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

Defroster Nozzle Shape Optimization Using the Continuous Adjoint Method

Diploma Thesis

Lefki A. Germanou

Advisor
Kyriakos C. Giannakoglou, Professor NTUA

Athens, February 2016

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

Defroster Nozzle Shape Optimization Using the Continuous Adjoint Method

Diploma Thesis

Lefki Germanou

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, February 2016

Abstract

This thesis presents the use of the continuous adjoint method, developed by the Parallel CFD & Optimization Unit of NTUA in the OpenFOAM environment, for the shape optimization of a passenger car defroster nozzle, including experimental validation performed at Toyota Motor Europe (TME). The defroster nozzle plays a major role in the demisting–defogging of the windshield, by blowing high velocity hot air jets supplied by the HVAC (heating, ventilation and air conditioning) unit of the vehicle.

Since the basic performance requirements for a defroster nozzle are clear-up speed and clear-up pattern, the time required for dispelling condensation or frost on a windshield must be reasonable and the nozzle must have the capability to perform uniform defrosting from the bottom of the windshield to its top, so that it becomes clear without patches of condensation. In view of the above, an appropriate objective function, to be minimized, is the integral of the difference of the air velocity from a target (desirable) one over a thin control volume appropriately defined close to the windshield, inside the car cabin.

To set up the optimization problem, the shape of an existing defroster nozzle is allowed to vary using a volumetric NURBS tool developed by NTUA; the latter is also used for deforming the computational mesh at each optimization cycle, by adapting it to the changed defroster shape. The CFD analysis is based on RANS, using the k-ϵ turbulence model. The optimization loop uses the gradient of the

objective function with respect to the coordinates of the volumetric NURBS lattice, computed using the continuous adjoint method.

Experimental tests performed to measure the actual velocity pattern on the windshield include velocity measurements with a hot-wire anemometer. A convincing comparison between CFD analysis and measurements is presented. Finally, the improved demisting performance of the geometry resulted from the adjoint optimization is validated experimentally, using rapid prototyping to manufacture the defroster nozzle.

Major part of this diploma thesis was carried our in the premises of Toyota Motor Europe, in Belgium, with Mr. Antoine Delacroix as industrial advisor.

Acknowledgements

I am deeply grateful to all the people who supported me, not only during my occupation with my diploma thesis, but also throughout my undergraduate studies in NTUA. First of all, I would like to express my very profound gratitude to Prof. K. C. Giannakoglou, for being an inspiration to me since my first academic year, transmitting his knowledge and passion for his field of expertise and trusting me with this very interesting and innovative project. His advice and corrections were valuable.

A big part of this diploma thesis, was realised in the premises of Toyota Motor Europe, in the department of "Vehicle Performance Engineering". Taking this opportunity, I would like to wholeheartedly thank Mr. A. Delacroix, my industrial advisor, for his persistence and guidance throughout this endeavour. All these months, he was always giving me motivation with his enthousiasm and in every chance, he was sharing with me his knowledge and experience concerning the automotive industry. Moreover, I express my warm thanks to Mr. N. Yokoyama, my manager, for his encouraging comments and for having faith in me and in the project. I would also like to express my gratitude to Dr. K. Gkagkas for lending a hand many times in this venture.

In addition, I would like to thank Mr. V. Skaperdas and Mr. G. Fotiadis, engineers of BETA CAE Systems, for their contribution to the project and their valuable advice to me personally.

I would like to express my deepest appreciation to all the members of the Parallel CFD & Optimization Unit of National Technical University of Athens for their help throughout my studies. Particularly, I wish to express my sincere thanks to Dr. E.M. Papoutsis–Kiachagias for his continuous help and for all the time he spent to support me in this project.

This wonderful journey would have been so much harder and full with self–doubt, if not for the love, support and understanding of my friends and family. I owe many thanks to all of them, but especially to my parents and sister, who endlessly encouraged me to pursue my interests with persistence.

Acronyms

NTUA	National Technical University of Athens
PCopt	Parallel CFD & Optimization Unit
TME	Toyota Motor Europe
CFD	Computational Fluid Dynamics
CAD	Computer Aided Design
HVAC	Heating Ventilation and Air Conditioning
OpenFOAM	Open Field Operation And Manipulation
PDE	Partial Differential Equation

Contents

Chapter 1

Introduction

Automatic shape optimization loops for optimal performance, according to fluid dynamics or aerodynamic criteria, have been widely developed for aerospace applications and, nowadays, are also gaining more and more importance in the automotive industry. On the basis of the advanced progress of adjoint-based shape design, automatic optimization loop has been used to solve an interesting problem related to part of the HVAC system of a passenger car. The target was the improvement of the demisting performance of a passenger vehicle by modifying the shape of its defroster nozzle.

1.1 HVAC - Defroster Nozzle

The safety and thermal comfort of automotive passengers are the most important factors in the development of the automotive Heating Ventilation and Air Conditioning (HVAC) system, [11]. HVAC is responsible for the demisting and defrosting of the vehicle's windows and for creating/maintaining a pleasant climate inside the cabin, by controlling cabin air humidity and temperature.

The defroster nozzle, as part of the HVAC system of vehicles, plays a major role in the demisting-defrosting of the windscreen. Demisting refers the function intended to remove mist, a film of condensate on the inside face of the glazed surface of the windscreen, while defrosting refers to the function intended to eliminate frost or ice generally on the outside surface of the windscreen. The HVAC unit provides hot air to the nozzle which is, then, blowing high velocity air jets to the windscreen.

Windshield defrosting performance is an important factor on evaluating automotive HVAC systems but, on the same time, is a compulsory test according to national and international legislation since it has a significant impact on driving safety.

Particularly, the formation of frost on the windshield and front door glasses during cold season can be proved dangerous as it is veiling the driver's view and

disturbing driving. Therefore, defroster performance is seriously taken into consideration during the design of HVAC system in order to ensure the safety of the passengers.

In order to achieve an acceptable demisting performance, the HVAC system has to meet some performance requirements [2]. These are mainly the following:

1. Clear-up speed performance: The time required for dispelling condensation or frost on the windshield must be reasonable.

2. Clear-up pattern performance: The capability to perform uniform defrosting ideally on the whole surface of the windshield so that it becomes clear without being spotty, without any condensation patches.

The clear-up speed performance can be improved, to some extent by increasing the airflow volume of the blower or the heat exchange efficiency of the heater core (i.e. the temperature of the air being aimed at the windshield). At the same time, for improvement of the clear-up pattern performance, it is common that repetitive experiments are carried out with different draft shapes of defroster nozzles (air guide vanes and outlet area are very important parameters) in order to select the one providing an acceptable clear-up patterning. So far, the efficiency of the defroster depends strongly on the technicians' experience [2], [10].

However, irregular shape and complex structure makes the design difficult while experimental design method costs too much time and money. Consequently, time consuming experiments that are generally required to achieve an improved shape with better performance are gradually being replaced by CFD simulations and automatic optimization loops.

On the same time, it is known that clear–up patterning is correlated with the air velocity distribution on the windshield as mentioned in [2], [10]. Lately, distribution of airflow velocity on the surface of windshield has been investigated by numerical calculations and associated with defrosting performance resulting to be judged as adequate to reflect it, to some extent. The defroster nozzle must, therefore, be designed to provide optimal air velocity distribution. Based on this conclusion, we set this later as the optimization target, which in terms of the optimization method is translated into an objective function.

Based on some previous research [4], [10] it was concluded that CFD could be used to carry out the majority of the design process not only of the defroster nozzle, but, also, of other ducts and HVAC components. It presents significant advantages to the engineer, compared to traditional techniques based on trial and error, which depend too much on experience. One of them is that features of the airflow can be studied, resulting to deeper understanding of the flow to help determining the source of the problems. Another benefit is that the vehicle, or any prototype part is no more needed. Therefore, evaluation work can progress in parallel to the design

process. CFD can be used during the entire component design and evaluation phase, leaving actual testing for the final homologation test. Defrost testing should be reserved for the final legislative performance test. Last but not least, design space exploration is time–consuming and the improvement in shape is difficult to reach its full potential. For these reasons, in order to produce a new defroster that gives an improved velocity pattern on the windshield, a CFD–based optimization is preferred and, in this diploma thesis, this relies upon the continuous adjoint method to compute the gradient of the objective function with respect to the design variables determining the shape of the defroster duct.

