourier transformation needs to be included in Eq. 4.3.3 , subtracting the time-averaged value of F_k and Q from their instantaneous values, along with a multiplication with a Hanning window $\mathcal{H}(t)$ before transforming them into the frequency domain. The Fourier transformation for an arbitrary signal $s(t)$ is performed as follows

$$\hat{s}(\omega) = \frac{1}{T_O}\int\limits_{T_O}\mathcal{H}(t)\Big[s(t) - \frac{1}{T_O}\int\limits_{T_O}s(t)dt\Big]e^{-2i\pi\omega t}dt \tag{4.3.4}$$

where T_O is the optimization time window during which F_k and Q are computed and stored. In practice, T_O is a part of T_s. For example, in periodic flows, the transient phase is included in T_s though excluded from T_O.

Including Eq. 4.3.4 into Eq. 4.3.3 and permuting time and frequency integrals, $\frac{\delta J}{\delta b_i}$ reads:

$$\frac{\delta J}{\delta b_i} = -\int\limits_{T_O}\int\limits_{f=0}\left[A_k\frac{\delta F_k}{\delta b_i} + B\frac{\delta Q}{\delta b_i}\right]dSdt \tag{4.3.5}$$

with

$$\begin{aligned}
A_k = \frac{1}{T_O}\Bigg\{&\int\limits_{\omega}\left[\frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|}\left(\frac{\delta\hat{G}}{\delta x_{s_k}}\right)_{\mathrm{Re}} + \frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|}\left(\frac{\delta\hat{G}}{\delta x_{s_k}}\right)_{\mathrm{Im}}\right][\mathcal{H}(t)\cos(2\pi\omega t) - H_c(\omega)]\,d\omega \\
&+\int\limits_{\omega}\left[\frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|}\left(\frac{\delta\hat{G}}{\delta x_{s_k}}\right)_{\mathrm{Im}} - \frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|}\left(\frac{\delta\hat{G}}{\delta x_{s_k}}\right)_{\mathrm{Re}}\right][\mathcal{H}(t)\sin(2\pi\omega t) - H_s(\omega)]\,d\omega\Bigg\} \\
B = \frac{1}{T_O}\Bigg\{&\int\limits_{\omega}\omega\left(\frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|}\hat{G}_{\mathrm{Re}} - \frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|}\hat{G}_{\mathrm{Im}}\right)[\mathcal{H}(t)\cos(2\pi\omega t) - H_c(\omega)]\,d\omega \\
&+\int\limits_{\omega}\omega\left(\frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|}\hat{G}_{\mathrm{Im}} - \frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|}\hat{G}_{\mathrm{Re}}\right)[\mathcal{H}(t)\sin(2\pi\omega t) - H_s(\omega)]\,d\omega\Bigg\} \\
H_c(\omega) = &\frac{1}{T_O}\int\limits_{T_O}\mathcal{H}(t)\cos(2\pi\omega t)dt \\
H_s(\omega) = &\frac{1}{T_O}\int\limits_{T_O}\mathcal{H}(t)\sin(2\pi\omega t)dt
\end{aligned} \tag{4.3.6}$$

Equation 4.3.5 contains a double time/surface integral over the FW-H surface. In order to eliminate the expensive derivatives of the flow variables w.r.t. b_i, when solving the adjoint equations, these terms contribute as sources in the cells lying along the FW-H surface and their neighbors. Derivatives of F_k and Q w.r.t. b_i read

$$\begin{aligned}
\frac{\delta F_k}{\delta b_i} &= \underbrace{\left[\frac{\delta_{kj}(\gamma-1)|\vec{v}|^2}{2} - v_j v_k\right] n_j^{\text{FWH}}}_{\mathcal{C}_k^{\text{FWH1}}} \frac{\delta U_1}{\delta b_i} + \delta_{kj}(\gamma-1)\frac{\delta U_5}{\delta b_i} n_j^{\text{FWH}} \\
&\quad + \underbrace{\left[\left(v_k - 2v_{\infty k}\right) n_m^{\text{FWH}} + v_j n_j^{\text{FWH}} \delta_{km} - \delta_{kj}(\gamma-1) n_j^{\text{FWH}} v_m^A\right]}_{\mathcal{C}_{km}^{\text{FWH2}}} \frac{\delta U_{m+1}}{\delta b_i} - \frac{\delta \tau_{kj}}{\delta b_i} n_j^{\text{FWH}} \\
\frac{\delta Q}{\delta b_i} &= \frac{\delta U_{k+1}}{\delta b_i} n_k^{\text{FWH}}
\end{aligned} \tag{4.3.7}$$

where

$$\begin{aligned}
\frac{\delta \tau_{kj}}{\delta b_i} n_j^{\text{FWH}} = & \underbrace{-\left(\mu+\mu_t\right)\left[\frac{\partial\left(\frac{v_k^A}{\rho}\right)}{\partial x_j} + \frac{\partial\left(\frac{v_j^A}{\rho}\right)}{\partial x_k} - \frac{2}{3}\delta_{kj}\frac{\partial\left(\frac{v_m^A}{\rho}\right)}{\partial x_m}\right] n_j^{\text{FWH}}}_{\mathcal{E}_k^{\text{FWH1}}} \frac{\delta U_1}{\delta b_i} \\
& \underbrace{+\left(\mu+\mu_t\right)\left[\frac{\partial\left(\frac{1}{\rho}\right)}{\partial x_i} n_i^{\text{FWH}} \delta_{km} + \frac{\partial\left(\frac{1}{\rho}\right)}{\partial x_k} n_m^{\text{FWH}} - \frac{2}{3}\frac{\partial\left(\frac{1}{\rho}\right)}{\partial x_m} n_k^{\text{FWH}}\right]}_{\mathcal{E}_{km}^{\text{FWH2}}} \frac{\delta U_{m+1}}{\delta b_i} \\
& \underbrace{-\frac{\left(\mu+\mu_t\right)}{\rho}\left[v_k^A n_m^{\text{FWH}} + v_j^A n_j^{\text{FWH}} \delta_{mk} - \frac{2}{3} n_k^{\text{FWH}} v_m^A\right] \frac{\partial}{\partial x_m}\left(\frac{\delta U_1}{\delta b_i}\right)}_{\mathcal{I}_{\text{FWH\_vis1}}} \\
& \underbrace{+\frac{\left(\mu+\mu_t\right)}{\rho}\left[n_j^{\text{FWH}} \delta_{km} + n_m^{\text{FWH}} \delta_{kj} - \frac{2}{3} n_k^{\text{FWH}} \delta_{mj}\right] \frac{\partial}{\partial x_j}\left(\frac{\delta U_{m+1}}{\delta b_i}\right)}_{\mathcal{I}_{\text{FWH\_vis2}}} \\
& \underbrace{+\frac{\tau_{kj}}{\mu+\mu_t} n_j^{\text{FWH}} \left[\mathcal{P}\left(\mu_t,\tilde{\mu}\right) + \mathcal{P}\left(\mu_t, f_{v_1}\right)\mathcal{P}\left(f_{v_1},\chi\right)\mathcal{P}\left(\chi,\tilde{\mu}\right)\right]}_{\mathcal{E}_k^{\text{FWH3}}} \frac{\delta\tilde{\mu}}{\delta b_i}
\end{aligned} \tag{4.3.8}$$

The last two terms in Eq. 4.3.8 include the spatial gradient of $\frac{\delta \vec{U}}{\delta b_i}$ requiring the application of the Green-Gauss theorem to the finite volumes formed around any node P on the FW-H surface. For such a node, the spatial gradient of any variable

Φ can be obtained by

$$\frac{\partial\Phi}{\partial x_j} \simeq \frac{1}{\Omega_P}\int\limits_{\Omega_P}\frac{\partial\Phi}{\partial x_j}\mathrm{d}\Omega_P = \frac{1}{\Omega_P}\int\limits_{\partial\Omega_P}\Phi\hat{n}_j\mathrm{d}(S) \simeq \frac{1}{\Omega_P}\Phi^{PQ_l}n_j^{PQ_l} \tag{4.3.9}$$

where Ω_P is the area/volume of the finite volume formed around P, $l = 1, ..., W$ with W being the number of the neighbors Q_l of P and $n_j^{PQ_l}$ is the jth component of the normal vector with magnitude equal to the length of the edge shared by the finite volumes of P and Q_l. Φ^{PQ_l} is approximated by $\Phi^{PQ_l} = \frac{\Phi^P+\Phi^{Q_l}}{2}$. Using Eq. 4.3.9 , terms $\mathcal{I}_{\text{FWH\_vis1}}$ and $\mathcal{I}_{\text{FWH\_vis2}}$ become

$$\begin{aligned}
\mathcal{I}_{\text{FWH\_vis1}} &= -\frac{(\mu+\mu_t)}{2\rho\Omega_P}\left[v_k^A n_m^{\text{FWH}} + v_j^A n_j^{\text{FWH}}\delta_{mk} - \frac{2}{3}n_k^{\text{FWH}}v_m^A\right]\left(\frac{\delta U_1}{\delta b_i}\right)^{Q_l} n_m^{PQ_l} \\
\mathcal{I}_{\text{FWH\_vis2}} &= +\frac{(\mu+\mu_t)}{2\rho\Omega_P}\left[n_j^{\text{FWH}}\delta_{km} + n_m^{\text{FWH}}\delta_{kj} - \frac{2}{3}n_k^{\text{FWH}}\delta_{mj}\right]\left(\frac{\delta U_{m+1}}{\delta b_i}\right)^{Q_l} n_j^{PQ_l}
\end{aligned} \tag{4.3.10}$$

Finally, the variation in the objective function can be written as

$$\frac{\delta J}{\delta b_i} = \underbrace{-\int\limits_{T_O}\int\limits_{f=0}(S_l^{\text{FWH}} + S_l^{\text{Q\_FWH}})\frac{\delta U_l}{\delta b_i}dSdt}_{\text{FAE}}\underbrace{-\int\limits_{T_O}\int\limits_{f=0}S^{\text{SA\_FWH}}\frac{\delta\tilde{\mu}}{\delta b_i}dSdt}_{\text{FAE\_SA}} \tag{4.3.11}$$

where S_l^{FWH}, $S^{\text{SA\_FWH}}$ and $S_l^{\text{Q\_FWH}}$ appear as three sets of source terms, with the first two valid on the finite volumes formed around the FW-H surface nodes and the latter valid at the neighbors of the FW-H nodes; these are given by

$$S^{\text{FWH}} = \begin{bmatrix} (\mathcal{C}_k^{\text{FWH1}} - \mathcal{E}_k^{\text{FWH1}})A_k \\ (\mathcal{C}_{k1}^{\text{FWH2}} - \mathcal{E}_{k1}^{\text{FWH2}})A_k + n_1^{\text{FWH}}B \\ (\mathcal{C}_{k2}^{\text{FWH2}} - \mathcal{E}_{k2}^{\text{FWH2}})A_k + n_2^{\text{FWH}}B \\ (\mathcal{C}_{k3}^{\text{FWH2}} - \mathcal{E}_{k3}^{\text{FWH2}})A_k + n_3^{\text{FWH}}B \\ (\gamma-1)n_k^{\text{FWH}}A_k \end{bmatrix} \tag{4.3.12}$$

$$S^{\text{Q\_FWH}} = \begin{bmatrix} \{\sum\limits_{\forall P\in\mathcal{N}(Q)} \frac{(\mu+\mu_t)}{2\rho\Omega_P}\left(-v_k^A n_m^{\text{FWH}} - v_j^A n_j^{\text{FWH}}\delta_{mk} + \frac{2}{3}n_k^{\text{FWH}}v_m^A\right) n_m^{PQ}\}A_k \\ \{\sum\limits_{\forall P\in\mathcal{N}(Q)} \frac{(\mu+\mu_t)}{2\rho\Omega_P}\left(n_j^{\text{FWH}}\delta_{k1} + n_1^{\text{FWH}}\delta_{kj} - \frac{2}{3}n_k^{\text{FWH}}\delta_{1j}\right) n_j^{PQ}\}A_k \\ \{\sum\limits_{\forall P\in\mathcal{N}(Q)} \frac{(\mu+\mu_t)}{2\rho\Omega_P}\left(n_j^{\text{FWH}}\delta_{k2} + n_2^{\text{FWH}}\delta_{kj} - \frac{2}{3}n_k^{\text{FWH}}\delta_{2j}\right) n_j^{PQ}\}A_k \\ \{\sum\limits_{\forall P\in\mathcal{N}(Q)} \frac{(\mu+\mu_t)}{2\rho V_P}\left(n_j^{\text{FWH}}\delta_{k3} + n_3^{\text{FWH}}\delta_{kj} - \frac{2}{3}n_k^{\text{FWH}}\delta_{3j}\right) n_j^{PQ}\}A_k \\ 0 \end{bmatrix} \tag{4.3.13}$$

$$S^{\text{SA\_FWH}} = \mathcal{E}_k^{\text{FWH3}} A_k \tag{4.3.14}$$

where $P \in \mathcal{N}(Q)$ is any FW-H node connected with Q by an edge. $S^{\text{Q\_FWH}}$ is zero at nodes that are not neighbor to any FW-H node; S^{FWH} and $S^{\text{SA\_FWH}}$ are non-zero only for the nodes along the FW-H surface, Fig. 4.1.

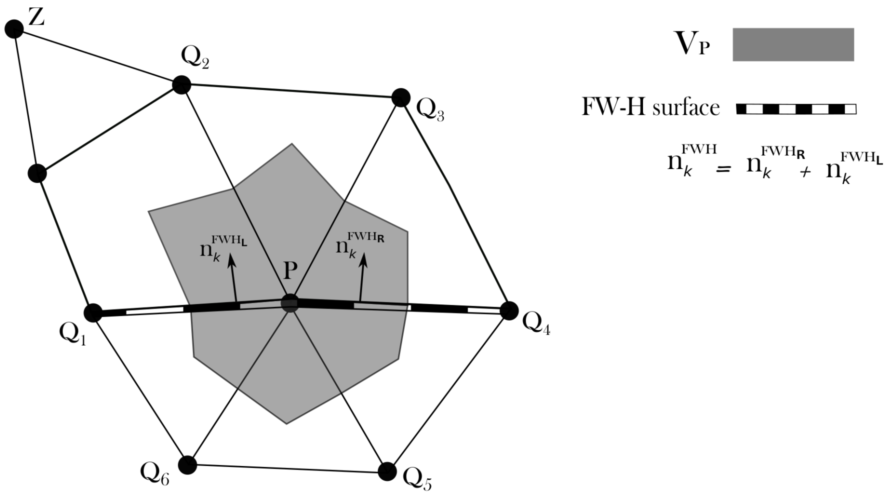


Figure 4.1: Schematic of the finite volume formed around node P, located on the permeable FW-H surface. n_k^{FWH} are pointing outwards the FW-H surface. S^{FWH} and $S^{\text{SA\_FWH}}$ are non-zero only at Q_1, P and Q_4, while $S^{\text{Q\_FWH}}$ is non-zero for all nodes except Z which is not neighbor to any FW-H node.

4.4 Field Adjoint Equations

So far, all terms in Eq. 4.9 are differentiated. The next step is to merge all terms denoted by FAE and setting them to zero, which gives rise to the mean flow field adjoint PDEs. Similarly, terms denoted as FAE_SA vanish, resulting to the adjoint SA PDE. The FAE_DISTANCE terms are either computed directly by the parameterization and mesh morphing technique (assuming that the distance of a particular node from the wall is always measured from the same closest to this node point on the wall) or included into the adjoint distance equation (by also differentiating the Eikonal equation) [102].

$$-\frac{\partial \Psi_m}{\partial t} - A_{nmk}\frac{\partial \Psi_n}{\partial x_k} - \mathcal{K}_m + S_m^{\text{adj}} + \mathcal{K}_m^{\text{SA}} - S_m^{\text{FWH}} - S_m^{\text{Q\_FWH}} = 0 \tag{4.4.1}$$

where terms $\mathcal{K}_m$, S_m^{adj} and $\mathcal{K}_m^{\text{SA}}$ read

$$\mathcal{K}_m = (\frac{\partial \tau_{kq}^{\text{adj}}}{\partial x_k} - \tau_{kq}\frac{\partial \Psi_5}{\partial x_k})\frac{\partial v_q^A}{\partial U_m} + \frac{\partial q_k^{\text{adj}}}{\partial x_k}\frac{\partial T}{\partial U_m} \tag{4.4.2}$$

$$S_m^{\text{adj}} = \rho \varepsilon_{n\ell k}\Psi_{n+1}\omega_\ell \frac{\partial v_k^A}{\partial U_m} + \varepsilon_{n\ell k}\Psi_{n+1}\omega_\ell v_k^A \frac{\partial \rho}{\partial U_m} \tag{4.4.3}$$

$$\mathcal{K}_m^{\text{SA}} = \mathcal{K}_m^{\text{SA},\rho}\frac{\partial \rho}{\partial U_m} + \mathcal{K}_m^{\text{SA},v_k^A}\frac{\partial v_k^A}{\partial U_m} \tag{4.4.4}$$

with

$$\begin{aligned}
\mathcal{K}_m^{\text{SA},\rho} =& -\frac{\tilde{\nu}_a}{\sigma}\left\{\frac{\partial}{\partial x_k}\left[(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_k}\right] + c_{b_2}\frac{\partial \tilde{\nu}}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k}\right\} + \frac{1}{\sigma}\frac{\partial(\rho\tilde{\nu}_a)}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k}\mathcal{P}(\nu,\rho) \\
& - \frac{1}{\sigma}\frac{\tilde{\nu}}{\rho}\left(\frac{\partial(\rho\tilde{\nu}_a)}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k}(1+c_{b_2}) - \frac{\partial}{\partial x_k}\left\{[\nu + (1+c_{b_2})\tilde{\nu}]\frac{\partial(\rho\tilde{\nu}_a)}{\partial x_k}\right\}\right) \\
& - \frac{c_{b_2}}{\sigma}\frac{\tilde{\nu}}{\rho}\left\{\rho\tilde{\nu}_a\frac{\partial^2 \tilde{\nu}}{\partial x_k^2} + \frac{\partial}{\partial x_k}\left[\frac{\partial(\tilde{\nu}_a\tilde{\mu})}{\partial x_k}\right]\right\} \\
& - \tilde{\nu}_a\left(c_{w_1}f_w - \frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\tilde{\nu}}{\Delta}\right)^2 \\
& - \left\{\tilde{\nu}(c_{w_1}\tilde{\nu}_a\mathcal{C}_4 + \tilde{\nu}_a\mathcal{C}_7\mathcal{P}(\chi,\tilde{\nu})) - \rho\tilde{\nu}_a\mathcal{C}_7\mathcal{P}(\chi,\nu)\mathcal{P}(\nu,\rho)\right\} \\
& + \rho\tilde{\nu}_a\mathcal{C}_6\mathcal{C}_1\mathcal{P}(\chi,\nu)\mathcal{P}(\nu,\rho) - \tilde{\nu}\tilde{\nu}_a\mathcal{C}_6\{\mathcal{P}\left(\tilde{S},\tilde{\nu}\right) + \mathcal{C}_1\mathcal{P}(\chi,\tilde{\nu})\} \\
\mathcal{K}_m^{\text{SA},v_k^A} =& -\rho\tilde{\nu}\frac{\partial \tilde{\nu}_a}{\partial x_k} - \frac{\partial}{\partial x_\ell}\left[\rho\tilde{\nu}_a\mathcal{C}_6\mathcal{P}\left(\tilde{S},S\right)\frac{1}{S}\varepsilon_{k\ell m}\varepsilon_{kqr}\frac{\partial v_r^A}{\partial x_q}\right]
\end{aligned}$$

Similarly, the field adjoint equation for the Spalart-Allmaras model reads

$$-\frac{\partial \tilde{\nu}_a}{\partial t} - \frac{\partial \tilde{\nu}_a}{\partial x_k}v_k^R + \mathcal{G}^{\text{SA,diff}} + \mathcal{G}^{\text{SA,src}} + \mathcal{G}^{\mu_t}\frac{\partial \mu_t}{\partial \tilde{\nu}} - S^{\text{SA\_FWH}} = 0 \tag{4.4.5}$$

where

$$\begin{aligned}
\mathcal{G}^{\mathrm{SA},diff} &= \frac{1}{\sigma\rho}\left(\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\left(1+c_{b_2}\right)-\frac{\partial}{\partial x_k}\left\{\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\right\}\right)\\
&+\frac{c_{b_2}}{\rho\sigma}\left\{\rho\tilde{\nu}_a\frac{\partial^2\tilde{\nu}}{\partial x_k^2}+\frac{\partial}{\partial x_k}\left[\frac{\partial\left(\tilde{\nu}_a\tilde{\mu}\right)}{\partial x_k}\right]\right\}\\
\mathcal{G}^{\mathrm{SA},src} &= \tilde{\nu}_a\left[-c_{b_1}\left(1-f_{t_2}\right)\tilde{S}+2\left(c_{w_1}f_w-\frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\tilde{\nu}}{\Delta^2}\right)\right]\\
&+\left\{c_{w_1}\tilde{\nu}_a\mathcal{C}_4+\tilde{\nu}_a\mathcal{C}_7\mathcal{P}\left(\chi,\tilde{\nu}\right)\right\}+\tilde{\nu}_a\mathcal{C}_6\{\mathcal{P}\left(\tilde{S},\tilde{\nu}\right)+\mathcal{C}_61\mathcal{P}\left(\chi,\tilde{\nu}\right)\}\\
\mathcal{G}^{\mu_t} &= \frac{\tau_{km}}{\mu+\mu_t}\left[\mathcal{P}\left(\mu_t,\tilde{\mu}\right)+\mathcal{P}\left(\mu_t,f_{v_1}\right)\mathcal{P}\left(f_{v_1},\chi\right)\mathcal{P}\left(\chi,\tilde{\mu}\right)\right](\frac{\partial\Psi_{m+1}}{\partial x_k}+\frac{\partial\Psi_5}{\partial x_k}v_m^A+\frac{\mathcal{C}_p}{\mathrm{Pr}_t}\frac{\partial\Psi_5}{\partial x_k})
\end{aligned}$$

The numerical solution of Eqs. 4.4.5 and 4.4.1 is similar to the procedure used for the flow equations. However, the time integration of these two equations has to be performed backward in time. This calls for the storage of the computed unsteady flow field, requiring considerable memory particularly for industrial cases. Herein, depending on the unsteady nature of the problem, two different storage and solution methods are used. For periodic cases with a known period (apriori) which is not changing during the optimization (such as a plunging or pitching airfoil), the flow field over a single period of the phenomena is stored and, the unsteady adjoint equations are repeatedly integrated backwards over the single period, until a periodic adjoint solution be established. On the other hand, if the period of the case changes during the optimization (such as a vortex shedding cylinder), the flow field over the whole solution time window is stored (including the transient phase) and unsteady adjoint equations are solved backwards during this time window.

4.5 Adjoint Boundary Conditions

So far, all volume integrals arising from the differentiation of J_{aug}, have been treated and the remaining terms are the surface integrals. These integrals are marked with S.I_1-S.I_12 in Eq. 4.1.3 to Eq. 4.2.10 . The surface integrals are split into two groups depending on whether they include variations in the geometric quantities or flow variables w.r.t. the design variables. The former contribute into sensitivity derivatives while the latter must be eliminated, giving rise to the adjoint boundary conditions. However, the flow boundary conditions must also be taken into account, meaning that variations in the imposed quantities using Dirichlet condition are zero. The remaining flow quantity variations are grouped together and their multipliers are set to zero.

When the steady equations are solved in a relative reference frame, the adjoint boundary conditions can be found in the PhD Thesis of K. Tsiakas [132]. Therefore, herein, the adjoint boundary conditions are given for an unsteady case without a moving frame i.e. $v^A = v^R$. Boundary conditions are given for wall, split to slip or no-slip, and far-field boundaries.

During the development, terms that contribute to the adjoint boundary condition and sensitivity derivative formula are marked by ABC and SD, respectively, while those being zero are eliminated using an oblique Strikethrough.

4.5.1 Slip Wall Boundaries

For slip walls, the flow boundary conditions are no-penetration ($v_m^A \mathsf{n}_m = v_k^{grid} \mathsf{n}_k$) together with an adiabatic thermal condition ($q_k \mathsf{n}_k = 0$). If the flow is turbulent, a mirror condition for $\tilde{\nu}$ is imposed, namely $\frac{\partial \tilde{\nu}}{\partial x_k} \mathsf{n}_k = 0$. These are taken into account when developing the adjoint boundary conditions.

Starting from the first two surface integrals, S.I_1 and S.I_2 read

$$
\begin{aligned}
\text{S.I\_1} &= -\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \Psi_n \frac{\delta U_n}{\delta b_i} v_m^{grid} \mathsf{n}_m \mathrm{d}S\mathrm{d}t \\
&= -\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} v_m^{grid} \mathsf{n}_m (\Psi_1 + v_m^A \Psi_{m+1} + \frac{|v^A|^2}{2}\Psi_5) \frac{\delta \rho}{\delta b_i} \mathrm{d}S\mathrm{d}t \\
&\quad -\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} v_m^{grid} \mathsf{n}_m \Big\{ \rho (\Psi_{k+1} + v_k^A \Psi_5) \frac{\delta v_k^A}{\delta b_i} + \frac{1}{\gamma - 1} \Psi_5 \frac{\delta p}{\delta b_i} \Big\} \mathrm{d}S\mathrm{d}t
\end{aligned}
$$

$$
\begin{aligned}
\text{S.I\_2} = &\int\limits_{T_{sol}}\int\limits_{S_{\text{Slip}}} \Psi_n \frac{\delta f_{nk}^{inv}}{\delta b_i} \mathsf{n}_k \mathrm{d}S\mathrm{d}t \\
= &-\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \Psi_n f_{nk}^{inv} \frac{\delta \mathsf{n}_k}{\delta b_i} \mathrm{d}S\mathrm{d}t + \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} v_m^{grid} \mathsf{n}_m (\Psi_1 + v_m^A \Psi_{m+1} + \frac{|v^A|^2}{2}\Psi_5) \frac{\delta \rho}{\delta b_i} \mathrm{d}S\mathrm{d}t \\
&+ \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \Big\{ v_m^{grid} \mathsf{n}_m \rho (\Psi_{k+1} + v_k^A \Psi_5) \frac{\delta v_k^A}{\delta b_i} + (v_m^{grid} \mathsf{n}_m \frac{\gamma}{\gamma - 1} \Psi_5 + \Psi_{m+1} \mathsf{n}_m) \frac{\delta p}{\delta b_i} \Big\} \mathrm{d}S\mathrm{d}t \\
&+ \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} (\Psi_1 \rho + \Psi_k \rho v_k^A + \Psi_5 \rho h_t) \frac{\delta (v_m^{grid} \mathsf{n}_m)}{\delta b_i} \mathrm{d}S\mathrm{d}t + \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} p \Psi_{k+1} \frac{\delta \mathsf{n}_k}{\delta b_i} \mathrm{d}S\mathrm{d}t
\end{aligned}
$$

Variation in density and velocities cancel by adding S.I_1 and S.I_2. Therefore

$$\text{S.I\_1} + \text{S.I\_2} = \underbrace{-\int\int_{T_s S_{\text{Slip}}} \Big\{ (\Psi_n f^{inv}_{nk} - p\Psi_{k+1}) \frac{\delta \mathsf{n}_k}{\delta b_i} - (\Psi_1 \rho + \Psi_k \rho v^A_k + \Psi_5 \rho h_t) \frac{\delta (v^{grid}_m \mathsf{n}_m)}{\delta b_i} \Big\} \mathrm{d}S\mathrm{d}t}_{\text{SD}}$$
$$+ \underbrace{\int\int_{T_s S_{\text{Slip}}} (v^{grid}_m \mathsf{n}_m \Psi_5 + \Psi_{m+1} \mathsf{n}_m) \frac{\delta p}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^p} \tag{4.5.1.1}$$

The S.I_3 term is expanded as follows

$$\text{S.I\_3} = -\int\int_{T_s S_{\text{Slip}}} \Psi_n \frac{\delta f^{vis}_{nk}}{\delta b_i} \mathsf{n}_k \mathrm{d}S\mathrm{d}t = \underbrace{-\int\int_{T_s S_{\text{Slip}}} \Psi_n \frac{\delta (f^{vis}_{nk} \mathsf{n}_k)}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{S.I\_3}^{\text{Mom\&Ener}}} + \underbrace{\int\int_{T_s S_{\text{Slip}}} \Psi_n f^{vis}_{nk} \frac{\delta \mathsf{n}_k}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \tag{4.5.1.2}$$

where

$$\text{S.I\_3}^{\text{Mom\&Ener}} = \underbrace{-\int\int_{T_s S_{\text{Slip}}} \Psi_{m+1} \mathsf{n}_m \frac{\delta}{\delta b_i} (\tau_{k\ell} \mathsf{n}_k \mathsf{n}_\ell) \, \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{(\tau_{k\ell}\mathsf{n}_k\mathsf{n}_\ell)}} \underbrace{- \int\int_{T_s S_{\text{Slip}}} \Psi_{m+1} \tau_{k\ell} \mathsf{n}_k \mathsf{n}_\ell \frac{\delta (\mathsf{n}_m)}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}}$$
$$\cancel{- \int\int_{T_s S_{\text{Slip}}} \Psi_{m+1} \frac{\delta}{\delta b_i} (\tau_{k\ell} \mathsf{n}_k \mathsf{t}_\ell \mathsf{t}_m) \, \mathrm{d}S\mathrm{d}t}$$
$$\cancel{- \int\int_{T_s S_{\text{Slip}}} \Psi_5 \frac{\delta}{\delta b_i} \left(v^A_\ell \tau_{km} \mathsf{n}_k \mathsf{t}_m \mathsf{t}_l \right) \mathrm{d}S\mathrm{d}t} \underbrace{- \int\int_{T_s S_{\text{Slip}}} \Psi_5 v^{grid}_\ell \mathsf{n}_\ell \frac{\delta}{\delta b_i} (\tau_{km} \mathsf{n}_k \mathsf{n}_m) \, \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{(\tau_{mk}\mathsf{n}_k\mathsf{n}_m)}}$$
$$\underbrace{- \int\int_{T_s S_{\text{Slip}}} \Psi_5 \tau_{km} \mathsf{n}_k \mathsf{n}_m \frac{\delta}{\delta b_i} \left(v^{grid}_\ell \mathsf{n}_\ell \right) \mathrm{d}S\mathrm{d}t}_{\text{SD}} \cancel{- \int\int_{T_s S_{\text{Slip}}} \Psi_5 \frac{\delta}{\delta b_i} (q_k \mathsf{n}_k) \, \mathrm{d}S\mathrm{d}t} \tag{4.5.1.3}$$

S.I_4 reads

$$\begin{aligned}\text{S.I\_4} = \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tau^{\text{adj}}_{mk}\mathsf{n}_k \frac{\delta v^A_m}{\delta b_i}\mathrm{d}S\mathrm{d}t = & \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tau^{\text{adj}}_{k\ell}\mathsf{n}_k\mathsf{n}_\ell \frac{\delta\left(v^{grid}_m\mathsf{n}_m\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}} - \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tau^{\text{adj}}_{k\ell}\mathsf{n}_k\mathsf{n}_\ell v^A_m \frac{\delta \mathsf{n}_m}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}} \\ & \underbrace{- \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tau^{\text{adj}}_{k\ell}\mathsf{n}_k\mathsf{t}_\ell \frac{\delta\left(v^A_m\mathsf{t}_m\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\left(v^A_m\mathsf{t}_m\right)}} - \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tau^{\text{adj}}_{k\ell}\mathsf{n}_k\mathsf{t}_\ell v^A_m \frac{\delta \mathsf{t}_m}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}}\end{aligned} \tag{4.5.1.4}$$

Term S.I_5 is zero due to the adiabatic condition on the solid walls.

The sum of S.I_6 and S.I_7 yields

$$\text{S.I\_6} + \text{S.I\_7} = \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tilde{\nu}_a\tilde{\mu}\frac{\delta(v^{grid}_m\mathsf{n}_m)}{\delta b_i}\mathrm{d}s\mathrm{d}t - \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tilde{\nu}_a\tilde{\mu}v^A_k\frac{\delta\mathsf{n}_k}{\delta b_i}\mathrm{d}s\mathrm{d}t}_{\text{SD}} \tag{4.5.1.5}$$

The rest of the terms are analyzed as follows

$$\begin{aligned}\text{S.I\_8} = & \underbrace{\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \rho\tilde{\nu}_a\left[\nu + (1+c_{b_2})\,\tilde{\nu}\right]\frac{\partial\tilde{\nu}}{\partial x_k}\frac{\delta\mathsf{n}_k}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}} \\ & \cancel{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \rho\tilde{\nu}_a\frac{\delta}{\delta b_i}\left\{\left[\nu + (1+c_{b_2})\,\tilde{\nu}\right]\frac{\partial\tilde{\nu}}{\partial x_k}\mathsf{n}_k\right\}\mathrm{d}S\mathrm{d}t}\end{aligned} \tag{4.5.1.6}$$

$$\text{S.I\_9} + \text{S.I\_11} = \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \left\{\frac{1}{\sigma}\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\left[\nu + (1+c_{b_2})\,\tilde{\nu}\right] - \frac{c_{b_2}}{\sigma}\frac{\partial\left(\rho\tilde{\nu}_a\tilde{\nu}\right)}{\partial x_k}\right\}\mathsf{n}_k\frac{\delta\tilde{\nu}}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\tilde{\nu}}} \tag{4.5.1.7}$$

$$\text{S.I\_10} = \cancel{\frac{c_{b_2}}{\sigma} \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tilde{\nu}_a \tilde{\mu} \frac{\delta}{\delta b_i} \left(\frac{\partial \tilde{\nu}}{\partial x_k} \mathsf{n}_k \right) \mathrm{d}S\mathrm{d}t} - \underbrace{\frac{c_{b_2}}{\sigma} \int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \tilde{\nu}_a \tilde{\mu} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\delta \mathsf{n}_k}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \tag{4.5.1.8}$$

$$\begin{aligned} \text{S.I\_12} = & \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \rho \tilde{\nu}_a \mathcal{C}_6 \mathcal{P}\left(\tilde{S}, S\right) \frac{1}{S} \varepsilon_{k\ell m} \varepsilon_{kqr} \frac{\partial v_r^A}{\partial x_q} \mathsf{n}_l \mathsf{n}_m \frac{\delta (v_s^{grid} \mathsf{n}_s)}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \\ & + \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \rho \tilde{\nu}_a \mathcal{C}_6 \mathcal{P}\left(\tilde{S}, S\right) \frac{1}{S} \varepsilon_{k\ell m} \varepsilon_{kqr} \frac{\partial v_r^A}{\partial x_q} \mathsf{n}_l v_s^{grid} \mathsf{n}_s \frac{\delta \mathsf{n}_m}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \\ & + \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \rho \tilde{\nu}_a \mathcal{C}_6 \mathcal{P}\left(\tilde{S}, S\right) \frac{1}{S} \varepsilon_{k\ell m} \varepsilon_{kqr} \frac{\partial v_r^A}{\partial x_q} \mathsf{n}_l \mathsf{t}_m \frac{\delta (v_s^A \mathsf{t}_s)}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\left(v_s^A \mathsf{t}_s\right)}} \\ & + \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{Slip}}} \rho \tilde{\nu}_a \mathcal{C}_6 \mathcal{P}\left(\tilde{S}, S\right) \frac{1}{S} \varepsilon_{k\ell m} \varepsilon_{kqr} \frac{\partial v_r^A}{\partial x_q} \mathsf{n}_l v_s^A \mathsf{t}_s \frac{\delta \mathsf{t}_m}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \end{aligned} \tag{4.5.1.9}$$

To eliminate of both ABC^p and ABC$^{(\tau_{k\ell} \mathsf{n}_k \mathsf{n}_\ell)}$ the following adjoint boundary condition

$$v_m^{grid} \mathsf{n}_m \Psi_5 + \Psi_{m+1} \mathsf{n}_m = 0 \tag{4.5.1.10}$$

must be imposed. Terms ABC$^{\left(v_m^A \mathsf{t}_m\right)}$ and ABC$^{\tilde{\nu}}$ are eliminated by the following adjoint boundary conditions

$$\rho \tilde{\nu}_a \mathcal{C}_6 \mathcal{P}\left(\tilde{S}, S\right) \frac{1}{S} \varepsilon_{k\ell m} \varepsilon_{kqr} \frac{\partial v_r^A}{\partial x_q} \mathsf{n}_l \mathsf{t}_m - \tau_{k\ell}^{\text{adj}} \mathsf{n}_k \mathsf{t}_\ell = 0 \tag{4.5.1.11}$$

$$\left\{\frac{1}{\sigma}\frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\left[\nu+\left(1+c_{b_2}\right)\tilde{\nu}\right]-\frac{c_{b_2}}{\sigma}\frac{\partial\left(\rho\tilde{\nu}_a\tilde{\nu}\right)}{\partial x_k}\right\}\mathsf{n}_k=0 \tag{4.5.1.12}$$

4.5.2 No-Slip Wall Boundaries

Along no-slip walls, $v_m^A=v_m^{grid}$ and an adiabatic thermal condition ($q_k\mathsf{n}_k=0$) are applied. The boundary condition imposed for the Spalart-Allmaras variable is $\tilde{\nu}=0$. S.I_1 and S.I_2 are expanded similarly to the slip-wall.

The S.I_3 integrals are expanded as follows

$$\text{S.I\_3}=\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_n\frac{\delta(f_{nk}^{vis}\mathsf{n}_k)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{S.I\_3}^{\text{Mom\&Ener}}}+\underbrace{\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_n f_{nk}^{vis}\frac{\delta\mathsf{n}_k}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}} \tag{4.5.2.1}$$

where

$$\begin{aligned}\text{S.I\_3}^{\text{Mom\&Ener}}=&\cancel{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_{q+1}\mathsf{n}_q\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\frac{\delta\left(\mathsf{t}_m\mathsf{n}_m\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}+\underbrace{\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_{q+1}\mathsf{n}_q\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\mathsf{t}_m\frac{\delta\mathsf{n}_m}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}}\\
&\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_{m+1}\mathsf{n}_m\frac{\delta\left(\tau_{\ell k}\mathsf{n}_k\mathsf{n}_\ell\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\left(\tau_{\ell k}\mathsf{n}_k\mathsf{n}_\ell\right)}}\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_{m+1}\mathsf{t}_m\frac{\delta\left(\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\left(\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\right)}}\\
&\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_{m+1}\tau_{\ell k}\mathsf{n}_k\mathsf{n}_\ell\frac{\delta\mathsf{n}_m}{\delta b_i}\mathrm{d}S\mathrm{d}t-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_{q+1}\mathsf{t}_q\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\mathsf{t}_m\frac{\delta\mathsf{t}_m}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}}\\
&\cancel{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_5 v_q^{grid}\mathsf{n}_q\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\frac{\delta\left(\mathsf{n}_m\mathsf{t}_m\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}+\underbrace{\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_5 v_q^{grid}\mathsf{n}_q\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\mathsf{t}_m\frac{\delta\mathsf{n}_m}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}}\\
&\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_5 v_m^{grid}\mathsf{n}_m\frac{\delta\left(\tau_{\ell k}\mathsf{n}_k\mathsf{n}_\ell\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\left(\tau_{\ell k}\mathsf{n}_k\mathsf{n}_\ell\right)}}\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}}\Psi_5 v_m^{grid}\mathsf{t}_m\frac{\delta\left(\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\right)}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^{\left(\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell\right)}}\end{aligned}$$

$$\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \Psi_5 v_m^{grid} \tau_{\ell k} \mathsf{n}_k \mathsf{n}_\ell \frac{\delta \mathsf{n}_m}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}}$$

$$\underbrace{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \Psi_5 v_q^{grid} \mathsf{t}_q \tau_{\ell k} \mathsf{n}_k \mathsf{t}_\ell \mathsf{t}_m \frac{\delta \mathsf{t}_m}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \cancel{-\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \Psi_5 \frac{\delta \left(q_k \mathsf{n}_k\right)}{\delta b_i} \mathrm{d}S\mathrm{d}t} \tag{4.5.2.2}$$

Analyzing S.I_4, we get

$$\text{S.I\_4} = \underbrace{\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \tau^{\text{adj}}_{mk} \mathsf{n}_k \frac{\delta v_m^{grid}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \tag{4.5.2.3}$$

Term S.I_5 is zero due to the adiabatic condition on the wall and term S.I_6 + S.I_7 is expanded the same was as for the slip-wall. The rest of the terms read

$$\text{S.I\_8} = \underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \rho\tilde{\nu}_a \left[\nu + \left(1 + c_{b_2}\right)\tilde{\nu}\right] \frac{\delta}{\delta b_i}\left(\frac{\partial \tilde{\nu}}{\partial x_k}\mathsf{n}_k\right) \mathrm{d}S\mathrm{d}t}_{\text{ABC}\left(\frac{\partial\tilde{\nu}}{\partial x_k}\mathsf{n}_k\right)}$$

$$\underbrace{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \rho\tilde{\nu}_a \frac{\partial \tilde{\nu}}{\partial x_k}\mathsf{n}_k \mathcal{P}\left(\nu,\rho\right)\mathcal{P}\left(\rho,p\right)\frac{\delta p}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{ABC}^p} \cancel{-\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \rho\tilde{\nu}_a \frac{\partial \tilde{\nu}}{\partial x_k}\mathsf{n}_k \left(1 + c_{b_2}\right)\frac{\delta \tilde{\nu}}{\delta b_i}\mathrm{d}S\mathrm{d}t}$$

$$\underbrace{+\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \rho\tilde{\nu}_a \left[\nu + \left(1 + c_{b_2}\right)\tilde{\nu}\right] \frac{\partial \tilde{\nu}}{\partial x_k}\frac{\delta \mathsf{n}_k}{\delta b_i}\mathrm{d}S\mathrm{d}t}_{\text{SD}} \tag{4.5.2.4}$$

$$\text{S.I\_9} = \cancel{\frac{1}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \frac{\partial\left(\rho\tilde{\nu}_a\right)}{\partial x_k}\left[\nu + \left(1 + c_{b_2}\right)\tilde{\nu}\right]\mathsf{n}_k\frac{\delta\tilde{\nu}}{\delta b_i}\mathrm{d}S\mathrm{d}t} \tag{4.5.2.5}$$

$$\text{S.I\_10} = \cancel{\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \rho\tilde{\nu}_a\tilde{\nu}\mathsf{n}_k\frac{\partial}{\partial b_i}\left(\frac{\partial\tilde{\nu}}{\partial x_k}\right)\mathrm{d}S\mathrm{d}t} \tag{4.5.2.6}$$

$$\text{S.I\_11} = \cancel{-\frac{c_{b_2}}{\sigma}\int\limits_{T_s}\int\limits_{S_{\text{NoSlip}}} \frac{\partial\left(\rho\tilde{\nu}_a\tilde{\nu}\right)}{\partial x_k}\mathsf{n}_k\frac{\delta\tilde{\nu}}{\delta b_i}\mathrm{d}S\mathrm{d}t} \tag{4.5.2.7}$$

$$\text{S.I\_12} = \underbrace{\int\limits_{T_s}\int\limits_{\partial\Omega} \rho\tilde{\nu}_a \mathcal{C}_6 \mathcal{P}\left(\tilde{S}, S\right) \frac{1}{S} \varepsilon_{k\ell m} \varepsilon_{kqr} \frac{\partial v_r^A}{\partial x_q} \mathsf{n}_\ell \frac{\delta v_m^{grid}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{SD}} \tag{4.5.2.8}$$

Collecting and eliminating the ABC^p terms, the following condition

$$\Psi_{m+1}\mathsf{n}_m + \Psi_5 v_m^{grid}\mathsf{n}_m - \frac{1}{\sigma}\rho\tilde{\nu}_a \frac{\partial\tilde{\nu}}{\partial x_m}\mathsf{n}_m \mathcal{P}\left(\nu,\rho\right)\mathcal{P}\left(\rho,p\right) = 0 \tag{4.5.2.9}$$

results. Elimination of the ABC$^{(\tau_{\ell k}\mathsf{n}_k\mathsf{n}_\ell)}$ and ABC$^{(\tau_{\ell k}\mathsf{n}_k\mathsf{t}_\ell)}$ terms leads to

$$\Psi_{m+1}\mathsf{n}_m + \Psi_5 v_m^{grid}\mathsf{n}_m = 0 \tag{4.5.2.10}$$

$$\Psi_{m+1}\mathsf{t}_m + \Psi_5 v_m^{grid}\mathsf{t}_m = 0 \tag{4.5.2.11}$$

Elimination of terms ABC$^{\left(\frac{\partial\tilde{\nu}}{\partial x_k}\mathsf{n}_k\right)}$ leads to the following adjoint condition

$$\rho\tilde{\nu}_a\left[\nu + \left(1 + c_{b_2}\right)\tilde{\nu}\right] = 0 \tag{4.5.2.12}$$

4.5.3 Far–Field Boundaries

Along the far–field, it is assumed that the total variation in the viscous stresses and heat flux is zero. Hereby, the surface integrals that arise from the differentiation of the viscous terms are neglected there. In addition, the far–field boundary is fixed during the optimization, resulting to zero variation in the geometric quantities there.