1.2 CFD–based Optimization

The constituents needed for running an automated shape optimization loop include the flow (CFD) solver, the geometry parameterization (the parameters of which act as the design variables), an optimization method capable of computing the optimal values of the design variables and a way to adapt (or regenerate) the computational mesh for each candidate solution.

During the CFD–based optimization loop, the shape of the geometry under consideration is controlled by a number of design variables. For instance, these can be the coefficients of the Bezier–Bernstein polynomials parameterizing the shape of the body to be designed. The quality of the new shape produced is evaluated by computing a quantity (usually an integral), known as the objective or cost function of the optimization problem. For example, in the case of the aerodynamic design of an airfoil (which is the standard problem used to validate all relevant methods), this could be either the drag or lift coefficient of the airfoil. The objective function value depends on the values of the flow variables, obtained through the solution of the flow equations used to simulate the flow inside (internal aerodynamics) or around (external aerodynamics) the shape under consideration. The flow equations can also be considered as the flow constraints of the optimization problem and may include the solution of the incompressible or compressible, steady or unsteady, inviscid or viscous flow equations. Each optimization problem aims at the minimization of the objective function with respect to (w.r.t.) the design variables. Maximization problems can easily be reformulated to minimization ones and can be solved using the same tools.

1.2.1 Optimization Methods

CFD–based optimization methods can be classified into two main categories, i.e. deterministic and stochastic optimization methods. Deterministic methods, known also as gradient–based, require the calculation of the derivatives of the objective

function. To get the new values of the design variables, a steepest–descent algorithm (or Newton or quasi–Newton methods) must be used, based on the computed gradient. On the other hand, stochastic methods do not require any further information than the value of the objective function.

Stochastic optimization algorithms, [30], [31], [32], a representative example of which is an evolutionary algorithm (EA), have the significant advantage that it is almost unlikely for them to trapped into local minima, due to the fact that candidate solutions are being searched in a randomized way. However, a great number of evaluations, CFD simulations in our case, are usually necessary , provided that the optimization runs for a adequately high number of generations, before obtaining the optimal one. That increases the computational cost.

Deterministic algorithms are quite vulnerable to entrapment to local minima as the solution can be erroneously considered as the optimum, while in this is nothing more than a local extremum. However, as the direction of updating the design variables is dictated by the sensitivity derivatives, rather than a random, or even randomised choice, each cycle of the algorithm produces an improved result, formulating in general an automatic process that requires much less evaluations than those required by EAs.

Though ideally the target would be to acquire optimal solutions, especially in an industrial environment,, improved (compared to the configuration which is currently in use) solutions are highly welcome too. Thus, in this application, gradient–based optimization methods are preferred.

The efficiency of gradient–based methods mainly depends on the technique that is used to compute the sensitivity derivatives. Common techniques for that are the finite differences method (FD), the complex variable method (CV) and the direct differentiation method (DD) [33].

The gradient can be computed, in a straightforward manner, using finite differences. For a second order FD scheme, the derivatives of an objective function F w.r.t. to the design variables $\overrightarrow{b}$ are given by

$$\frac{\delta F(\overrightarrow{b})}{\delta b_n} = \frac{F(b_1, b_2, ..., b_n + \epsilon, ..., b_{N-1}, b_N) - F(b_1, b_2, ..., b_n - \epsilon, ..., b_{N-1}, b_N)}{2\epsilon} \quad (1.1)$$

Even though it is straightforward to implement, since it only requires the re–computation of the value of the objective function, the cost of FD scales with the number of the design variables, N $(b_n, n \in [1, N])$, making its use unfeasible in large scale optimization problems. Moreover, the choice of the infinitesimally small quantity ϵ can affect the result and round–off errors are very common, as the two F values are too close due to the small ϵ. Consequently, a trial and error process must be employed in order to ensure that ϵ–independent derivatives have been computed.

This last disadvantage of the FD method can be circumvented by using the

complex variable method where the sensitivities are computed as

$$\frac{\delta F(\overrightarrow{b})}{\delta b_n} = \frac{Im[F(b_1, b_2, ..., b_n + i\epsilon, ..., b_{N-1}, b_N)]}{\epsilon} \tag{1.2}$$

where $i = \sqrt{-1}$. Since the sensitivity derivatives are no longer computed as a function of the difference of two very close values of F, round–off errors cease to exist, providing with an non ϵ–dependent result. Nevertheless, the cost of CV is still scaled linearly with N. An added problem is the fact that the use of complex variables requires an intervention into the source code of the flow solver, so that it can handle complex variables.

The next alternative is direct differentiation, according to which the flow equations are differentated w.r.t. to $\overrightarrow{b}$ and the resulting N linear systems are solved for the derivatives of the flow variables w.r.t. the design ones. The sensitivity derivatives are expressed as a function of these fields. DD does not only scale with N but it is also harder to implement than FD. However, this method is indispensable part of algoritms used to compute high–order sensitivity derivatives.

These methods can be really accurate or indispensable in some cases as explained above, however they share the same important drawback. Their cost scales linearly with the number of the design variables, N , making them impractical for large scale optimization problems. Thankfully, there is a fourth alternative, the adjoint method,[34], [35], which is overcoming this issue due to the fact that practically, the cost of computing the necessary sensitivity derivatives independent from the number of the design variables. This is making the adjoint method the most appropriate method to compute gradients.

1.2.2 Adjoint Method

The adjoint method is a mathematical tool that computes the gradient of the objective function w.r.t. the design variables by also taking into consideration that each solution of the optimization problem has to satisfy the Navier–Stokes equations.

Let F be the objective function which is generally expressed as

$$F = F(\overrightarrow{U}(\overrightarrow{b}), \overrightarrow{b}) \tag{1.3}$$

where $\overrightarrow{U}$ is the vector of the flow variables and $\overrightarrow{b}$ the vector of the design variables, which in the case of shape optimization define, for example, the shape of an airfoil. Practically, any change of the values of the design variables $\overrightarrow{b}$, modifies the shape of the airfoil, resulting to a new flow field $\overrightarrow{U}$ around it and to a new value

of the objective function. Consequently, the variation of F w.r.t. $\overrightarrow{b}$ is

$$\frac{dF}{d\overrightarrow{b}} = \frac{\partial F}{\partial \overrightarrow{b}} + \frac{\partial F}{\partial \overrightarrow{U}} \frac{d\overrightarrow{U}}{d\overrightarrow{b}} \tag{1.4}$$

The augmented objective function is introduced

$$F_{aug} = F + \overrightarrow{\psi}^T \overrightarrow{R} \tag{1.5}$$

where $\overrightarrow{\psi}$ is the vector of Lagrange multipliers or the so–called adjoint variables. The system of equations that describe the physical problem are noted as $\overrightarrow{R} = \overrightarrow{R}(\overrightarrow{U}, \overrightarrow{b})$. Since $\overrightarrow{R} = 0$, then $F = F_{aug}$. As a consequence, minimizing F is equivalent to minimizing F_{aug};

$$\frac{dF_{aug}}{d\overrightarrow{b}} = \frac{dF}{d\overrightarrow{b}} + \overrightarrow{\psi}^T \frac{d\overrightarrow{R}}{d\overrightarrow{b}} \tag{1.6}$$

$$\frac{d\overrightarrow{R}}{d\overrightarrow{b}} = \frac{\partial R}{\partial \overrightarrow{b}} + \frac{\partial R}{\partial \overrightarrow{U}} \frac{d\overrightarrow{U}}{d\overrightarrow{b}} = 0 \tag{1.7}$$

Equation 1.6, using 1.4 and 1.7, turns into

$$\begin{aligned} \frac{dF_{aug}}{d\overrightarrow{b}} &= \frac{\partial F}{\partial \overrightarrow{b}} + \frac{\partial F}{\partial \overrightarrow{U}} \frac{d\overrightarrow{U}}{d\overrightarrow{b}} + \overrightarrow{\psi}^T \left(\frac{\partial R}{\partial \overrightarrow{b}} + \frac{\partial R}{\partial \overrightarrow{U}} \frac{d\overrightarrow{U}}{d\overrightarrow{b}} \right) \\ &= \left(\frac{\partial F}{\partial \overrightarrow{U}} + \overrightarrow{\psi}^T \frac{\partial R}{\partial \overrightarrow{U}} \right) \frac{d\overrightarrow{U}}{d\overrightarrow{b}} + \left(\frac{\partial F}{\partial \overrightarrow{b}} + \overrightarrow{\psi}^T \frac{\partial R}{\partial \overrightarrow{b}} \right) \end{aligned} \tag{1.8}$$

Matrix $\frac{d\overrightarrow{U}}{d\overrightarrow{b}}$ has high computational cost so it is desirable to avoid computing it. Consequently, the multipliers $\overrightarrow{\psi}$ are computed instead so as to set to zero the multiplier of this matrix on the equation 1.8. After having computed $\overrightarrow{\psi}$ the Lagrange multipliers are used on the computation of the sensitivities according to

$$\frac{dF}{d\overrightarrow{b}} = \frac{dF_{aug}}{d\overrightarrow{b}} = \frac{\partial F}{\partial \overrightarrow{b}} + \overrightarrow{\psi}^T \frac{\partial \overrightarrow{R}}{\partial \overrightarrow{b}} \tag{1.9}$$

The computation of the sensitivity derivatives through equation 1.9 demands the solution of only one linear system for the computation of $\overrightarrow{\psi}$ while the computation through 1.4 demands the solution of N systems $\frac{d\overrightarrow{R}}{d\overrightarrow{b}}$ for the computation of the matrix $\frac{d\overrightarrow{U}}{d\overrightarrow{b}}$ (the latter refers to the aforementioned DD method in sec. 1.2.1).

The adjoint method appears in two, substantially different, variants:: the discrete [36], [37], [38] and the continuous, [34], [35], [39] adjoint method. Discrete adjoint includes the discretization of the equations of $\overrightarrow{R} = 0$ and, then, their integration in equations 1.8 and 1.9, [26]. The Lagrange multipliers are also in discrete form. On the contrary, according to the continuous adjoint method, the flow equations in

continuous form are firstly differentiated and the adjoint PDEs are derived as closed form expressions. The latter are, then, discretized and numerically solved.

In other words, the discrete adjoint approach works with the algebraic equations that come from the discretisation of the fluid dynamic equations while, in the continuous adjoint approach, the adjoint PDEs are formulated and then discretised and solved. The adjoint solver developed by PCOpt/NTUA on the OpenFOAM software is using the continuous adjoint method and this software is used in this diploma thesis.

The above equations that described the formulation of the adjoint problem in general, actually refer to the discrete adjoint method. This approach is preferred at this point as it is considered easier for the reader to become familiar with the adjoint method and its advantages. The continuous adjoint method will be analyzed in chapter 2.

1.2.3 Shape Morphing Methods

The computed sensitivity derivatives from the adjoint solver are translated into displacements of the control points using algorithms such as steepest descent, BFGS, Newton, quasi–Newton and others [24], [29], [13]. In CFD shape optimization, the design variables can be the nodes of the surface mesh or more usually, control points that parameterize the desired geometry. So the optimization loop uses the gradient of the objective function w.r.t. the coordinates of control points, computed using the continuous adjoint method; then, after translating the derivatives into displacements of the control points, a morphing tool changes the surface according to the changed control points.

In general, shape parameterization techniques can be divided into the following eight categories [7]: basis vector, domain element, discrete, analytical, free–form deformation (FFD), partial differential equation (PDE), polynomial and spline, and Computer Aided Design (CAD). Among them, only CAD–based and FFD techniques seem to be efficient and suitable enough for complex geometries.

As far as the former technique is concerned, the need to differentiate the code of the commercial CAD software appears; this, however, is most of the times not available to users. Assuming that the source code of the CAD software can be accessed, another issue is that the differentiation of large codes is not a trivial task as, even the use of automatic differentiation tools, cannor replace the necessity of complex coding. Furthermore, the result is a code that requires a lot of computer memory to perform the necessary calculations. Therefore, the calculation of the analytical sensitivity derivatives of the geometry w.r.t. the design variables could prove to be difficult within a commercial CAD environment. Nevertheless, there is ongoing academic research in this field.

The latter technique, FFD, has the advantage of parameterizing and deforming

both the surface and volume mesh. This is positive since that it eliminates the need of using a different tool to deform the volume mesh after updating the surface. Moreover the topology of the mesh is preserved and thus, remeshing is avoided and at the same time this allows the initialization of the flow from solutions obtained in the previous cycle. As a result, significant reduction in the computational cost is achieved. Examples of this method are volumetric B–splines or NURBS, Radial Basis Functions (RBFs), the harmonic coordinates method, etc. In the current application, the PCOpt/NTUA morpher that parameterizes the desired $3D$ space using volumetric B–splines is used.

1.3 Target and Structure of the Thesis

The target of this diploma thesis is the application of shape optimization methods, using the continuous adjoint method, to the defroster nozzle of a Toyota vehicle. The objective of this optimization procedure was to achieve an improved defrosting performance. To do so, a suitable objective function is, firstly, formulated to describe the engineering goal. The adjoint method provides the corresponding sensitivity derivatives that gave qualitative information about the suggested shape deformation of the defroster. Then, the coupled shape morpher translates this information appropriately to produce a new defroster nozzle geometry, giving better defrosting performance than the initial one. The whole procedure is carried out iteratively until a convergence criterion is met.

The structure of this thesis is organised in the following chapters:

- In chapter 2, the flow equations, as well as the continuous adjoint method for incompressible, steady–state flow is presented. The objective function, the adjoint equations and their boundary conditions and sensitivity derivatives are formulated. The procedures of discretising and, then, solving the equations are explained in detail.

- In chapter 3, the shape morpher is presented along with the optimization algorithm.

- In chapter 4, a test for pressure measurement and the simulation of the equivalent CFD model are presented. CFD–based results are validated through comparison with experimental measurements.

- In chapter 5, the performed velocity pattern test and the computed flow in the cabin are presented. The test and CFD correlation is validated.

- In chapter 6, results of the automated optimization process are presented. Several cases are studied since there are manufacturing and topology constraints

which are not included in the optimization loop. Those are taken into consideration only at the end of each optimization project. The most suitable new defroster nozzle is, then, manufactured using rapid prototyping techniques and its improved performance is evaluated and compared to the one of the initial shape through the presented defrost shape.

- In chapter 7, results and conclusions are summarized.

Chapter 2

The Continuous Adjoint Method for Incompressible, Steady–State Flow

As mentioned before, the target for this optimization problem, is the improvement of the demisting and defogging performance of a car. According to [4], CFD can be used with confidence to simulate defrost performance. The realistic case would be to simulate the transient, thermal, phase changing conditions that happen during the procedure of defogging the windshield. However, the engineer can evaluate the defrost performance from an interpretation of the velocity distribution close to the windshield [2], [4]. As a consequence, it is also acceptable to run isothermal steady state simulations and, in this way, simplify the problem complexity and decrease the computational cost.

The automotive cases solved in this thesis are governed by the Reynolds Averaged Navier–Stokes equations for steady–state, incompressible flow, coupled with the turbulence model equations. The k–ϵ turbulence model is applied. The mean flow and turbulence model PDEs along with their boundary conditions are referred to as the primal (or state) equations of the optimization problem.

2.1 Flow Equations

The mean flow equations [21], [6], [28], the continuity equation and the momentum equations are, respectively, written as

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{2.1}$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \nu_t\right)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] + \frac{\partial p}{\partial x_i} = 0 \qquad i = 1, 2, 3 \tag{2.2}$$

where v_i are the velocity components, p stands for static pressured divided by the constant density ρ, ν is the constant bulk viscosity and ν_t is the turbulent viscosity. Turbulence viscosity results from by the solution of the turbulence model PDEs.