Two sets of adjoint far–field conditions are derived depending on the far–field flow condition (standard or non–reflecting). In the standard method, the flow far–field condition is treated as inlet and outlet. In the following subsections, two sets of adjoint boundary conditions are derived.

4.5.3.1 Far–Field Boundaries as Inlet/Outlet

Herein, the inlet and outlet conditions are presented for subsonic cases. A set of local flow quantities $\left(V_\ell^{\text{loc}}, \ell = 1, \ldots, 5\right)$ is defined at these boundaries. For subsonic inlet boundaries ($S_{I^{\text{sub}}}$) the value of a one flow quantity, say V_5^{loc}, is extrapolated from the interior domain while, for the rest, a Dirichlet condition is imposed, meaning $\frac{\delta V_\ell^{\text{loc}}}{\delta b_i} = 0, \ell = 1, \ldots, 4$. As a result, term S.I_2 leads to

$$\text{S.I\_2} = \underbrace{\int\limits_{T_s}\int\limits_{S_{I^{\text{sub}}}} \Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_5^{\text{loc}}} \frac{\delta V_5^{\text{loc}}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{V_5^{\text{loc}}}} \tag{4.5.3.1}$$

Eliminating term $\text{ABC}^{V_5^{\text{loc}}}$ leads to the following expression

$$\Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_5^{\text{loc}}} = 0 \tag{4.5.3.2}$$

The rest of the terms vanish by taking the Dirichlet boundary condition imposed on $\tilde{\nu}$ at the inlet boundaries into account and imposing $\tilde{\nu}_a = 0$.

For a subsonic outlet ($S_{O^{\text{sub}}}$), V_5^{loc} is imposed by a Dirichlet condition, resulting to $\frac{\delta V_5^{\text{loc}}}{\delta b_i} = 0$. $V_\ell^{\text{loc}}, (\ell = 1, \dots, 4)$ are extrapolated from the flow domain. Taking this into account, term S.I_2 reads

$$\begin{aligned} \text{S.I\_2} = & \underbrace{\int\limits_{T_s}\int\limits_{S_{O^{\text{sub}}}} \Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_1^{\text{loc}}} \frac{\delta V_1^{\text{loc}}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{V_1^{\text{loc}}}} + \underbrace{\int\limits_{T_s}\int\limits_{S_{O^{\text{sub}}}} \Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_1^{\text{loc}}} \frac{\delta V_2^{\text{loc}}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{V_2^{\text{loc}}}} \\ & + \underbrace{\int\limits_{T_s}\int\limits_{S_{O^{\text{sub}}}} \Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_1^{\text{loc}}} \frac{\delta V_3^{\text{loc}}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{V_3^{\text{loc}}}} + \underbrace{\int\limits_{T_s}\int\limits_{S_{O^{\text{sub}}}} \Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_1^{\text{loc}}} \frac{\delta V_4^{\text{loc}}}{\delta b_i} \mathrm{d}S\mathrm{d}t}_{\text{ABC}^{V_4^{\text{loc}}}} \end{aligned} \tag{4.5.3.3}$$

Eliminating the $\text{ABC}^{V_\ell^{\text{loc}}}, (\ell = 1, \dots, 4)$ terms requires the following conditions

$$\Psi_n \mathsf{n}_k \frac{\partial f_{nk}^{inv}}{\partial U_m} \frac{\partial U_m}{\partial V_\ell^{\text{loc}}} = 0 \tag{4.5.3.4}$$

A zero Neumann condition is imposed on $\tilde{\nu}$, meaning that $\dfrac{\delta}{\delta b_i}\left(\dfrac{\partial \tilde{\nu}}{\partial x_k}\mathsf{n}_k\right) = 0$. The remaining surface integrals eliminated by setting the multiplier of $\frac{\delta \tilde{\nu}}{\delta b_i}$ equal to zero.

4.5.3.2 Far–Field Boundaries, Non–Reflecting

In case of using a non–reflecting far–field boundary condition for flow solution, the far–field adjoint conditions are derived based on the non-reflecting boundary conditions presented in [130] for hyperbolic PDEs. For simplicity, let us assume that the far-field boundary is aligned with the x-direction; thus, y stands for the normal to the far-field boundary. Field adjoint equation can be re-written as

$$-\frac{\partial \Psi_m}{\partial t} - L_{np}\Lambda_{pq}L^{-1}_{qm}\frac{\partial \Psi_n}{\partial y} = A_{nm1}\frac{\partial \Psi_n}{\partial x} - \mathcal{R}^{adj}_{visc,m} - \mathcal{R}^{adj}_{source,m} \tag{4.5.3.5}$$

where $\mathcal{R}^{adj}_{visc}$, $\mathcal{R}^{adj}_{source}$ are the adjoint diffusion and source terms, respectively. A_{nm1}, A_{nm2} (diagonalized as $A_{nm2} = L_{np}\Lambda_{pq}L^{-1}_{qm}$) are the Cartesian components of the flux Jacobian. Multiplying both sides with L_{mk} and introducing $\mathfrak{D}\breve{\Psi}_k = L_{mk}\mathfrak{D}\Psi_m$, where $\mathfrak{D}$ refers to any partial derivative, Eq. 4.5.3.5 takes the form

$$-\frac{\partial \breve{\Psi}_k}{\partial t} - \lambda_k\frac{\partial \breve{\Psi}_k}{\partial y} = L_{mk}\left(A_{nm1}\frac{\partial \Psi_n}{\partial x} - \mathcal{R}^{adj}_{visc,m} - \mathcal{R}^{adj}_{source,m}\right) \tag{4.5.3.6}$$

Based on the sign of the eigenvalues λ_k (diagonal entries of Λ), the incoming and outgoing adjoint characteristic variables $\breve{\Psi}_k$ are identified. The outgoing ones contribute to the residuals at the far-field nodes while the incoming ones are set to zero. The above can readily be generalized for far-field boundaries not aligned with x.

Once the flow and adjoint fields are obtained, all terms denoted by SD are used to compute the sensitivity derivatives that are necessary for the G-B optimization. The CFD-CAA optimization workflow is shown in Fig. 4.2.

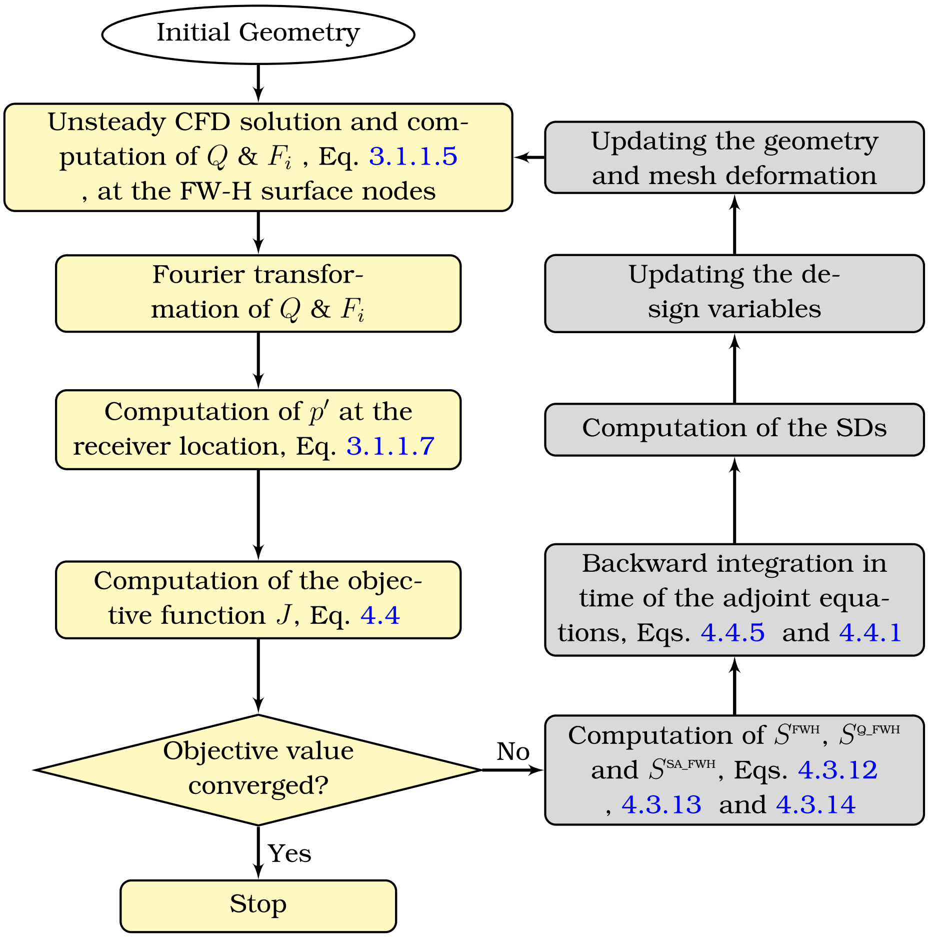

Initial Geometry
Unsteady CFD solution and computation of Q & F_i , Eq. 3.1.1.5 , at the FW-H surface nodes
Fourier transformation of Q & F_i
Computation of p' at the receiver location, Eq. 3.1.1.7
Computation of the objective function J, Eq. 4.4
Objective value converged?
Yes
Stop
No
Computation of S^{FWH}, $S^{\mathrm{Q\_FWH}}$ and $S^{\mathrm{SA\_FWH}}$, Eqs. 4.3.12 , 4.3.13 and 4.3.14
Backward integration in time of the adjoint equations, Eqs. 4.4.5 and 4.4.1
Computation of the SDs
Updating the design variables
Updating the geometry and mesh deformation

Figure 4.2: Work-flow of the CFD-CAA Optimization. Primal and adjoint parts of the workflow are in yellow and gray, respectively.

Chapter 5

Verification of the Hybrid CFD-FW-H solver

In this section, two verification cases for the coupled CFD–CAA solver are presented for 2D and 3D problems. They both refer to the sound field created by a monopole source. The purpose of these test cases are to verify the implementation of the FW-H analogy and its differentiation. Since the flow field is given by an analytical solution these test cases provide the opportunity of exclusively testing the developed acoustic software. Validation of the software is also included in the next chapter where the FW-H results are compared to other results computed by just numerically integrating the URANS equations.

5.1 2D Monopole in uniform flow

In this section, the implementation of the FW-H integral is verified in a 2D problem. To do so, results of the FW-H integral, Eq. 3.1.1.7, are compared with the analytical solution of the sound field generated by a monopole source in a uniform flow. Additionally, in order to verify the part of the code that differentiates the FW-H integral, derivatives of the objective function, Eq. 4.4 , w.r.t. the coordinates of the monopole source, $\vec{x}_s$, are computed by the code and compared with the outcome of a central second–order FD based on the analytical solution. The monopole source is located at the origin of the coordinate system and is exposed to a uniform flow v_∞ along the $+x$ direction. The complex velocity potential of the case is [83]

$$\phi(\vec{x}_o, \vec{x}_s, \omega) = \frac{Ai}{4\beta} \exp i(\omega t + M_\infty k \bar{x}_1/\beta^2) H_0^{(2)} \left(\frac{k}{\beta^2} \sqrt{\bar{x}_1^2 + \beta^2 \bar{x}_2^2} \right) \tag{5.1.1}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the same as in Eq. 3.1.1.9. The perturbation field of flow variables are obtained from the real parts of

$$p' = -\rho_0(\frac{\partial \phi}{\partial t} + v_{\infty 1}\frac{\partial \phi}{\partial x}), \quad v' = \nabla \phi, \quad \rho' = p'/{c_\infty}^2 \tag{5.1.2}$$

These variables are used to compute the flow data on the FW-H surface which are required to compute F_i and Q in the FW-H integral. In this case, $M_\infty = 0.6$, $A = 0.02\ m^2/s$ and $\omega = 0.162\ rad/s$. The FW-H surface is a circle with radius $4m$ having its center in the center of the coordinate and 150 nodes on it. Fig.5.1 compares the time history of the pressure fluctuations p' for a receiver located at $\vec{x}_o = (500m, 0m)$ and the directivity plot of the root mean squared (r.m.s.) of p' at the radius of $R = 500m$ with their analytical solution. The results of the FW-H integral exactly match the analytical solution. This is a convincing verification of the implementation of the 2D FW-H formulation, in problems with a uniform mean flow. The derivatives w.r.t. to the source coordinates are also compared in Fig. 5.2. There is a very good agreement between FDs of the analytical solution and the differentiation of the FW-H. Being a saddle point, the derivative w.r.t. x_{s_2} is zero.

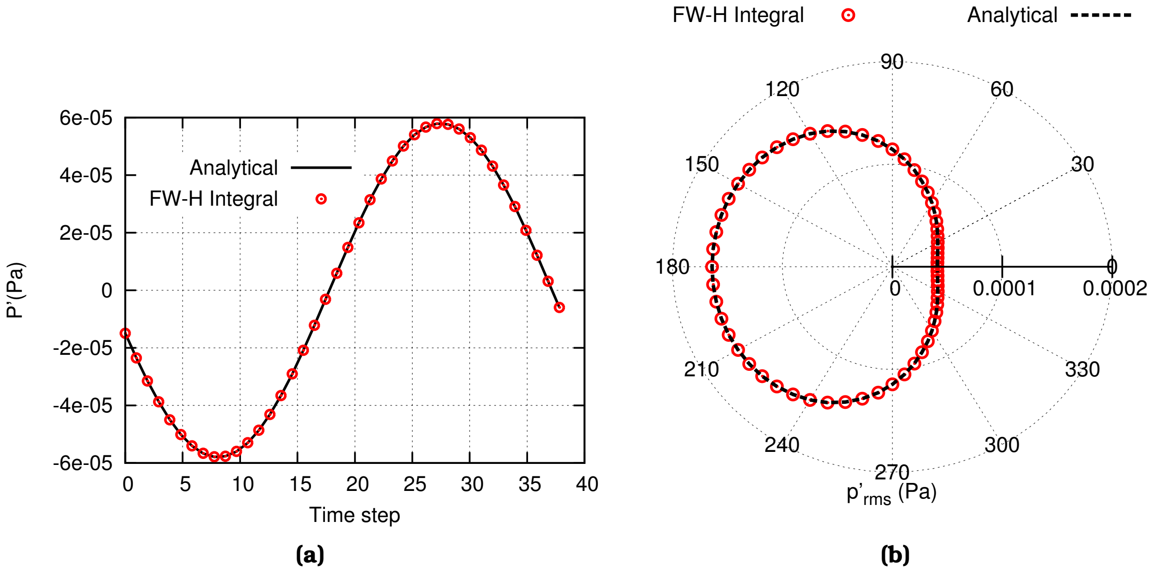


Figure 5.1: *2D Monopole in uniform flow*: (a) Comparison of the time history of pressure fluctuation within a period, for a receiver located at $(500m, 0m)$. (b) Comparison of the directivity plots at $R = 500m$.

5.2 3D Monopole in uniform flow

This section is to verify the implementation of the FW-H analogy for 3D problems. To do so, results of the FW-H integral are compared to a well-known analytical

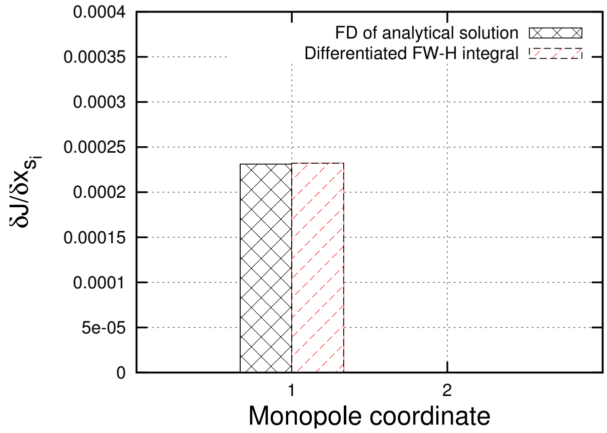


Figure 5.2: *2D Monopole in uniform flow*: Comparison of the derivatives of the acoustic objective function w.r.t. to the three coordinates ($x_{s_i}, i = 1, 2$) of the monopole computed by the differentiation of the FW-H and FDs applied to the closed–form expressions.

solution of the sound field generated by a monopole source in a uniform 3D flow. Similarly to the 2D case in section 5.1, the stationary monopole source is located at the origin of the coordinate system and there is a uniform flow v_∞ along the $+x$ direction. The complex velocity potential of the case is

$$\phi(\vec{x}_o, \vec{x}_s, \omega) = A \exp(i\omega t) \frac{\exp(-ikr^+)}{4\pi r^*} \tag{5.2.1}$$

where r^+ and r^* are the same as in Eq. 3.1.2.2. The perturbation fields of flow variables are computed the same as Eq. 5.1.2 . In this case, $M_\infty = 0.5$, $A = 0.004$ m^2/s and $\omega = 3.095$ rad/s. The FW-H surface is a cube extending from $-0.5m$ to $0.5m$ in all three direction, with 20 nodes on each edge (400 on each face). A sketch of the case is shown in Fig. 5.3. Fig.5.4a compares the time history of p' at a receiver located at $\vec{x}_o = (10m, 0m, 10m)$. The results of the FW-H integral exactly match the analytical solution. This is a convincing verification of the implementation of the 3D FW-H formulation, in problems with a uniform mean flow. The derivatives w.r.t. to the source coordinates are compared in Fig. 5.4b. The agreement between FDs of the analytical solution and the differentiation of the FW-H is very good. Since this is a saddle point, the derivative w.r.t. x_{s_2} is zero. The analytical derivatives can be found in Appendix B.

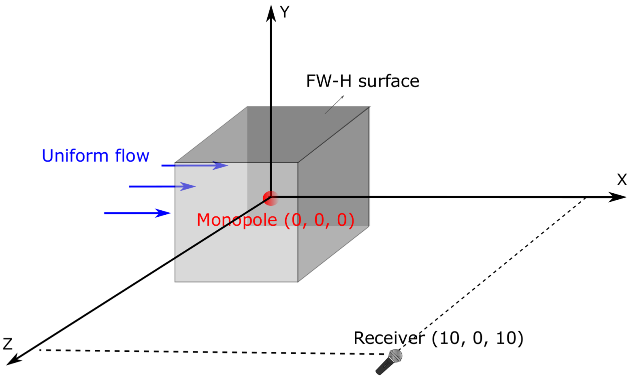


Figure 5.3: *3D Monopole in uniform flow*: Schematic illustration of the monopole source, the FW-H surface and the receiver.

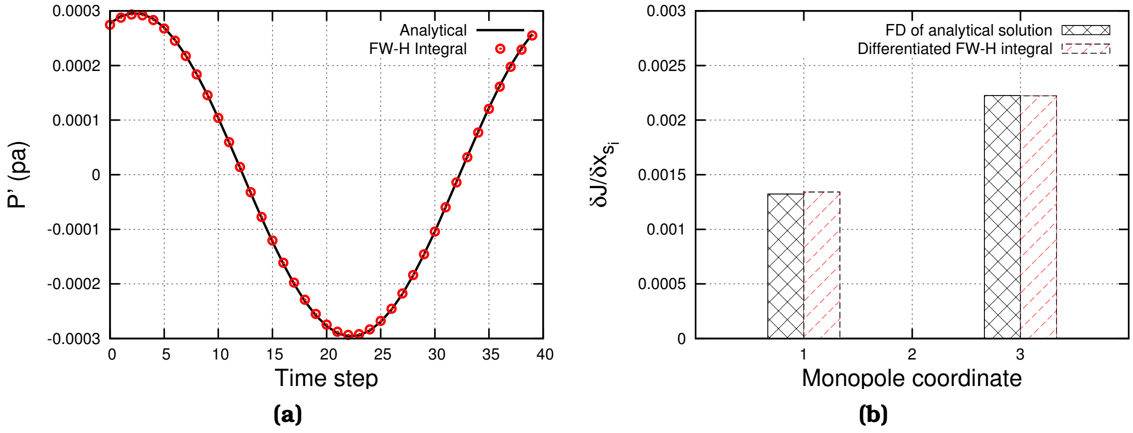


Figure 5.4: *3D Monopole in uniform flow*: (a) Comparison of the time history of pressure fluctuation within a period, for a receiver located at $(10m,\, 0m,\, 10m)$. (b) Comparison of the derivatives of the acoustic objective function w.r.t. to the three coordinates $(x_{s_i}, i = 1, 2, 3)$ of the monopole computed by the differentiation of the FW-H and FDs applied to the closed–form expressions.

Chapter 6

Aeroacoustic (& Aerodynamic) Optimization Cases

In this chapter, the continuous adjoint technique (developed in Chapter 4) supported by shape parameterization methods used to perform gradient-based methods (such as steepest-descent) for the aeroacoustic shape optimization. Since the accuracy of flow data on the FW-H surface affects the application of the FW-H analogy, the implementation of the FW-H analogy is verified, in most of the cases in this chapter, by comparing its results to those achieved by post–processing the flow quantities' time–series computed from the numerical solution of the URANS equations.

6.1 Unconstrained Aerodynamic & Aeroacoustic Optimization of a Pitching NACA12 Airfoil - Inviscid Flow

This test case is concerned with a NACA12 isolated airfoil which is pitching about the quarter–chord point in an inviscid flow, with a 2 deg amplitude and frequency of f $= 10Hz$. The free–stream Mach number and flow angle are $M = 0.6$ and 0 deg, respectively. These correspond to a reduced frequency of $\omega_{red} = \frac{2\pi \mathrm{f} C}{2v_\infty} = 0.153$ where C is the airfoil chord length. A 2D unstructured grid which extends 50 chords away from the airfoil's mid–chord is used, with 51000 nodes overall, among which 202 nodes on the airfoil contour. The airfoil is parameterized using two Bezier curves with 8 Control Points (CPs) each, Fig. 6.1a. The simulation computes the solution at 40 time steps per period of pitching, with 200 pseudo–time steps per time step, which is sufficient for the residuals to drop at least 6 orders of magnitude at each time step. The numerical solution of a period takes ~ 9 min on a single NVIDIA Tesla K40 GPU. After the first 3 transient periods, the monopole and dipole source terms of the FW-H analogy are computed and stored during the 4^{th} period. These data are then used to compute the FW-H integral.

Both aerodynamic and aeroacoustic shape optimizations are performed in this case. Since the exact period of pitching is known a priori, the primal fields are stored only during the last (4^{th}) period in order this to be used for the unsteady adjoint solution. For both optimizations, the adjoint solver runs for 4 periods backwards in time to reach a periodic state. Since 200 pseudo–time steps are performed within each adjoint time step, the adjoint solution takes almost the same time as the primal. An algebraic–based model is used for the grid displacement due to both pitching motion and optimization, and the displacement terminates at $R = 3C$. In what follows, the results of aerodynamic and aeroacoustic optimizations are separately presented in two subsections.

6.1.1 Aerodynamic Optimization

In order to verify the implementation of unsteady continuous adjoint solver, a shape optimization of a pitching isolated NACA12 for time–averaged lift force is performed. Figure 6.1b compares the SDs, obtained by the adjoint solver w.r.t. those computed by second–order accurate central FDs. For the latter, the CPs are perturbed by $10^{-7}C$. SDs from the two methods are in an excellent agreement. Then, the so–computed adjoint SDs are used in shape optimization. Figure 6.2a shows how the lift force has increased from its initial value which is zero (a symmetric airfoil pitching around zero angle of attack) after 4 optimization cycles, by changing the shape basically at the trailing edge, Fig. 6.2b. Since the flow is inviscid, the lift force increases almost linearly during the optimization and since there is no constraint imposed, it was decided to perform only 4 optimization cycles.

6.1.2 Aeroacoustic Optimization

Next, the optimization method is used for aeroacoustic noise reduction. The case is the same as that about lift maximization while the objective function is defined by Eq. 4.4. The FW-H surface is placed at $R = 4C$ (outside the grid displacement zone) from the airfoil mid–chord $(0.5C, 0)$. The receiver is located at $20C$ underneath the leading edge, $\vec{x}_r = (0, -20C)$. In order to verify the SDs of the aeroacoustic objective function obtained by the unsteady continuous adjoint solver, they are compared with the outcome of FDs. FDs are performed as in subsection 6.1.1. Figure 6.3a shows that the two methods are in a good agreement.

The so–computed SDs are then used to perform aeroacoustic shape optimization. Three different sub-cases are considered here. In Case 1, the whole shape of the airfoil can be changed during the optimization, while in Case 2 the optimization is allowed to change only the shape of the suction side. In Case 3, the whole shape of the airfoil, excluding though the trailing edge which is fixed, can be changed. As illustrated in Fig. 6.3b, after 4 design cycles, the noise objective

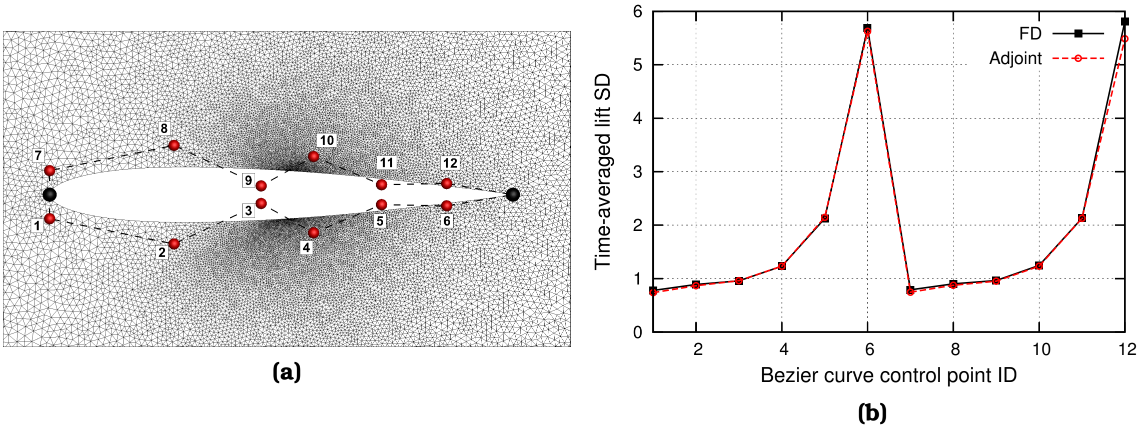


Figure 6.1: *Unconstrained Aerodynamic Optimization of a Pitching NACA12 Airfoil - Inviscid Flow*: Results after 4 design cycles. (a) A view of the grid close to the airfoil and control points of the Bezier curves. FD computations were performed only for the labeled CPs, colored in red. (b) Comparison of the time-averaged lift sensitivity derivative computed using the adjoint method and FDs.

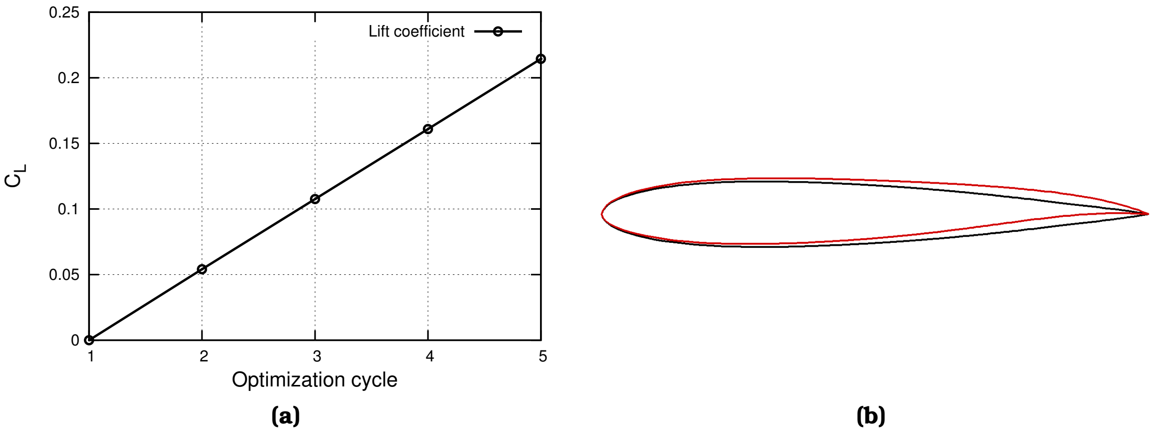


Figure 6.2: *Unconstrained Aerodynamic Optimization of a Pitching NACA12 Airfoil - Inviscid Flow*: (a) Evolution of the time-averaged lift coefficient during the first 4 optimization cycles. (b) Baseline and optimized (after 4 cycles) airfoils in black and red, respectively.

function, Eq. 4.4 is reduced by about 2%, 8% and 20%, in Cases 3, 2 and 1, respectively. This is expected as the optimization gain increases by increasing the degrees of freedom. Also, the reduction in the objective value results in pressure fluctuations with a lower amplitude, as it can be seen in Fig. 6.4a, with the lowest amplitude occurring in Case 1. Comparison of the sound directivity plots of the baseline and optimized airfoil of Case1, in Fig. 6.4b, shows an omni-directional sound reduction. Figure 6.5 compares the baseline airfoil with the optimized shapes. It can be seen that shapes optimized for noise become slightly thinner close to the leading edge and much thicker at the trailing edge (apart, of course, from Case 3 in which the trailing edge is not allowed to change). This is, of course, the outcome of an optimization for the selected receiver's location. The effect of the trailing edge being the main mechanism in noise generation, reflects on the difference between the outcomes of Cases 2 and 3; Case 3 by keeping the trailing edge fixed during the optimization, resulted to a lower noise drop even though a great part of the airfoil can be reshaped. Regarding lift, the baseline airfoil has a zero lift coefficient due to its symmetric shape and the pitching around the horizontal axis. The time-averaged lift of the noise optimized shape in Case 1 becomes negative and equal to -0.132. The reduced design spaces in Cases 2 and 3 increase the time-averaged lift coefficient to 0.64 and 0.06, respectively.

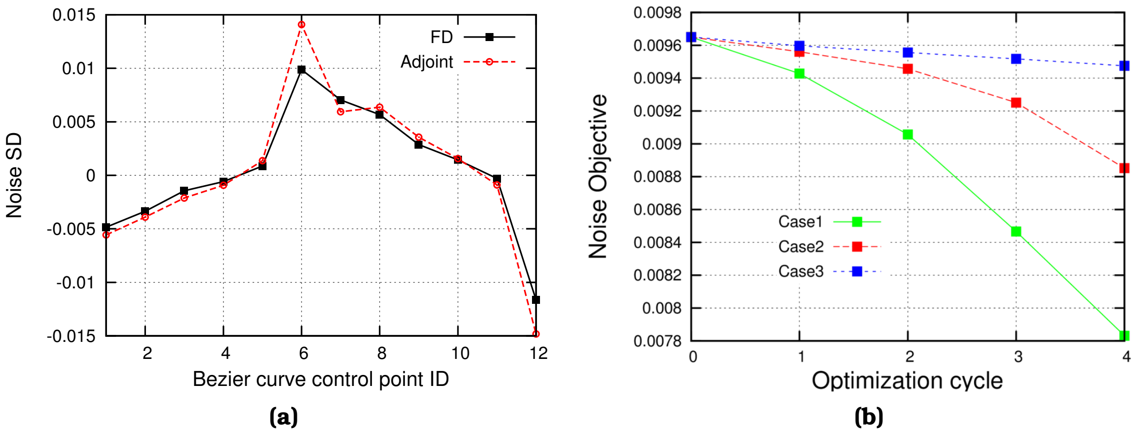


Figure 6.3: *Unconstrained Aeroacoustic Optimization of a Pitching NACA12 Airfoil - Inviscid Flow*: (a) Comparison of the noise SDs computed using the adjoint to the hybrid method and FDs. (b) Evolution of the noise objective value during the first 4 optimization cycles.

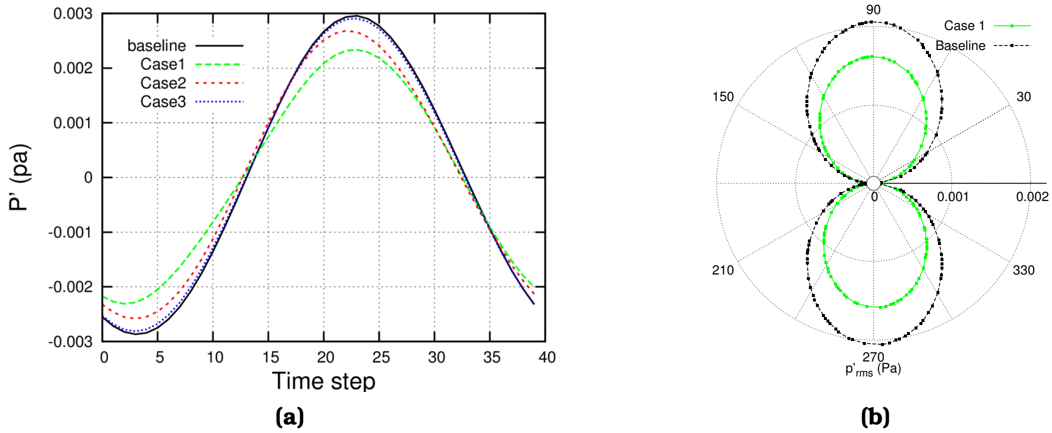


Figure 6.4: *Unconstrained Aeroacoustic Optimization of a Pitching NACA12 Airfoil - Inviscid Flow*: (a) Time-history of pressure fluctuation in a period for the receiver at $\vec{x}_r = (0, -20C)$. (b) Comparison of the directivity plots of p'_{rms} at radius $R = 20C$, between the baseline and optimized airfoils.

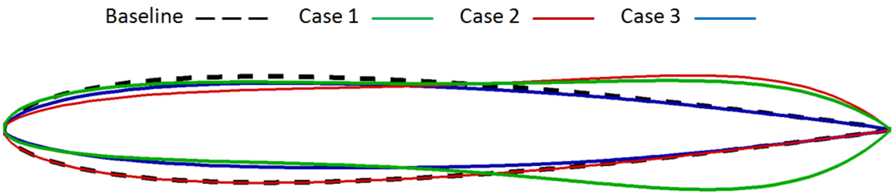


Figure 6.5: *Unconstrained Aeroacoustic Optimization of a Pitching NACA12 Airfoil - Inviscid Flow*: Shape of the baseline (NACA12) and optimized airfoils after 4 optimization cycles.

6.2 Lift–Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow

This test case is based on a RAE2822 isolated airfoil which is pitching about the quarter–chord point in an inviscid flow. The amplitude and frequency of the pitching motion is 2 deg and f $= 10Hz$. The free–stream Mach number and flow angle are $M = 0.6$ and 0 deg, respectively. These correspond to a reduced frequency of $\omega_{red} = 0.153$. A view of the grid and location of the FW-H surface is shown in Fig. 6.6. The 2D unstructured grid extends 50 chords away from the airfoil and includes 51000 nodes, among which 202 nodes are located on the airfoil contour. The FW-H surface is placed at $R = 4C$ from the airfoil mid–chord ($0.5C$, 0) (origin) with 151 nodes on it. Each period of pitching is formed by 40 time steps. 200 pseudo–time steps per time step are performed and this is sufficient for the residuals of the primal and adjoint equations to drop by at least 6 orders of magnitude at each time step.

The numerical solution of a period of either the primal or adjoint problem takes $\sim 9 \text{ min}$ on a single NVIDIA Tesla K40 GPU. The numerical solution is carried for 4 periods of pitching, and the FW-H integral is computed by accounting for the monopole and dipole source terms computed during the 4^{th} period. Only the 4^{th} period's flow field is stored and used during the backward in time integration of the adjoint equations. The adjoint solution is continued for 5 periods to reach a periodic state. Similar to the previous case, grid displacement for both pitching and optimization is performed up to $R = 3C$ using an algebraic-based model.

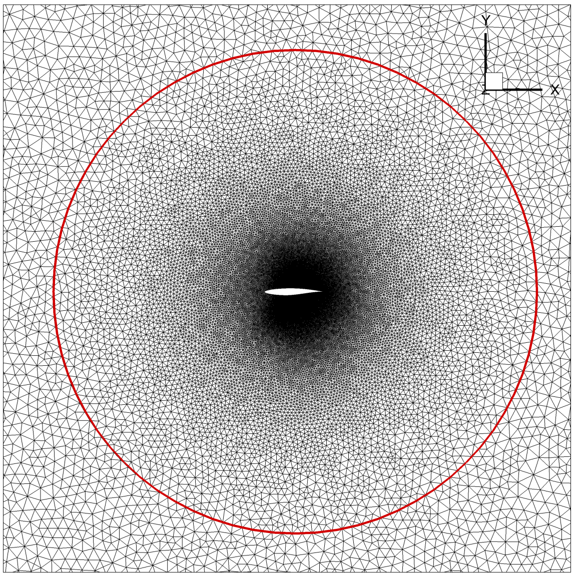


Figure 6.6: *Lift-Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow*: A view of the grid close to the location of the FW-H surface, shown in red, at radius $R = 4C$.

6.2.1 Comparison Between the Hybrid method and URANS

To verify the developed acoustic tool in this case, a comparison between the hybrid solver and the outcome of a pure URANS-based aeroacoustic simulation is performed. The directivity pattern of the p'_{rms} at R=$9C$ is plotted in Fig. 6.7 and shows a very good agreement between results of the unsteady CFD (post-processing of the computed pressure time-series at grid nodes lying along a circle with $R = 9C$) and the application of the FW-H integral on the flow time-series computed on the FW-H surface.

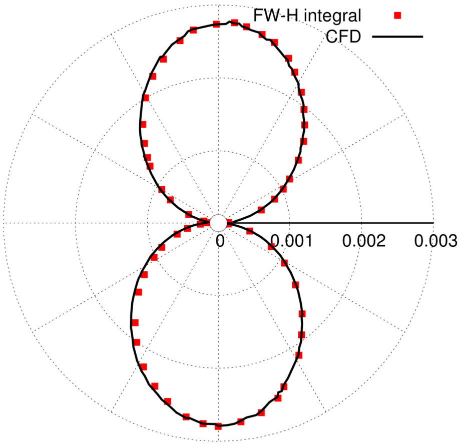


Figure 6.7: *Lift-Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow*: Comparison of the directivity plots of p'_{rms} at radius $R = 9C$ between the hybrid method and pure CFD solvers.

6.2.2 Adjoint–Based Aeroacoustic Optimization

In the aeroacoustic shape optimization, the objective function is defined using Eq. 4.4, integrated over all resolved frequencies. The airfoil pressure and suction sides are parameterized using two Bezier curves, with 20 CPs each, which are free to move in the y direction. Since the first and last CPs are fixed, this case has 36 design variables. The receiver is located at $\vec{x}_o = (0, -20C)$. In this case, an equality constraint is imposed on the time-averaged lift so as to respect the initial lift force during the optimization. This is done using a gradient projection method with a deferred correction (see Appendix A).

In order to verify the computed SDs of the function of Eq. 4.4 using the adjoint solver, these are compared with those obtained by second-order accurate central FDs where CPs were perturbed by $10^{-7}C$. Result in Fig. 6.8a shows a good agreement between the two methods. There are some discrepancies at the CPs at the trailing and leading edge areas; however, even for those control points, the SDs obtained by the two methods have the same signs.

Then, the adjoint-based shape optimization takes place. As illustrated in Fig. 6.8b, after 18 design cycles, the noise objective function is reduced by more than 60%. This figure also shows that the proposed constraint imposition method keeps the time-averaged lift value almost constant, as it changes about 3% by the end of the optimization. As expected, the reduction in the objective value results in a lower amplitude in pressure fluctuations, as shown in Fig. 6.10a.

Figure 6.9 compares the baseline and the optimized airfoil shapes. It shows that the airfoil's shape is changed mainly close to the trailing edge while the rest of it remains almost intact. Figure 6.10b compares the p'_{rms} directivity plot of the baseline and the optimized airfoils at the radius of R=20C and shows that the reduction in noise is omni-directional. Finally, in Fig. 6.11, iso-areas of pressure fluctuations are compared between the baseline and optimized geometries at different time steps. Fluctuation amplitudes in the optimized geometry are lower than in the baseline one.

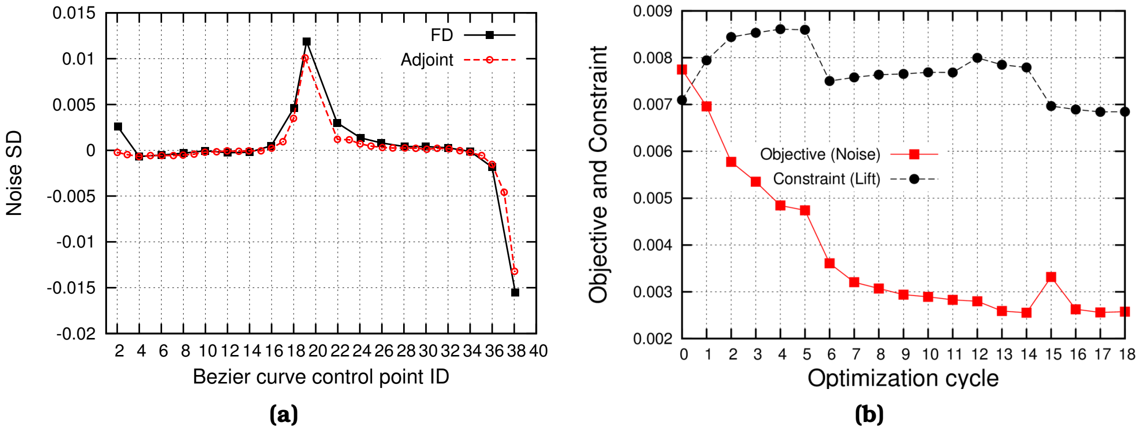


Figure 6.8: *Lift-Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow*: Results after 18 design cycles. (a) Comparison of the noise (Eq. 4.4) SDs for some CPs, using the adjoint to the hybrid solver and FDs. (b) Convergence of the objective and constraint functions. The constraint is imposed on the initial lift value (0.007) and the captured value at the end of the optimization is (0.0068).

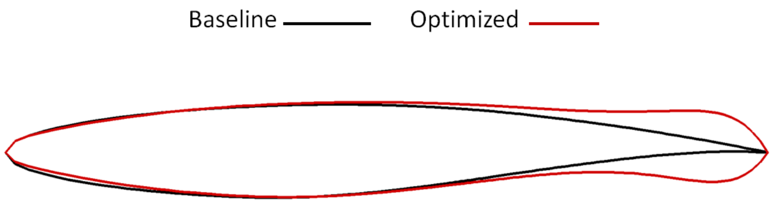


Figure 6.9: *Lift-Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow*: Shapes of the baseline and optimized airfoils.

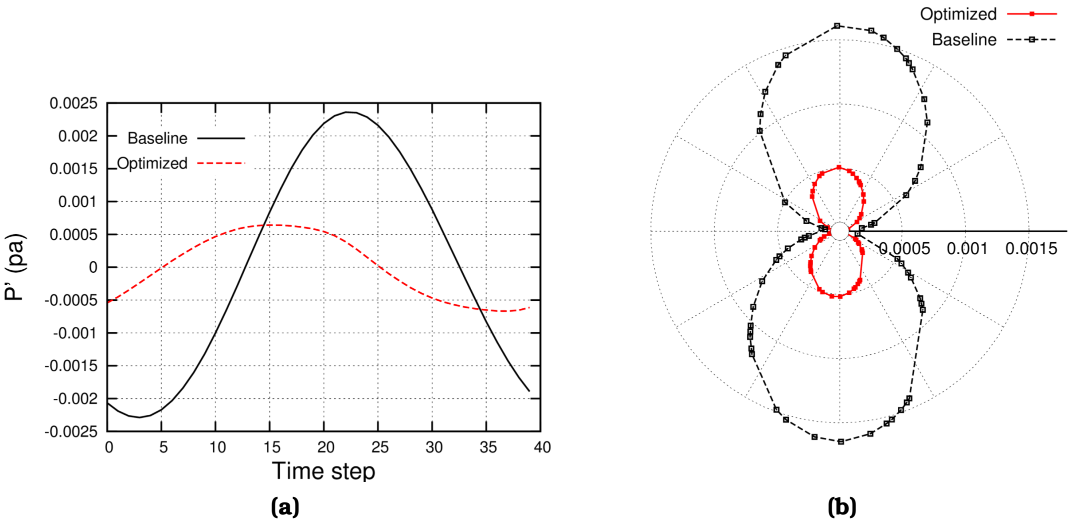


Figure 6.10: *Lift-Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow*: Results after 18 optimization cycles. (a) Time history of pressure fluctuation within a period at the receiver's location. (b) Comparison of the directivity plots of p'_{rms} for the baseline and optimized airfoils, at $R = 20C$.

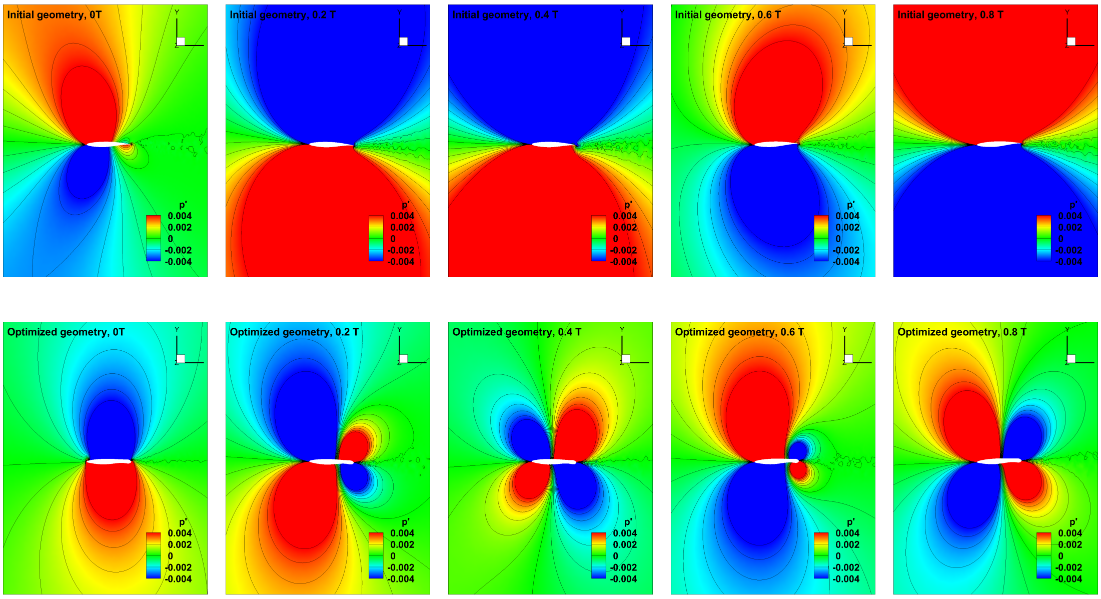


Figure 6.11: *Lift-Constrained Aeroacoustic Optimization of a Pitching RAE2822 Airfoil - Inviscid Flow*: Iso-areas of pressure fluctuations around the baseline (top row) and aeroacoustically optimized airfoil (bottom row) at different instants within a period.

6.3 Gradient Verification of a Pitching NACA12 Airfoil - Turbulent Flow

This test case is referred to a 2D isolated NACA12 airfoil in turbulent flow and aims exclusively at verifying the adjoint based sensitivities w.r.t. FDs in turbulent flows. The airfoil is pitching about the quarter-chord point with an amplitude equal to 2.0 deg and frequency f $= 10Hz$. The Reynolds and Mach numbers are $Re = 4.8 \times 10^6$ and $M = 0.2$, respectively. A 2D hybrid grid which extends 50 chords away from the airfoil is used, with 67000 nodes overall, among which 602 nodes on the airfoil contour and 151 nodes on the FW-H surface which is located at radius $R = 3C$ from the airfoil's mid-chord. The Spalart-Allmaras turbulence model is employed and the non-dimensional distance (y^+) of the first node off the wall distance is below 1.

The simulation computes 40 time steps per period of pitching with 200 iteration per time step making sure at least 6 orders of magnitude drop in residuals within each time step of primal and adjoint solution. After 4 periods that the transient phase is passed, one period of pitching is stored. The numerical solution of each period of pitching takes ~ 5 min, for either the primal or the adjoint equation, on a single NVIDIA Tesla P100 GPU. The adjoint solution is performed for 4 periods to reach a periodic state.

The pressure and suction sides of the airfoil are parameterized using 8 Bezier CPs each, which are free to move in the y direction. Since the first and the last CPs are fixed, this case has 12 design variables. The grid displacement is based on the linear spring analogy [32]. The displacement is confined in an area with with radius $R = 2.5C$.

In order to verify the computed gradients of aeroacoustic (Eq. 4.4 integrated over all resolved frequencies) and aerodynamic (time-averaged lift and drag) objective functions using the adjoint solver, these are compared with those obtained by central FDs where CPs were perturbed by $10^{-7}C$. As presented in Fig. 6.13, the gradients of the time-averaged lift and noise objective functions, obtained by the adjoint solver, perfectly match FDs; for the time-averaged drag, the agreement is quite satisfactory.

In Fig. 6.14, iso-areas of adjoint density is plotted for each objective function at different time instants within a period of pitching. Streamlines of the adjoint velocities are also drawn. One may notice that the adjoint field resulted from the aeroacoustic objective function has much stronger unsteadiness compared to the adjoints for the time-averaged lift and drag.

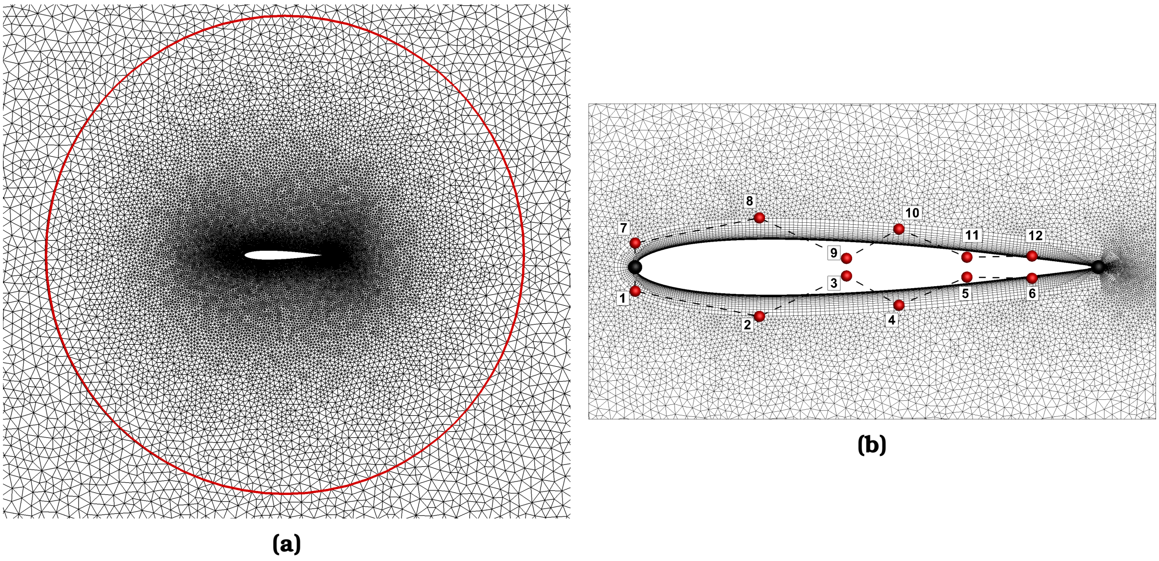
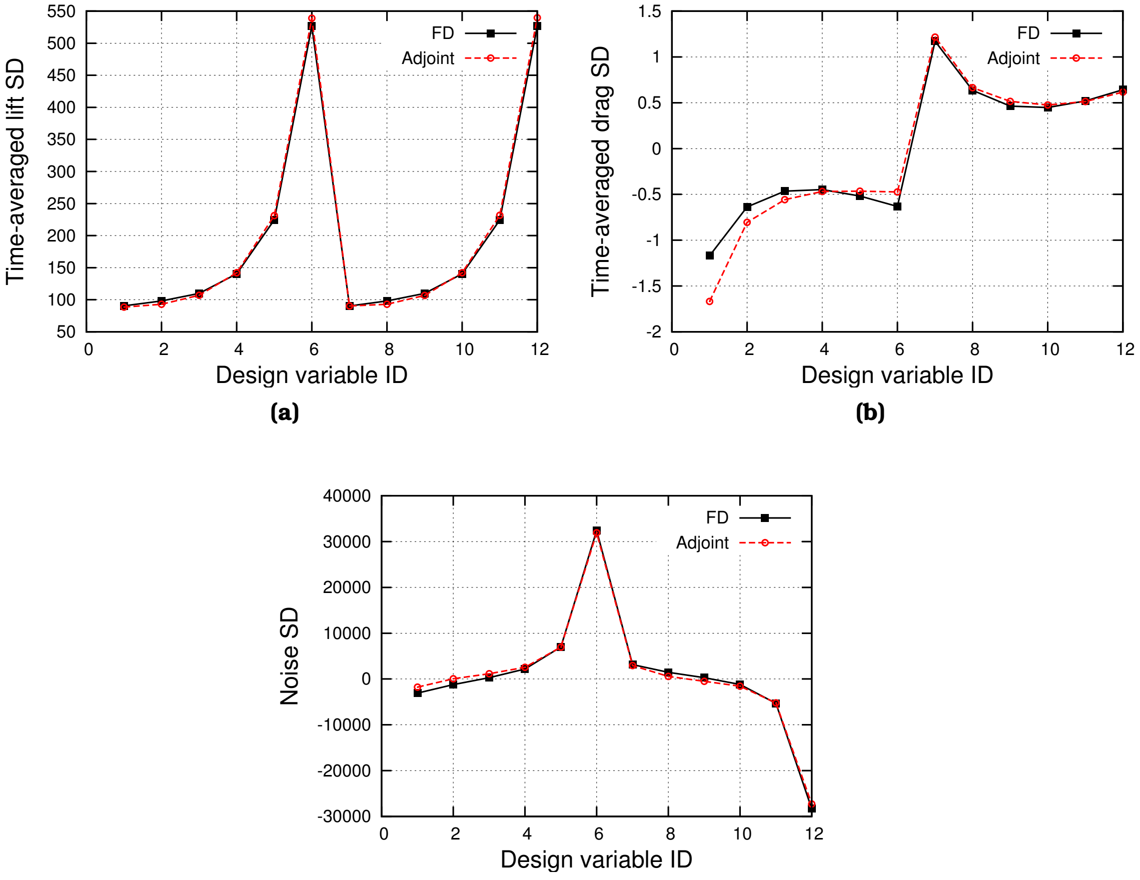


Figure 6.12: *Gradient Verification of a Pitching NACA12 Airfoil - Turbulent Flow*: (a) A view of the grid close to the location of the FW-H surface shown in red at radius $R = 3C$, in red. (b) A view of the grid close to the airfoil and control points of the Bezier curves. FDs are performed for the labeled CPs which are marked in red.



(a)

(b)

(c)

Figure 6.13: *Gradient Verification of a Pitching NACA12 Airfoil - Turbulent Flow*: Comparison of the SDs using the adjoint to the hybrid solver and FDs for (a) time–averaged lift, (b) time–averaged drag and (c) noise objective (Eq. 4.4).

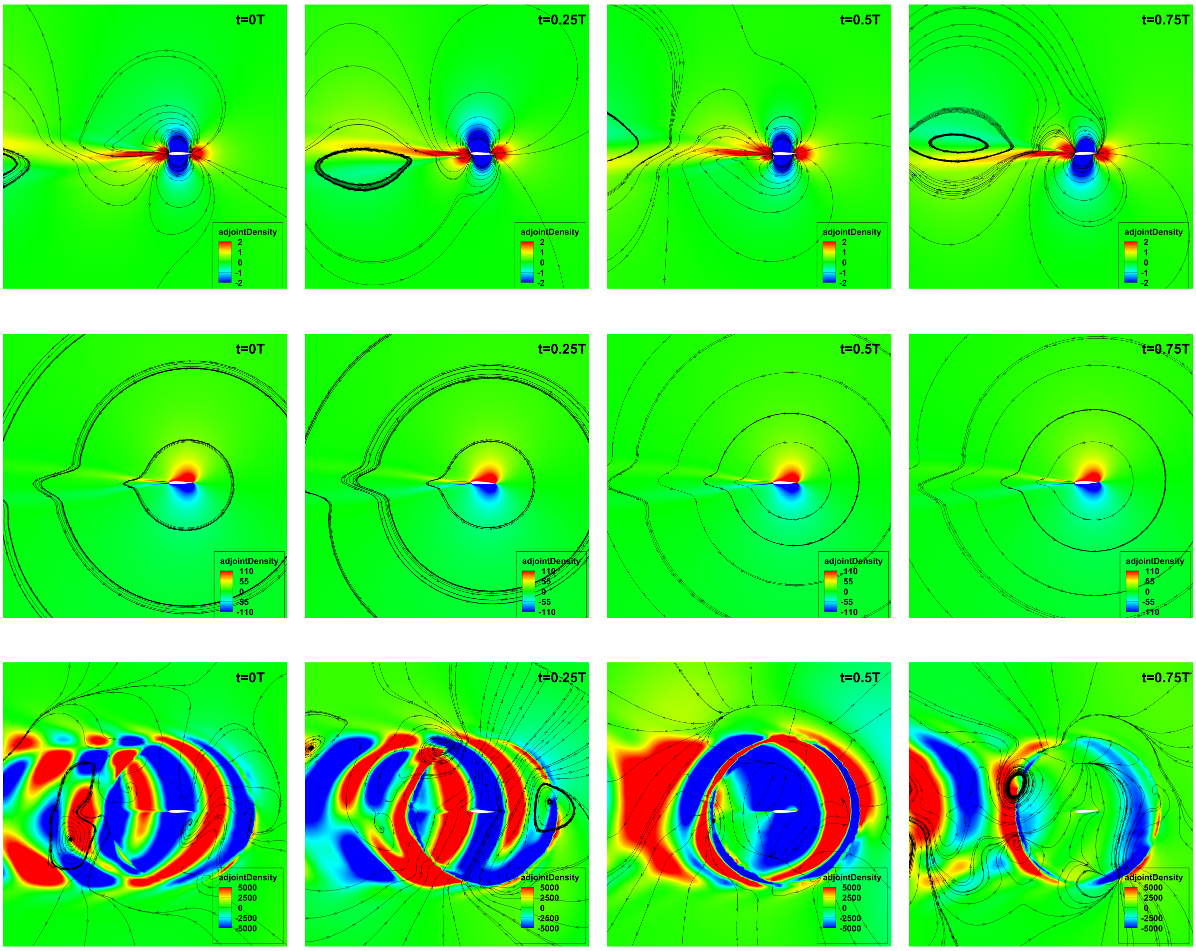


Figure 6.14: *Gradient Verification of a Pitching NACA12 Airfoil - Turbulent Flow*: Iso-areas of adjoint density together with the streamlines of adjoint velocities for the (top row) time–averaged drag, (middle row) time–averaged drag and (bottom row) aeroacoustic objective functions at different time instants within a period of pitching.

6.4 Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow

This test case is concerned with a 2D isolated plunging NACA 64A010 airfoil in transonic flow. The airfoil is plunging with an amplitude equal to 5% of the chord length ($C = 1m$) and period equal to 0.2 seconds. The Reynolds number of the case is $Re = 4.9 \times 10^6$ and the free-stream Mach number is $M = 0.8$ with a zero angle of attack. A hybrid grid is used with a number of structured layers close to the airfoil walls, followed by an unstructured mesh extending up to the circular far-field, located at a $50C$ distance from the airfoil's mid-chord. The 2D mesh, shown in Fig. 6.15, includes almost $60K$ nodes overall, among which 302 nodes are on the airfoil contour and 151 nodes on the FW-H surface at radius $R = 3C$. The Spalart-Allmaras turbulence model is used and the wall distance of the first node off the wall is below 1 ($y^+ < 1$).

The simulation computes 50 time steps per period of pitching (T) with 500 pseudo-time steps within each time step. This is sufficient for the primal and adjoint residuals to drop by at least 6 orders of magnitude at each time. The numerical solution of a period takes ~ 14 min on a single NVIDIA Tesla P100 GPU. After the first 4 periods that the transient phase is passed, the instantaneous flow fields during the 5^{th} period are stored to be used for computing the monopole and dipole source terms of the FW-H analogy and unsteady adjoint solution. The adjoint code runs for 4 periods backwards in time to reach a periodic state. Figure 6.16 illustrates the iso-Mach number contours around the baseline airfoil at different time instants. It shows two shocks (one over each side of the airfoil) located between the mid-chord and the trailing edge.

In this case, both EA and the developed continuous adjoint to the hybrid solver are used to perform (aerodynamic and) aeroacoustic shape optimizations. The former optimization is performed using the Evolutionary Algorithm software EASY developed by PCOpt/NTUA [3].

The airfoil shape is parameterized using a Free-Form Deformation (FFD) box comprising 5×8 CPs, among which 24 are free to move in the y direction during the optimization, Fig. 6.17. The leading and training edges are kept fixed being associated with still CPs. This parameterization remains the same for both optimization methods (EA and adjoint). The linear spring analogy method is used for grid displacement, being confined inside a region with $R = 2.5C$, due to either the plunging motion or shape optimization.

6.4.1 Comparison Between the Hybrid Method and URANS

Since the accuracy of the acoustic analogy depends on the accuracy of flow data on the FW-H surface, in Fig. 6.18, the directivity plots of the pure CFD results

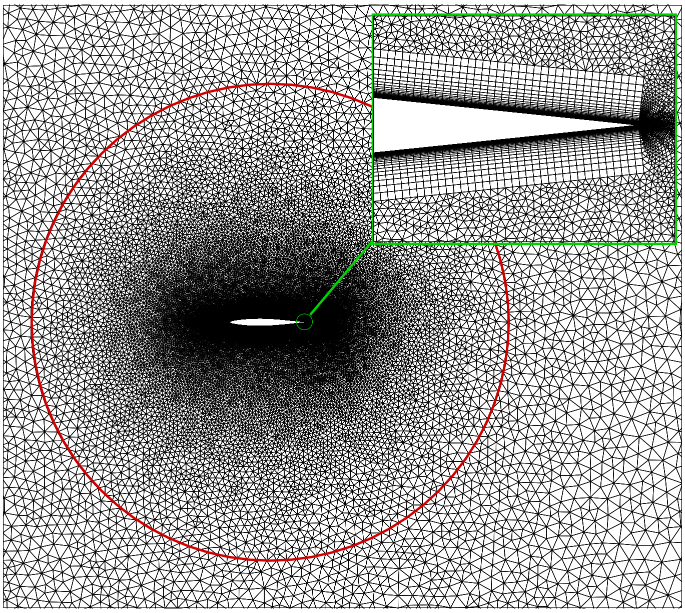


Figure 6.15: *Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: View of part of the computational grid along with the FW-H surface (shown in red, located at $R=3C$ from the airfoil's mid-chord) and close-up view of the grid close to the trailing edge.

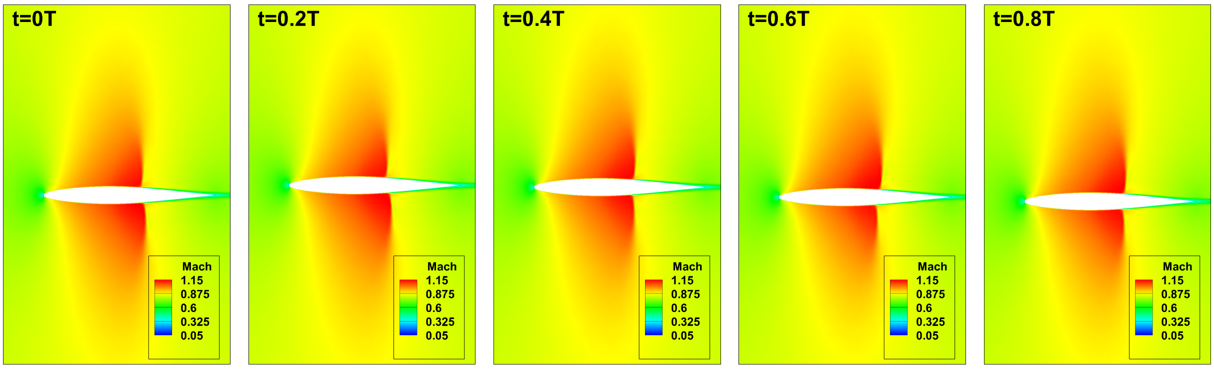


Figure 6.16: *Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Isoareas of Mach number around the baseline airfoil at different time instants within a period of plunging.

and those obtained by the hybrid method at the area relatively close to the FW-H surface, $R = 6C$ and $R = 9C$, are compared. The directivity patterns show a very good agreement between the two methods at both radii.

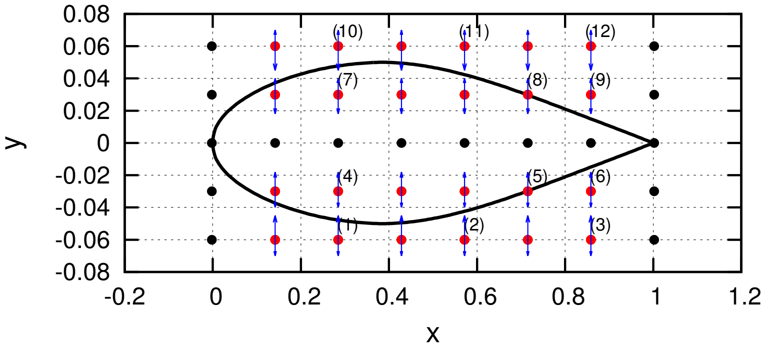


Figure 6.17: *Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: CPs of the FFD box. Only the red CPs are allowed to move during the optimization and each control point can be displaced as determined by the blue arrows. To reduce the cost, the FD-based computation of gradients was performed only for the labeled CPs. (x, y) axes not in scale.

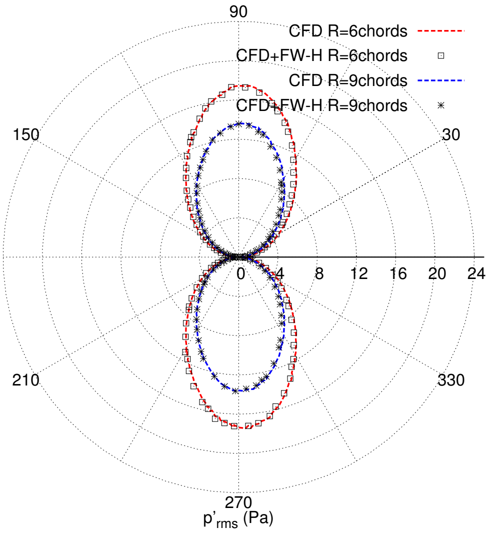


Figure 6.18: *Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Comparison of the directivity plot of p'_{rms} between the hybrid method and that computed by post-processing the CFD results, at $R\!=\!6C$ and $R\!=\!9C$.

6.4.2 EA–Based Shape Optimization

The most common way to increase the efficiency of EA is the use of surrogate evaluation models or metamodels, which gives rise to the MAEA. [16, 68]. However, the high number of design variables increases the metamodel(s) training cost and deteriorates their prediction ability. This is known as curse of dimensionality. This problem can be tackled through dimensionality reduction techniques, such as the PCA of appropriate, dynamically-updated, individual sets recomputed regularly during the evolution. The PCA can be used during the LCPE phase of MAEA [41].

Based on above, the software EASY is used to run the EA-based shape optimization. For the previously described case, a two-objective problem was solved for min. noise (expressed by Eq. 4.4 for a receiver located at $\vec{x}_r = (1C,\ 30C)$) and max. time-averaged lift. A $(10, 18)$ PCA-driven MAEA is used. On-line trained metamodels are activated after the first 35 evaluated solutions are stored in the data base. During the LCPE phase of the MAEA, only the 4 top individuals, according to the metamodel, are re-evaluated. The metamodel training for each individual uses 16 inputs after the PCA-driven truncation of the design space.

For the sake of a fair comparison, it was decided to keep the computational cost of the EA run comparable to that of the adjoint-based optimization. Thus, having first conducted the adjoint optimization, it was decided to keep the computational budget of the EA-based optimization run to 200 evaluations. However, such a very low-budget run was not expected to give a converged front of non-dominated solutions, i.e. the Pareto front, and for this reason, the EA run was finally stopped after performing 450 evaluations. This is why two fronts are shown in Fig. 6.19. The noise is normalized by the corresponding value of the baseline geometry; in contrast, lift cannot be normalized in this way as the initial lift is zero due to the symmetric baseline shape. All non-dominated solutions are characterized by improved performance in terms of noise and lift. No need to say that the run terminated at 450 evaluations provided a better front that dominates that obtained after 200 evaluations. For the latter, maximum noise mitigation occurs for solution *C* having nose less than 30% of the baseline whereas a great increase in the lift is observed for solution *A*. After 450 evaluations, the EA reaches almost the same noise reduction as with 200 evaluations; however, the lift performance of the optimal solutions is greatly improved.

Solutions corresponding to max. lift (*A*), min. noise (*C*) and a selected mid-point (*B*) after 200 evaluations are further examined. Table 6.1, shows that, compared to the baseline airfoil, the time-averaged drag coefficients of the optimized ones are at least 2.5 times bigger, with the biggest increase resulted from airfoil *B*. This is inline with other studies for noise reduction of a blunt trailing edge in the subsonic flow [115], and a pitching airfoil in transonic flow regime [153], that the drag and noise are reported as competing objectives. As shown in Fig. 6.20, the shapes of these three airfoils altered in a similar way, becoming thinner close to the leading edge and thicker close to mid-chord, while the trailing edge area is shifted upward. The leading edge and mid-chord of the solution yielding the lowest noise is slightly thicker than the other two. As shown in Fig. 6.21a, the amplitudes of pressure fluctuations resulted from these three airfoils are reduced compared to the baseline one, with the biggest reduction achieved by airfoil *C*. This can also be seen by comparing the directivity plots at $R = 9C$, Fig. 6.21b,

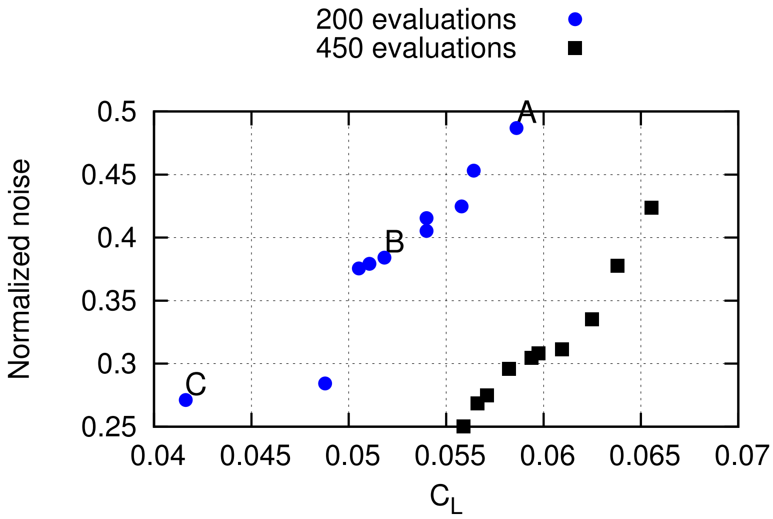


Figure 6.19: *EA-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow:* Front of non–dominated solutions after 200 and 450 evaluations. *A* and *C* are the solutions on the first front with max. lift and min. noise, respectively.

according to which airfoil *C* has the lowest p'_{rms}. The noise reduction can also be seen in Fig. 6.22 for airfoil *C* where pressure fluctuations are compared to the baseline geometry at different time steps. As seen, the amplitude of fluctuations in the optimized geometry is lower compared to the baseline.

Table 6.1: *EA-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow:* Drag coefficients of the baseline and optimized airfoil.

Airfoil	Baseline	A	B	C
C_d	3.5×10^{-3}	9.7×10^{-3}	10.3×10^{-3}	10.1×10^{-3}

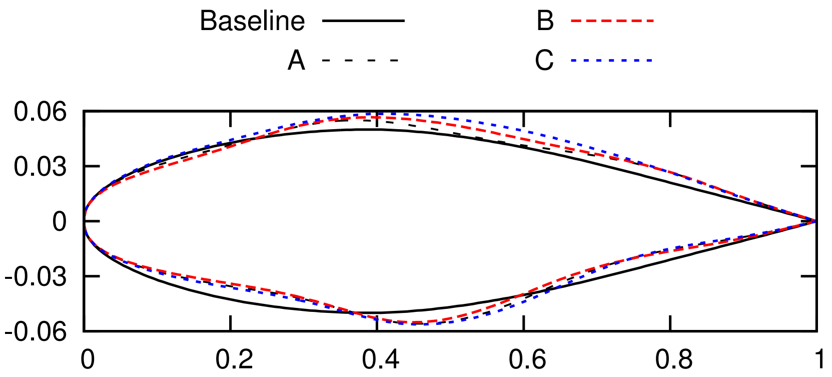


Figure 6.20: *EA-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow:* Comparison of the baseline and three optimized airfoils (*A*, *B* and *C*)

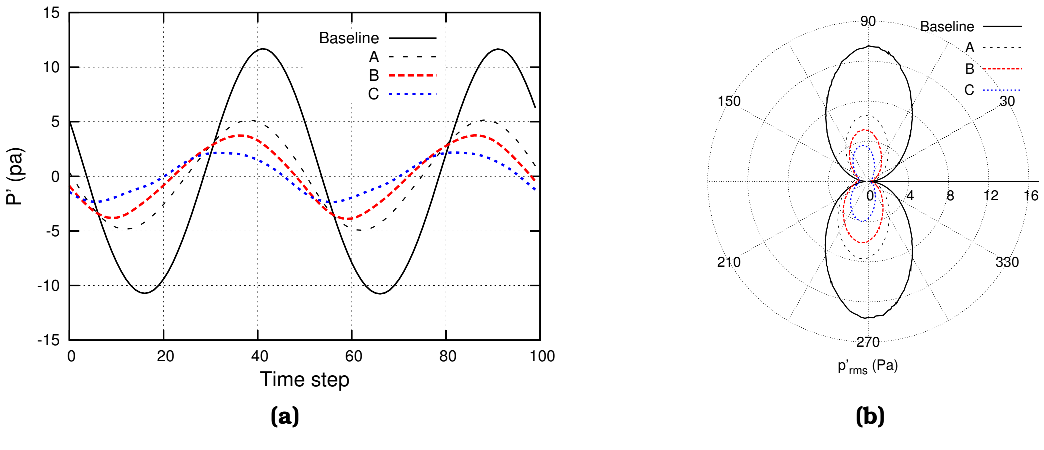


Figure 6.21: *EA-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow:* (a) Time–history of the pressure fluctuation at $\vec{x}_r = (1C,\ 30C)$. (b) Directivity plots of p'_{rms} at $R = 9C$.

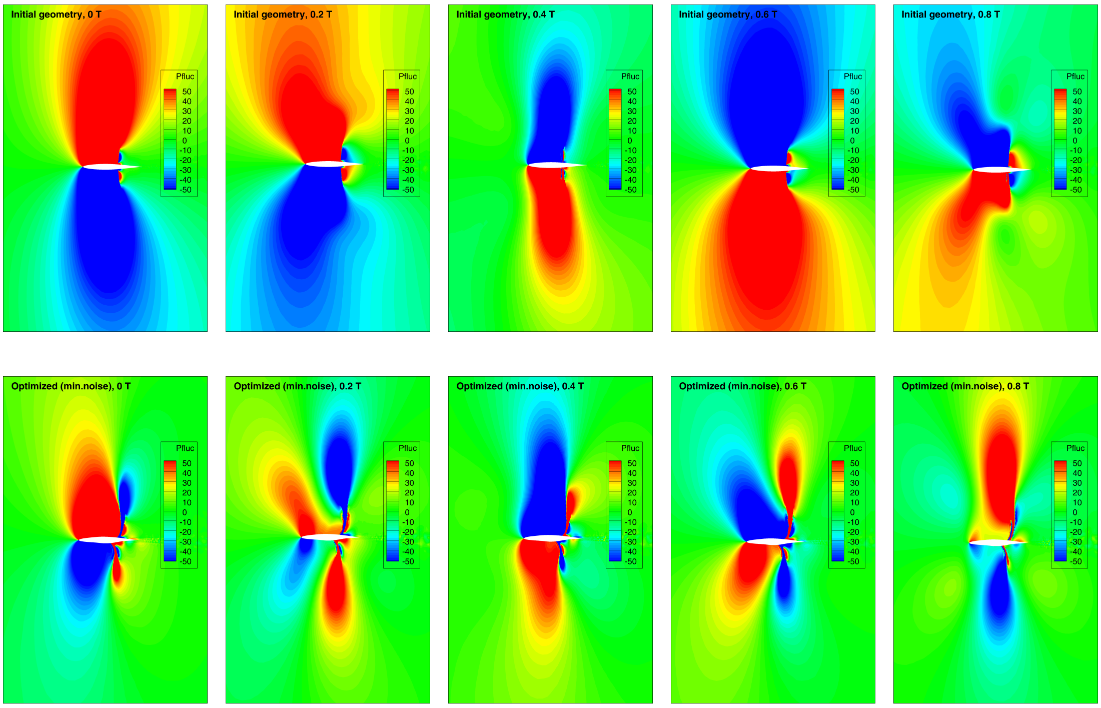


Figure 6.22: *EA-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow:* Iso-areas of pressure fluctuations around the baseline (top row) and airfoil *C* (bottom row) at a number of time instants within a period.

6.4.3 Adjoint–Based Shape Optimization Results

The same problem is also solved using a gradient–based method supported by the continuous adjoint to the hybrid solver. The baseline geometry, parameterization, receiver location and case set–up are as in the previous section. Figure 6.23 depicts the contour of the adjoint density at time instants $0.2\,T$ and $0.6\,T$ around the baseline airfoil. A strong gradient of adjoint density can be seen in the vicinity of the FW-H surface, where the sources to the adjoint equations are defined.

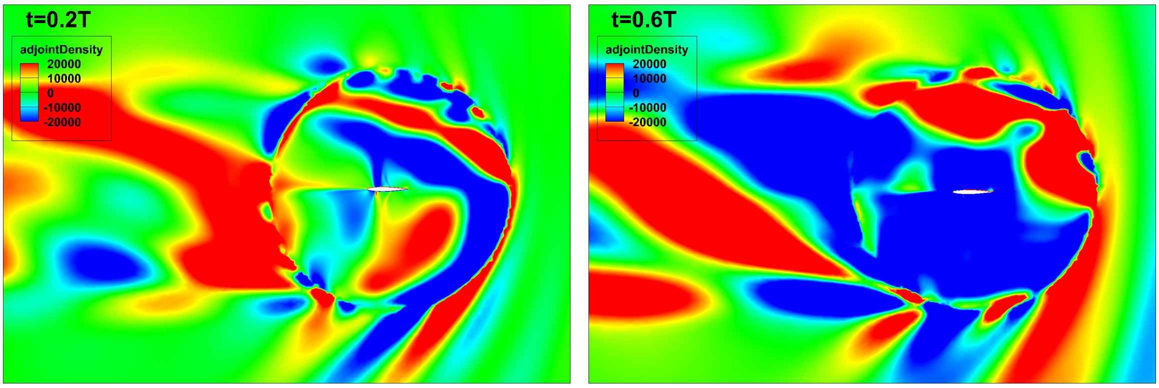
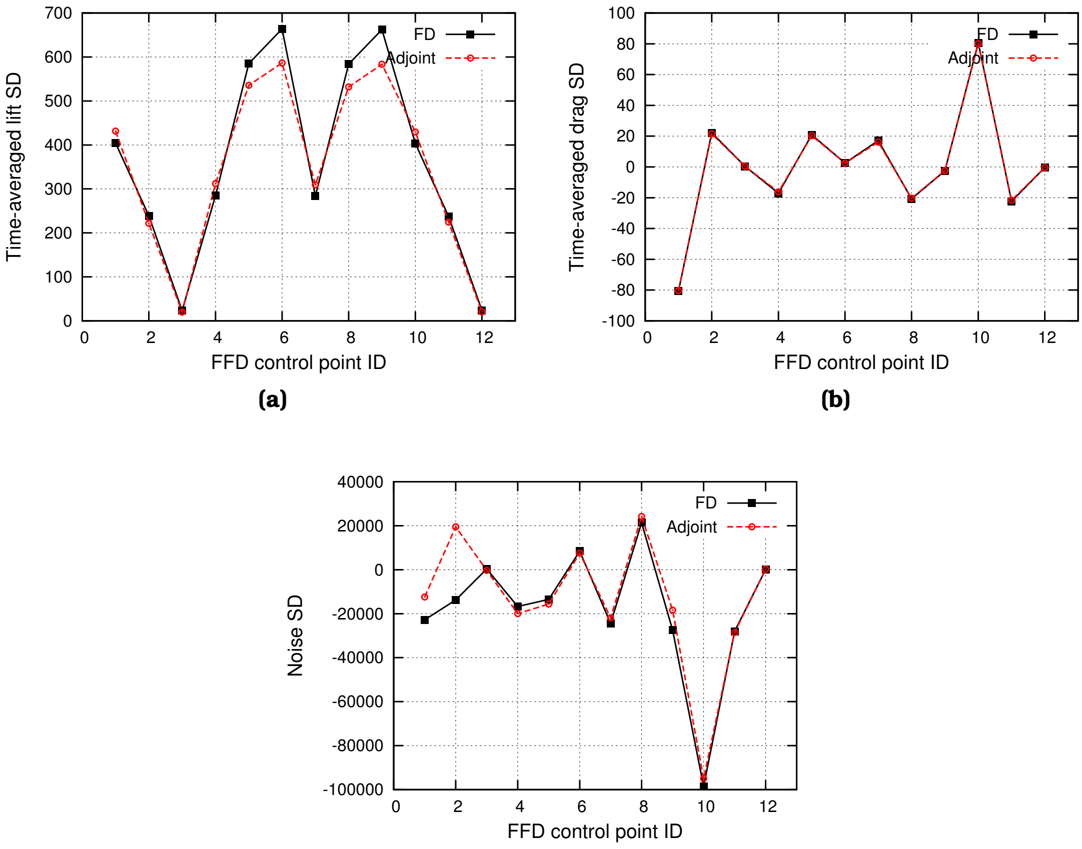


Figure 6.23: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Iso–areas of adjoint density around the baseline airfoil at two different time instants within a period of plunging.