Repeated indices denote summation over the spatial dimensions of the problem (Einstein convention). Equation 2.1, is therefore, a shorthand representation of: $\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} = 0$.

The turbulence model k-ϵ equations [23],[21],[28] are given by

$$R^k = v_i \frac{\partial k}{\partial x_i} - \Big(\nu + \frac{\nu_t}{P_{r_k}}\Big)\frac{\partial^2 k}{\partial x_i^2} - P + \epsilon = 0 \tag{2.3}$$

$$R^\epsilon = v_i \frac{\partial \epsilon}{\partial x_i} - \Big(\nu + \frac{\nu_t}{P_{r_\epsilon}}\Big)\frac{\partial^2 \epsilon}{\partial x_i^2} - C_1 \frac{\epsilon}{k} P + C_2 \frac{\epsilon^2}{k} = 0 \tag{2.4}$$

$$\nu_t = C_\mu \frac{k^2}{\epsilon} \tag{2.5}$$

where k is the turbulent kinetic energy, ϵ the turbulence energy dissipation term and P stands for the turbulence production term

$$P = \nu_t \Big(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\Big)\frac{\partial v_j}{\partial x_i} \tag{2.6}$$

The constant coefficients of the model are: $C_1 = 1.44, C_2 = 1.92, C_\mu = 0.09$ and the turbulent Prandtl numbers $P_{r_\epsilon} = 1.3, P_{r_k} = 1.0$

2.2 Boundary Conditions

The boundary conditions that "close" the primal problem are:

- Dirichlet boundary condition for v_i and the turbulence model variables (k,ϵ) and zero Neumann boundary condition for p, at the inlet S_I.

- Zero Neumann boundary condition for v_i and the turbulence model variables (k,ϵ) and zero Dirichlet boundary condition for p, at the outlet S_O.

- Zero Dirichlet boundary condition for v_i (no-slip condition). Empirical relations based on the friction velocity (wall functions) for the turbulence model variables (k,ϵ) and zero Neumann boundary condition for p at the walls S_W. Empirical information using the wall function technique was used to approximate viscous stresses at the wall [23], [21].

  The boundaries in the main case of interest are shown in fig. 2.1. The details of the computational model are later explained in chapter 5.

2.3 Objective Function

Usually, in aerodynamic problems, the objective function is a scalar quantity like the lift or drag force exerted on an aerodynamic body. In case there are more than one objective functions that happen to be convex, the Multi Objective Optimization (MOO) problem can be turned into a Single Objective Optimization (SOO) problem by simply adding the objectives and multiplying them with a different weight decided by the engineer.

As previously mentioned, in our case, it is desirable that the velocity pattern of the air close to the windshield (and not exactly on it, as the no–slip condition implies zero velocity on the walls) meets some criteria (as listed in 1.1) and on the same time it improves the demisting and defogging performance. In other words, the aim is to achieve more uniform distribution of the velocity and increase its magnitude on the weakest areas which is most of the times the upper part of the windshield. This target is mathematically expressed as

$$F_{obj} = \frac{1}{2}\int_{\Omega_{tar}} \left(v_i^2 - v_{tar}^2\right)^2 d\Omega_{tar} \tag{2.7}$$

The above objective function describes the wish to force the magnitude of velocity over a thin volume close to the windshield, Ω_{tar}, fig. 6.1 to reach a certain value or to have a pre–defined distribution. Ω_{tar} is often refered as target volume, in the scope of this thesis.

The starting point of the adjoint problem is the use of the convenient augmented function F_{aug} and its use, instead of using F directly, see sec. 1.2.2 and [6]. So, F_{aug} is defined as

$$F_{aug} = F + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega \tag{2.8}$$

where Ω is the computational domain, fig. 2.1a, u_i are the adjoint velocity components, v_i the primal velocity components, q is the adjoint pressure and p is the primal pressure. Since the residuals of the primal equations 2.1, 2.2 are zero, F and F_{aug} are identical.

At this point, it is important to mention that within the scope of this thesis, in the analysis of the equations following, the effect of the solution to the equations of the turbulence model is neglected, for reasons of simplicity. Consequently, the commonly known as "frozen turbulence" assumption is made. Readers interested in the differentiation of turbulence can refer to [6], [13], [15],[14], [16], [18], [17], [19].

Differentiating the augmented objective function we get

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n}\int_{\Omega} u_i R_i^v d\Omega + \frac{\delta}{\delta b_n}\int_{\Omega} q R^p d\Omega \tag{2.9}$$

Then, the Leibniz theorem, which is used for the differentiation of the volume inte-

grals with variable boundaries, is applied

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_\Omega u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_\Omega q \frac{\partial R^p}{\partial b_n} d\Omega + \int_S (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS \tag{2.10}$$

where S is the boundary of the computational domain which is actually $S = S_I \cup S_O \cup S_W \cup S_{W_P}$. The boundaries S_I, S_O, S_W and S_{W_P} refer to the inlet, outlet, fixed and parameterized (varying or controlable) boundaries of the domain, respectively. However, only the parameterized boundaries may change ($\frac{\delta x_k}{\delta b_n} n_k$ corresponds to the deformation velocity of the surface in the normal direction) so we have

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_\Omega u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_\Omega q \frac{\partial R^p}{\partial b_n} d\Omega + \int_{S_{W_P}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS \tag{2.11}$$

As already mentioned, $F = F_{aug}$ and consequently $\frac{\delta F}{\delta b_n} = \frac{\delta F_{aug}}{\delta b_n}$.

At this point, it is necessary to make a clear distinction between the meaning of the partial and total derivative, as used from eq. 2.11 and below. $\frac{\delta \Phi}{\delta b_n}$ is used to indicate the total derivative of an arbitrary quantity Φ, which can be any of the flow variables or even the residual of the state equations and represents the total change in Φ by varying b_n. The partial derivative $\frac{\partial \Phi}{\partial b_n}$ denotes the variation in Φ due to changes in the flow variables (that are caused by geometry deformation) excluding contributions from the space deformation itself. In other words, the partial derivative represents the variation in Φ if the internal nodes of the computational domain remain unchanged. The expression that links the total and partial derivatives is

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k} \frac{\delta x_k}{\delta b_n} \tag{2.12}$$

However, if Φ is a quantity computed on the surface, like pressure, eq 2.12 slighty changes. As any small surface deformation can be seen as a normal displacement, only the normal surface deformation velocity $\frac{\delta x_k}{\delta b_n}$ contributes in the change of Φ so its tangential component can be neglected resulting in

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m \tag{2.13}$$

In order to express $\frac{\delta F}{\delta b_n}$ and to formulate the adjoint equations and their boundary conditions, the objective function is expressed in general form as a sum of both surface and volume integrals, following [6].

$$F = \int_S F_S dS + \int_{\Omega_{tar}} F_{\Omega_{tar}} d\Omega_{tar} \tag{2.14}$$

However, the objective function used 2.7 has the form of a field integral on a pres-

elected part of the flow domain and it does not include any surface integral. So, in our case,

$$F = \int_{\Omega_{tar}} F_{\Omega_{tar}} d\Omega_{tar} \tag{2.15}$$

The differentiation of F w.r.t. the design variables gives

$$\frac{\delta F}{\delta b_n} = \frac{\delta}{\delta b_n} \int_{\Omega_{tar}} F_{\Omega_{tar}} d\Omega_{tar} \tag{2.16}$$

The application of the Leibniz theorem yields

$$\frac{\delta}{\delta b_n} \int_{\Omega_{tar}} F_{\Omega_{tar}} d\Omega_{tar} = \int_{\Omega_{tar}} \frac{\partial F_{\Omega_{tar}}}{\partial b_n} d\Omega_{tar} + \int_S F_{\Omega_{tar}} n_k \frac{\delta x_k}{\delta b_n} dS \tag{2.17}$$

In our case, the velocity of the boundary $\frac{\delta x_k}{\delta b_n}$ is zero as the boundaries of the target volume in this application Ω_{tar} are fixed. Consequently, we have

$$\frac{\delta}{\delta b_n} \int_{\Omega_{tar}} F_{\Omega_{tar}} d\Omega_{tar} = \int_{\Omega_{tar}} \frac{\partial F_{\Omega_{tar}}}{\partial b_n} d\Omega_{tar} \tag{2.18}$$