In order to assess the accuracy of the adjoint solver, adjoint sensitivities are compared with FDs. In order to reduce the computational cost, FDs are performed only for the labeled CPs in Fig. 6.17, with a step size equal to $10^{-7}C$. The flow solver is initialized using the same stored flow field for both adjoint and FDs. As shown in 6.24, adjoint and FDs satisfactorily match each other for the time–averaged lift and noise objective functions while the agreement is excellent for the time–averaged drag. Part of the discrepancies in noise sensitivities can be due to the inaccuracies in FDs due to round–off error [152].

Next, the above verified adjoint solver is used for aeroacoustic and aerodynamic shape optimizations. The weighted sum of the noise, Eq. 4.4, and time–averaged lift, with different weights, is the objective function; hence, the objective is defined as $\mathcal{W}_{\text{noise}}\textit{noise} - \mathcal{W}_{\text{lift}}\textit{lift}$, where $\mathcal{W}_{\text{noise}}$ and $\mathcal{W}_{\text{lift}}$ are weight factors, given in table 6.2. According to this, case *I* corresponds to a pure noise whereas case *V* to a pure lift optimization. The minus sign of $\mathcal{W}_{\text{lift}}$ is due to the fact that lift is to be maximized while running a minimization problem.

The convergence of the objective function, illustrated in Fig. 6.25, shows that in case *I*, the objective value of the pure noise optimization is reduced by more than 80% after 20 cycles and, in case *V*, the time–averaged lift is considerably improved with the same computational cost. All five optimizations run for 20 cycles, as this



(a)

(b)

(c)

Figure 6.24: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Comparison of the adjoint and FD sensitivities for (a) time-averaged lift, (b) time-averaged drag and (c) noise perceived by a receiver located at $\vec{x}_r = (1C,\ 30C)$.

Table 6.2: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Weight factors of noise and lift used in the objective function.

Case name	*I*	*II*	*III*	*IV*	*V*
$\mathcal{W}_{\text{noise}}$	1	0.75	0.5	0.25	0
$\mathcal{W}_{\text{lift}}$	0	0.25	0.5	0.75	1

is enough for the objectives to converge, and the obtained solutions are compared with those obtained by the MAEA in Figure 6.26. Since the computational cost of adjoint and primal are almost the same, the total computational budget spent by the adjoint optimization is ~ 260 primal solutions (this computational cost could be lowered as in almost the last 5 cycles the objective values are already converged). A comparison with the one obtained by the EA after 200 evaluations shows that, at a similar noise (or lift) level, the adjoint-based optimization offers an improved lift (or noise) compared to the EA (comparison based on the same computational budget). For example, airfoil *I* has a lower noise level compared to airfoil *C* of the EA-based optimization with a slightly better lift. Another example is airfoil *II* which has the same lift coefficient (~ 0.053) as the airfoil *B* while its noise level is significantly lower. It can also be noticed that the trend of adjoint results are closer to those obtained by EA after 450, rather than 200, evaluations. One may easily notice that the lowest noise was achieved by one of the adjoint runs. The time-averaged drag coefficients are given in table 6.3. As seen, similar to the EA-based results, all optimized airfoils are marked with increased drag, with the biggest rise occurred in the airfoil corresponding to min. noise (*I*).

Figure 6.27 (top) compares the baseline and optimized airfoils *I*, *III* and *V*. The trailing edge area of the optimized airfoils are shifted upward and at mid-chord they became thicker. It can also be seen that the leading edge of the airfoil having max. lift is thicker than the other two. It should be noted that some control points reach their maximum bound. In addition, this figure, makes a comparison between the optimized shapes obtained by the adjoint and EAs. It compares the two airfoils corresponding to min. noise (*I* from the adjoint and the *C* from EA) (middle) and, also, airfoils *II* and *B* from adjoint and EA, respectively (bottom), which have comparable lift value.

As expected, the reduction in the objective value results in a lower amplitude in pressure fluctuations, Fig. 6.28. Comparison of the directivity plot of the baseline and optimized airfoils at $R = 9C$, Fig. 6.29, shows that noise reduction occurs in almost all directions, except for airfoil *V* between angle $180°$ and $230°$. As seen, unlike the baseline geometry, the directivity plot is not symmetric in the optimized geometries as airfoil shapes are no more symmetric.

Table 6.3: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Drag coefficients of the baseline and optimized airfoil.

Airfoil	Baseline	*I*	*II*	*III*	*IV*	*V*
C_d	3.5×10^{-3}	11.5×10^{-3}	11.02×10^{-3}	10.2×10^{-3}	9.2×10^{-3}	9.03×10^{-3}

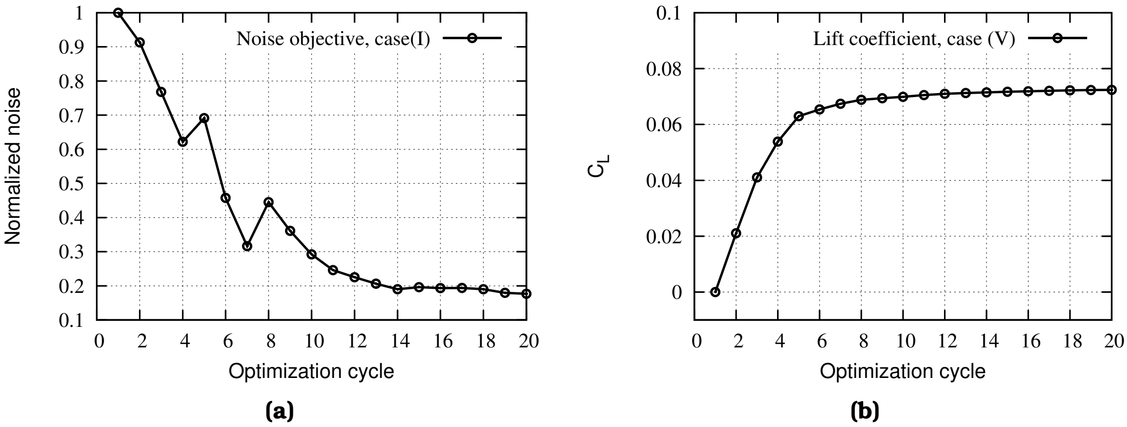


Figure 6.25: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: (a) Convergence of the noise objective value (case *I*) for a receiver located at $\vec{x}_r = (1C,\ 30C)$ normalized by its initial value. (b) Convergence of the time-averaged lift coefficient during the lift-only optimization (case *V*).

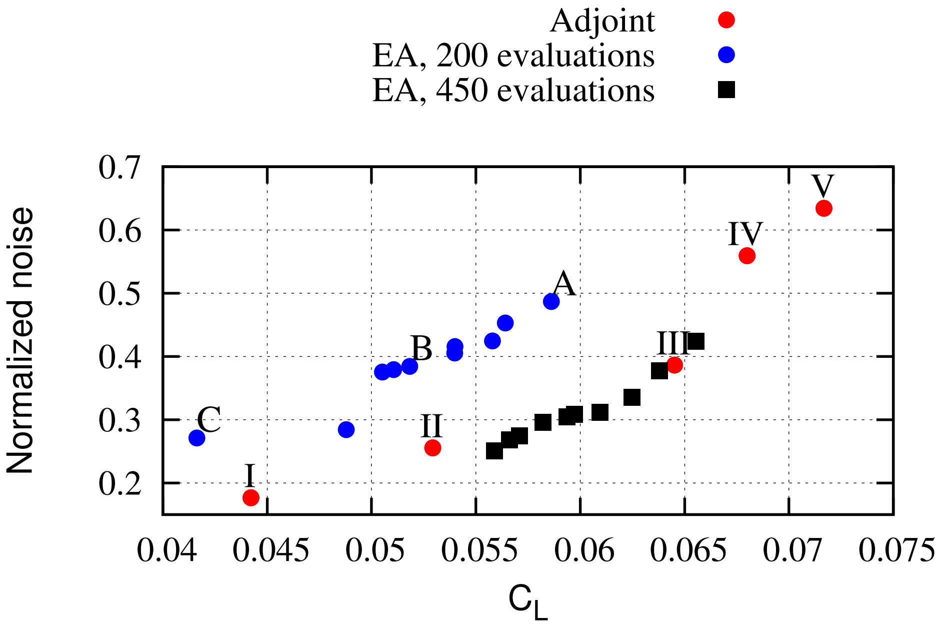


Figure 6.26: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Front of non-dominated solutions after 20 optimization cycles per case (five runs, red circles) compared to those obtained by EA after 200 (blue circles) and 450 evaluations (black squares).

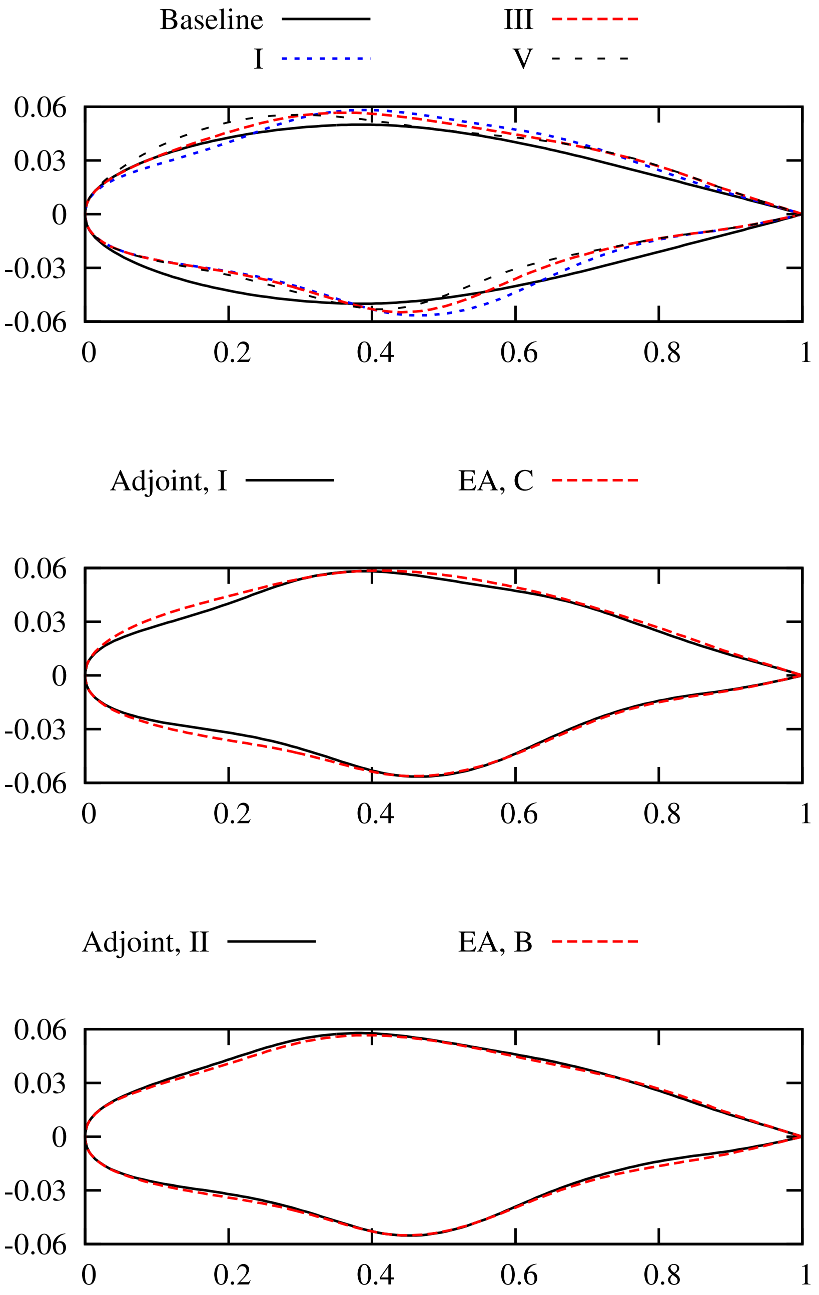


Figure 6.27: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: (top) Comparison of the shape of the baseline and three optimized airfoils. Comparison of some of the optimized geometries resulted from adjoint (black line) and EA (red dashed line). (middle) Airfoils *I* and *C*. (bottom) Airfoils *II* and *B*.

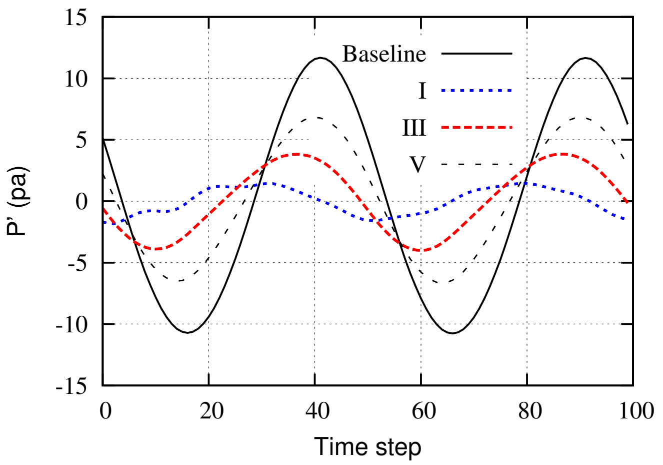


Figure 6.28: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Comparison of the time-history of the pressure fluctuation at $\vec{x}_r = (1C,\ 30C)$ between the initial and optimized geometries.

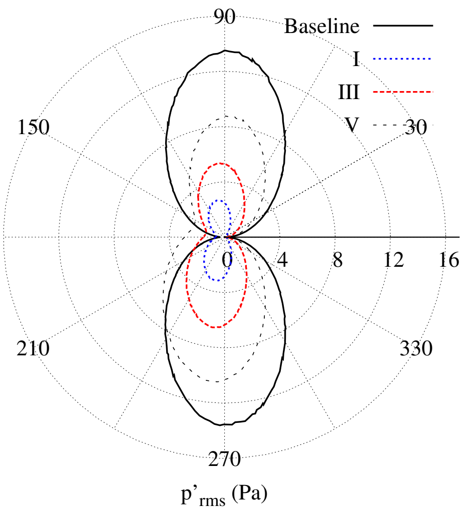


Figure 6.29: *Adjoint-Based Shape Optimization of a Plunging NACA64A010 Airfoil - Transonic Flow*: Comparison of directivity plots of p'_{rms} at $R = 9C$ between the baseline and optimized geometries.

6.5 Optimization of a Vortex Shedding Cylinder - Laminar Flow

A 2D cylinder in a laminar flow undergoing vortex shedding is studied. The cylinder's diameter is $D = 0.019m$ and the free–stream Mach number is $M = 0.2$ with zero angle of attack. This corresponds to the Reynolds number of $Re = 1000$. The O-type grid (generated as structured, treated as unstructured), shown in Fig. 6.30, contains 49280 nodes among which 308 are located on the cylinder and 160 in the radial direction. The grid extends $35D$ from the center and the FW-H surface is located at the radius of $R = 3.5D$ containing 308 nodes. The time step size for the unsteady solution was set to 0.00122 seconds which includes almost 50 time steps per period of shedding. 500 pseudo–time steps are performed per time step of both the primal and adjoint solution, which is sufficient for the residuals to drop at least by 15 orders of magnitude at each time step. The computed Strouhal number based on the period of lift variation is $St = 0.24$ and the mean drag coefficient is $\bar{C}_d = 1.42$. These numbers are close to 0.234 and 1.56 of the URANS simulation presented in [22]. After the flow reaches periodic state, the 290 next time steps are used to perform the FW-H integration.

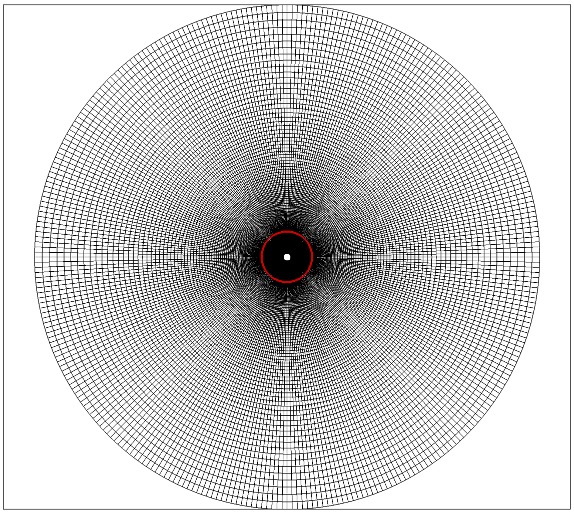


Figure 6.30: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: O-type grid around the cylinder and the location of the FW-H surface at $R = 3.5D$ shown in red.

6.5.1 Comparison Between the Hybrid Method and URANS

In order to verify the accuracy of the implementation of the FW-H analogy for this case, the outcome of the hybrid solver is compared with the results of a URANS simulation for two receivers located at $\vec{x}_{r1} = (-0.4D, 6.97D)$ and $\vec{x}_{r2} = (-3.15D, -6.23D)$, shown in Figs. 6.31a and 6.31b, respectively. The two methods are in good agreement and the discrepancies at the beginning and ending parts are due to the application of a Hanning window.

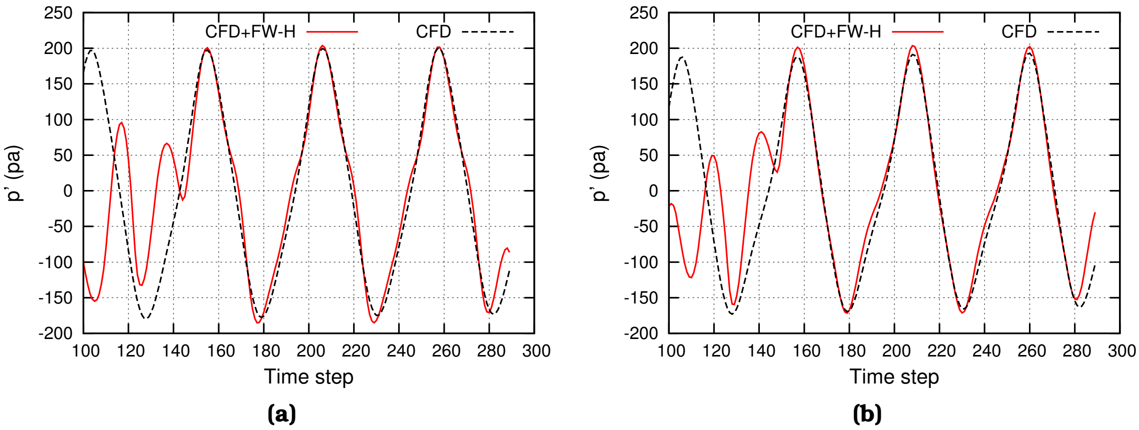


Figure 6.31: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Comparison of the time history of the pressure fluctuation computed using pure CFD and hybrid method for a receiver located at (a) $\vec{x}_{r1} = (-0.4, 6.97)$. (b) $\vec{x}_{r2} = (-3.15, -6.23)$.

6.5.2 Adjoint–Based Optimization Results

In contrast to the cases studied so far, here the exact period of the problem is not known a priori. Therefore, storing the flow field over a single period of the shedding, so as to integrate the unsteady adjoint equations repetitively over this period until it reaches the periodic state is not possible. Thus, in this case, the unsteady adjoint solution must be performed over a longer time window.

Each flow solution, during FDs or the adjoint optimization, is initialized using the same field in a fully periodic state, stored for 600 time steps. The unsteady adjoint equations are integrated backward in time during this time window. The numerical solution of either the primal or adjoint problem takes $\sim 87min$ on a single NVIDIA Tesla P100 GPU. The aerodynamic and aeroacoustic objective functions are defined over a specific time range, known as the optimization time window, outside of which there are no contributions to the adjoint equations. For all objective functions, the optimization time window starts at time step 300 and finishes at 590. The aeroacoustic objective function is defined for a receiver located at $\vec{x}_o = (-4D, -20D)$. The shape of the upper and lower half of the cylinder is parameterized using two Bezier curves with 7 CPs each; these are free to move only in the y direction during the optimization, Fig. 6.32. Since the first and last CPs are fixed, the problem includes 10 design variables in total.

Gradients of aerodynamic and aeroacoustic objective functions computed using the adjoint solver are compared with the outcome of FDs in Fig. 6.33. It shows that the continuous adjoint solver well-duplicates FDs results for the time-averaged lift (Fig. 6.33a) and drag (Fig. 6.33b) and satisfactorily matches FDs for the noise objective function (Figs. 6.33c). Figure 6.34 depicts the contours of density and adjoint density resulted from the noise objective function at time steps 220 and 251. High values of adjoint density can be seen close to the FW-H surface as well as the top and bottom of the cylinder at time steps 220 (Fig. 6.34b) and 251 (Fig. 6.34d), respectively.

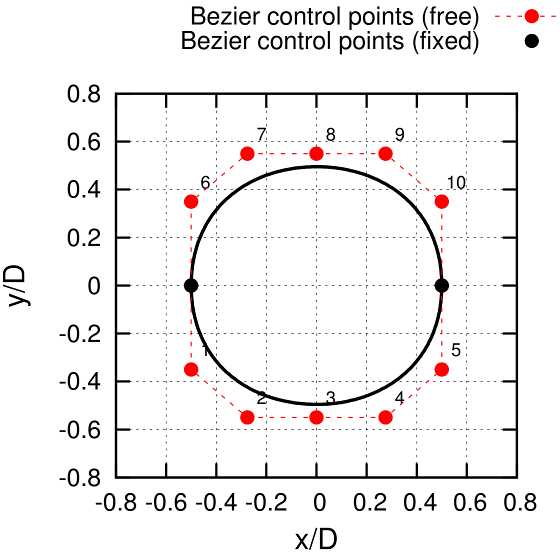
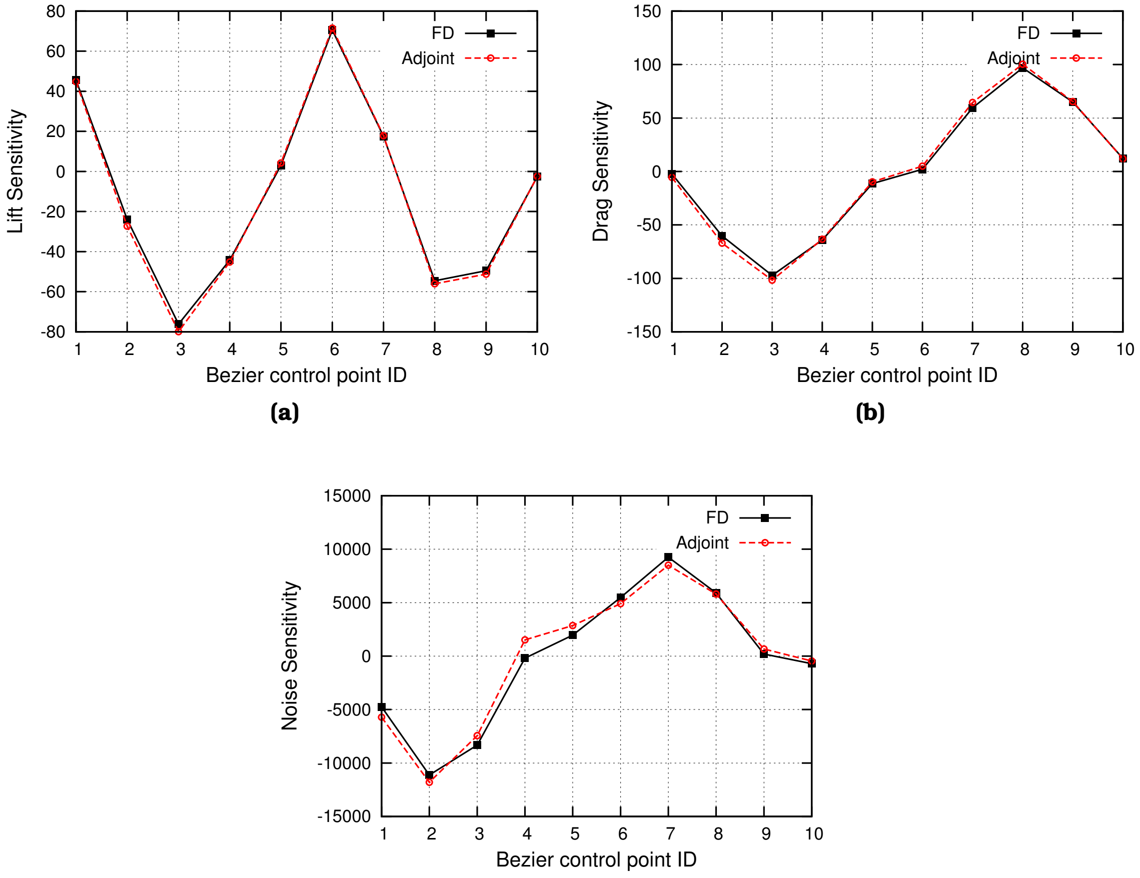


Figure 6.32: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Control points of the Bezier curves.

Next, the above-computed aeroacoustic sensitivities are employed within the adjoint-based shape optimization. The objective function is defined as the average of Eq. 4.4 over 4 receivers located at $\vec{x}_{r1} = (-4D, -20D)$, $\vec{x}_{r2} = (4D, -20D)$, $\vec{x}_{r3} = (-4D, 20D)$ and $\vec{x}_{r4} = (4D, 20D)$. A constraint on the minimum allowed area is imposed during the optimization. The grid displacement is based on the linear spring analogy, performed inside a circular area with $R = 3C$. As illustrated in Fig. 6.35, after 23 design cycles, the area of the optimized shape meets the imposed constraint and, then, stops becoming smaller. The noise objective function has reduced by more than 80%. As expected, the reduction in the objective value results in a lower amplitude in pressure fluctuations for all receivers, as shown in Fig. 6.36. Comparison of the iso-areas of the sound pressure between the initial and optimized geometries in Fig. 6.37 shows that the sound reduction is omni-directional.

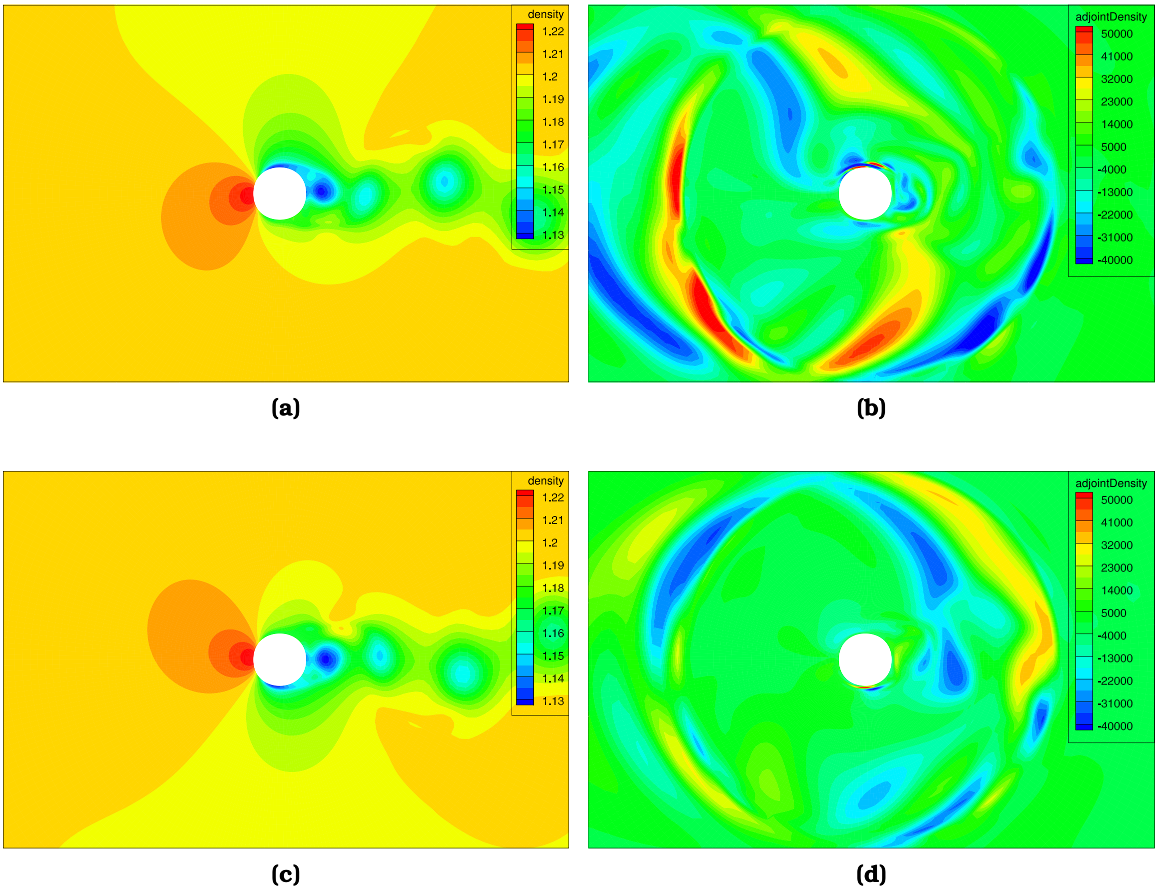


(a)

(b)

(c)

Figure 6.33: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Comparison of the sensitivities using the proposed adjoint method and FDs for (a) time-averaged lift, (b) time-averaged drag and (c) noise objective (Eq. 4.4).



(a)

(b)

(c)

(d)

Figure 6.34: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Contours of flow and adjoint densities. (a) Flow density at time step 220. (b) Adjoint density at time step 220. (c) Flow density at time step 250. (d) Adjoint density at time step 250.

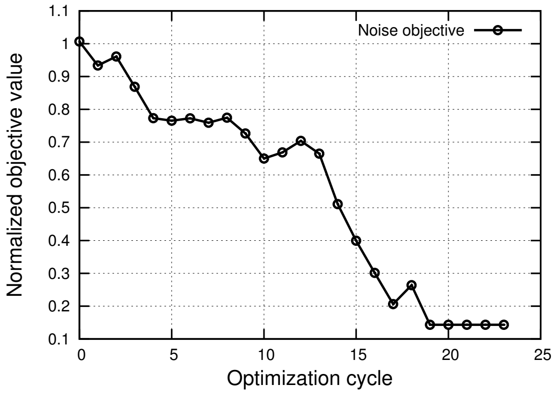


Figure 6.35: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Results after 23 design cycles. Convergence of the noise objective function (J as in Eq. 4.4) normalized by its initial value.

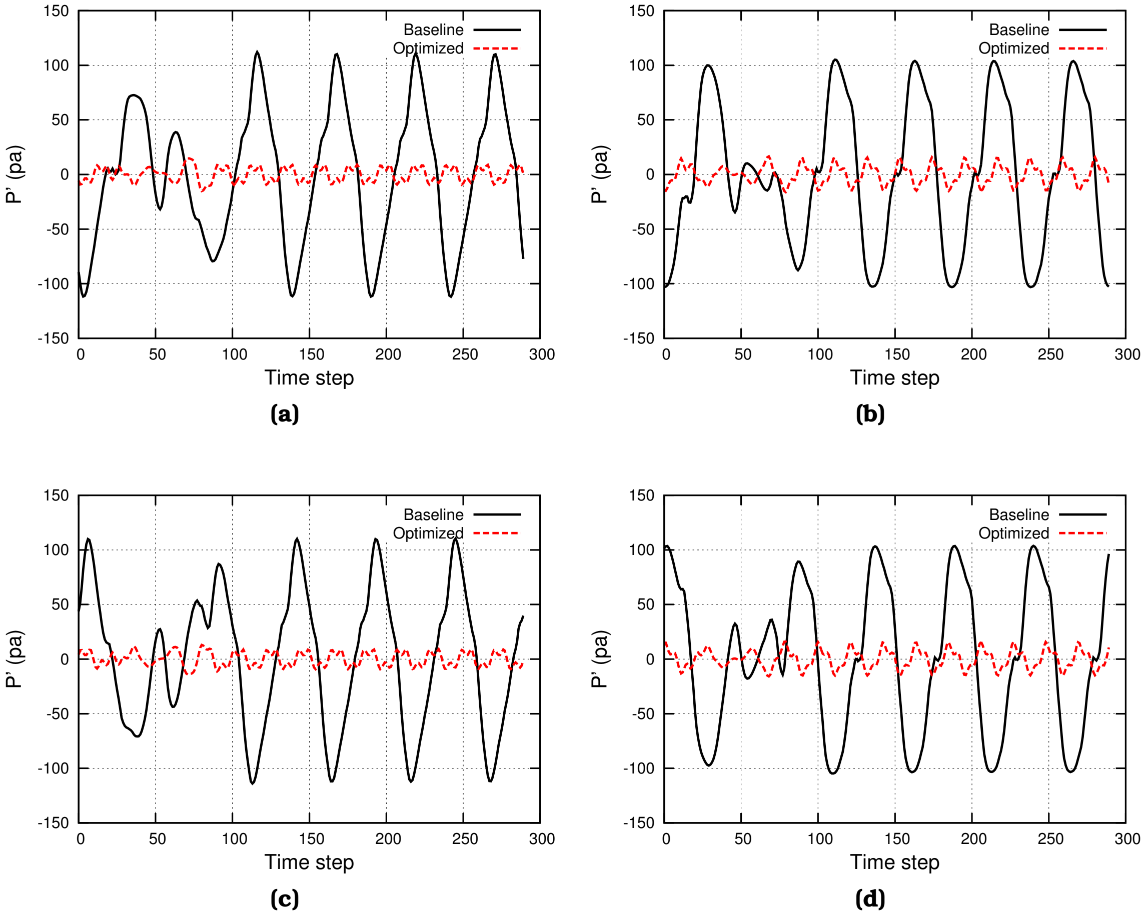


Figure 6.36: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Results after 23 design cycles. Comparison of time history of the pressure fluctuation between the baseline and optimized geometries at the receiver's locations (a) $\vec{x}_{r1} = (-4, -20)$, (b) $\vec{x}_{r2} = (4, -20)$, (c) $\vec{x}_{r3} = (-4, 20)$, (d) $\vec{x}_{r4} = (4, 20)$.

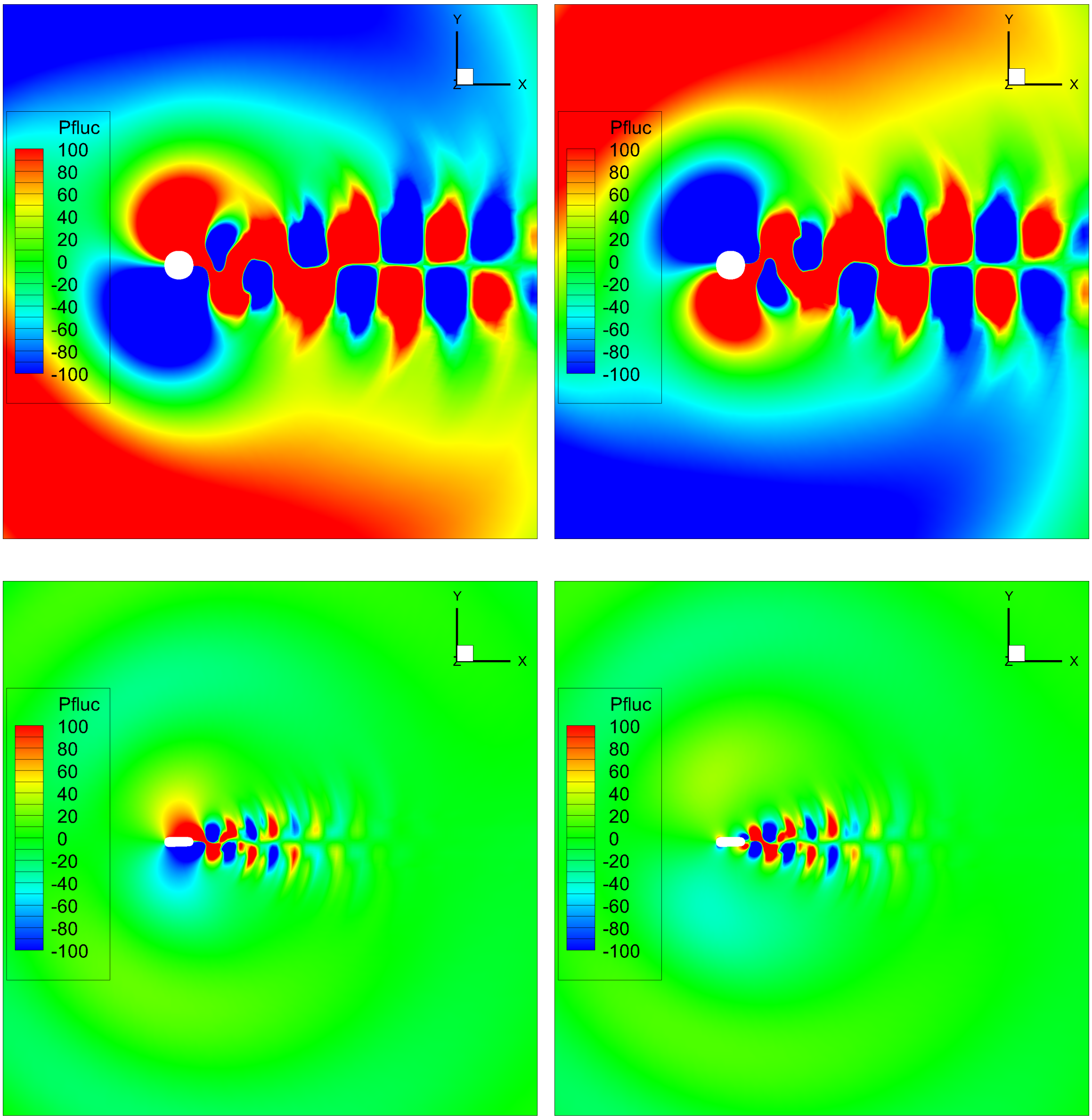


Figure 6.37: *Optimization of a Vortex Shedding Cylinder - Laminar Flow*: Results after 23 design cycles. Comparison of the iso-areas of the pressure fluctuation between the baseline (top row) and optimized geometries (bottom row) at two different time steps 550 (left column) and 575 (right column).

6.6 Optimization of a Rod-Airfoil configuration

The rod-airfoil configuration is selected as a test case not only due to the availability of experimental and numerical results for both the flow and acoustic fields [55] but, also, due to the flow features that make it an ideal test for numerical sound modeling in simplified air-frame components such as a landing gear. The periodic vortex shedding from the rod results to the quasi–tonal noise component while the impingement of vortices on the airfoil contributes to the broadband noise [45].

In this work, the configuration of [55] is used, as shown in Fig. 6.38. It comprises a NACA0012 airfoil with chord length $C = 0.1m$, located one chord downstream of a cylinder with diameter $D = \frac{C}{10}$. Figure 6.38 shows the computational domain and the location of the FW-H surface. The origin of the coordinate system is at the leading edge of the airfoil and the computational domain extends $6C$ upstream of and $7C$ downstream of the origin and, also, extends $\pm 3C$ in the crosswise direction. The grid close to the rod and airfoil is shown in Fig. 6.39. The grid is formed only by quadrilaterals with 161906 grid nodes, among which 256, 414 and 1016 nodes are located on the rod, airfoil and the FW-H surface, respectively. The first nodes off the wall are at the distance of approximately $10^{-4}C$ from the rod and $10^{-5}C$ from the airfoil, resulting to $y^+ < 1$.

The free–stream Mach number and density are $M = 0.2$. The Reynolds number based on the airfoil chord is 4.8×10^5 and the infinite flow angle is 0 deg The time step is 2×10^{-6} seconds and the simulation runs for 20000 time steps to reach a periodic state with 120 iteration within each time step that assures 15 orders of magnitudes drop in residuals. Afterwards, a time window of 2030 time steps is used to compute the FW-H integral as well as time–averaged aerodynamic data. This time window corresponds to about 7 shedding periods, according to the computed Strouhal number.

6.6.1 Verification of the CFD Results

In order to verify the solution of the flow field, the vortex shedding frequency, the mean stream-wise velocities and the mean pressure coefficient are compared with results available in the literature. The computed Strouhal number is 0.24 and agrees with the 2D URANS computation performed by Jacob [55], being different than the experimental Strouhal number of [55] which is about 0.19. According to [55], this overestimation is expected for URANS solutions which cannot accurately capture vortex dynamics. The profiles of the stream–wise component of the flow velocity are compared to the numerical results of Rousoulis [113] and Jacob [55] as well as the outcome of LES and measurements. The velocity profiles computed for two different stream–wise locations, $x/C = -0.25$ and $x/C = 0.25$, are shown in Fig. 6.40. From Fig. 6.40a, at $x/C = -0.25$, the velocity profile from the

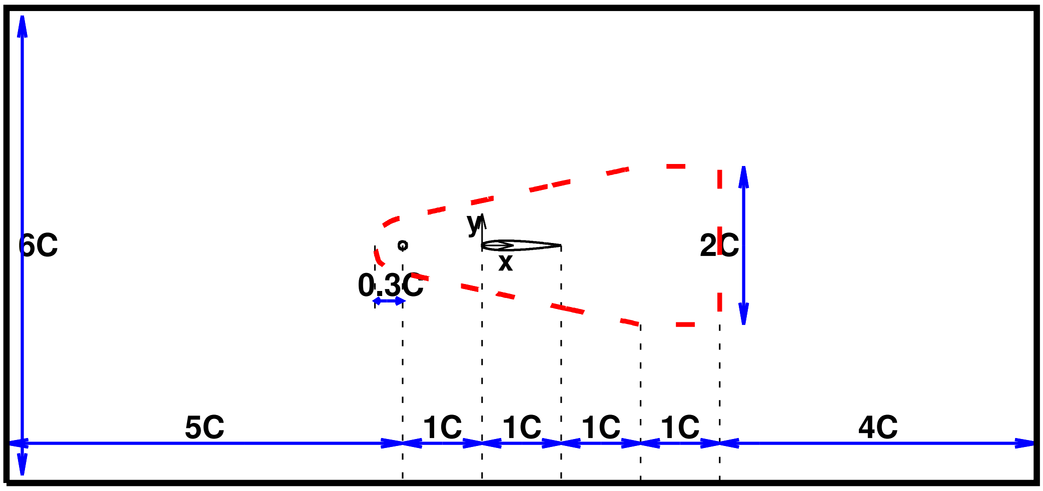


Figure 6.38: *Optimization of a Rod-Airfoil configuration*: Computational domain. The dashed line stands for the FW-H surface.

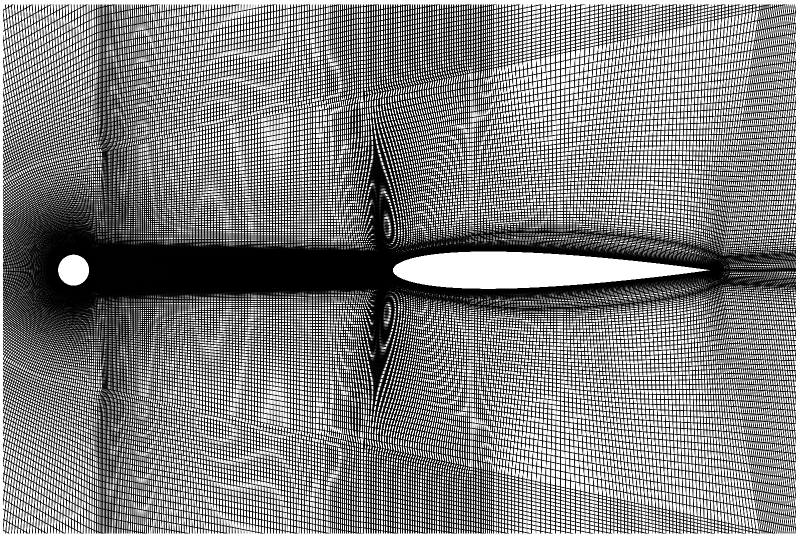


Figure 6.39: *Optimization of a Rod-Airfoil configuration*: CFD grid close to the rod and the airfoil.

current study agrees well with the 2D URANS results of [113], and seems to outperform the outcome of the 2D URANS of [55]. At $x/C = 0.25$ (Fig. 6.40b), a stronger overprediction of the measured distribution can be seen in all URANS simulations, compared to LES. However, the overall trend is adequately captured, and part of the slight disparities in the URANS results might be due to the different turbulence model used in this study (Spalart-Allmaras) compared to the k-ω SST of [113] and [55].

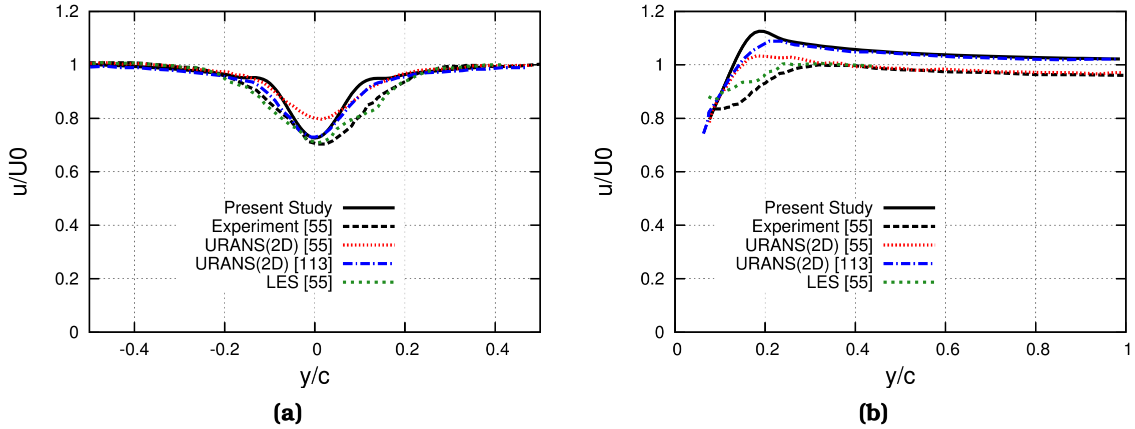


Figure 6.40: *Rod-Airfoil initial configuration.* Comparison of the stream-wise velocity profiles at (a) $x/C = -0.25$, (b) $x/C = 0.25$ with the URANS results of [55, 113], LES and measurements of [55].

Figures 6.41a and 6.41b compare the distribution of the mean and r.m.s. values of the pressure coefficient Cp on the airfoil's pressure side with the 3D LES results of [45] and 2D URANS of [96]. The mean Cp values are also compared with the experiment [54]. It can be seen that the present results are in a good agreement with the mean Cp of the LES while both overestimate the experimental results. Comparison of the r.m.s. Cp with the LES results in Fig. 6.41b shows that the present results are slightly better than those of other 2D URANS solutions. All these comparative studies confirm the acceptable accuracy of this set-up, including the grid, numerical parameters and the flow solver, which will be used in the remaining of this section.

6.6.2 Verification of the Acoustic Results

In order to verify the acoustic noise prediction tool, the outcome of the hybrid solver is compared to pure unsteady CFD as well as the above cited computational and experimental results. Figure 6.42 compares the time history of the pressure fluctuation computed by (a) applying the FW-H analogy and (b) the URANS only, for two receivers located outside, quite close though, to the FW-H surface. The

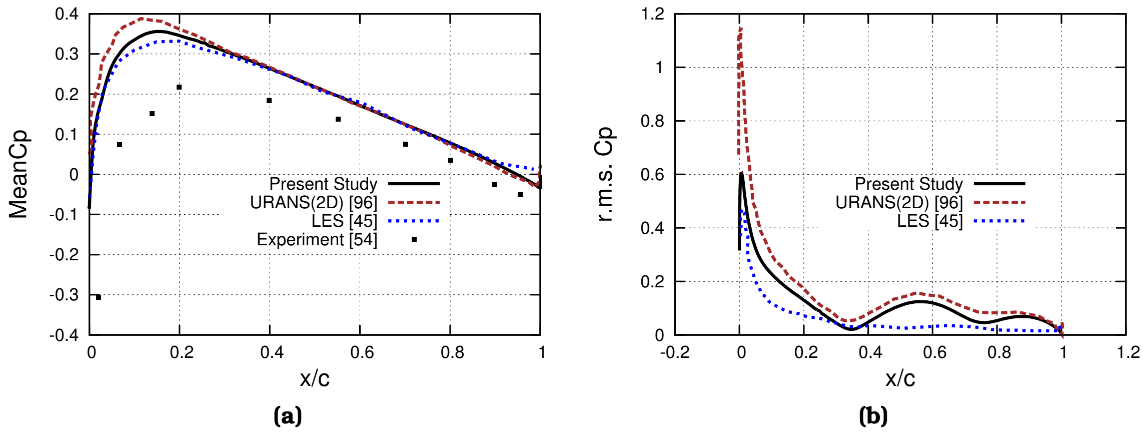


Figure 6.41: *Rod-Airfoil initial configuration.* Comparison of the mean and r.m.s. values of the pressure coefficient Cp on the pressure side of the airfoil with the other CFD results [45, 96] and measurements [54]. (a) Mean Cp. (b) r.m.s. Cp.

receiver locations are $\vec{x}_{r1} = (0.31m, 0.101m)$ and $\vec{x}_{r2} = (-0.009m, 0.17m)$. As shown in Figs. 6.42a and 6.42b, at both receiver locations, results from the two methods are in good agreement. The discrepancies at the beginning and ending part of the signal are expected due to the application of a Hanning window.

The power spectral density (PSD) for a far-field receiver located at $\vec{x}_r = (0.05m, 1.85m)$ is compared with the experimental and computational results of [55] and [113], Fig. 6.43 . As seen in Fig. 6.43, the main peak and corresponding frequency from the current study match the 2D URANS solution of [55] and are close to similar solutions in [113]. Compared to the experiment, the value of the main peak is adequately close, whereas the frequency of the tonal peak ($1728Hz$) is overestimated. As mentioned before, this overprediction of the shedding frequency is expected in any URANS simulation due to the weakness to accurately capture vortex dynamics [55]. Five peaks, corresponding to the first five harmonics, can be identified. In comparison with [113], harmonic frequencies estimated by the current study are in a slightly better agreement with the computational data of [55] whereas PSD peaks are of higher magnitude.

Figure 6.44 compares the directivity plot of the overall sound pressure level (OASPL) from the current study with 2D URANS results and experimental data in [96] and [55]. This figure also presents a grid dependency study. 100 receivers are located at radius $R = 1.85m$ from the mid-chord of the airfoil. It shows that the directivity plot has a quadrupole shape for the coarse grid (124000 nodes) whereas, for the medium size (161906 nodes) and dense (240000 nodes) grids, this tends to a dipole. Since results on the dense and medium grids are almost the same, the latter is used for this study. Figure 6.44, also shows that there is a good agreement between the current study and the 2D URANS results of [96] and that

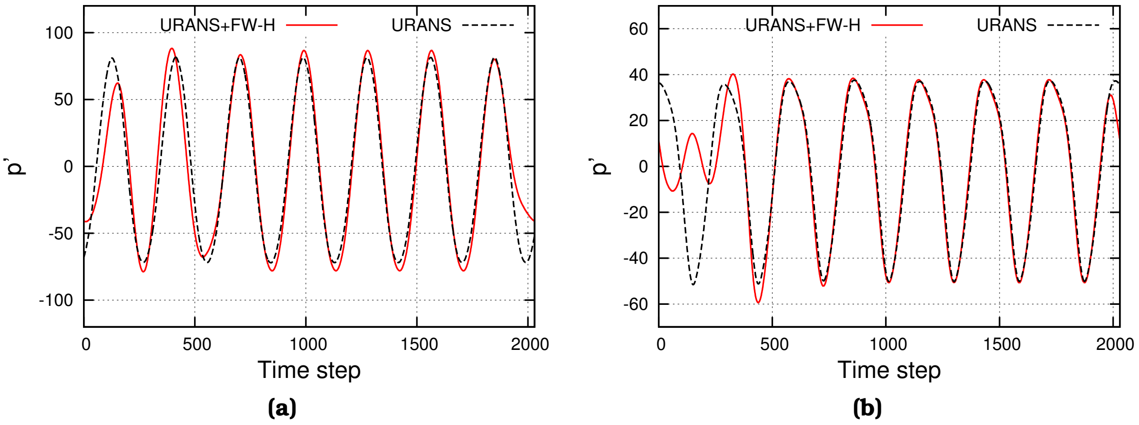


Figure 6.42: *Rod-Airfoil initial configuration.* Comparison of the time history of the pressure fluctuation computed using pure CFD and the hybrid solver at a receiver located at (a) $\vec{x}_{r1} = (0.31m, 0.104m)$, (b) $\vec{x}_{r2} = (-0.009m, 0.17m)$.

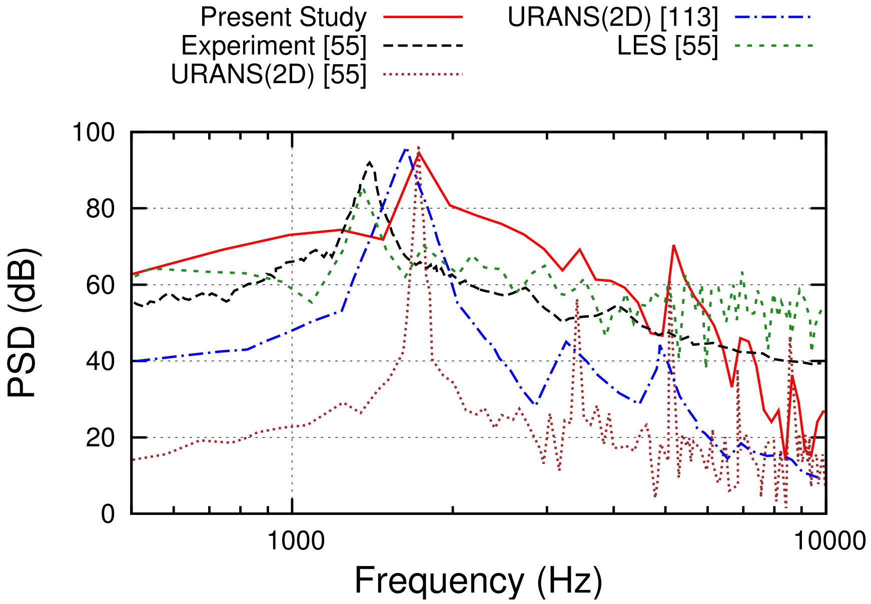


Figure 6.43: *Rod-Airfoil initial configuration.* Comparison of the power spectral density at a receiver located $1.85m$ above the center of the airfoil with other URANS results in [55, 113] and LES as well as measurements of [55].

both follow the trend of the experiment. Overprediction of the OASPL is expected in URANS simulation, see [121, 96].

Overall, despite some differences in the magnitude of the harmonics, the peak tone and the fundamental frequencies are well predicted. Given that such a problem is mainly dominated by the vortex shedding from the rod which governs

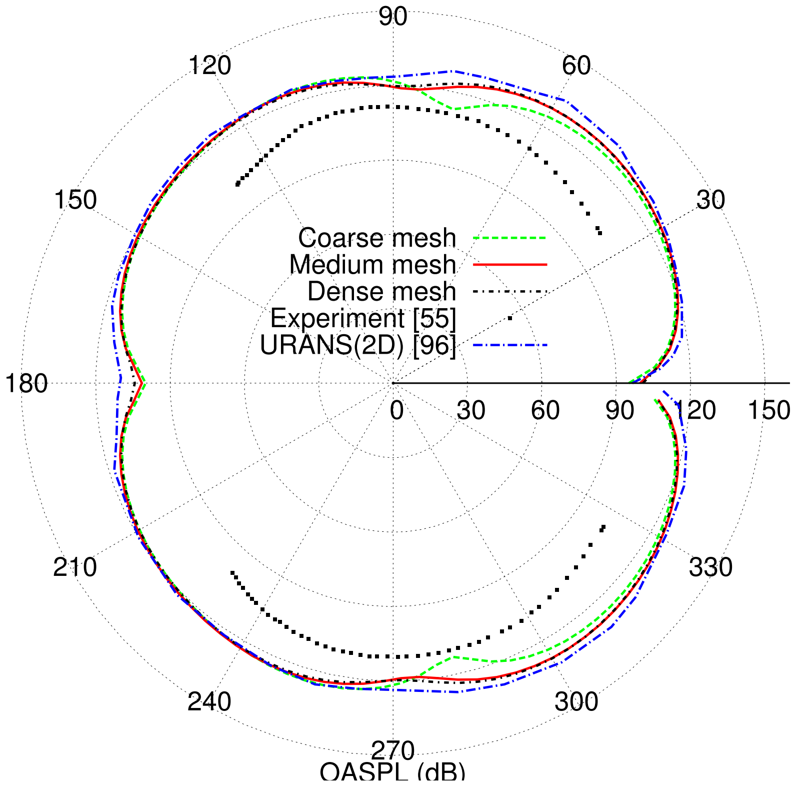


Figure 6.44: *Rod-Airfoil initial configuration.* Directivity of OASPL at $R = 1.85m$. Comparison of the directivity, computed using 3 different grids, with other URANS results in [96] and measurements in [55].

the tonal component of the generated sound, the present results capture the main flow and noise mechanisms accurately enough to be used in the context of an unsteady aeroacoustic shape optimization.

Adjoint–based aeroacoustic optimization is performed for two different objective functions, namely case A and case B. In case A, the aeroacoustic objective function, Eq. 4.4 , is integrated over all the resolved frequencies, while in case B it includes only the tonal component i.e. f $= 1728Hz$ and two other frequencies before and after the tonal one i.e. $1481Hz$ and $1975Hz$.

6.6.3 Adjoint–Based Aeroacoustic Optimization, Case A

As mentioned above, the objective function in case A is given by Eq. 4.4 which includes all the resolved frequencies. The receiver is located at $\vec{x}_r = (0.05m, 1.85m)$. During the optimization, only the airfoil shape may change while the rod remains intact. The airfoil shape is parameterized using a FFD box, with 108 control points (9×12) in total, out of which only 50 are free to move only in the y direction, Fig.

6.45. Thus, the optimization problem includes 50 design variables.

Every CFD solution, in either the optimization loop or FDs, starts from the same initial solution field in fully periodic state. As the shape changes during the optimization or FDs, the solver first runs for 5460 time steps to make sure that a periodic state is achieved and then 2030 time steps are sampled to perform the FW-H analogy, which is practically the optimization time window T_O. The numerical solution for almost 7500 time steps takes $\sim 12\ hrs$ on a single NVIDIA Tesla P100 GPU. The fact that the unsteady adjoint equations are integrated backward in time leads to memory issues in problems with many time steps since it demands considerable space to store the flow field at every time step, including the transient phase. However, the objective function contributes to the adjoint equation only within the optimization time window whereas, outside this window, the only contribution is from the unsteady term, meaning that the adjoint fields gradually decay, Fig. 6.46. Plotted in this figure are the contours of instantaneous adjoint density during the adjoint solution of the baseline geometry. As seen, for snapshots taken outside the optimization time window (time steps below 5460), the adjoint field decays, travels towards the inlet and has the trend to exit the domain, as time goes back. The computational time for 7490 time steps of adjoint solution (with 120 iteration per time step) is $\sim 11\ hrs$.

Decay of the adjoint field makes sensitivities reach a converged state, Fig. 6.47. This figure includes the reversed time series of the sensitivity for control points No. 11 to 14 during the adjoint solution and, as seen in the plot, after 460 time steps from the end of the optimization window, variations in sensitivities are greatly reduced. Hence, in order to reduce both computational cost and memory footprint of the optimization, yet with adequately accurate sensitivity derivatives, the unsteady adjoint solution continues for 460 time steps after the end of the optimization time window, covering 2490 time steps in total. Doing so, the computational time of an optimization cycle is $\sim 15.5\ hrs$ ($\sim 33\%$ faster than the full time horizon adjoint solution) and the required memory for storage of the primal and adjoint fields is $\sim 80\ GB$ (60% less than full time horizon adjoint solution). The so-computed gradients are further verified through comparison to central FDs with a step size of $10^{-7}C$. To reduce the computational cost, FDs are conducted only for control points No. 9 to 14 among which some high sensitivity values exist. As seen in Fig. 6.48, sensitivities computed by the hybrid adjoint method agree with FDs.

These continuous adjoint based computed aeroacoustic sensitivities are then used in the shape optimization of the airfoil. Design variable values are updated using a descent algorithm with a constant step size of $1.5 \times 10^{-12} \frac{m^2 s}{Pa}$. Convergence of the objective function, normalized by its initial value, during the optimization is illustrated in Fig. 6.49a. Its initial value reduces by 30%, after 12 optimization cycles. The biggest drop occurs at the first cycle followed by a small rise at the end of the second cycle. The objective value eventually converges to the optimal value

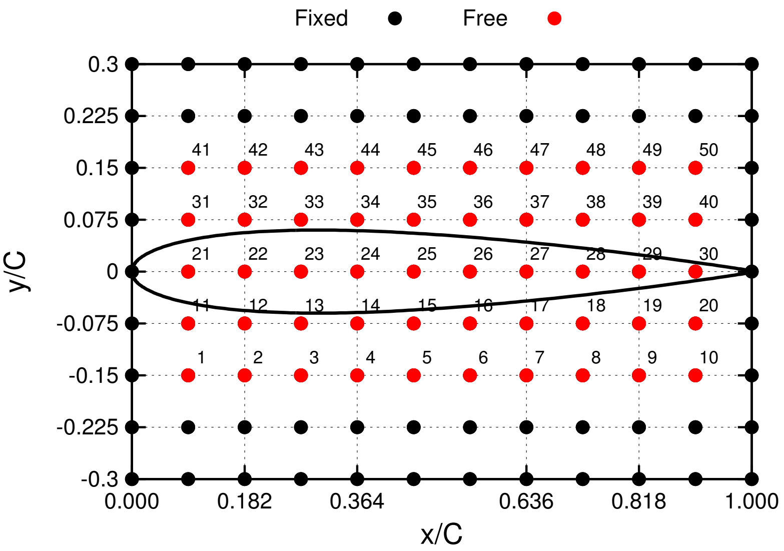


Figure 6.45: *Optimization of a Rod-Airfoil configuration*: Control points of the FFD box. Red/labeled control points are free to move during the optimization

at the 8^{th} cycle, where the SDs become small and, thereafter, fluctuates around this extremum. Figure 6.49b depicts the geometry of the optimized airfoil after 12 cycles. The resulted shape does not have an improved aerodynamic performance as the mean drag and lift coefficients of the airfoil change by $\Delta C_d = 0.011$ and $\Delta C_l = -0.018$; however, this work is focused on the aeroacoustic shape optimization and this is why lift or drag are not used as objective functions or constraints. Figure 6.50 plots the contours of instantaneous adjoint density during the adjoint solution of optimized geometry. It shows that the adjoint field decays similarly to what happens to the initial geometry (Fig. 6.46).

A comparative plot of the time series of the predicted sound pressure at the receiver location generated by the baseline and optimized geometries is displayed in Fig. 6.51a. It shows that pressure fluctuations in the optimized geometry are slightly shifted in time while their amplitude is significantly reduced. A similar comparison for the power spectrum, depicted in Fig. 6.51b, reveals that the optimization leads to reduced PSD levels for a great part of the frequency range below the third harmonic ($3456\,Hz$). The main peak, which is of the utmost importance as the current simulation is performed using the URANS model, is reduced by about $3.7\,dB$ and drops to $90.7\,dB$. Similarly, the second harmonic reduces by about $1.6\,dB$, whereas the third and fifth one increase by about 1.2 and $3.4\,dB$, respectively, and the fourth one remains unchanged. The figure also shows that all five harmonic frequencies remain unaffected by the optimization.

Figure 6.52 presents the OASPL directivity plot for the two geometries. As in the previous section, 100 receivers are located at radius $R = 1.85m$ from the mid-chord point of the airfoil. Compared to the initial geometry, the directivity cor-

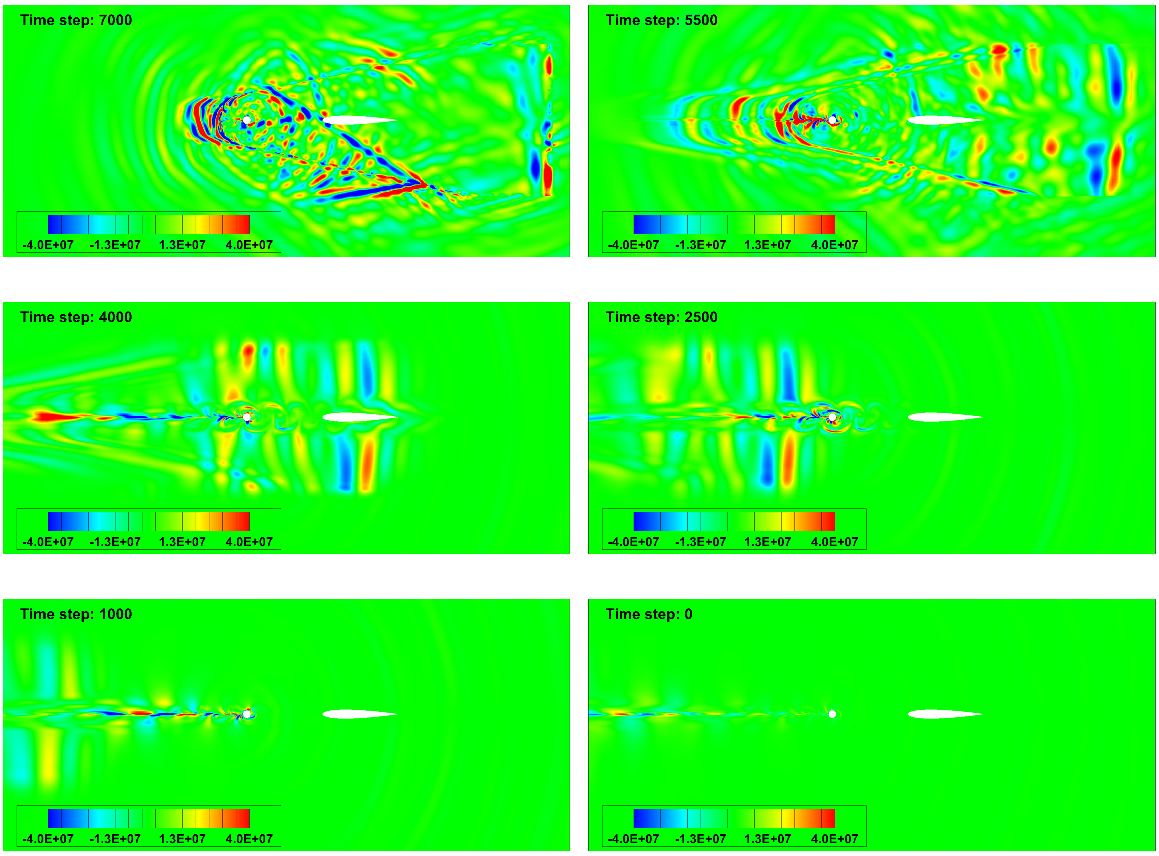


Figure 6.46: *Optimization of a Rod-Airfoil configuration*: Instantaneous adjoint density iso–areas for the baseline geometry.

responding to the optimized one has a stronger tendency to a quadrupole shape. Further study of the results in this graph shows insignificant noise reduction for receivers located between $300°$ and $60°$, together with a sound amplification from $135°$ to $195°$. At other angles, a considerable sound mitigation is observed, with the maximum reduction in the vicinity of $90°$. This is expected as this is the location of the receiver for which J (Eq. 4.4) has reduced. A similar sound reduction pattern can be seen in Fig. 6.53 where the iso-areas of the r.m.s. value of pressure fluctuations of the baseline and optimized geometries are compared.

Finally, contours of instantaneous Mach numbers at time steps 7000 and 6000 are depicted in Fig. 6.54 for both the baseline and optimized geometries. In the optimized geometry, low velocity areas appear close to the mid–chord of the suction side where re-circulation takes place, and trailing edge of the pressure side.

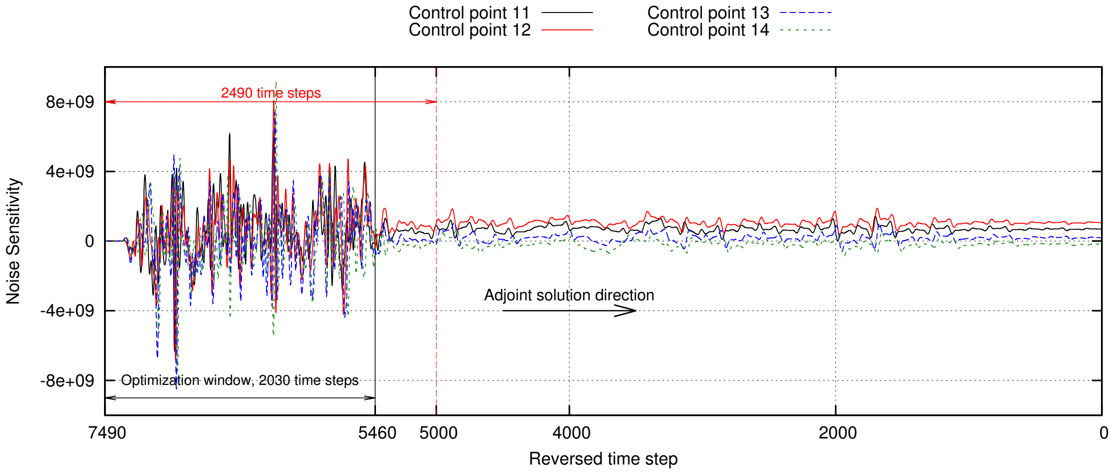


Figure 6.47: *Optimization of a Rod-Airfoil configuration*: Convergence of the sensitivity of the noise objective function for control points No. 11 to 14 during the unsteady adjoint solution. The backward in time integration of the adjoint equations starts at time step 7490. The optimization time window T_O during which the objective function is computed goes from time step 7490 back to 5460.

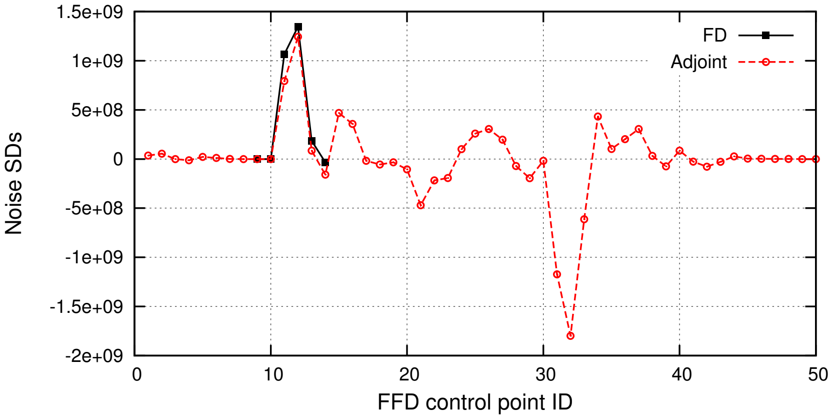


Figure 6.48: *Optimization of a Rod-Airfoil configuration*: Comparison of the sensitivities of the noise objective function computed by solving the adjoint equations for 2490 time steps backward in time and FDs for control points No. 9 to 14.

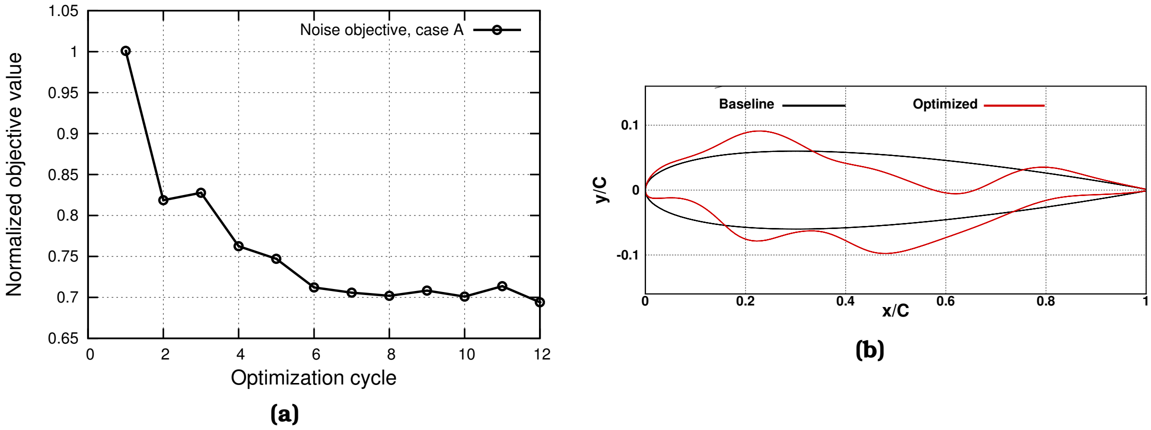


Figure 6.49: *Optimization of a Rod-Airfoil configuration*: Results after 12 optimization cycles. (a) Convergence of the noise objective function of case A normalized by its initial value. (b) Comparison of the baseline and optimized geometries.

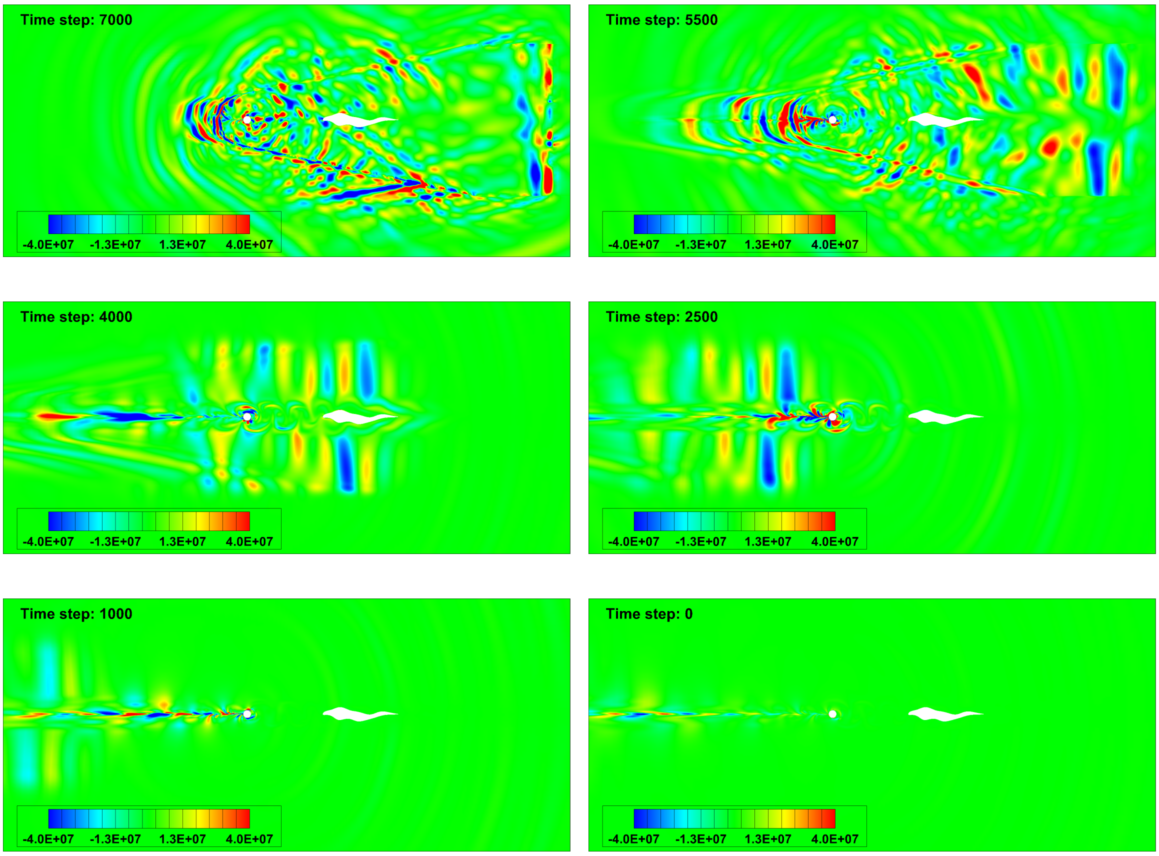


Figure 6.50: *Optimization of a Rod-Airfoil configuration*: Instantaneous adjoint density iso-areas for the optimized geometry.

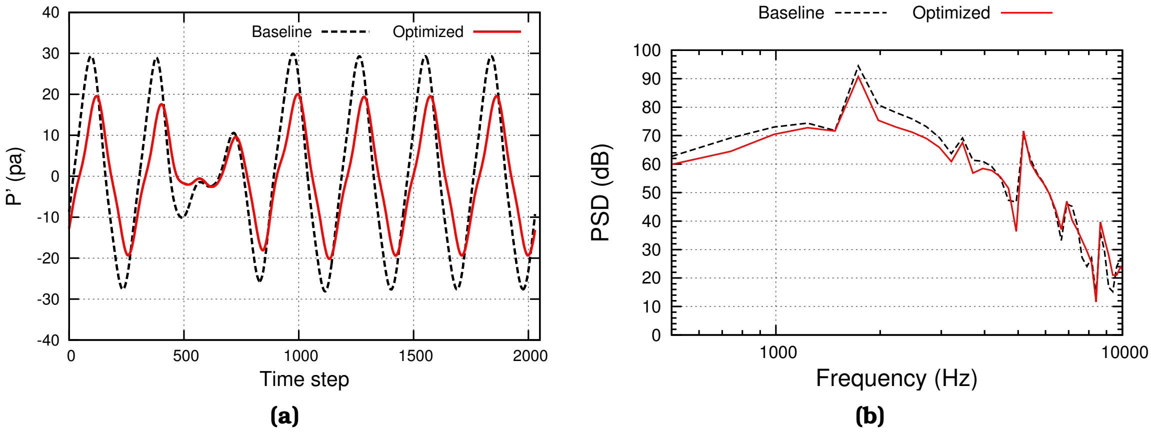


Figure 6.51: *Optimization of a Rod-Airfoil configuration*: Results after 12 optimization cycles. (a) Comparison of the time history of the pressure fluctuation between the baseline and optimized geometries at the receiver location $\vec{x}_r = (0.05m, 1.85m)$. (b) Comparison of the power spectral density of the pressure fluctuation resulted from the baseline and optimized geometries at the same location.

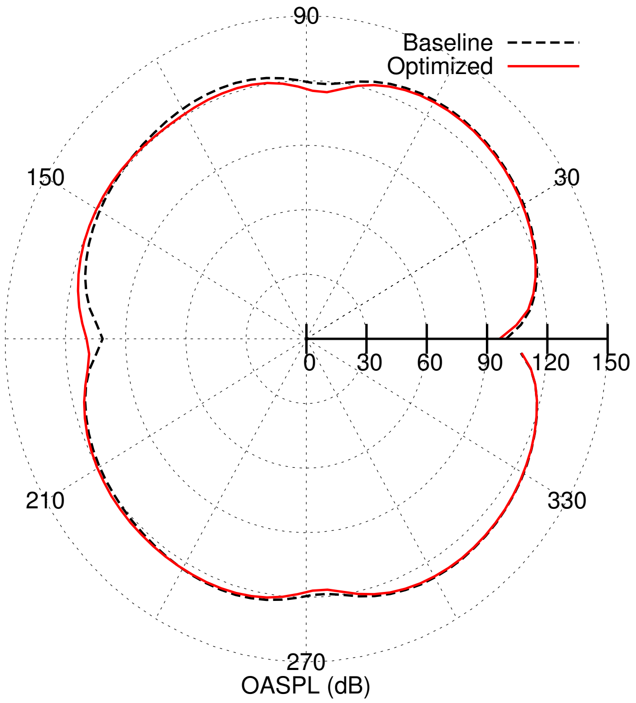


Figure 6.52: *Optimization of a Rod-Airfoil configuration*: Results after 12 optimization cycles. Comparison of the directivity plot of OASPL at radius $R = 1.85m$ from the mid-chord of the baseline and optimized geometries.

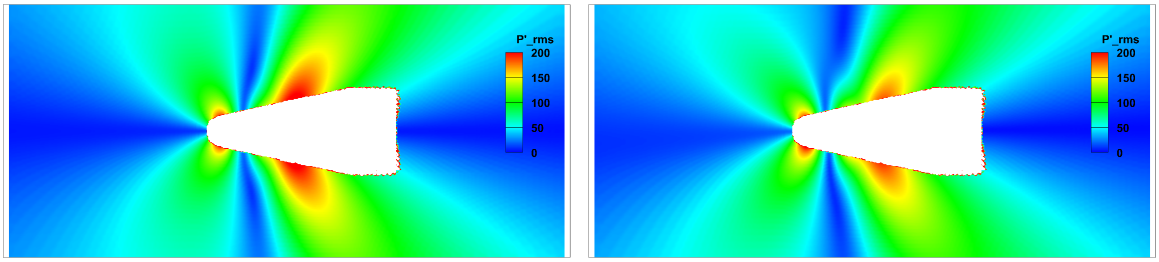


Figure 6.53: *Optimization of a Rod-Airfoil configuration*: Results after 12 optimization cycles. Comparison of the r.m.s. of pressure fluctuations outside the FW-H surface between the baseline (left) and optimized (right) geometries.