Since the objective function, equation 2.7 is exclusively expressed in terms of velocity, the use of the chain rule for $\int_{\Omega_{tar}} \frac{\partial F_{\Omega_{tar}}}{\partial b_n} d\Omega_{tar}$ w.r.t. v_i gives

$$\int_{\Omega_{tar}} \frac{\partial F_{\Omega_{tar}}}{\partial b_n} d\Omega_{tar} = \int_{\Omega_{tar}} \acute{F}^v_{\Omega_{tar},i} \frac{\partial v_i}{\partial b_n} d\Omega_{tar} \tag{2.19}$$

where $\acute{F}^v_{\Omega_{tar},i}$ is equal to $\frac{\partial F}{\partial v_i}$ which, for the objective function of equation 2.7 is

$$\acute{F}^v_{\Omega_{tar},i} = \frac{\partial F}{\partial v_i} = 2\big(v_k^2 - v_{tar}^2\big) v_i \tag{2.20}$$

According to 2.19 the derivative of our objective function is

$$\frac{\delta F_{obj}}{\delta b_n} = 2 \int_{\Omega_{tar}} \left(v_k^2 - v_{tar}^2 \right) v_i \frac{\partial v_i}{\partial b_n} d\Omega_{tar} \tag{2.21}$$

and, in general form using, eqs. 2.16, 2.18 and 2.19, we have

$$\frac{\delta F}{\delta b_n} = \int_{\Omega_{tar}} \acute{F}^v_{\Omega_{tar},i} \frac{\partial v_i}{\partial b_n} d\Omega_{tar} \tag{2.22}$$

2.4 Adjoint Equations

The terms of eq. 2.11 that include the partial derivatives of the mean flow equations w.r.t. the design variables can be developed by applying the operator $\frac{\partial()}{\partial b_n}$ to eqs.

2.1 and 2.2. According to the comments made in sec. 2.3, the partial derivative reflects the changes in the flow variables over the initial domain without taking into account changes in its shape. Hence, the following permutation is allowed [6]

$$\frac{\partial}{\partial b_n}\left(\frac{\partial \Phi}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\partial \Phi}{\partial b_n}\right) \tag{2.23}$$

Using eq. 2.23 in the differentation of eqs. 2.1 and 2.2, we get

$$\frac{\partial R^p}{\partial b_n} = -\frac{\partial}{\partial x_j}\left(\frac{\partial v_j}{\partial b_n}\right) \tag{2.24}$$

$$\frac{\partial R_i^v}{\partial b_n} = \frac{\partial v_j}{\partial b_n}\frac{\partial v_i}{\partial x_j} + v_j\frac{\partial}{\partial x_j}\left(\frac{\partial v_i}{\partial b_n}\right) - \frac{\partial}{\partial x_j}\left[\left(\nu+\nu_t\right)\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_i}\frac{\partial p}{\partial b_n} \tag{2.25}$$

Applying the Green–Gauss theorem[1] [6] to eq.2.11 using 2.24, the $\int_\Omega q\frac{\partial R^p}{\partial b_n}d\Omega$ volume integral becomes

$$\int_\Omega -q\frac{\partial}{\partial x_j}\left(\frac{\partial v_j}{\partial b_n}\right)d\Omega = -\int_S q\frac{\partial v_j}{\partial b_n}n_j dS + \int_\Omega \frac{\partial q}{\partial x_j}\frac{\partial v_j}{\partial b_n}d\Omega \tag{2.26}$$

In a similar way, using eq.2.25 the inviscid terms of the volume integral $\int_\Omega u_i\frac{\partial R_i^v}{\partial b_n}d\Omega$ can be written as

$$\begin{aligned}\int_\Omega u_i\frac{\partial v_i}{\partial x_j}\frac{\partial v_j}{\partial b_n}d\Omega + \int_\Omega u_i v_j\frac{\partial}{\partial x_j}\left(\frac{\partial v_i}{\partial b_n}\right)d\Omega = \\ \int_\Omega u_j\frac{\partial v_j}{\partial x_i}\frac{\partial v_i}{\partial b_n}d\Omega + \int_S u_i v_j n_j\frac{\partial v_i}{\partial b_n}dS - \int_\Omega \frac{\partial}{\partial x_j}(u_i v_j)\frac{\partial v_i}{\partial b_n}d\Omega\end{aligned} \tag{2.27}$$

$$\int_\Omega u_i\frac{\partial}{\partial x_i}\frac{\partial p}{\partial b_n}d\Omega = \int_S u_i n_i\frac{\partial p}{\partial b_n}dS - \int_\Omega \frac{\partial u_i}{\partial x_i}\frac{\partial p}{\partial b_n}d\Omega \tag{2.28}$$

The viscous terms of the same volume integral (using also eq. 2.23) can be written as

$$\begin{aligned}-\int_\Omega u_i\frac{\partial}{\partial x_j}\left[\left(\nu+\nu_t\right)\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\right]d\Omega = \\ = -\int_S u_i\left(\nu+\nu_t\right)\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)n_j dS + \int_\Omega\left(\nu+\nu_t\right)\frac{\partial u_i}{\partial x_j}\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)d\Omega \\ = -\int_S u_i\left(\nu+\nu_t\right)\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)n_j dS + \int_\Omega\left(\nu+\nu_t\right)\frac{\partial u_i}{\partial x_j}\frac{\partial}{\partial x_j}\left(\frac{\partial v_i}{\partial b_n}\right)d\Omega \\ + \int_\Omega\left(\nu+\nu_t\right)\frac{\partial u_i}{\partial x_j}\frac{\partial}{\partial x_i}\left(\frac{\partial v_j}{\partial b_n}\right)d\Omega\end{aligned} \tag{2.29}$$

[1] Green–Gauss or Divergence theorem: $\int_V(\nabla\cdot\mathbf{F})dV = \oint_S(\mathbf{F}\cdot n)dS$ where n is the normal vector

The two volume integrals in the last expression of eq. 2.29 become

$$\int_\Omega \big(\nu+\nu_t\big)\frac{\partial u_i}{\partial x_j}\frac{\partial}{\partial x_j}\Big(\frac{\partial v_i}{\partial b_n}\Big)d\Omega = \int_S \big(\nu+\nu_t\big)\frac{\partial u_i}{\partial x_j}n_j\frac{\partial v_i}{\partial b_n}dS - \int_\Omega \frac{\partial}{\partial x_j}\Big(\big(\nu+\nu_t\big)\frac{\partial u_i}{\partial x_j}\Big)\frac{\partial v_i}{\partial b_n}d\Omega \tag{2.30}$$

$$\begin{aligned}\int_\Omega \big(\nu+\nu_t\big)\frac{\partial u_i}{\partial x_j}\frac{\partial}{\partial x_i}\Big(\frac{\partial v_j}{\partial b_n}\Big)d\Omega &= \int_S \big(\nu+\nu_t\big)\frac{\partial u_i}{\partial x_j}n_i\frac{\partial v_j}{\partial b_n}dS - \int_\Omega \frac{\partial}{\partial x_i}\Big(\big(\nu+\nu_t\big)\frac{\partial u_i}{\partial x_j}\Big)\frac{\partial v_j}{\partial b_n}d\Omega \\ &= \int_S \big(\nu+\nu_t\big)\frac{\partial u_j}{\partial x_i}n_j\frac{\partial v_i}{\partial b_n}dS - \int_\Omega \frac{\partial}{\partial x_j}\Big(\big(\nu+\nu_t\big)\frac{\partial u_j}{\partial x_i}\Big)\frac{\partial v_i}{\partial b_n}d\Omega\end{aligned} \tag{2.31}$$

By substituting eqs. 2.27 to 2.4 and also using 2.22 into the expression of the differantiated arbitrary objective function F_{aug} 2.11 we get