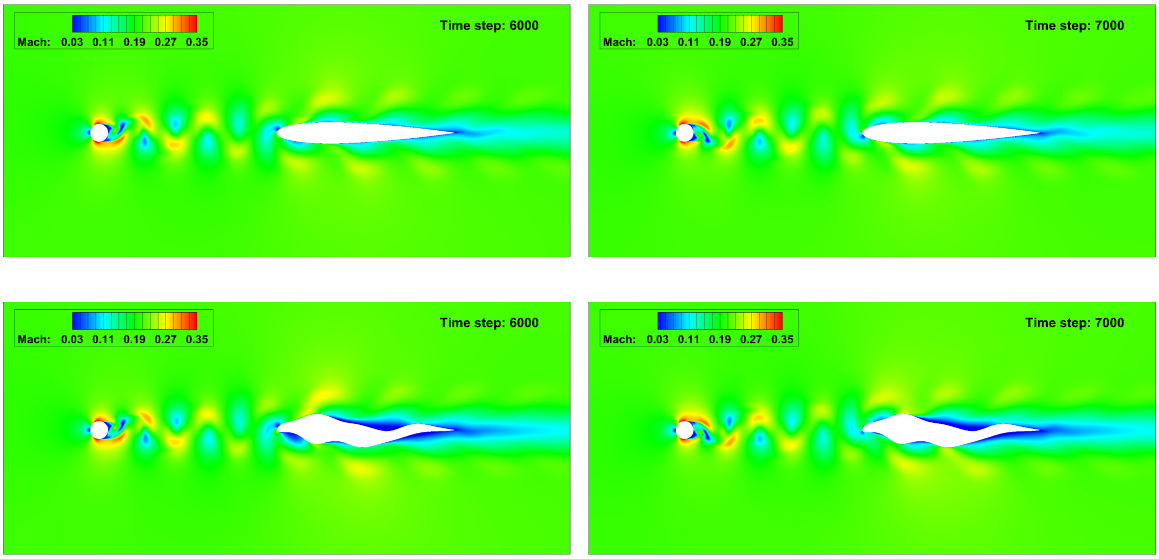


Figure 6.54: *Optimization of a Rod-Airfoil configuration*: Instantaneous Mach iso-areas at different time steps for the baseline (top row) and optimized (bottom row) geometries.

6.6.4 Adjoint–Based Aeroacoustic Optimization, Case B

In case B, the primal and adjoint solution set-ups, parameterization and step size are exactly the same as the case A in subsection 6.6.3 while the only difference is the objective function. As already mentioned, in case B, the aeroacoustic objective function Eq. 4.4 includes only the tonal component i.e. f $= 1728Hz$ and the two immediate neighboring frequencies i.e. $1481Hz$ and $1975Hz$. The purpose of this case is to check the ability of the developed aeroacoustic optimization tool in targeting a specific frequency range.

As seen in Fig. 6.55a, the value of the objective function is reduced by 35% after 13 design cycles. In Fig. 6.55b, the optimized shape from this case is compared to the one from case A and the baseline geometry. Despite the differences in geometries, the two optimized cases have quite similar acoustic performance which can be seen in Figs. 6.56 to 6.58. Similarity in the acoustic results between two cases is expected to some extent, given that the acoustic radiation of the rod-airfoil case is mainly governed by the tonal component. As a result, reduction in the objective value in case A, which includes all the resolved frequencies, is achieved mostly by reducing the tonal component, similar to case B.

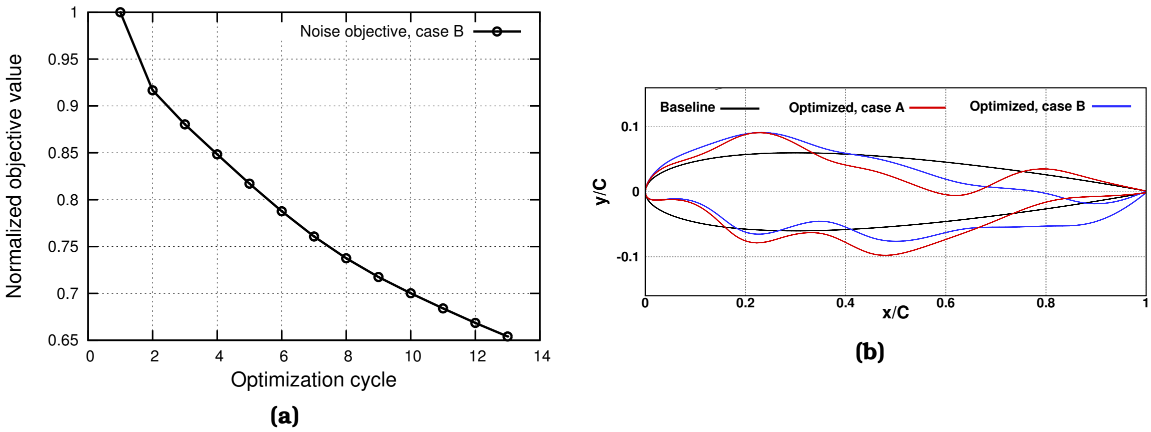


Figure 6.55: *Optimization of a Rod-Airfoil configuration*: Results after 13 optimization cycles. (a) Convergence of the noise objective function of case B normalized by its initial value. (b) Comparison of the baseline and two optimized geometries.

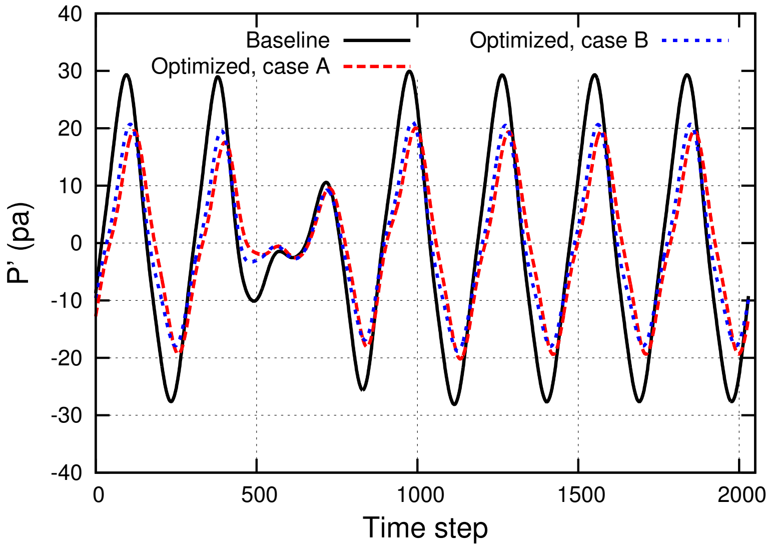


Figure 6.56: *Optimization of a Rod-Airfoil configuration*: Results after 13 optimization cycles. Comparison of the time history of the pressure fluctuation between the baseline and two optimized geometries at a receiver located at $\vec{x}_r = (0.05m, 1.85m)$.

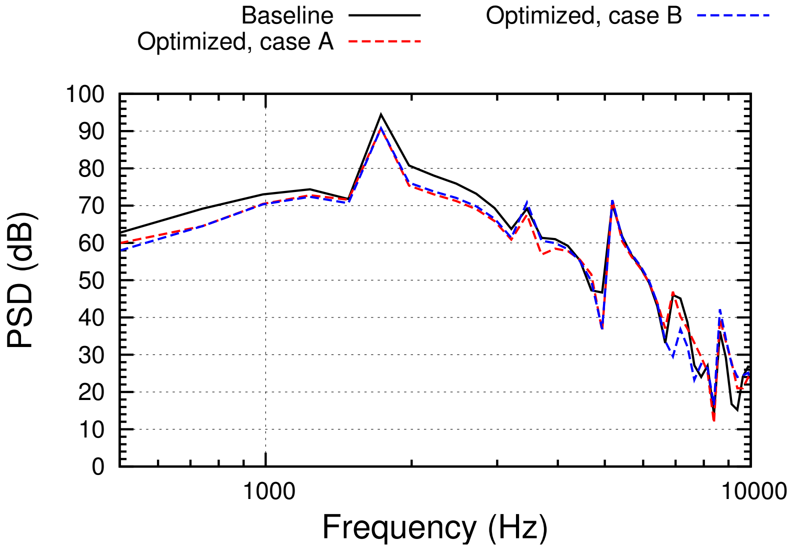


Figure 6.57: *Optimization of a Rod-Airfoil configuration*: Results after 13 optimization cycles. Comparison of the power spectral density of the pressure fluctuation resulted from the baseline and two optimized geometries at the same location.

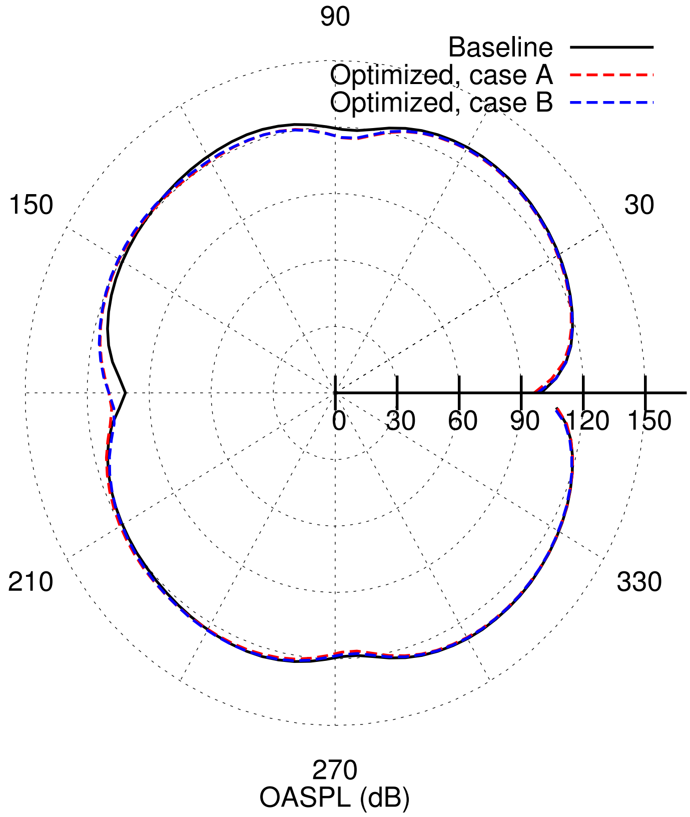


Figure 6.58: *Optimization of a Rod-Airfoil configuration*: Results after 13 optimization cycles. Comparison of the directivity plot of OASPL at radius $R = 1.85m$ from the mid-chord of the baseline and two optimized geometries.

6.7 Shape Optimization of a Vortex Shedding Sphere

The first 3D case is concerned with flow around a sphere. The free-stream flow is aligned with the $+x$ direction with Mach number $M = 0.18$. The Reynolds number based on the sphere diameter is $Re = 40000$. A hybrid grid, with almost 172000 nodes is used. 2912 nodes are located on the FW-H surface at radius $R = D$ where D is the sphere's diameter. Shown in Fig. 6.59 is the surface mesh on the sphere. The time step size for the unsteady solution is set to 0.00001375 seconds. First, the equations are integrated for an adequate number of time steps to establish a periodic state, which is used for initialization and, from that point on, solution goes on for 800 more time steps. On a single Tesla V100 GPU, the numerical solution for 800 time steps takes $\sim 6.5hrs$ and $\sim 10hrs$ for the primal and adjoint equations, respectively. In order for the residuals to drop at least 10 orders of magnitude at each time step, 500 and 1000 pseudo-time steps are performed per time step during the primal and adjoint solution.

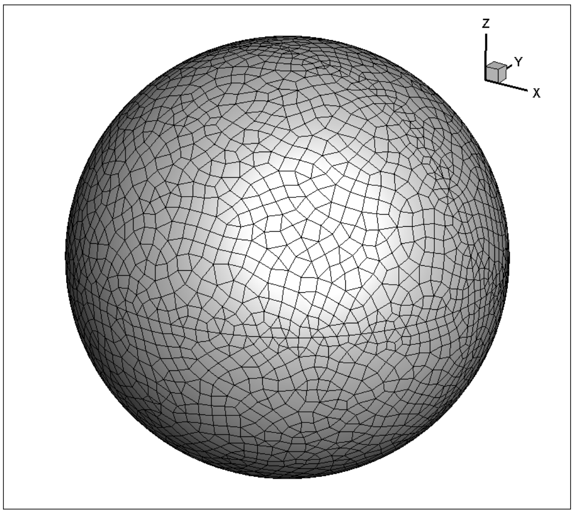


Figure 6.59: *Shape Optimization of a Vortex Shedding Sphere*: Surface grid on the sphere.

6.7.1 Comparison Between the Hybrid Method and URANS

Similar to the previous cases, in order to verify the accuracy of the implementation of the FW-H analogy in this case, results of the hybrid solver are compared with the outcome of the URANS equations' solver. This is done for 6 different receivers' locations shown in Fig. 6.60. As shown in Fig. 6.61, apart from some slight discrepancies for receivers 2 and 6, results of two methods are in a very good agreement for all receivers.

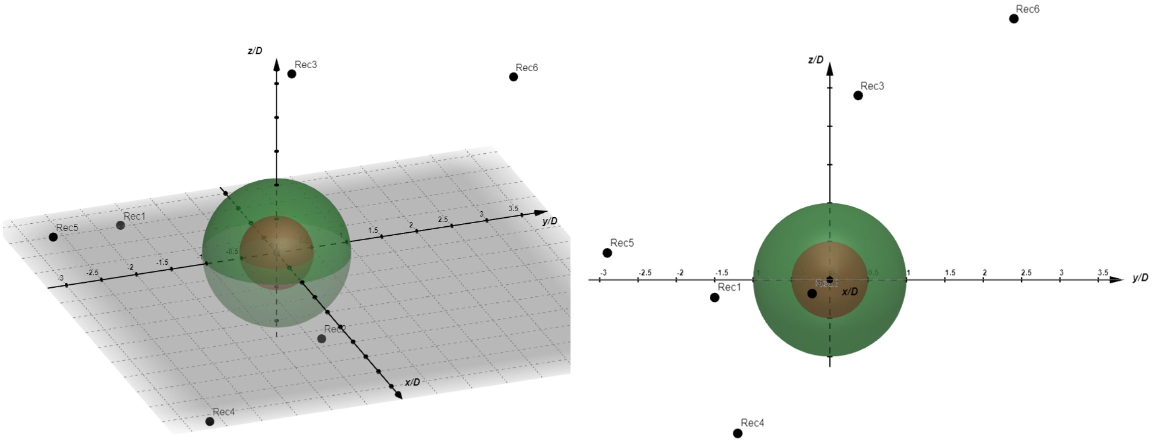


Figure 6.60: *Shape Optimization of a Vortex Shedding Sphere*: Comparison of the time history of the pressure fluctuation computed using pure CFD and hybrid method for a receiver located at

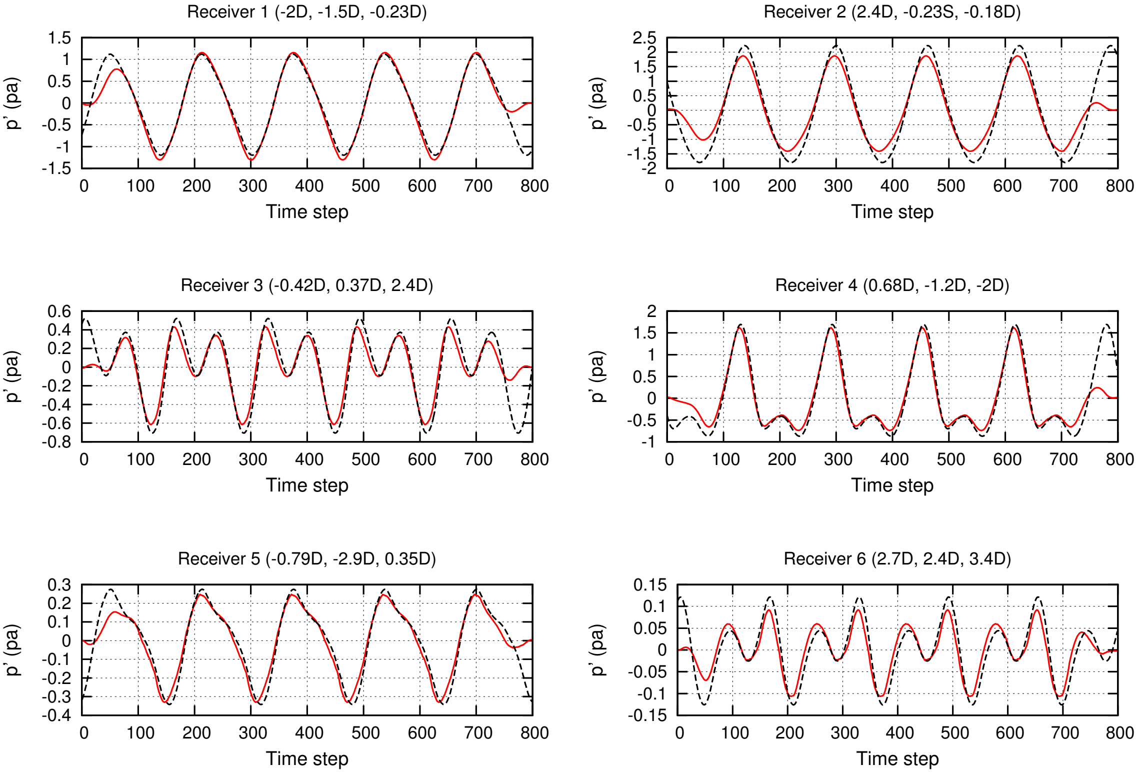
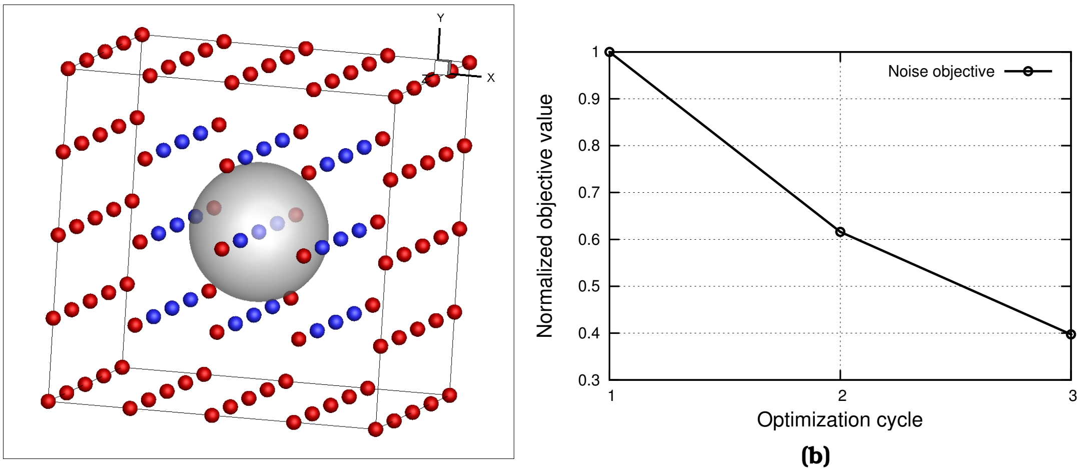


Figure 6.61: *Shape Optimization of a Vortex Shedding Sphere*: Comparison of the time history of the pressure fluctuations computed using URANS (- - -) and the hybrid solver (—) at different receivers' locations.

6.7.2 Adjoint–Based Optimization Results

Next, an aeroacoustic shape optimization is performed. The main goal is to ensure that the continuous adjoint solver is able to compute accurate gradients which can be used to reduce the objective value. Therefore no more than 3 optimization cycles are performed. The objective function is defined as Eq. 4.4 , integrated over all resolved frequencies, for a receiver located at $\vec{x}_r = (0, 0, 30D)$. The grid displacement is based on the linear spring analogy, performed inside an area with $R = 0.8DC$. In each cycle, the flow field is initialized using a stored periodic field corresponding to the initial geometry, and runs for 800 time steps among which the last 400 ones form the optimization time window. A $5 \times 5 \times 5$ FFD box is employed for parameterization with 27 control points being allowed to move in all 3 directions during the optimization, resulting to 54 design variables, Fig. 6.62a. As illustrated in Fig. 6.62b, after 3 design cycles, the objective value is reduced by 60 %, meaning that the hybrid adjoint solver predicted correct gradients for optimization. The baseline and optimized shapes are drawn in Fig. 6.63b where it shows that the optimized geometry is shrunk in the z direction while lengthened in the y direction. This, can also be in Fig. 6.64, where the iso-areas of normal to the surface displacement of the initial geometry to match the optimized one are shown. Figure 6.65 compares the iso-areas of pressure on the sphere and streamlines between the initial and optimized geometries at 3 different time instants.



(a)

(b)

Figure 6.62: *Shape Optimization of a Vortex Shedding Sphere*: (a) Control points of the FFD box. Blue control points are free to move during the optimization. (b) Convergence of the aeroacoustic objective function for 3 design cycles.

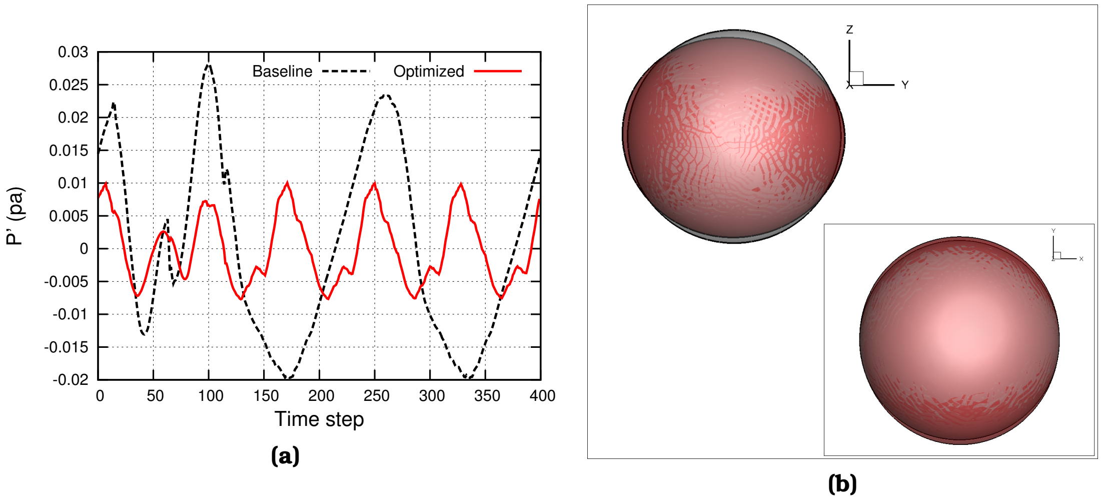


Figure 6.63: *Shape Optimization of a Vortex Shedding Sphere*: Results after 3 optimization cycles. (a) Comparison of the time history of the pressure fluctuation between the baseline and optimized geometries. (b) Two different views of the baseline (gray) and optimized (red) geometries.

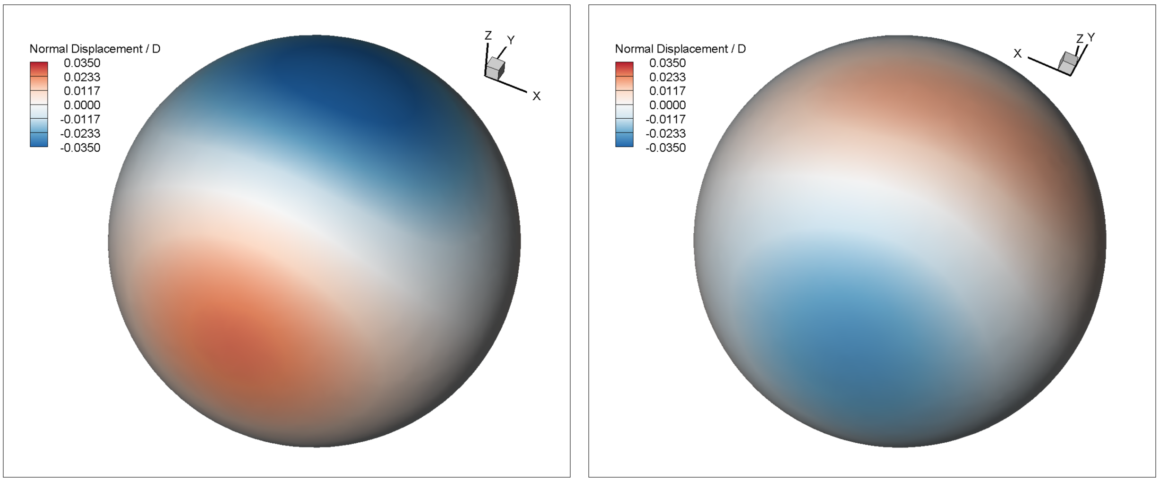


Figure 6.64: *Shape Optimization of a Vortex Shedding Sphere*: Results after 3 optimization cycles. Two different views of the surface displacement (normalized by diameter) of the baseline geometry in the normal to the surface direction to match the optimized geometry.

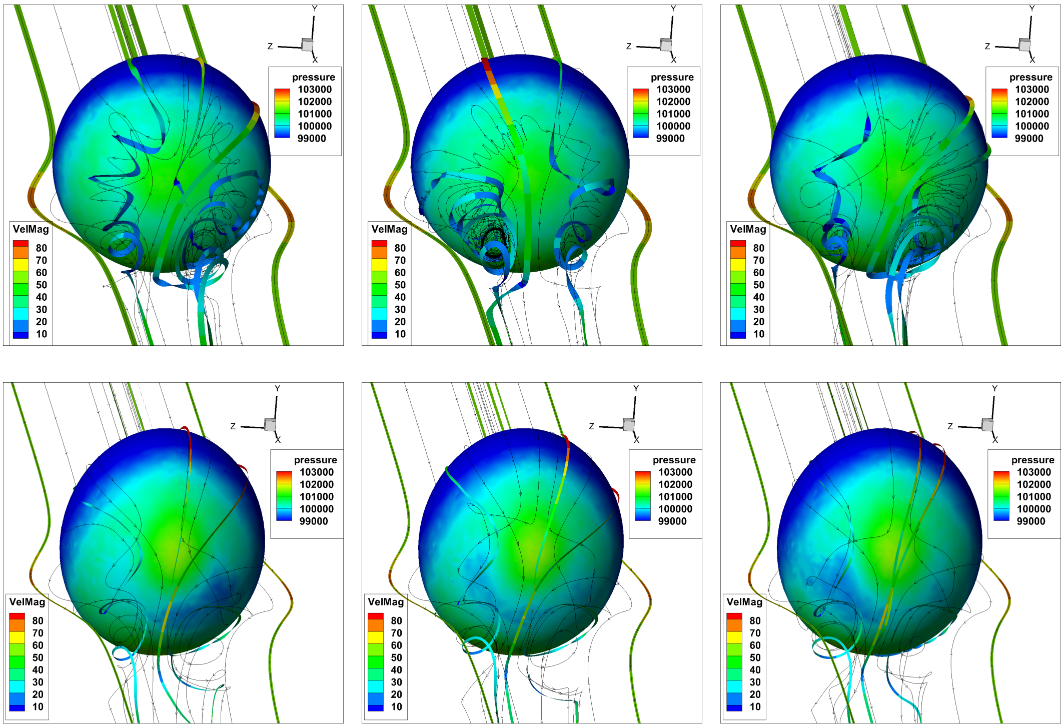


Figure 6.65: *Shape Optimization of a Vortex Shedding Sphere*: Results after 3 design cycles. Comparison of the iso-areas of pressure on the sphere and flow streamlines between the baseline (top row) and optimized (bottom row) geometries at three different time steps of 600 (left column), 700 (middle column) and 800 (right column).

6.8 Optimization of an Aero–Engine Intake

This is an industrial application of the developed optimization method, dealing with the aeroacoustic shape optimization of an aero–engine intake. This case became available thanks to the participation of the PCOpt/NTUA in the MADELEINE project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 769025. The initial intake geometry and the boundary conditions of the fan inlet were provided by Rolls-Royce plc. and the Institute of Sound and Vibration Research of the University of Southampton (ISVR), respectively.

Shown in Fig. 6.66 is a perspective view of the engine–intake. As shown in the previous sections, adjoint–based optimization in unsteady flows may become very demanding in terms of time and memory which is even more challenging for industrial cases. This is the main reason why the use of adjoint methods is comparatively restricted in aeroacoustic shape optimization that are unsteady by nature.

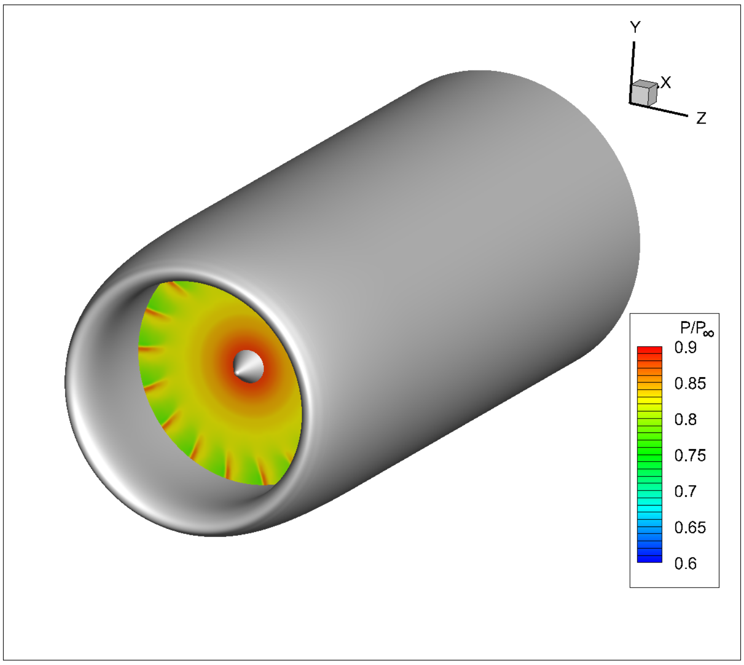


Figure 6.66: *Optimization of an Aero-Engine Intake*: Perspective view of the engine intake with a snapshot of the static pressure distribution, normalized by the far-field pressure, at the engine inlet serving as an instantaneous boundary condition.

In this particular application, in order to reduce the cost and since the intake geometry is axisymmetric, a CFD domain that corresponds to a single blade passage of the fan with appropriate periodicity conditions is modeled. The flow and adjoint equations are solved in a rotating (with the rotational speed of the engine) frame, thus leading to a steady-state solution for both the flow and adjoint problems. In order to assure a periodic adjoint solution, a continuous circumferential distribution of receivers at a given radius and axial position is used for the computation of the objective function of Eq. 4.4 . The unsteady flow fields are required for computing the FW-H integral. These are achieved by properly rotating the steady flow field, while also transforming the kinematic field into the absolute reference system. Therefore, the work–flow of the aeroacoustic shape optimization is the one shown in Fig. 6.67.

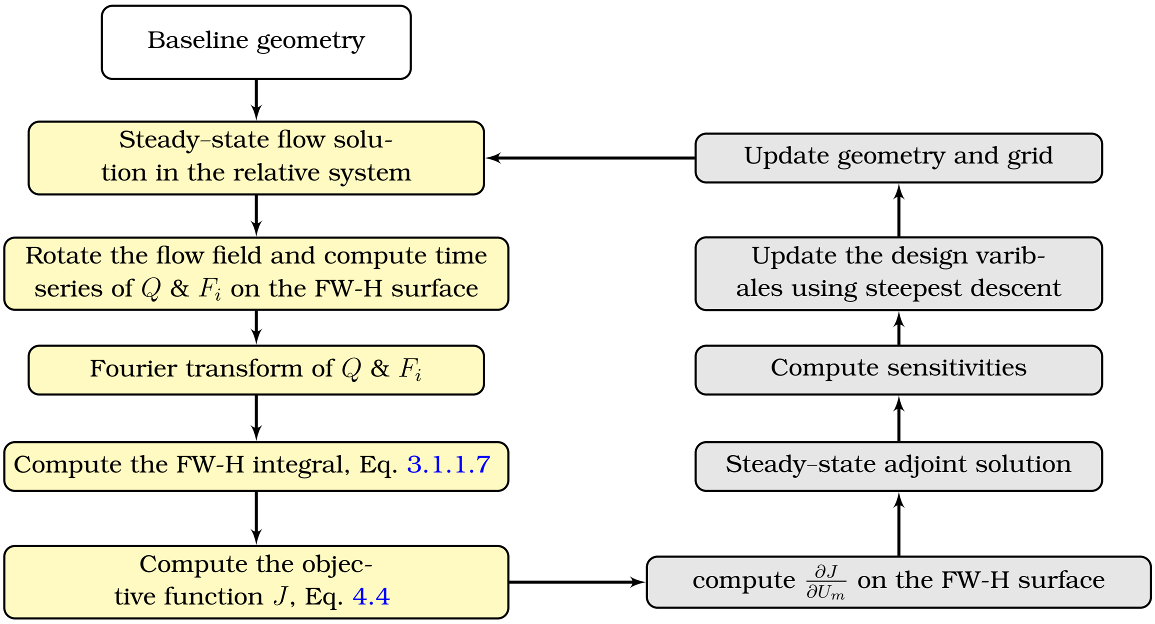

Baseline geometry
Steady–state flow solution in the relative system
Rotate the flow field and compute time series of Q & F_i on the FW-H surface
Fourier transform of Q & F_i
Compute the FW-H integral, Eq. 3.1.1.7
Compute the objective function J, Eq. 4.4
compute $\frac{\partial J}{\partial U_m}$ on the FW-H surface
Steady–state adjoint solution
Compute sensitivities
Update the design variables using steepest descent
Update geometry and grid

Figure 6.67: *Optimization of an Aero-Engine Intake*: Work-flow of the aeroacoustic optimization using the hybrid method of this thesis. Primal and adjoint are shown in yellow and gray, respectively.

Some basic dimensions of the aero–engine, normalized by the fan intake radius R, can be found in Fig. 6.68 where the generatrix of the engine is plotted. The geometry of a single blade passage, shown in Fig. 6.69, has $\sim 3.7M$ nodes arranged on 100 meridional planes. Air at far–field is still and the pressure distribution on the fan–inlet, provided by ISVR, is used as boundary condition. The Spalart-Allmaras turbulence model and its adjoint solver are employed. The numerical solution takes $\sim 8\ hrs$ on a single NVIDIA Tesla V100 GPU, and all residuals drop by 5 orders of magnitude. The pressure contours on the nacelle and mid–plane of the engine intake are plotted in Fig. 6.70.

The parameterized part of the nacelle, colored in red, is shown in Fig. 6.69. An axisymmetric parameterization model is adopted for the nacelle. The generatrix is firstly reconstructed using NURBS, giving rise to 15 design parameters controlling the shape of the nacelle lips and the throat area, leaving its outer shape and the part close to the fan intact. The NURBS control points are allowed to vary in both

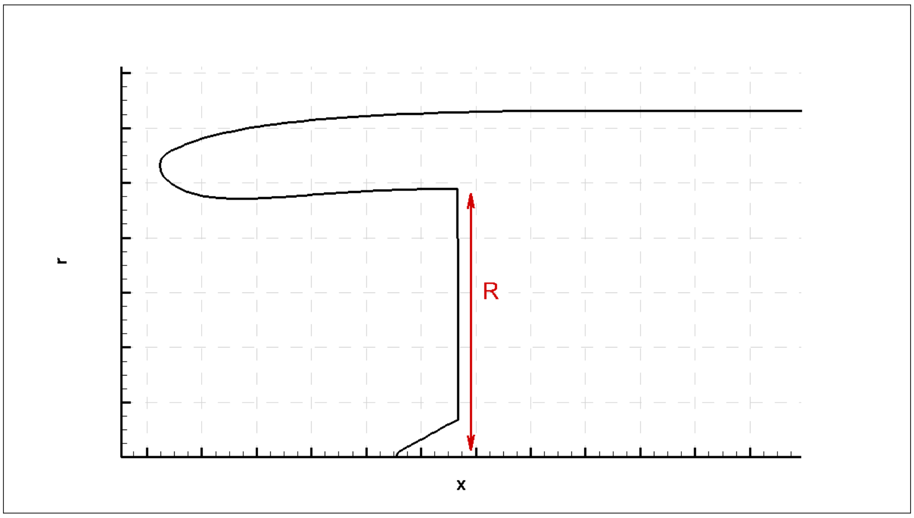


Figure 6.68: *Optimization of an Aero-Engine Intake*: Generatrix of the aero-engine, normalized by R.

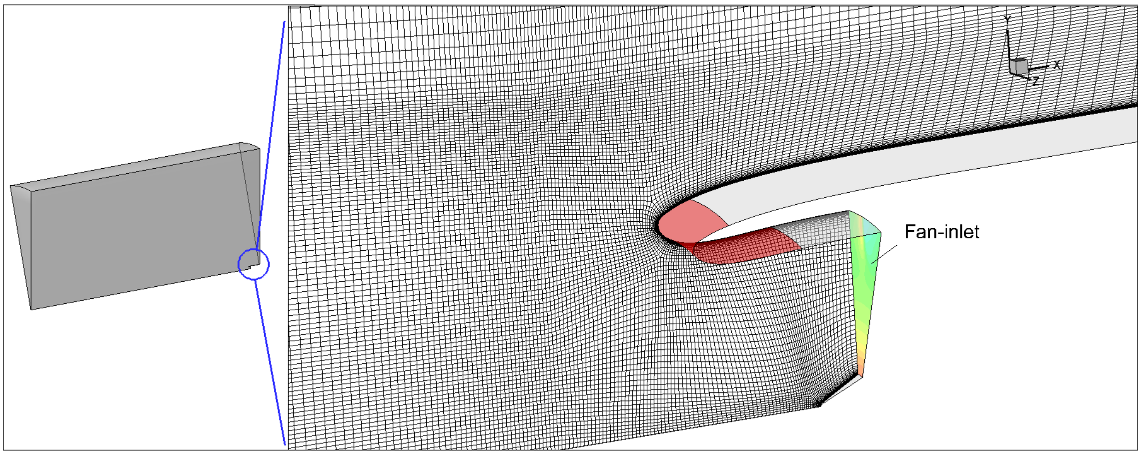


Figure 6.69: *Optimization of an Aero-Engine Intake*: Entire and close-up views of the CFD domain and the surface grid on one of periodic boundaries. The parameterized part of the nacelle is colored in red.

the axial and radial direction. Since the first and last control points are fixed, the optimization has $2{\times}13{=}26$ design variables, Fig. 6.71.

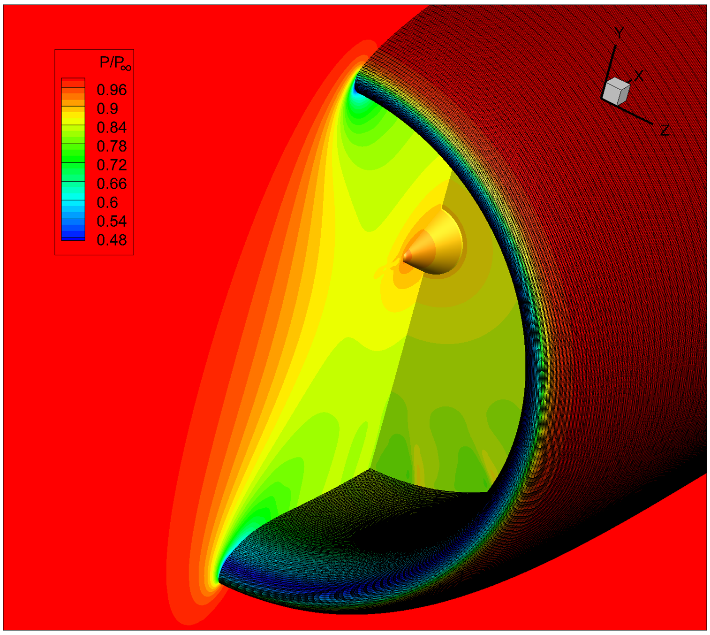


Figure 6.70: *Optimization of an Aero-Engine Intake*: Iso-areas of pressure on the nacelle, mid-plane and engine inlet.

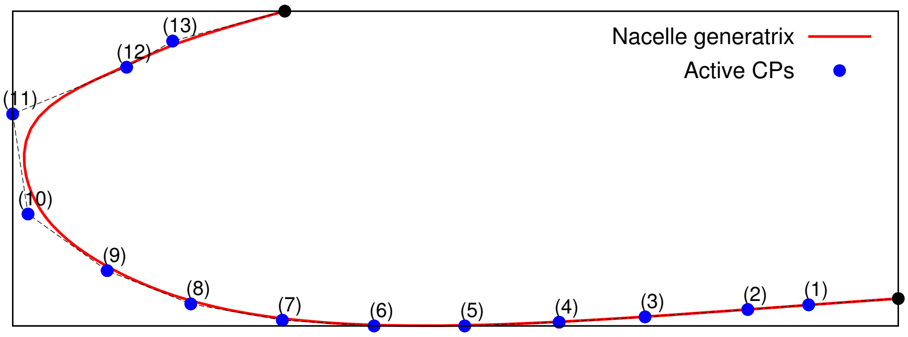


Figure 6.71: *Optimization of an Aero-Engine Intake*: Distribution of the control points for the parameterization of the nacelle generatrix using NURBS. The first and last control points are fixed during the optimization.

6.8.1 Comparison Between the Hybrid method and URANS

The FW-H surface, illustrated in Fig. 6.72, has 16000 nodes. To perform the FW-H integral, the FW-H surface should rotate to cover the full circumference. However, as a cheaper alternative, the receivers (instead of the FW-H surface) are rotated and the acoustic pressure field is retrieved by superimposing pressure signals from each receiver. Before proceeding to the aeroacoustic optimization, the acoustic results of the case are verified, as both the grid and numerical set-up may affect acoustic results. To do so, the acoustic pressure computed based on the hybrid model is compared with that of the unsteady CFD code at 3 receiver locations, shown in Fig. 6.72. In order to reduce the inaccuracies of the pressure signals computed by pure CFD, the receivers are located relatively close to the FW-H surface. As illustrated in Fig. 6.73, there is a reasonable agreement between the results of the FW-H analogy and CFD, given that part of the discrepancies may be due to the neglected quadrupole terms, representing the noise due to viscous effects and turbulence.

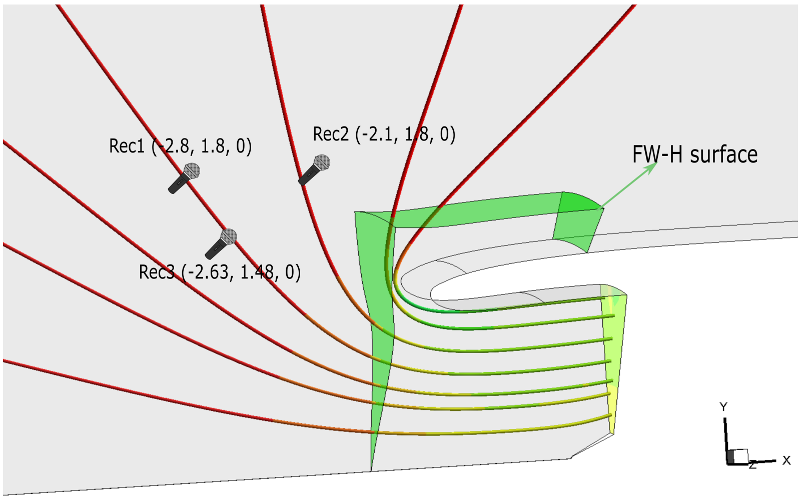


Figure 6.72: *Optimization of an Aero-Engine Intake*: Location of the FW-H surface, in green, and 3 receivers within the CFD domain. Streamlines are colored based on the pressure values. Coordinates are normalized by the fan intake radius R.

6.8.2 Adjoint–Based Aeroacoustic Optimization, Single Receiver

In this optimization, the aeroacoustic objective function to be minimized, i.e. that of Eq. 4.4, includes only the value corresponding to the blade passing frequency (BPF). As already mentioned, a continuous circumferential distribution of receivers is used for the computation of the objective function. In this optimization, a single circumferential row of receivers (located at a single radial and axial position) is used, as shown in Fig. 6.74. The convergence history of the aeroacoustic objective function is presented in Fig. 6.75a. A clear reduction in the objective value after 13 optimization cycles can be seen. This reduction is also obvious in Fig. 6.75b which compares the amplitude of the sound pressure at the receiver in the baseline and the optimized geometries. Plotted in Fig. 6.76 are the directivities of SPL at the BPF. The SPL values are normalized by the initial value of the aeroacoustic objective function and this is why the normalized SPL of the initial geometry at $\theta = 90^\circ$ is unit. As expected, the maximum reduction occurs

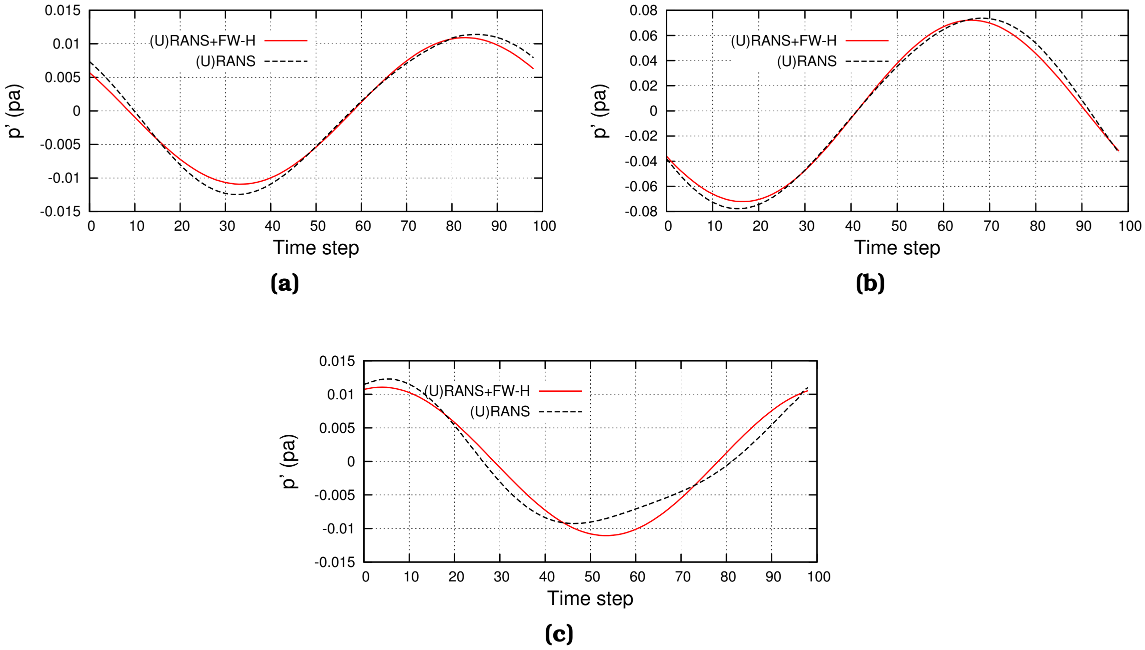


Figure 6.73: *Optimization of an Aero-Engine Intake*: Comparison of the pressure fluctuations computed by the hybrid method with those obtained by pure CFD, at 3 different receivers locations (a) $\vec{x}_{Rec1} = (-2.8, 1.8, 0)$, (b) $\vec{x}_{Rec2} = (-2.1, 1.8, 0)$, (a) $\vec{x}_{Rec3} = (-2.63, 1.48, 0)$. Coordinates are normalized by the fan intake radius R.

at $\theta = 90^\circ$ where the objective function is defined. However, this considerable reduction is not omni-directional as the SPL is increased between 100° and 120° and below 80°. This is due to the use of a single circumferential row of receivers in the aeroacoustic optimization.

Changes of the parameterized part of the nacelle are illustrated in Figs. 6.77. As seen, the biggest geometrical change occurs at the nacelle lip which is pushed downward, while minor changes occur elsewhere. Regarding the aerodynamic performance, the optimized shape increases the total pressure losses by 0.8% compared to the initial one. Worsening the aerodynamic performance could be expected as an aeroacoustic objective was only considered.

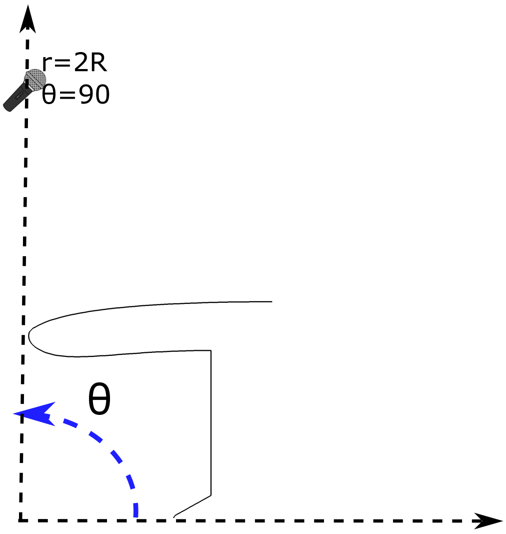


Figure 6.74: *Optimization of an Aero-Engine Intake*: Schematic of the circumferential row of receivers based on which the aeroacoustic objective function is defined. Generatrix and receiver location are not in scale.

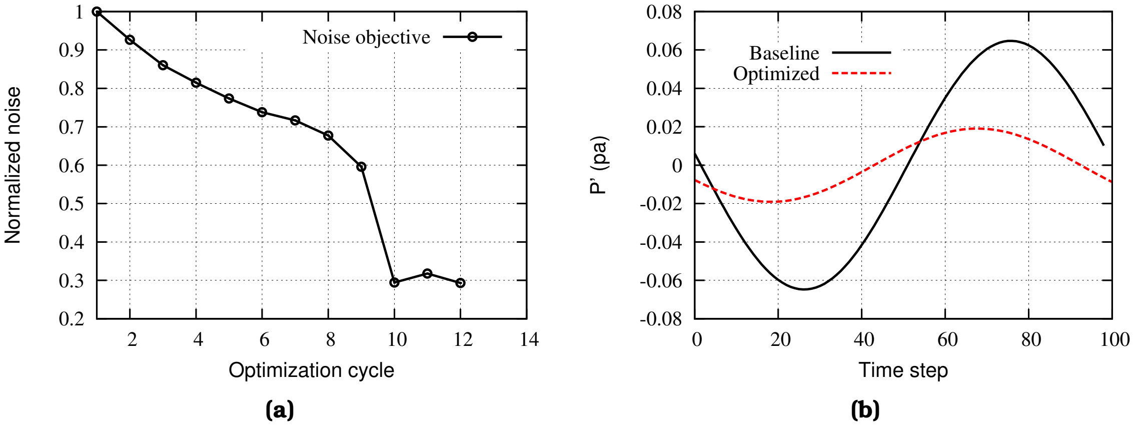


Figure 6.75: *Optimization of an Aero-Engine Intake*: Results after 13 aeroacoustic optimization cycles. (a) Convergence of the aeroacoustic objective value, normalized by its initial value. (b) Comparison of the time history of the pressure fluctuation within a period at one receiver.

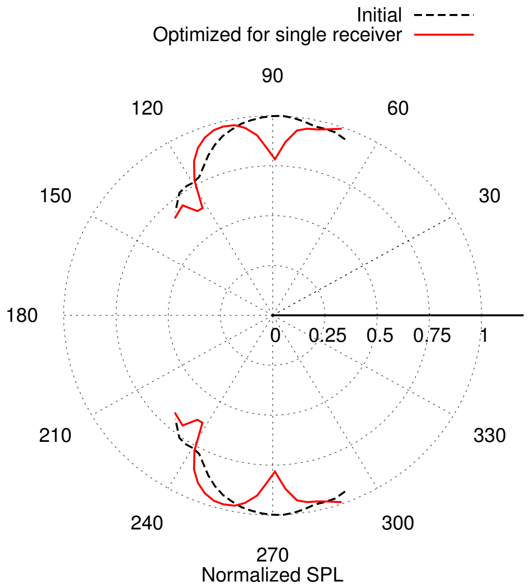


Figure 6.76: *Optimization of an Aero-Engine Intake*: Results after 13 aeroacoustic optimization cycles. Comparison of the directivity plot of the SPL at the BPF for receivers at $r = 2R$ normalized by the initial value of the aeroacoustic objective function.

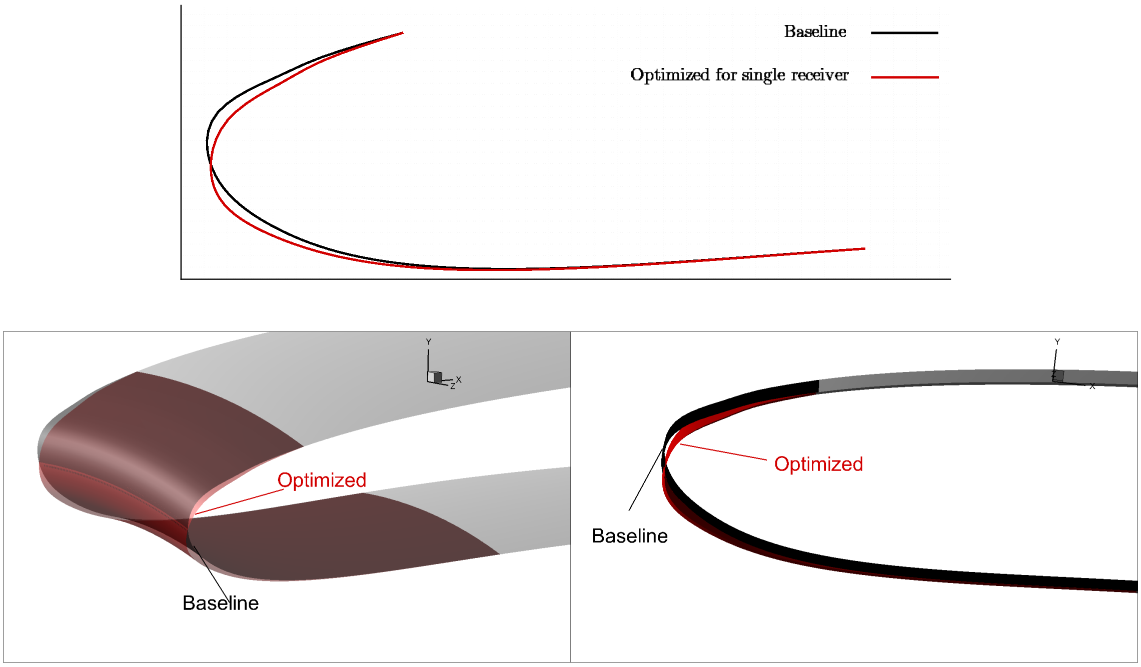


Figure 6.77: *Optimization of an Aero-Engine Intake*: Results after 13 aeroacoustic optimization cycles. (Top) The generatrix of the baseline and optimized geometry of the parameterized part of the nacelle. (Bottom) 3D views of the baseline and optimized geometry.

6.8.3 Adjoint–Based Aeroacoustic Optimization, Multi Receiver

The previous optimization showed that the noise reduction strongly depends on the receiver location as it results in a sudden noise reduction in the vicinity of the receiver and increase in other directions. Herein, with a goal to achieve a smooth noise reduction pattern for a wider range of direction, the previous optimization is repeated, this time with an objective function which takes receivers located on 3 different circumferential rows into account, Fig. 6.78. This means that the objective function is defined as the average of Eq. 4.4 over 3 receiver locations. The receivers are located at a same radius of $8.8R$ and different angles of $90°$, $105°$ and $120°$. As already mentioned, a continuous circumferential distribution of receivers is used at each receiver location.

The convergence history of the aeroacoustic objective function is presented in Fig. 6.79a. The objective value is reduced by 20% after 12 optimization cycles which is less than the reduction in the previous case. This is expected, due to the involvement of more circumferential rows of receivers in the new optimization, as objective values at different rows can be competing. This reduction is also obvious in Fig. 6.79b - 6.79d that compare the amplitude of the sound pressure at the receivers in the baseline and the optimized geometries. Plotted in Fig. 6.80 are the directivities of normalized SPL at the BPF resulted from two optimized geometries and the initial one. Despite the first optimization which reduces noise only at $\theta = 90°$ direction, the second optimization results to an almost omni–directional noise reduction.

Changes in the parameterized part of the nacelle are illustrated in Figs. 6.81. As seen, the biggest geometrical change occurs at the nacelle lip which is pushed forward, in contrast of the previous optimization. Regarding the aerodynamic performance, the optimized shape has lower total pressure losses by 0.12% compared to the initial one; this can be attributed to the slightly shorter axial length.

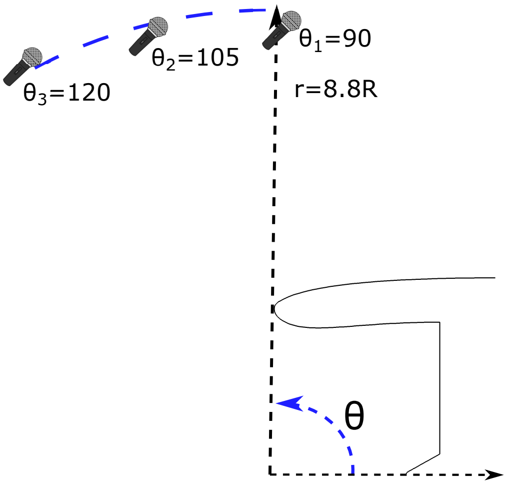


Figure 6.78: *Optimization of an Aero-Engine Intake*: Schematic of the receiver locations based on which the aeroacoustic objective function is defined. The generatrix and receiver locations are not in scale.

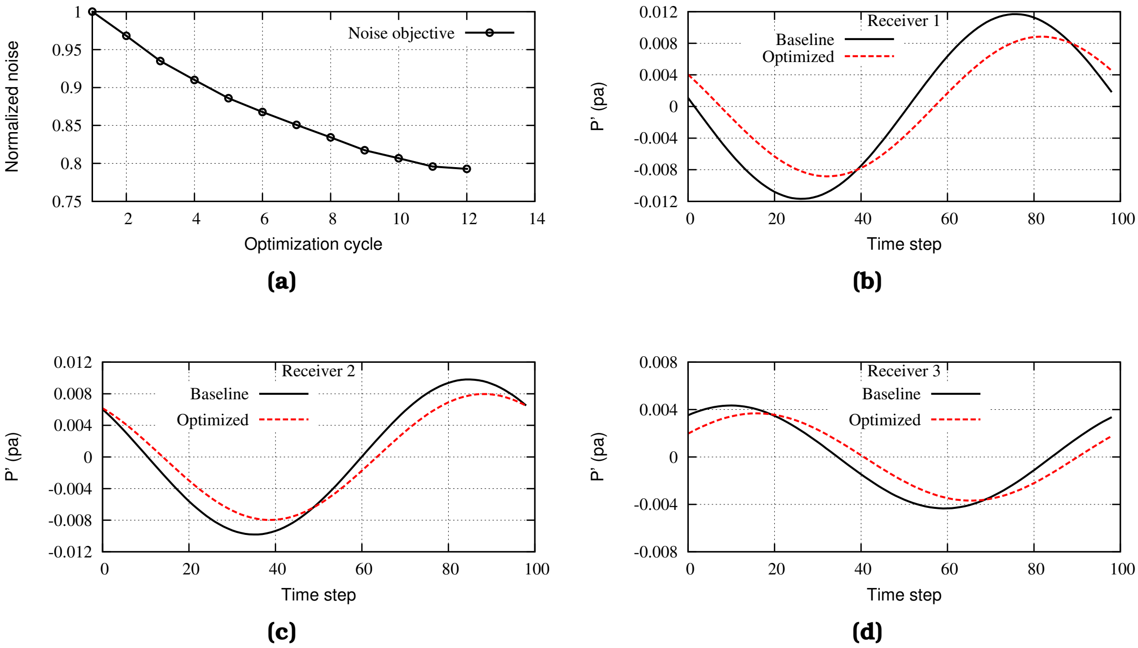


Figure 6.79: *Optimization of an Aero-Engine Intake*: Results after 12 aeroacoustic optimization cycles. (a) Convergence of the aeroacoustic objective value, normalized by its initial value. (b) to (d) Comparisons of the time histories of the pressure fluctuation within a period at the 3 circumferential row locations.

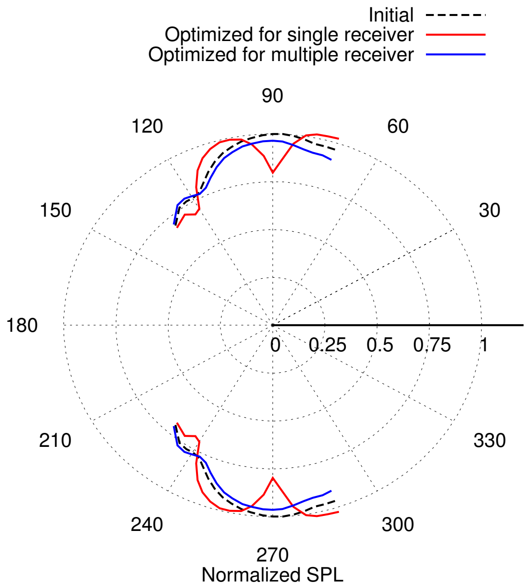


Figure 6.80: *Optimization of an Aero-Engine Intake*: Results after 12 aeroacoustic optimization cycles. Comparison of the directivity plot of the SPL at the BPF for receivers at $r = 8.8R$ normalized by the initial value of the aeroacoustic objective function.

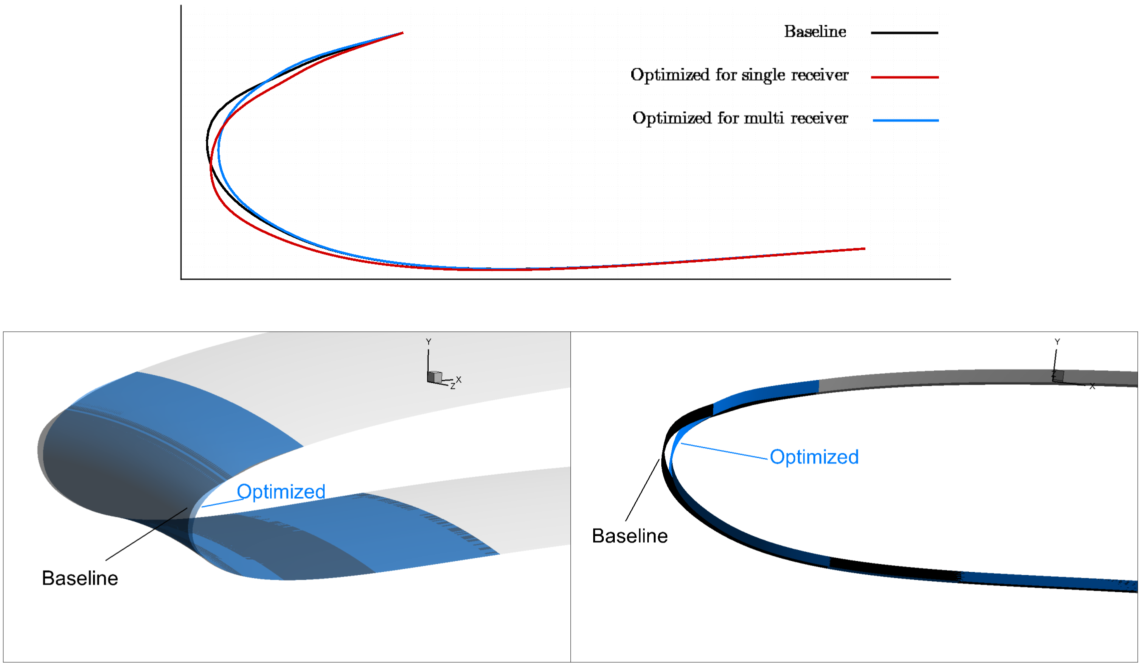


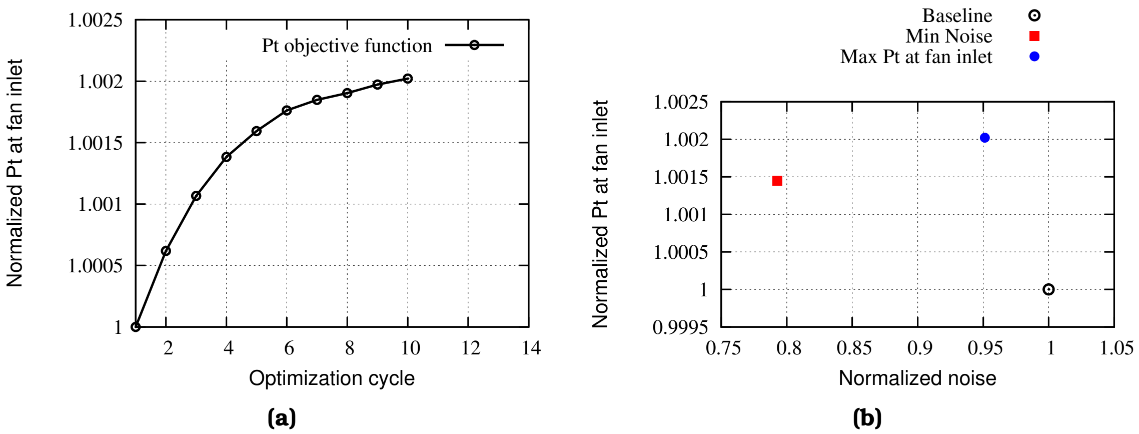
Figure 6.81: *Optimization of an Aero-Engine Intake*: Results after 12 aeroacoustic optimization cycles. (Top) The generatrix and of the baseline and optimized geometry of the parameterized part of the nacelle. (Bottom) 3D views of the baseline and optimized geometries.

6.8.4 Adjoint–Based Aerodynamic Optimization

Finally, an aerodynamic optimization is also performed in order to reduce the total pressure loss. This is equivalent to increasing the total pressure at the fan inlet, as the far–field pressure remains constant. The aeroacoustic performance is evaluated using the same distribution of receivers as in the previous case in subsection 6.8.3.

The convergence history of the aerodynamic objective function in Fig. 6.82a shows that the total pressure at the fan inlet is increased by 0.2% after 10 optimization cycles. Results of the current aerodynamic optimization are compared to the previous aeroacoustic one in Fig. 6.82b. As seen, aerodynamic performance resulted from two optimizations are close to each other, while aeroacoustic results are different, yet being improved compared to the baseline.

Figure 6.81 compares the optimized nacelle geometry resulted from the current optimization with all previous aeroacoustic optimizations and the baseline. It shows that the present optimization (green dashed line) leads to a similar geometry to the previous aeroacoustic optimization (blue line). This explains the similar aerodynamic performance. It also indicates that, compared to the total pressure, the aeroacoustic objective function is much more sensitive to the nacelle shape.



(a) (b)

Figure 6.82: *Optimization of an Aero-Engine Intake*: Results after 10 aerodynamic optimization cycles. (a) Convergence of the total pressure at fan inlet normalized by its initial value. (b) Comparison of the results from current optimization (aerodynamic) with the previous one (aeroacoustic).

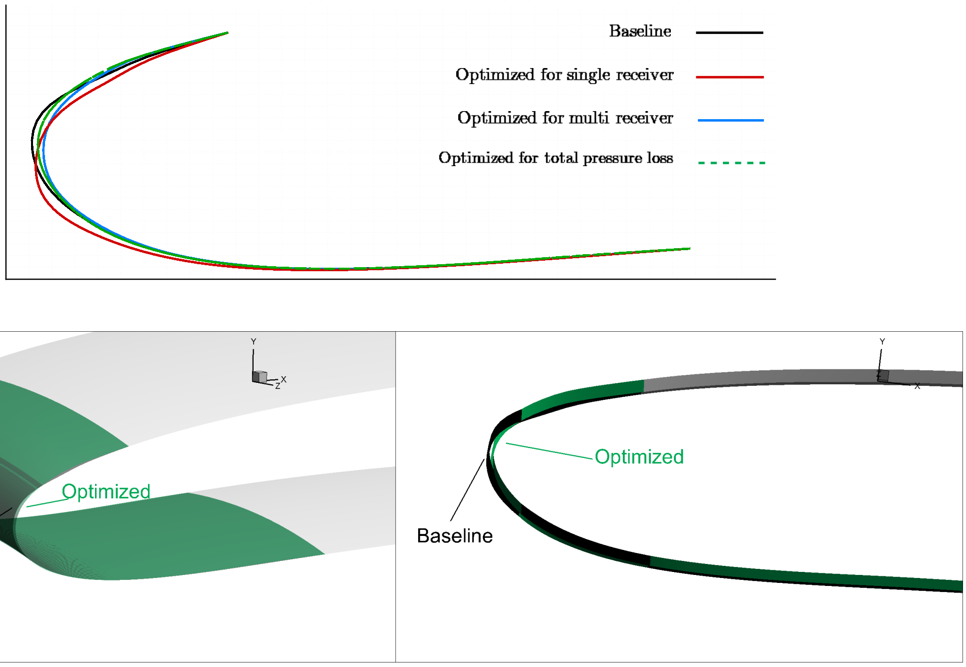


Figure 6.83: *Optimization of an Aero-Engine Intake*: Results after 10 aerodynamic optimization cycles. (Top) The generatrix and of the baseline and optimized geometry of the parameterized part of the nacelle. (Bottom) 3D views of the baseline and optimized geometries.

Chapter 7

Closure

7.1 Summary-Conclusion

The aim of this doctoral thesis was to develop an aeroacoustic shape optimization tool, efficient and accurate enough, with an affordable computational cost even in large–scale problems. The mathematical formulation, programming and verification of the continuous adjoint method to a hybrid noise prediction tool, being the backbone of the developed G-B optimization method, is the most important contribution of this PhD thesis.

A hybrid method for noise prediction starts by simulating the unsteady flow which, in this work, was performed using the GPU–enabled URANS equations' solver PUMA of PCOpt/NTUA. All tools developed during this PhD have been incorporated into this solver, and thus programming was entirely made in CUDA/C++.

The hybrid noise prediction chain is completed by implementing the FW-H analogy. The permeable version of this analogy, in the frequency domain, was developed and coupled with the PUMA solver. The equations governing the physical mechanisms of noise creation and transmission to far–field have been described in chapter 3. Comparison to a well–known analytical solution of the sound field from a monopole source in uniform flow verified the implementation of the FW-H integral.

Prior the continuous adjoint development, the differentiation of the FW-H integral was verified in the above mentioned analytical test case. Then, the continuous adjoint for the hybrid noise prediction tool was developed and coupled with the unsteady adjoint solver of PUMA, presented in chapter 4. With this, the final software was made able to compute the gradients of aeroacoustic objective functions in addition to aerodynamic objective functions (such as lift, drag and total pressure losses). The gradients of which could already be computed with accuracy by the method and software which were available in the beginning of this thesis (work made in the past in the PCOpt/NTUA group and included in the recent PhD thesis [132]). For the purpose of this thesis, an aeroacoustic objective function, expressed in the frequency domain, was defined to quantify the energy contained in the sound pressure spectrum. With the developed continuous adjoint method, the computation of the SDs of aeroacoustic objective functions, defined not only over all resolved frequencies but also over a specific frequency range, w.r.t. design variables, at a cost independent of their number was made available.

The FI adjoint approach, which is characterized with high accuracy, have been used to derive the SDs formula, including surface and volume integrals of the flow and adjoint fields as well as of grid sensitivities. Variations in turbulent quantities and distance from the wall have been taken into account by differentiating the SA turbulence model and the Eikonal equation. Moreover, it was shown that incorporation of the FW-H analogy in the adjoint development leads to sets of source terms in the adjoint mean–flow and adjoint SA equations. By defining the permeable FW-H surface outside the grid displacement area, a simpler mathematical development results. In addition, chapter 4 paid attention to the time window over which the simulation results (time–series) must be stored and the time window over which the objective function is evaluated. If the period of the unsteady phenomenon is constant during the optimization, the two aforementioned time windows are equal to a single period of the problem. Otherwise, the flow solution over the whole time window must be stored, with the objective time window being a part of it, resulting to memory- and time-wise expensive computations.

The developed software was applied to a series of aeroacoustic shape optimization problems. The cases with strong tonal noise component were selected, as the flow simulation was based on the URANS equations. It was shown that, in all applications, the acoustic pressure predicted by the hybrid solver is in good agreement with the one computed by a pure URANS equation model. Moreover, the good agreement between adjoint–based computed SDs and FDs verified the accuracy of the hybrid adjoint solver in a number of problems including pitching and plunging airfoils at different flow conditions. For these airfoils, omni–directional noise reductions were computed upon completion of the aeroacoustic shape optimization. Besides, an equality constraint handling method was developed based on a gradient projection with a deferred correction. The method proved to be effective in a lift–constrained noise minimization of a pitching airfoil, where it successfully retained the mean lift at its initial level while reducing the noise.

Comparison of the EA-based and adjoint–based shape optimization of a plunging airfoil in transonic flow by considering a multi–objective function (noise and lift) showed that the PCA–driven MAEA is able to provide results comparable to the adjoint, however, with a higher optimization cost, as expected. In this case, in agreement with earlier studies [115, 153], the drag and noise objective were shown to be competing.

Concerning cases with varying period during the optimization, a 2D vortex shedding cylinder in laminar flow and the rod–airfoil benchmark were considered. For the former, the optimizer successfully suppressed vortex shedding during the shape optimization. For the latter, an extensive comparison of aerodynamic and aeroacoustic results with available data in the literature verified the validity of the noise prediction tool. For such a computationally expensive case, early termination of the unsteady adjoint solution was shown to reduce the solution time by 33% and storage requirement by 60% while providing acceptable SDs w.r.t. FDs; this is very encouraging for large–scale industrial applications. An unconventional airfoil was the outcome of the optimization. This was partly due to the unconstrained optimization and the use of a high number of design variable in the parameterization. We should keep in mind that airfoils with unconventional wavy shape for this application can also be found in the literature [151]. The peak SPL was reduced by $3.7dB$ while its frequency remained untouched. In this case, sound reduction was not omni–directional. A second optimization with an objective function defined only in the area of the tonal frequency led to similar acoustic results, with a different geometry though which reveals the complexity of the design space for this problem. This, also, highlighted tonal component noise domination of such a case as well as the shortcoming of URANS solvers in computing broadband noise.

Validity of the developed tool for 3D application was shown first, by performing aeroacoustic shape optimization after verifying the acoustic results for flow around a sphere. Then, the shape of an aero–engine intake was optimized as an industrial application of the developed software. Combination of periodic boundary conditions and solution in a moving reference frame was already available for aerodynamic optimizations. This application showed that, this combination in PUMA can been extended to the aeroacoustic problems, leading to steady flow and adjoint runs, resulting to significant reduction in computational cost. The software was extended to perform the proper rotation of flow field to generate the corresponding unsteady fields required for computing the FW-H integral. It was shown that the proposed method can provide acoustic results in a good agreement with the outcome of a URANS equations' solver. In order to assure a periodic adjoint solution, a continuous circumferential distribution of receivers was used for the computation of the objective function. The first aeroacoustic optimization using a single circumferential distribution of receivers resulted to a directional noise reduction while incorporating more receivers in the second optimization successfully reduced the noise in an almost omni–directional pattern.

7.2 Novel Contributions

The novel contributions of this PhD thesis are summarized below:

- Formulation, development and verification of the unsteady continuous adjoint for a hybrid aeroacoustic prediction tool. To the authors knowledge, it is the first time in the literature that the continuous adjoint is developed for a FW-H based noise prediction tool.

- Contribution of the acoustic analogy to the adjoint Spalart-Allmaras turbulence model has been taken into account.

- The use of the permeable version of the FW-H analogy offered simpler adjoint development. However, this resulted to some terms including spatial derivatives of variations in flow variables w.r.t. the design variables. Use of the discretized form of these terms has been shown to be a way to incorporate these terms into the adjoint development.

- The proposed equality constraint handling has been shown to work properly for cases with non–linear constraints.

- This PhD thesis is the first work using PUMA for shape optimization in unsteady problems, therefore, it improved the capabilities of the unsteady adjoint solver of PUMA not only for aeroacoustic but also for aerodynamic shape optimization.

- A novel contribution is the extension of the developed tool to handle rotating cases such as the industrial application of this work. The aeroacoustic adjoint solver has been extended to be consistent with a MRF–based steady primal solution, by performing proper rotation of flow fields. By doing so, steady flow and adjoint problems were solved and this reduced the computational cost.

7.3 Publications & Conference Presentations

- M. Monfaredi, X. Trompoukis, K. Tsiakas, and K. Giannakoglou. Continuous Adjoint for Aerodynamic–Aeroacoustic Optimization Based on the Ffowcs Williams and Hawkings Analogy. In *Fluid-Structure-Sound Interactions and Control: Proceedings of the 5th Symposium on Fluid-Structure-Sound Interactions and Control 5*, 329-334. Springer, 2021. (Chapter in book)

- M. Monfaredi, X. Trompoukis, K. Tsiakas, and K. Giannakoglou. An Unsteady Aerodynamic/Aeroacoustic Optimization Framework Using Continuous Adjoint. In *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences: Proceedings of EUROGEN 2019*, 147-162. Springer, 2021. (Chapter in book)

- M. Monfaredi, V. Asouti, X. Trompoukis, K. Tsiakas, and K. Giannakoglou. Continuous Adjoint–Based Aeroacoustic Shape Optimization of an Aero–

Engine Intake. In 14^{th} *International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control. EUROGEN 2021*, 28-30 June, Athens, 2021.

- K. Giannakoglou, M. Monfaredi, V. Asouti, X. Trompoukis and K. Tsiakas. Aeroacoustic mitigation through shape optimization. In *Von Karman Institute Lectures Series on Advanced concepts for the reduction of flow-induced noise generation, propagation and transmission*, November 23-26, 2020.

- M. Monfaredi, X. Trompoukis, K. Tsiakas, and K. Giannakoglou. Unsteady Continuous Adjoint to URANS Coupled with FW-H Analogy for Aeroacoustic Shape Optimization. In *Journal of Computers and Fluids*,2021 (Accepted)

7.4 Future Work

This PhD thesis was based on the previous works by the PCOpt/NTUA regarding the PUMA code and its continuous adjoint tool for aerodynamic and hydrodynamic shape optimization, and enhanced this software with new tools and novel developments. Some suggestions for future work are listed below.

- Turbulent Boundary Layer (TBL) models [117, 46] can provide fair fast predictions of far–field sound based on low–cost RANS simulations. Therefore, the aeroacoustic module of PUMA can be extended by incorporating these models for an initial sound prediction for some applications. Developing continuous adjoint for these models is also an interesting/challenging task as TBL models are based on averaged boundary layer data.

- In order to explicitly reduce the broadband noise, optimizations must be performed on the basis of turbulence-resolving simulations. Extending PUMA and its adjoint solver to include simulations such IDDES or DDES can be considered as a future work, which aeroacoustic shape optimization could certainly profit it.

- Some interesting fields of research for unsteady adjoint applications are related to the cost reduction techniques for storing/recomputing intermediate flow solutions, needed to perform the adjoint solution backwards in time. Data compression and check pointing are two efficient methods that related works regarding them are in progress in two PhD theses in PCOpt/NTUA. However, their application in PUMA offers a new opportunity for future work. Moreover, use of temporal/spatial coarsening for the unsteady adjoint solution is an interesting cost reduction method to be considered as a future development for PUMA.

- A high fidelity simulation tool such as LES can be used to re–evaluate the acoustic performance of the URANS–based optimized shape and provide further insight to the physical mechanism of noise reduction as well as the effects of such optimization on broadband noise.

- Current continuous adjoint development of aeroacoustic problem is based on the permeable version of the FW-H analogy. However, an interesting future work is to repeat the continuous adjoint development for the non–permeable version of this analogy, giving contribution to SD expressions and adjoint boundary conditions, and compare the accuracy of the two developments.

- Combining EAs and adjoint into a two–level hybrid optimization tool, taking advantages of both, is worthy to be considered as a future work. In this case, a fist level EA–based optimization can be performed using a few number of design variables in order to obtain the initial design for the second (adjoint–based) optimization level where higher number of design variables are used.

Appendix A

Constraint Imposition using Gradient Projection with a Deferred Correction

In the constrained optimization case presented in section 6.2, a gradient projection method with an additional deferred correction term is used to impose an equality constraint. The standard gradient projection method operates by eliminating the component of the computed gradient of the objective function which is aligned with the gradient of the constraint function. This works effectively when, close to the current design, the constraint function is almost linear w.r.t. the design variables, b_n $(n = 1, ..., N)$. However, in case of a strongly non-linear constraint, even if the optimization happens to meet the constraint value at an optimization cycle, be it the first or i^{th} cycle, the method fails making solution updates still meet the constraint and gradually deviates from it. To overcome this, the standard gradient projection method is enhanced with a deferred correction.

The method is graphically explained in Fig. A.1. Let J be the objective function to be minimized, subjected to the equality constraint $C = C_{cnst}$, where C_{cnst} is a known value. The design variables $\vec{b}$ are updated using steepest descent with a step η. In cycle $i-1$, let the objective function value be J^{i-1} and the equality constraint met. The proposed deferred correction term is applied only if the newly updated solution no more meets the equality constraint. Should this be the case, in cycle $i-1$, a standard gradient projection is performed and design variables are updated as

$$\vec{b}^{i} = \vec{b}^{i-1} + \delta\vec{b}^{i-1}_{projected} \tag{A.1}$$

where

$$\delta\vec{b}^{i-1}_{projected} = -\eta \left[\nabla J^{i-1} - (\nabla J^{i-1} \cdot \nabla^{*} C^{i-1}) \nabla^{*} C^{i-1} \right] \tag{A.2}$$

where ∇ is the gradient operator and $\nabla^{*} C = \frac{\nabla C}{|\nabla C|}$. Depending on the non-linearity of the constraint function, this may cause a deviation from the constraint value in cycle i, $\Delta C^{i} = C^{i} - C_{cnst}$, as shown in Fig. A.1. ΔC^{i} becomes known after evaluating the lastly updated solution, as long as this constraint is related to the performance of the corresponding flow. The idea of the proposed method is to partly compensate this "loss" using a deferred correction $\delta\vec{b}^i_{DC}$ which is indexed with i as it will be performed in the the i^{th} cycle of the optimization loop. Since the correction must be able to partly bridge the gap between the current constraint value and the threshold (L_{cnst}), it must satisfy $\delta\vec{b}^i_{DC} \cdot \nabla C^i = \Delta C^i$. This equation, however, includes N unknowns namely the N components of $\delta\vec{b}^i_{DC}$. Therefore, $\delta\vec{b}^i_{DC}$ is chosen to be aligned with the constraint gradient i.e. $\delta\vec{b}^i_{DC} = \gamma^i \nabla^* C^i$, shown by a green arrow in Fig. A.1. γ^i can be found as $\gamma^i = \frac{\Delta C^i}{|\nabla C^i|}$.