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_n} &= \int_S \Big[u_i v_j n_j + \big(\nu+\nu_t\big)\Big(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\Big)n_j - q n_i\Big]\frac{\partial v_i}{\partial b_n}dS \\ &+ \int_S (u_j n_j)\frac{\partial p}{\partial b_n}dS + \int_S (-u_i n_j)\frac{\partial \tau_{ij}}{\partial b_n}dS + \int_{S_{W_P}} (u_i R_i^v + q R^p)\frac{\delta x_k}{\delta b_n}n_k dS \\ &+ \int_\Omega \Big\{u_j\frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big[\big(\nu+\nu_t\big)\Big(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\Big)\Big] + \frac{\partial q}{\partial x_i} + \acute{F}^v_{\Omega,i}\Big\}\frac{\partial v_i}{\partial b_n}d\Omega \\ &+ \int_\Omega \Big(-\frac{\partial u_j}{\partial x_j}\Big)\frac{\partial p}{\partial b_n}d\Omega\end{aligned} \tag{2.32}$$

It is necessary to avoid the computation of the partial derivatives $\frac{\partial v_i}{\partial b_n}$ and $\frac{\partial p}{\partial b_n}$ as they require the solution of N systems of equations. To do so, their multipliers in the volume integrals of eq. 2.32 are set to zero, resulting in the so called adjoint mean flow equations

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \tag{2.33}$$

$$R_i^u = u_j\frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big[\big(\nu+\nu_t\big)\Big(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\Big)\Big] + \frac{\partial q}{\partial x_i} + \acute{F}^v_{\Omega_{tar},i} = 0 \qquad i = 1,2,3 \tag{2.34}$$

where u_i is adjoint velocity and q is adjoint pressure. As a result, the remaining terms of the expression of 2.32 that will give the adjoint boundary conditions and the final expression of the sensitivity derivatives are

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_n} &= \int_S \Big[u_i v_j n_j + \big(\nu+\nu_t\big)\Big(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\Big)n_j - q n_i\Big]\frac{\partial v_i}{\partial b_n}dS \\ &+ \int_S (u_j n_j)\frac{\partial p}{\partial b_n}dS + \int_S (-u_i n_j)\frac{\partial \tau_{ij}}{\partial b_n}dS + \int_{S_{W_P}} (u_i R_i^v + q R^p)\frac{\delta x_k}{\delta b_n}n_k dS\end{aligned} \tag{2.35}$$

2.5 Adjoint Boundary Conditions

The adjoint boundary conditions derive from the expression 2.35 by setting to zero the first three integrals, see [6].

Inlet Boundaries As discussed in sec. 2.2, at the inlet boundaries S_I, the Dirichlet boundary condition on velocity implies that $\frac{\delta v_i}{\delta b_n} = 0$. As this boundary is fixed, we have $\frac{\delta x_k}{\delta b_n} = 0$ so, according to 2.13, also $\frac{\partial v_i}{\partial b_n} = 0$. Eliminating the rest second and third integrals of 2.35 we get

$$u_j n_j = u_{\langle n \rangle} = 0 \tag{2.36}$$

$$u_i n_j = u_i t_i^l = u_{\langle t \rangle}^l = 0 \tag{2.37}$$

but, since no boundary condition for q can be derived from the previous expression, a zero Neumann boundary condition is used.

Outlet Boundaries Having a zero Dirichlet condition for p on the fixed outlet boundaries S_O, it is $\frac{\delta p}{\delta b_n} = \frac{\partial p}{\partial b_n} = 0$. Due to the distance of the outlet boundary from the parameterized area, see fig. 2.1 (where the parameterized area is a part of the defroster nozzle), an almost uniform velocity profile can be assumed along S_O leading to eliminating the third term of 2.35. However, so as to eliminate the first term of the equation, it should

$$u_i v_j n_j + \big(\nu + \nu_t\big)\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) n_j - q n_i = 0 \tag{2.38}$$

By multiplying 2.38 with the normal surface vector n_i, results namely a Dirichlet boundary condition for q

$$q = u_{\langle n \rangle} v_{\langle n \rangle} + 2\big(\nu + \nu_t\big)\frac{\partial u_{\langle n \rangle}}{\partial n} = 0 \tag{2.39}$$

The tangential adjoint velocity components can be obtained by multiplying eq. 2.38 with the tangent to the surface vectors $t_i^l, l = 1, 2$,resulting to a Robin type boundary condition

$$v_{\langle n \rangle} u_{\langle t \rangle}^l + \big(\nu + \nu_t\big)\left(\frac{\partial u_{\langle t \rangle}^l}{\partial n} + \frac{\partial u_{\langle n \rangle}}{\partial t^l}\right) = 0 \tag{2.40}$$

while for the normal adjoint velocity components zero Neumann boundary condition is applied.

Fixed Wall Boundaries Similar to the inlet boundaries S_I, at the fixed walls of the domain S_W, the boundary condition for u_i is Dirichlet resulting to

$$u_j n_j = u_{\langle n \rangle} = 0 \tag{2.41}$$

$$u_i n_j = u_i t_i^l = u_{\langle t \rangle}^l = 0 \tag{2.42}$$

and zero Neumann boundary condition for q is imposed.

Parameterized Wall Boundaries The main difference between fixed and parameterized walls is that the boundaries of the latter may change during optimization, hence $\frac{\delta x_k}{\delta b_n} \neq 0$. Consequently, since $v_i = 0$, its total variation is zero $\frac{\delta v_i}{\delta b_n} = 0$ resulting in

$$\frac{\partial v_i}{\partial b_n} = -\frac{\partial v_i}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m \tag{2.43}$$

which makes the remaining first term of 2.35 along S_{W_P} to turn into

$$\begin{aligned} &\int_{S_{W_P}} \left[u_i v_j n_j + \big(\nu + \nu_t\big)\Big(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\Big) n_j - q n_i \right] \frac{\partial v_i}{\partial b_n} dS = \\ -&\int_{S_{W_P}} \left[u_i v_j n_j + \big(\nu + \nu_t\big)\Big(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\Big) n_j - q n_i \right] \frac{\partial v_i}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m dS \end{aligned} \tag{2.44}$$

Since the integral on the r.h.s. of eq. 2.44 includes only the surface variation and primal and adjoint fields, its value can be computed and added to the final expression of the sensitivity derivatives.

To summarize, according to [6], for the objective function of eq. 2.7 that is only dependent on v_i, the adjoint boundary conditions are the following:

- Zero Dirichlet boundary condition on u_i and zero Neumann boundary condition on q at the inlet.
- Robin type boundary condition on the tangential adjoint velocity components, zero Neumann boundary condition on normal adjoint velocity components and Dirichlet boundary condition on q, at the outlet.
- Zero Dirichlet boundary condition on u_i and zero Neumann boundary condition on q at the fixed and parameterized walls.

2.6 Sensitivity Derivatives

After satisfying the adjoint mean flow equations and their boundary conditions, the adjoint variables are given by the adjoint solver. Then, it is possible to compute the sensitivity derivatives by the expression

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} = -\int_{S_{W_P}} &\left[\big(\nu + \nu_t\big)\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) n_j - q n_i \right] \frac{\partial v_i}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m dS \\ &+ \int_{S_{W_P}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS \end{aligned} \tag{2.45}$$

where, in the first term, inside the brackets there are only primal and adjoint variables while the outside part comes from the differentiation of the geometry, that is produced by the morpher.

2.7 Discretisation and Solution of the Equations

The systems of the Partial Differential Equations (PDEs) of the primal (eq. 2.1, 2.2) and adjoint problem (eq. 2.33, 2.34) should be discretised in order to be transformed into a corresponding system of algebraic equations and, then, numerically solved [6].

The steady state, mean flow equations for incompressible, turbulent flows are given by eqs. 2.1, 2.2. In order to complete the system of primal equations, the turbulence model, used to compute ν_t, should also be taken into account. In the SIMPLE algorithm presented the turbulence model equations are solved in a segregated manner from the mean flow ones.

The primal (already existed in the OpenFOAM CFD toolbox) and adjoint (developed by PCopt/NTUA) solvers are programmed in the OpenFOAM environment; they solve the equations using the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm and a cell-centered finite-volume discretization scheme, on unstructured grids. For the convection terms, second-order upwind (primal) and downwind (adjoint) schemes were used. For the computation of spatial gradients, the Green–Gauss theorem is used and the involved quantities are interpolated using a linear scheme. For more information about the discretization schemes, readers can refer to [25], [28].

2.7.1 The SIMPLE algorithm

In what follows, the variant of the SIMPLE algorithm, already programmed in OpenFOAM, for the solution of the primal equations is presented, [6].

The momentum equations, 2.2, can be written in a semi-discretized form as

$$a_P v_{P,i} = \sum_{N=1}^{NB(P)} a_N v_{N,i} - \frac{\partial p}{\partial x_i} \tag{2.46}$$

where P is used to denote both the cell index and its centroid in which the momentum equations are discretized and $NB(P)$ are its adjacent cells, fig. 2.3. The coefficients a_P and a_N result from the discretization of the convection and diffusion terms in 2.2. It should be noted that the coefficient a_P is the same for all the components of the momentum equations. The iterative solution of 2.46, using the pressure field p^*, obtained through the previous iteration, results in a velocity field, v_i^*, which does not satisfy the continuity equation. Moreover, there is no equation from which the pressure field can be updated (the continuity equation includes only

the velocity components). For that reason, it appears the need to derive an equation for the computation of p and correct the velocity field to also satisfy the continuity equation.

Let the velocity and pressure fields that satisfy the momentum and continuity equations be denoted by v_i and p. As already discussed, v_i^* stand for the velocity components resulting from the iterative solution of the momentum equations (which do not, however, satisfy the continuity equation) based on the pressure field p^*, made available from the previous step of the solution algorithm. Based on the two above-mentioned sets of fields, the semi-discretized momentum equations, 2.46, can be written as

$$a_P v_{P,i} = \sum_{N=1}^{NB(P)} a_N v_{N,i} - \frac{\partial p}{\partial x_i} \tag{2.47}$$

$$a_P v^*_{P,i} = \sum_{N=1}^{NB(P)} a_N v^*_{N,i} - \frac{\partial p^*}{\partial x_i} \tag{2.48}$$

Once 2.48 have been numerically solved in the current iteration of the solution algorithm, a prediction of the velocity components is given by

$$v^*_{P,j} = \hat{v}_{P,j} \frac{1}{a_p} \frac{\partial p^*}{\partial x_j} \tag{2.49}$$

where

$$\hat{v}_{P,j} = \frac{1}{a_P} H_{P,j}(\mathbf{v}^\star) \tag{2.50}$$

$$H_{P,j}(\mathbf{v}^\star) = \sum_{N=1}^{NB(P)} a_N v^*_{N,j} \tag{2.51}$$

Let $v_i^{'}$ and $p^{'}$ correspond to the velocity and pressure corrections that need to be added to v_i^* and p^* in order to also satisfy the continuity equation, i.e.,

$$v_i = v_i^* + v_i^{'} \tag{2.52}$$

$$p = p^* + p^{'} \tag{2.53}$$

Subtracting 2.48 from 2.47 and assuming that the discretization coefficients (a_P, a_N) are (approximately) the same in eq.2.47 and eq.2.48, the following relation between $v_i^{'}$ and $p^{'}$, holds

$$a_P v^{'}_{P,i} = \sum_{N=1}^{NB(P)} a_N v^{'}_{N,i} - \frac{\partial p^{'}}{\partial x_i} \tag{2.54}$$

After additionally assuming that the first term on the r.h.s. of eq. 2.54 is negligible

compared to the gradient of the pressure correction, eq. 2.54 is simplified to

$$v_{P,i}^{'} = -\frac{1}{a_P}\frac{\partial p^{'}}{\partial x_i} \tag{2.55}$$

Then, by taking 2.52 into account, the continuity equation, 2.1, is written as

$$\frac{\partial v_j}{\partial x_j} = 0 \Rightarrow \frac{\partial v_j^{'}}{\partial x_j} = \frac{\partial v_j^{*}}{\partial x_j} \tag{2.56}$$

Substituting 2.55 into 2.56 yields

$$\frac{\partial}{\partial x_j}\left(\frac{1}{a_p}\frac{\partial p^{'}}{\partial x_j}\right) = \frac{\partial v_j^{*}}{\partial x_j} \tag{2.57}$$

Eq.2.57 can be further processed after taking 2.49 into account,

$$\begin{aligned} \frac{\partial}{\partial x_j}\left(\frac{1}{a_p}\frac{\partial p^{'}}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\hat{v}_{P,j} - \frac{1}{a_P}\frac{\partial p^{*}}{\partial x_j}\right) \Rightarrow \\ \frac{\partial}{\partial x_j}\left(\frac{1}{a_p}\frac{\partial p}{\partial x_j}\right) = \frac{\partial \hat{v}_{P,j}}{\partial x_j} \end{aligned} \tag{2.58}$$

giving rise to the pressure equation. It should be noted that the presented algorithm utilized by OpenFOAM to solve the primal equations, formulates eq.2.58, from which the pressure field, rather than the pressure correction one (usually computed through more traditional variants of SIMPLE, [25], [28], [40]), is obtained. The integration of eq.2.58 over a cell P with $nb(P)$ faces yields

$$\sum_{f=1}^{nb(P)} \overline{\hat{v}_{f,j}} S_{f,j} = \sum_{f=1}^{nb(P)} \frac{1}{\overline{a_f}}\left(\frac{\partial p}{\partial x_j} S_j\right)_f \tag{2.59}$$

In eq.2.59, $S_{f,j}$ are the components of the (dimensional) face normal vector, the norm of which is equal to the face area. Overlines indicate a linear interpolation of the variable under consideration to face f, using the values computed at the neighbouring cells P and N. For an arbitrary variable ϕ, the linear interpolation is written as

$$\phi_f = w_P\phi_P + (1 - w_P)\phi_N + \underbrace{\overline{\left.\frac{\partial \phi}{\partial x_j}\right|_{f'} m_j}}_{\text{skew correction}} \tag{2.60}$$

where

$$w_P = \frac{d_2}{d_1 + d_2} \qquad d_1 = \overrightarrow{Pf} \cdot \overrightarrow{S_f} \qquad d_2 = \overrightarrow{fN} \cdot \overrightarrow{S_f} \qquad \overrightarrow{m} = \overrightarrow{f^{'}f} \tag{2.61}$$

Symbols in eqs. 2.60 and 2.61 are explained in fig.2.4. For the sake of simplicity,

it will be assumed that vector **m** has a negligible length compared to the vector connecting points P and N (low skewness values) and the last term on the r.h.s. of eq. 2.60 will be neglected hereafter. After numerically solving eq. 2.59, the new pressure field is acquired. The velocity field, computed at the mesh cell centres, is corrected based on the newly acquired pressure field

$$\begin{aligned} v_{P,i} &= v^*_{P,i} + v^{'}_{P,i} \Rightarrow \\ v_{P,i} = \hat{v}_{P,i} - \frac{1}{a_p}\frac{\partial p^*}{\partial x_j} - \frac{1}{a_P}\frac{\partial p^{'}}{\partial x_j} \\ &= \hat{v}_{P,i} - \frac{1}{a_P}\frac{\partial p}{\partial x_j} \end{aligned} \tag{2.62}$$

Finally, the fluid volume flux at the mesh faces is updated through

$$m_f = v_{f,j}S_{f,j} = \overline{\hat{v}_{f,j}}S_{f,j} - \frac{1}{\overline{a_f}}\left(\frac{\partial p}{\partial x_j}S_j\right)_f \tag{2.63}$$

The normal pressure derivative at each face is evaluated as

$$\left(\frac{\partial p}{\partial x_j}S_j\right)_f = (p_N - p_P)\frac{n_{f,k}S_{f,k}}{n_{f,m}d_{PN,m}} + \overline{\left.\frac{\partial p}{\partial x_j}\right|_f}\left(S_{f,j} - \frac{n_{f,k}S_{f,k}}{n_{f,m}d_{PN,m}}d_{PN,j}\right) \tag{2.64}$$

where **n** is a unit vector, normal to face f and the rest of the symbols in eq. 2.65 are explained in 2.5. The normal pressure gradient at each face is evaluated as

$$\left(\frac{\partial p}{\partial x_j}S_j\right)_f = (p_N - p_P)\frac{n_{f,k}S_{f,k}}{n_{f,m}d_{PN,m}} + \overline{\left.\frac{\partial p}{\partial x_j}\right|_f}\left(S_{f,j} - \frac{n_{f,k}S_{f,k}}{n_{f,m}d_{PN,m}}d_{PN,j}\right) \tag{2.65}$$

where **n** is a unit vector, normal to face f and the rest of the symbols in eq. 2.65 are explained in fig. 2.5. To sum up, the steps of the SIMPLE algorithm used to solve the primal equations are:

1. Solve eq. 2.48 in order to acquire a velocity field, v^*_i. The computation of v^*_i is based on the pressure field p^* and the volume flux m_f (required for the computation of a_P and a_N) of the previous iteration of the algorithm (or the initialization).