In fact, instead of using Eq. A.1, the design variables are updated by

$$\vec{b}^{i+1} = \vec{b}^i + \delta\vec{b}^i_{corrected} \tag{A.3}$$

where

$$\delta\vec{b}^i_{corrected} = \delta\vec{b}^i_{projected} - \delta\vec{b}^i_{DC} \tag{A.4}$$

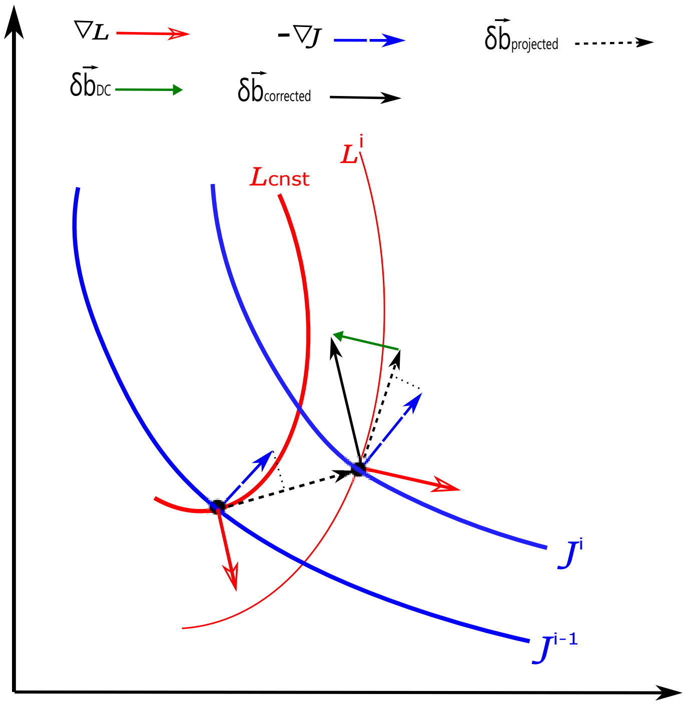


Figure A.1: Gradient Projection with a Deferred Correction. Each axes stands for a design variable. Iso-objective and constraint values are colored in blue and red, respectively. Gradients of the objective function, ∇J, and constraint function, ∇C, are shown by blue and red arrows, respectively. The green arrow represents the correction term. The black dashed arrow shows the update of the design variable after projection. The black continuous arrow shows the update of the design variable after projection and correction.

The proposed method is demonstrated in a simple example where $J = (b_1 - 2)^2 + (b_2 - 1)^2$ is the objective function to be minimized subjected to the constraint $C = (b_1)^2 + (b_2)^2 - 2 = C_{cnst}$ where $C_{cnst} = 0$. b_1 and b_2 are the design variables with the initial values of $b_1^1 = 1$ and $b_2^1 = 1$. Therefore the constraint is met in the beginning. A step size of $\eta = 0.2$ is used during the optimization. Figure A.2 illustrates the result after 10 optimization cycles and shows how the optimization follows the equality constraint iso-line. The values of the design variables, objective and constraint functions and their gradients are given in table A.1.

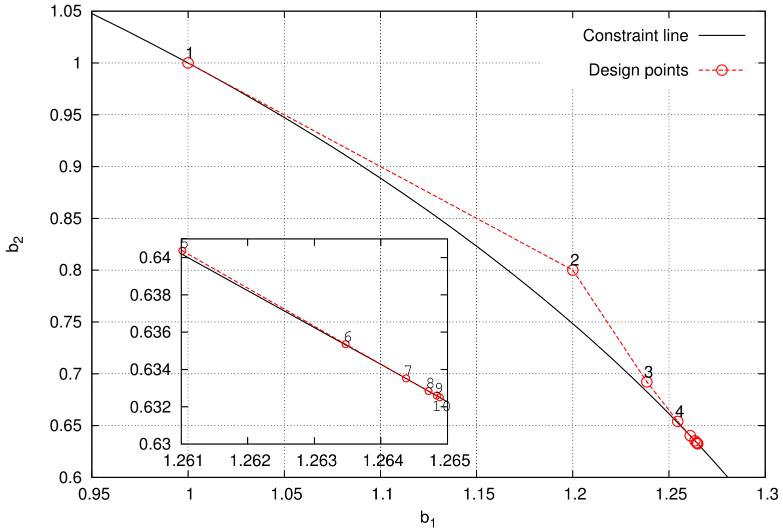


Figure A.2: Gradient Projection with a Deferred Correction. Result of the demonstration example for 10 optimization cycles, labeled by cycle ID. A closed-up view of the above in cycles 5 to 10 is also plotted.

Table A.1: Gradient Projection with a Deferred Correction. Result of the demonstration example for 10 optimization cycles.

Cycle	b_1	b_2	J	C	∇J	∇C
1	1.0000	1.0000	1.0000	$0.000e+0$	$(-2.000,\ 0.000)$	$(2.000,\ 2.000)$
2	1.2000	0.8000	0.6799	$8.000e-2$	$(-1.600,\ -0.3999)$	$(2.400,\ 1.600)$
3	1.2384	0.6923	0.6746	$1.307e-2$	$(-1.523,\ -0.6153)$	$(2.476,\ 1.384)$
4	1.2545	0.6540	0.6753	$1.719e-3$	$(-1.490,\ -0.6918)$	$(2.509,\ 1.308)$
5	1.2610	0.6403	0.6754	$2.303e-4$	$(-1.477,\ -0.7192)$	$(2.522,\ 1.280)$
6	1.2634	0.6353	0.6754	$3.108e-5$	$(-1.473,\ -0.7292)$	$(2.526,\ 1.270)$
7	1.2643	0.6335	0.6754	$4.199e-6$	$(-1.471,\ -0.7329)$	$(2.528,\ 1.267)$
8	1.2647	0.6328	0.6754	$5.672e-7$	$(-1.470,\ -0.7343)$	$(2.529,\ 1.265)$
9	1.2648	0.6325	0.6754	$7.662e-8$	$(-1.470,\ -0.7348)$	$(2.529,\ 1.265)$
10	1.2648	0.6325	0.6754	$1.035e-8$	$(-1.470,\ -0.7349)$	$(2.529,\ 1.265)$

Appendix B

Differentiation of the Analytical Solution for Monopole Source in 3D Flow

In order to validate the part of the code that differentiates the FW–H integral, in chapter 5, the variation in the objective function, Eq. 4.4 , w.r.t. the location of the monopole source, $\vec{x}_s$, is computed by the code and compared with analytical differentiation. The latter, for 3D cases, is detailed in this Appendix.

The complex velocity potential of the case is given by Eq. 5.2.1 . In what follows, receivers location, $\vec{x}_r$, is replaced by $\vec{x}$. The perturbed flow field is achieved as follows

$$\begin{aligned} p' &= \frac{\rho_0 A}{4\pi r^*}\left[sin(\omega t - kr^+)(\omega - u_{\infty 1}k\frac{\partial r^*}{\partial x_1}) + cos(\omega t - kr^+)\frac{u_{\infty 1}}{r^*}\frac{\partial r^*}{\partial x_1}\right] \\ u_i' &= \frac{A}{4\pi r^*}\left[k\frac{\partial r^+}{\partial x_i}sin(\omega t - kr^+) - \frac{1}{r^*}\frac{\partial r^*}{\partial x_i}cos(\omega t - kr^+)\right] \\ \rho' &= \frac{p'}{c_\infty^2} \end{aligned}$$

which are enough to compute the derivatives of J by performing FDs. In the second method, the FW–H integral is differentiated as follows.

$$\frac{\partial J}{\partial b_n} = -\frac{1}{T_s}\int\limits_{T_s}\oint\limits_{f=0}\left[(A_k + B_k)\frac{\partial F_k}{\partial b_n} + (C + D)\frac{\partial Q}{\partial b_n}\right] dsdt$$

where

$$
\begin{aligned}
A_k &= \int\limits_{\omega} \left(\frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|} \left(\frac{\partial \hat{G}}{\partial x_k} \right)_{\mathrm{Re}} + \frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|} \left(\frac{\partial \hat{G}}{\partial x_k} \right)_{\mathrm{Im}} \right) (\cos(2\pi\omega t)) d\omega \\
B_k &= \int\limits_{\omega} \left(\frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|} \left(\frac{\partial \hat{G}}{\partial x_k} \right)_{\mathrm{Im}} + \frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|} \left(\frac{\partial \hat{G}}{\partial x_k} \right)_{\mathrm{Re}} \right) (\sin(2\pi\omega t)) d\omega \\
C &= \int\limits_{\omega} \omega \left(\frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|} \hat{G}_{\mathrm{Re}} - \frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|} \hat{G}_{\mathrm{Im}} \right) \cos(2\pi\omega t) d\omega \\
D &= \int\limits_{\omega} \omega \left(\frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|} \hat{G}_{\mathrm{Im}} - \frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|} \hat{G}_{\mathrm{Re}} \right) \sin(2\pi\omega t) d\omega
\end{aligned}
$$

Please note that b_n are the x_{s_n} coordinates, so

$$
\begin{aligned}
\frac{\partial F_k}{\partial x_{s_n}} &= \left[\delta_{kn} \frac{\partial p}{\partial x_{s_n}} + (v_k - 2v_{\infty k}) \, (v_j \frac{\partial \rho}{\partial x_{s_n}} + \rho \frac{\partial v_j}{\partial x_{s_n}}) + \rho u_j \frac{\partial v_k}{\partial x_{s_n}} \right] n_j^{\text{FWH}} \\
\frac{\partial Q}{\partial x_{s_n}} &= (\rho \frac{\partial v_k}{\partial x_{s_n}} + v_k \frac{\partial \rho}{\partial x_{s_n}}) n_k^{\text{FWH}}
\end{aligned}
$$

Next, the variation in p, $\vec{u}$ and ρ w.r.t. x_{s_n} should be found based on the analytical solution

$$
\begin{aligned}
\frac{\partial p'}{\partial x_{s_n}} &= \frac{-1}{r^*} p' \frac{\partial r^*}{\partial x_{s_n}} \\
&+ \frac{\rho_0 A}{4\pi r^*} \Big\{ - k \frac{\partial r^+}{\partial x_{s_n}} cos(\omega t - kr^+)(\omega - v_{\infty 1} k \frac{\partial r^*}{\partial x_1}) \\
&\qquad - v_{\infty 1} k \frac{\partial}{\partial x_{s_n}} (\frac{\partial r^*}{\partial x_1}) sin(\omega t - kr^+) + k \frac{\partial r^+}{\partial x_{s_n}} sin(\omega t - kr^+) \frac{v_{\infty 1}}{r^*} \frac{\partial r^*}{\partial x_1} \\
&\qquad + cos(\omega t - kr^+)[- \frac{v_{\infty 1}}{r^{*2}} \frac{\partial r^*}{\partial x_{s_n}} \frac{\partial r^*}{\partial x_1} + \frac{v_{\infty 1}}{r^*} \frac{\partial}{\partial x_{s_n}} (\frac{\partial r^*}{\partial x_1})] \Big\}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial v'_i}{\partial x_{s_n}} = \frac{-1}{r^*} v'_i \frac{\partial r^*}{\partial x_{s_n}} + \frac{A}{4\pi r^*} \Big\{ & k \frac{\partial}{\partial x_{s_n}} (\frac{\partial r^+}{\partial x_i}) sin(\omega t - kr^+) - (k^2 \frac{\partial r^+}{\partial x_i} \frac{\partial r^+}{\partial x_{s_n}} + \frac{1}{r^{*2}} \frac{\partial r^*}{\partial x_{s_n}} \frac{\partial r^*}{\partial x_i} \\
& - \frac{1}{r^*} \frac{\partial}{\partial x_{s_n}} (\frac{\partial r^*}{\partial x_i})) cos(\omega t - kr^+) - \frac{1}{r^*} \frac{\partial r^*}{\partial x_i} k \frac{\partial r^+}{\partial x_{s_n}} sin(\omega t - kr^+) \Big\}
\end{aligned}
$$

where the following relations hold

$$\frac{\partial r^*}{\partial x_i} = \frac{(\vec{M} \cdot \vec{r})M_i + \beta^2(x_i - x_{s_i})}{r^*}$$

$$\frac{\partial r^*}{\partial x_{s_i}} = -\frac{\partial r^*}{\partial x_i}$$

$$\frac{\partial r^+}{\partial x_i} = \frac{1}{\beta^2}(\frac{\partial r^*}{\partial x_i} - M_i)$$

$$\frac{\partial r^+}{\partial x_{s_i}} = \frac{1}{\beta^2}(\frac{\partial r^*}{\partial x_i} + M_i)$$

$$\frac{\partial}{\partial x_{s_i}}(\frac{\partial r^*}{\partial x_k}) = -\frac{1}{r^*}\frac{\partial r^*}{\partial x_k}\frac{\partial r^*}{\partial x_{s_i}} - \frac{1}{r^*}(M_i M_k + \beta^2 \delta_{ik})$$

Bibliography

[1] Madeleine project homepage. https://www.madeleine-project.eu/.

[2] SmartAnswer project homepage. https://www.h2020-smartanswer.eu/.

[3] The EASY (Evolutionary Algorithms SYstem) software, http://velos0.ltt.mech.ntua.gr/EASY, 2008.

[4] W. Anderson, J. Newmann, D. Whitfield, and E. Nielsen. Sensitivity Analysis for Navier-Stokes Equations on Unstructured Meshes using Complex Variables. AIAA Journal, 39(1):56–63, 2001.

[5] W. Anderson and V. Venkatakrishnan. Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation. Computers & Fluids, 28(4):443 – 480, 1999.

[6] V.G. Asouti. Aerodynamic analysis and design methods at high and low speed flows, on multiprocessor platforms. PhD thesis, National Technical University of Athens, 2009.

[7] V.G. Asouti, X.S. Trompoukis, I.C. Kampolis, and K.C. Giannakoglou. Unsteady cfd computations using vertex-centered finite volumes for unstructured grids on graphics processing units. International Journal for Numerical Methods in Fluids, 67(2):232–246, 2011.

[8] H.M. Atassi, M. Dusey, and C.M. Davis. Acoustic radiation from a thin airfoil in nonuniform subsonic flows. AIAA journal, 31(1):12–19, 1993.

[9] T. Barth and C. Jespersen. The Design and Application of Upwind Schemes on Unstructured Meshes. AIAA paper, 0366, 02 1989.

[10] R. Bellman. Dynamic Programming. Dover Publications, Inc., New York, NY, USA, 2003.

[11] A. Bozorgi, L. Siozos-Rousoulis, S.A. Nourbakhsh, and Gh. Ghorbaniasl. A two-dimensional solution of the fw-h equation for rectilinear motion of sources. Journal of Sound and Vibration, 388:216–229, 2017.

[12] T. Brandvik and G. Pullan. Acceleration of a 3D Euler Solver Using Commodity Graphics Hardware. In 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 7 January–10 January, 2008.

[13] T. Brandvik and G. Pullan. An accelerated 3d navier–stokes solver for flows in turbomachines. Journal of turbomachinery, 133(2), 2011.

[14] K.S. Brentner and F. Farassat. Analytical comparison of the acoustic analogy and kirchhoff formulation for moving surfaces. AIAA journal, 36(8):1379–1386, 1998.

[15] D. Büche, N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with Gaussian process fitness function models. IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews, 35(2):183–194, May 2005.

[16] D. Buche, N.N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 35(2):183–194, 2005.

[17] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, and P. Hanrahan. Brook for gpus: stream computing on graphics hardware. ACM transactions on graphics (TOG), 23(3):777–786, 2004.

[18] A. Bueno-Orovio, C. Castro, F. Palacios, and E. Zuazua. Continuous adjoint approach for the Spalart-Allmaras model in aerodynamic optimization. AIAA journal, 50(3):631–646, 2012.

[19] A. Capitao Patrao, D. Lindblad, and T. Grönstedt. Aerodynamic and aeroacoustic comparison of optimized high-speed propeller blades. In 2018 Joint Propulsion Conference, page 4658, 2018.

[20] J. Christophe. Application of hybrid methods to high frequency aeroacoustics. Université Libre de Bruxelles, 2011.

[21] NVIDIA Corporation. NVIDIA CUDA Homepage. developer.nvidia.com/cuda-zone, 2021.

[22] J.S. Cox, K.S. Brentner, and C.L. Rumsey. Computation of vortex shedding and radiated sound for a circular cylinder: subcritical to transcritical reynolds numbers. Theoretical and Computational Fluid Dynamics, 12(4):233–253, 1998.

[23] N. Curle. The influence of solid boundaries upon aerodynamic sound. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 231(1187):505–514, 1955.

[24] P Di Francescantonio. A new boundary integral formulation for the prediction of sound radiation. Journal of Sound and Vibration, 202(4):491–509, 1997.

[25] T. Economon, F. Palacios, and J. Alonso. A coupled-adjoint method for aerodynamic and aeroacoustic optimization. In 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, page 5598, 2012.

[26] H. Elbanna and L. Carlson. Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes. In 27th Aerospace Sciences Meeting, Reno, NV, USA, 9 January - 12 January 1989.

[27] M. Emmerich, A. Giotis, M. Ozdemir, Bäck T., and K.C. Giannakoglou. Metamodel-assisted evolution strategies. In Parallel Problem Solving from Nature – PPSN VII, pages 361–370, Granada, Spain, 2002. Springer.

[28] E. Fabiano and D. Mavriplis. Adjoint-based aeroacoustic design-optimization of flexible rotors in forward flight. Journal of the American Helicopter Society, 62(4):1–17, 2017.

[29] E. Fabiano, A. Mishra, D. Mavriplis, and K. Mani. Time-dependent aeroacoustic adjoint-based shape optimization of helicopter rotors in forward flight. In 57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, page 1910, 2016.

[30] F. Farassat. Derivation of Formulations 1 and 1A of Farassat. Technical Report TM-2007-214853, NASA, 2007.

[31] F. Farassat and M.K. Myers. Extension of kirchhoff's formula to radiation from moving surfaces. Journal of Sound and Vibration, 123(3):451–460, 1988.

[32] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. Computer Methods in Applied Mechanics and Engineering, 163(1-4):231–245, 1998.

[33] R. Fletcher. Practical Methods of Optimization; (2nd Ed.). New York, NY, USA.

[34] A. Fosso Pouangué, M. Sanjosé, S. Moreau, G. Daviller, and H. Deniau. Subsonic jet noise simulations using both structured and unstructured grids. AIAA Journal, 53(1):55–69, 2015.

[35] N. Foster and G. Dulikravich. Three-Dimensional Aerodynamic Shape Optimization Using Genetic and Gradient Search Algorithms. Journal of Spacecraft and Rockets, 34(1):36–42, 1997.

[36] H.M. Frank and C. Munz. Direct aeroacoustic simulation of acoustic feedback phenomena on a side-view mirror. Journal of Sound and Vibration, 371:132–149, 2016.

[37] J.B. Frenud. Noise sources in a low-reynolds-number turbulent jet at mach 0.9. Journal of Fluid Mechanics, 438:277{305, 2001.

[38] J.B. Freund. Adjoint-based optimization for understanding and suppressing jet noise. Journal of Sound and Vibration, 330(17):4114–4122, 2011.

[39] S. Galdeano, S. Barré, and N. Réau. Noise radiated by a rod-airfoil configuration using des and the Ffowcs-Williams & Hawkings analogy. In 16th AIAA/CEAS Aeroacoustics Conference, page 3702, 2010.

[40] Gh. Ghorbaniasl and Ch. Lacor. A moving medium formulation for prediction of propeller noise at incidence. Journal of Sound and Vibration, 331(1):117–137, 2012.

[41] K.C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. Progress in Aerospace Sciences, 38(1):43–76, 2002.

[42] K.C. Giannakoglou, M. Monfaredi, V.G. Asouti, X.S. Trompoukis, and K.T. Tsiakas. Aeroacoustic mitigation through shape optimization. In Von Karman Institute Lectures Series on Advanced concepts for the reduction of flow-induced noise generation, propagation and transmission. 2020.

[43] M. Giles and N. Pierce. An introduction to the adjoint approach to design. Flow, Turbulence and Combustion, 65:393–415, 2000.

[44] J. Gill, X. Zhang, and P. Joseph. Single velocity-component modeling of leading edge turbulence interaction noise. The Journal of the Acoustical Society of America, 137(6):3209–3220, 2015.

[45] J.C. Giret, A. Sengissen, S. Moreau, M. Sanjosé, and J.C. Jouhaud. Noise source analysis of a rod–airfoil configuration using unstructured large-eddy simulation. AIAA Journal, 53(4):1062–1077, 2015.

[46] M. Goody. Empirical spectral model of surface pressure fluctuations. AIAA journal, 42(9):1788–1794, 2004.

[47] A. Griewank and A. Walther. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2008.

[48] J. Hambrey, D. Feszty, S.A. Meslioui, and J. Park. Acoustic prediction of high speed propeller noise using urans and a ffowcs williams-hawkings solver. In 35th AIAA Applied Aerodynamics Conference, page 3917, 2017.

[49] A.L. Hansell, M. Blangiardo, L. Fortunato, S. Floud, K. de Hoogh, D. Fecht, R.E. Ghosh, H.E. Laszlo, C. Pearson, L. Beale, et al. Aircraft noise and cardiovascular disease near heathrow airport in london: small area study. Bmj, 347, 2013.

[50] S.I. Hariharan, J.R. Scott, and K.L. Kreider. A potential-theoretic method for far-field sound radiation calculations. Journal of Computational Physics, 164(1):143–164, 2000.

[51] M. Hartmann, J. Ocker, T. Lemke, A. Mutzke, V. Schwarz, H. Tokuno, R. Toppinga, P. Unterlechner, and G. Wickern. Wind noise caused by the side-mirror and a-pillar of a generic vehicle model. In 18th AIAA/CEAS aeroacoustics conference (33rd AIAA aeroacoustics conference), page 2205, 2012.

[52] D. Heimann and R. Karle. A linearized euler finite-difference time-domain sound propagation model with terrain-following coordinates. The Journal of the Acoustical Society of America, 119(6):3813–3821, 2006.

[53] R. Içke, O. Baysal, L.V. Lopes, B.Y. Zhou, B. Diskin, and A. Moy. Toward adjoint-based aeroacoustic optimization for propeller and rotorcraft applications. In AIAA Aviation 2020 Forum, page 3140, 2020.

[54] M. Jacob, M. Ciardi, L. Gamet, B. Greschner, Y. Moon, and I. Vallet. Assessment of CFD broadband noise predictions on a rod-airfoil benchmark computation. In 14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference), page 2899, 2008.

[55] M.C. Jacob, J. Boudet, D. Casalino, and M. Michard. A rod-airfoil experiment as a benchmark for broadband noise modeling. Theoretical and Computational Fluid Dynamics, 19(3):171–196, 2005.

[56] A. Jameson. Aerodynamic design via control theory. Journal of scientific computing, 3(3):233–260, 1988.

[57] A. Jameson. Aerodynamic Design via Control Theory. Journal of Scientific Computing, 3(3):233–260, 1988.

[58] A. Jameson and J. Reuther. Control Theory based Airfoil Design using the Euler Equations. In 5th Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, FL, USA, 7 September - 9 September 1994.

[59] H.K. Jawahar, S.A. Ali, M. Azarpeyvand, and C.R. da Silva. Aerodynamic and aeroacoustic performance of high-lift airfoil fitted with slat cove fillers. Journal of Sound and Vibration, 479:115347, 2020.

[60] S. Kallas, M. Geoghegan-Quinn, M. Darecki, C. Edelstenne, T. Enders, E. Fernandez, and P. Hartman. Flightpath 2050 europe's vision for aviation. Report of the high level group on aviation research, European Commission, Brussels, Belgium, Report No. EUR, 98, 2011.

[61] M. Kaltenbacher, M. Escobar, S. Becker, and I. Ali. Numerical simulation of flow-induced noise using LES/SAS and Lighthill's acoustic analogy. International journal for numerical methods in fluids, 63(9):1103–1122, 2010.

[62] M. Kaltenbacher, A. Hüppe, A. Reppenhagen, M. Tautz, S. Becker, and W. Kuehnel. Computational aeroacoustics for HVAC systems utilizing a hybrid approach. SAE International Journal of Passenger Cars-Mechanical Systems, 9(2016-01-1808):1047–1052, 2016.

[63] I.C. Kampolis and K.C. Giannakoglou. A multilevel approach to single- and multi-objective aerodynamic optimization. Computer Methods in Applied Mechanics and Engineering, 197(33-40):2963–2975, 2008.

[64] I.C. Kampolis, X.S. Trompoukis, V.G. Asouti, and K.C. Giannakoglou. CFD-based analysis and two-level aerodynamic optimization on graphics processing units. Computer Methods in Applied Mechanics and Engineering, 199(9-12):712–722, 2010.

[65] C.S. Kapellos. The continuous adjoint method for Automotive aeroacoustic shape optimization. PhD thesis, NTUA, 2020.

[66] C.S. Kapellos, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou, and M. Hartmann. The unsteady continuous adjoint method for minimizing flow-induced sound radiation. Journal of Computational Physics, 392:368–384, 2019.

[67] K.T. Kapsoulis, D.H.and Tsiakas, X.S. Trompoukis, V.G. Asouti, and K.C. Giannakoglou. Evolutionary multi-objective optimization assisted by metamodels, kernel PCA and multi-criteria decision making techniques with applications in aerodynamics. Applied Soft Computing, 64:1–13, 2018.

[68] M. K Karakasis and K.C. Giannakoglou. On the use of metamodel-assisted, multi-objective evolutionary algorithms. Engineering Optimization, 38(8):941–957, 2006.

[69] M.K. Karakasis, A.P. Giotis, and K.C. Giannakoglou. Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization. International Journal for Numerical Methods in Fluids, 43(10-11):1149–1166, 2003.

[70] M.K. Karakasis, D.G. Koubogiannis, and K.C. Giannakoglou. Hierarchical distributed evolutionary algorithms in shape optimization. International Journal for Numerical Methods in Fluids, 53(3):455–469, 2007.

[71] I.S. Kavvadias. Continuous adjoint methods for steady and unsteady turbulent flows with emphasis on the accuracy of sensitivity derivatives. PhD thesis, National Technical University of Athens, 2016.

[72] I.S. Kavvadias, E.M. Papoutsis-Kiachagias, G. Dimitrakopoulos, and K.C. Giannakoglou. The continuous adjoint approach to the k–ω sst turbulence model with applications in shape optimization. Engineering Optimization, 47(11):1523–1542, 2015.

[73] I.S. Kavvadias, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization. Journal of Computational Physics, 301:1–18, 2015.

[74] M. R. Khorrami, B. A. Singer, and M.E. Berkman. Time-accurate simulations and acoustic analysis of slat free shear layer. AIAA journal, 40(7):1284–1291, 2002.

[75] A. Kierkegaard, S. Boij, and G. Efraimsson. A frequency domain linearized navier–stokes equations approach to acoustic propagation in flow ducts with sharp edges. The Journal of the Acoustical Society of America, 127(2):710–719, 2010.

[76] F. Kuznik, C. Obrecht, G. Rusaouen, and J. Roux. LBM based flow simulation using GPU computing processor. Computers & Mathematics with Applications, 59(7):2380–2392, 2010.

[77] S. Kyriacou. Evolutionary Algorithm-based Design-Optimization Methods in Turbomachinery. PhD thesis, National Technical University of Athens, 2013.

[78] 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, 134:71–90, 07 1996.

[79] M.J. Lighthill. On sound generated aerodynamically II. turbulence as a source of sound. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 222(1148):1–32, 1954.

[80] J. Lions. Optimal Control of Systems Governed by Partial Differential Equations. Springer-Verlang, New York, NY, USA, 1971.

[81] H. Liu, M. Azarpeyvand, J. Wei, and Z. Qu. Tandem cylinder aerodynamic sound control using porous coating. Journal of Sound and Vibration, 334:190–201, 2015.

[82] D. Lockard and J. Casper. Permeable surface corrections for ffowcs williams and hawkings integrals. In 11th AIAA/CEAS Aeroacoustics Conference, page 2995, 2005.

[83] D.P. Lockard. An efficient, two-dimensional implementation of the Ffowcs Williams and Hawkings equation. Journal of Sound and Vibration, 229(4):897–911, 2000.

[84] C. Lozano, E. Andrés, M. Martin, and P. Bitrián. Domain versus boundary computation of flow sensitivities with the continuous adjoint method for aerodynamic shape optimization problems. Numerical Methods in Fluids, 70(10):1305–1323, 2012.

[85] A.S. Lyrintzis. Surface integral methods in computational aeroacoustics | from the (cfd) near-field to the (acoustic) far-field. International journal of aeroacoustics, 2(2):95–128, 2003.

[86] D. Marinc and H. Foysi. Investigation of a continuous adjoint-based optimization procedure for aeroacoustic control of plane jets. International journal of heat and fluid flow, 38:200–212, 2012.

[87] J. Martins, P. Sturdza, and J. Alonso. The Complex-step Derivative Approximation. ACM Transactions on Mathematical Software, 29(3):245–262, 2003.

[88] S. Mendez, M. Shoeybi, S.K. Lele, and P. Moin. On the use of the ffowcs williams-hawkings equation to predict far-field jet noise from large-eddy simulations. International Journal of Aeroacoustics, 12(1-2):1–20, 2013.

[89] M. Monfaredi, X. S. Trompoukis, K. T. Tsiakas, and K. C. Giannakoglou. An Unsteady Aerodynamic/Aeroacoustic Optimization Framework Using Continuous Adjoint, pages 147–162. Springer International Publishing, Cham, 2021.

[90] M. Monfaredi, X.S. Trompoukis, K.T. Tsiakas, and K.C. Giannakoglou. Continuous adjoint for aerodynamic-aeroacoustic optimization based on the ffowcs williams and hawkings analogy. In Fluid-Structure-Sound Interactions and Control: Proceedings of the 5th Symposium on Fluid-Structure-Sound Interactions and Control 5, pages 329–334. Springer, 2021.

[91] D.C. Montgomery. Design and Analysis of Experiments. John Wiley & Sons, New York, USA, 6th edition, 2005.

[92] T. Münzel, T. Gori, W. Babisch, and M. Basner. Cardiovascular effects of environmental noise exposure. European Heart Journal, 35(13):829–836, 2014.

[93] S. Nadarajah and A. Jameson. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. AIAA Paper, 25(30), 2001.

[94] S. Nagarajan, S. Hahn, and S. Lele. Prediction of sound generated by a pitching airfoil: a comparison of rans and les. In 12th AIAA/CEAS Aeroacoustics Conference (27th AIAA Aeroacoustics Conference), page 2516, 2006.

[95] A. Najafi-Yazdi, G.A. Brès, and L. Mongeau. An acoustic analogy formulation for moving sources in uniformly moving media. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 467(2125):144–165, 2011.

[96] S. Nimmagadda, T.D. Economon, J.J. Alonso, C. Silva, B.Y. Zhou, and T. Albring. Low-cost unsteady discrete adjoints for aeroacoustic optimization using temporal and spatial coarsening techniques. In 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, page 1911, 2018.

[97] J. Nocedal and S. Wright. Numerical Optimization. Springer-Verlang, New York, NY, USA, 2006.

[98] D.I. Papadimitriou and K.C. Giannakoglou. A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. Computers & Fluids, 36(2):325–341, 2007.

[99] D.I. Papadimitriou and K.C. Giannakoglou. A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. Computers & Fluids, 36(2):325–341, 2007.

[100] D.I. Papadimitriou and K.C. Giannakoglou. Total pressure loss minimization in turbomachinery cascades using a new continuous adjoint formulation. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, 221(6):865–872, 2007.

[101] N. Papaxanthos, E. Perrey-Debain, S. Bennouna, B. Ouedraogo, S. Moreau, and J.M. Ville. Pressure-based integral formulations of Lighthill–Curle's analogy for internal aeroacoustics at low mach numbers. Journal of Sound and Vibration, 393:176–186, 2017.

[102] E. M. Papoutsis-Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. Archives of Computational Methods in Engineering, 23(2):255–299, 2016.

[103] E.M. Papoutsis-Kiachagias. Adjoint Methods for Turbulent Flows, Applied to Shape or Topology Optimization and Robust Design. PhD thesis, National Technical University of Athens, 2013.

[104] E.M. Papoutsis-Kiachagias, V.G. Asouti, K.C. Giannakoglou, K. Gkagkas, S. Shimokawa, and E. Itakura. Multi-point aerodynamic shape optimization of cars based on continuous adjoint. Structural and Multidisciplinary Optimization, 59(2):675–694, 2019.

[105] E.M. Papoutsis-Kiachagias, N. Magoulas, J. Mueller, C. Othmer, and K.C. Giannakoglou. Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. Computers & Fluids, 122:223–232, 2015.

[106] E.M. Papoutsis-Kiachagias, A.S. Zymaris, I.S. Kavvadias, D.I. Papadimitriou, and K.C. Giannakoglou. The continuous adjoint approach to the k–ε turbulence model for shape optimization and optimal active control of turbulent flows. Engineering Optimization, 47(3):370–389, 2015.

[107] O. Pironneau. On optimum design in fluid mechanics. Journal of Fluid Mechanics, 64(1):97–110, 1974.

[108] O. Pironneau. Optimal Shape Design for Elliptic Systems. In System Modeling and Optimization, pages 42–66, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[109] S. Qiu and J.Y. Ying. A combined shape and liner optimization of a general aeroengine intake for tone noise reduction. Procedia Engineering, 99:5–20, 2015.

[110] Sh. Qiu. A continuous adjoint-based aeroacoustic shape optimization for multi-mode duct acoustics. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 232(21):3897–3914, 2018.

[111] Z. Rarata, G. Gabard, R. Sugimoto, J. Coupland, R.J. Astley, H. Namgoong, and P.J. Schwaller. Integrating cfd source predictions with time-domain caa for intake fan noise prediction. In 20th AIAA/CEAS Aeroacoustics Conference, page 2456, 2014.

[112] P. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. Journal of Computational Physics, 43(2):357 – 372, 1981.

[113] L.S. Rousoulis, Gh. Ghorbaniasl, and Ch. Lacor. Noise control by a rotating rod in a rod-airfoil configuration. American Institute of Aeronautics and Astronautics Inc, AIAA, 2016.

[114] S. Roy. Les and dns of multiphase flows in industrial devices: Application of high-performance computing. In Two-Phase Flow for Automotive and Power Generation Sectors, pages 223–247. Springer, 2019.

[115] M.P. Rumpfkeil and D.W. Zingg. A hybrid algorithm for far-field noise minimization. Computers & Fluids, 39(9):1516–1528, 2010.

[116] S. Salehian, L.D. Nguyen, V.V. Golubev, and R.R. Mankbadi. Comparison of far-field acoustic prediction techniques in application to tonal noise radiation of transitional airfoils. In 22nd AIAA/CEAS Aeroacoustics Conference, page 2849, 2016.

[117] R. Schlinker and R. Amiet. Helicopter rotor trailing edge noise. In 7th Aeroacoustics Conference, page 2001, 1981.

[118] S. Schoder and M. Kaltenbacher. Hybrid aeroacoustic computations: State of art and new achievements. Journal of Theoretical and Computational Acoustics, 27(04):1950020, 2019.

[119] S. Shih, D. Hixon, R. Mankbadi, A. Pilon, A. Lyrintzis, S. Shih, D. Hixon, R. Mankbadi, and A. Pilon. Evaluation of far-field jet noise prediction methods. In 35th Aerospace Sciences Meeting and Exhibit, page 282, 1997.

[120] B.A. Singer, D.P. Lockard, and K.S. Brentner. Computational aeroacoustic analysis of slat trailing-edge flow. AIAA journal, 38(9):1558–1564, 2000.

[121] L. Siozos-Rousoulis, C. Lacor, and Gh. Ghorbaniasl. A flow control technique for noise reduction of a rod-airfoil configuration. Journal of Fluids and Structures, 69:293–307, 2017.

[122] B. Spagnoli and Ch. Airiau. Adjoint analysis for noise control in a two-dimensional compressible mixing layer. Computers & fluids, 37(4):475–486, 2008.

[123] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. Recherche Aerospatiale, (1):5–21, 1994.

[124] P.R. Spalart and M.L. Shur. Variants of the ffowcs williams-hawkings equation and their coupling with simulations of hot jets. International journal of aeroacoustics, 8(5):477–491, 2009.

[125] P.R. Spalart, M.L. Shur, M.Kh. Strelets, and A.K. Travin. Initial noise predictions for rudimentary landing gear. Journal of Sound and Vibration, 330(17):4180–4195, 2011. COMPUTATIONAL AERO-ACOUSTICS (CAA) FOR AIRCRAFT NOISE PREDICTION { PART B.

[126] M. Stadler, M. B Schmitz, and W. Laufer. Gpgpu based aeroacoustic optimization of a contra-rotating fan. In Turbo Expo: Power for Land, Sea, and Air, volume 45578, page V01AT10A019. American Society of Mechanical Engineers, 2014.

[127] the Open-Source CFD Code 2018 SU2. SU2, The Open-Source CFD Code. su2code.github.io/.

[128] A. Tadamasa and M. Zangeneh. Numerical prediction of wind turbine noise. Renewable energy, 36(7):1902–1912, 2011.

[129] J. Thibault and I. Senocak. Cuda implementation of a navier-stokes solver on multi-gpu desktop platforms for incompressible flows. In 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, page 758, 2009.

[130] K.W. Thompson. Time dependent boundary conditions for hyperbolic systems. Journal of computational physics, 68(1):1–24, 1987.

[131] X.S. Trompoukis. Solving aerodynamic-aeroelastic problems on Graphics Processing Units. PhD thesis, National Technical University of Athens, 2012.

[132] K. Tsiakas. Development of Shape Parameterization Techniques, a Flow Solver and its Adjoint, for Optimization on GPUs. Turbomachinery and External Aerodynamics Applications. PhD thesis, NTUA, 2019.

[133] A. Uzun, A.S. Lyrintzis, and G.A. Blaisdell. Coupling of integral acoustics methods with les for jet noise prediction. International Journal of Aeroacoustics, 3(4):297–346, 2004.

[134] B. van Leer. Flux-Vector Splitting for the Euler Equations. In Eighth International Conference on Numerical Methods in Fluid Dynamics, pages 507–512, 1982.

[135] V. Venkatakrishnan. On the Accuracy of Limiters and Convergence to Steady State Solutions. AIAA paper, 31(08), 02 1993.

[136] C. Wagner, T. Hüttl, and P. Sagaut. Large-eddy simulation for acoustics, volume 20. Cambridge University Press, 2007.

[137] A. Walther and A. Griewank. Getting started with ADOL-C. In Combinatorial Scientific Computing, pages 181–202. Chapman-Hall CRC Computational Science, 2012.

[138] M. Wang, D. Angland, and X. Zhang. The noise generated by a landing gear wheel with hub and rim cavities. Journal of Sound and Vibration, 392:127–141, 2017.

[139] J. Weiss, J. Maruszewski, and W. Smith. Implicit Solution of Preconditioned Navier-Stokes Equations using Algebraic Multigrid. AIAA Journal, 25:29–36, 1999.

[140] J.E.F Williams and D.L. Hawkings. Sound generation by turbulence and surfaces in arbitrary motion. Philosophical Transactions for the Royal Society of London. Series A, Mathematical and Physical Sciences, pages 321–342, 1969.

[141] M.C. Wright and C.L. Morfey. On the extrapolation of acoustic waves from flow simulations with vortical out flow. International Journal of Aeroacoustics, 14(1-2):217–227, 2015.

[142] W. Yamazaki, K. Matsushima, and K. Nakahashi. Aerodynamic Design Optimization Using the Drag-Decomposition Method. AIAA Journal, 46(5):1096–1106, 2008.

[143] L. Yang and S. Nadarajah. Data Compression Algorithms for Adjoint-Based Sensitivity Studies of Unsteady Flows. In ASME 2018 5th Joint US-European Fluids Engineering Division Summer Meeting, Montreal, Quebec, Canada, July 15 - July 20 2018.

[144] Y.X. Yuan. Recent Advances in Trust Region Algorithms. Mathematical Programming, Series A and B, 151(1):249–281.

[145] A. Zarri, J. Christophe, and C.F. Schram. Low-order aeroacoustic prediction of low-speed axial fan noise. In 25th AIAA/CEAS Aeroacoustics Conference, page 2760, 2019.

[146] T. Zervogiannis. Optimization methods in aerodynamics and turbomachinery based on the adjoint technique , hybrid grids and the exact Hessian matrix. PhD thesis, National Technical University of Athens, 2011.

[147] J. Zhang, LW. Xu, and R.Z. Gao. Multi-island Genetic Algorithm Optimization of Suspension System. TELKOMNIKA Indonesian Journal of Electrical Engineering, 10, 11 2012.

[148] S. Zhong and X. Zhang. On the frequency domain formulation of the generalized sound extrapolation method. The Journal of the Acoustical Society of America, 144(1):24–31, 2018.

[149] S.Y. Zhong and X. Zhang. A sound extrapolation method for aeroacoustics far-field prediction in presence of vortical waves. J. Fluid Mech, 820:424–450, 2017.

[150] B. Zhou, T. Albring, and N. Gauger. Optimal flow actuation for separation control and noise minimization. In Proceedings of 9th International Conference on Computational Fluid Dynamics, 2016.

[151] B.Y. Zhou. Numerical Optimization for Airframe Noise Reduction. PhD thesis, RWTH Aachen University, 2018.

[152] B.Y. Zhou, T.A. Albring, N.R Gauger, T.D Economon, and J.J Alonso. An efficient unsteady aerodynamic and aeroacoustic design framework using discrete adjoint. In 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, page 3369, 2016.

[153] B.Y. Zhou, T.A. Albring, N.R. Gauger, T.D. Economon, F. Palacios, and J.J Alonso. A discrete adjoint framework for unsteady aerodynamic and aeroacoustic optimization. In 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, page 3355, 2015.

[154] B.Y. Zhou, T.A. Albring, N.R. Gauger, C. Ilario, T.D. Economon, and J.J. Alonso. A discrete adjoint approach for jet-flap interaction noise reduction. In 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, page 0130, 2017.

[155] B.Y. Zhou, N.R. Gauger, H. Yao, S.H. Peng, and L. Davidson. Towards adjoint-based broadband noise minimization using stochastic noise generation. In AIAA Scitech 2019 Forum, page 0002, 2019.

[156] B.Y. Zhou, S.R. Koh, N.R. Gauger, M. Meinke, and W. Schoeder. A discrete adjoint framework for trailing-edge noise minimization via porous material. Computers & Fluids, 172:97–108, 2018.

[157] A.S. Zymaris. Adjoint methods for the design of shapes with optimal aerodynamic performance in laminar and turbulent flows. PhD thesis, National Technical University of Athens, 2010.

[158] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart–Allmaras turbulence model for incompressible flows. Computers & Fluids, 38(8):1528–1538, 2009.

[159] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. Journal of Computational Physics, 229(13):5228–5245, 2010.