2. Compute $\hat{v}_{P,i}$ through eq. 2.50 and interpolate its values to the cell faces using a linear interpolation scheme.

3. Solve eq. 2.59 to compute the pressure field p.

4. Update the volume flux m_f through eq. 2.63. The volume flux will be used in the solution of the momentum equations in the next iteration of the algorithm.

5. Explicitly relax the pressure field, i.e.,

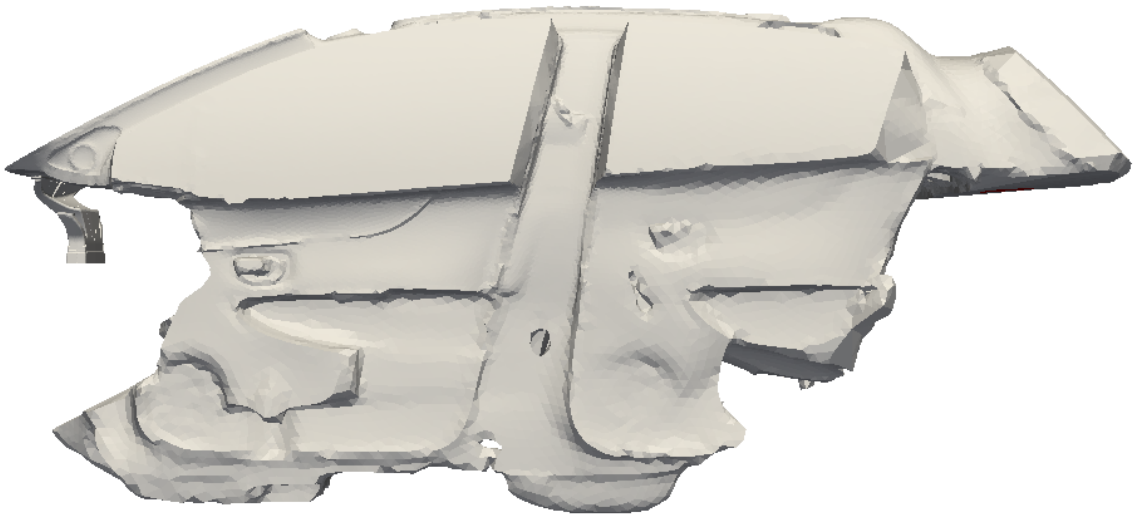
$$p_{new} = \alpha p + (1 - \alpha)p^*$$

where α is a user–defined relaxation factor. The new pressure field p_{new} will be used as p^* in step 1 of the next iteration of the algorithm.

6. Update the velocity fields v_i at the cell centres using the new pressure field p_{new}, eq. 2.62.

7. Solve the turbulence model equations, segregated from the mean flow ones.

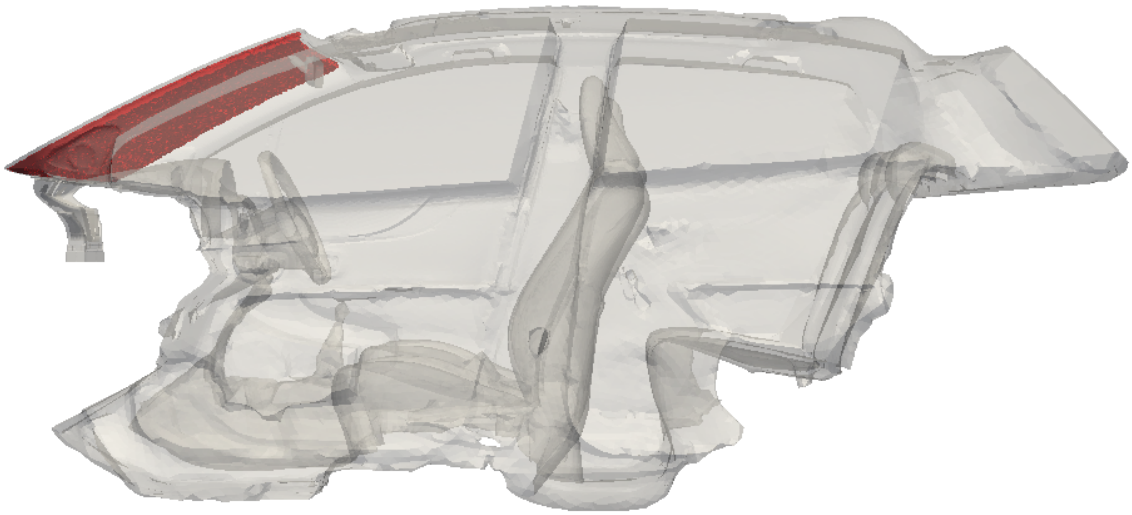
8. If the desired level of convergence is met for all equations, terminate the run. Otherwise, go to step 1.



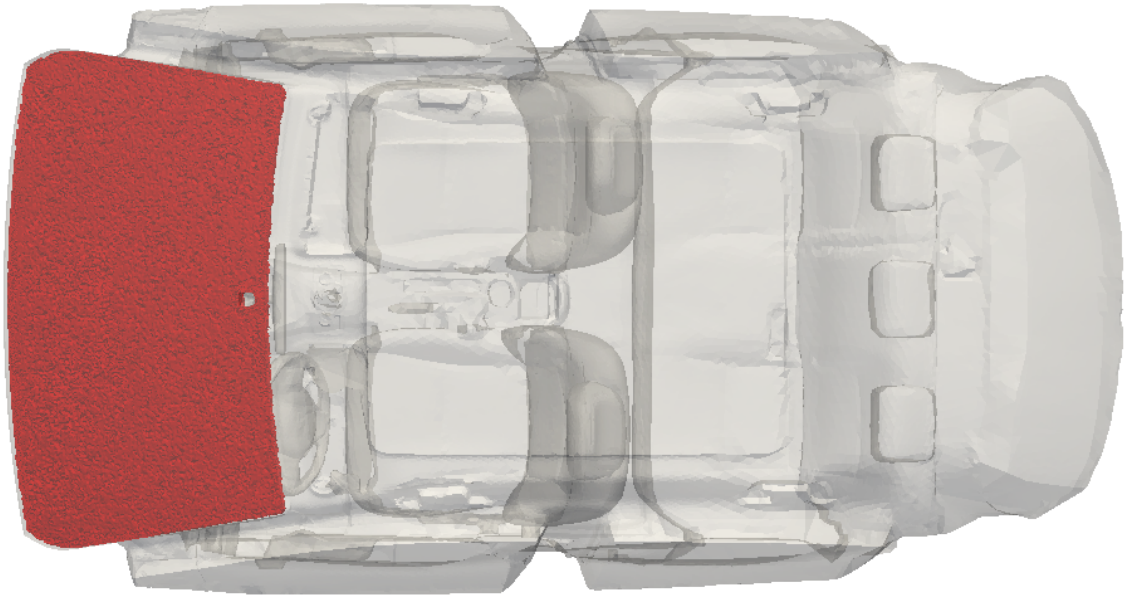
(a) *The computational domain Ω in the main case of interest. Side view.*

(b) *Bottom view of the computational domain. The inlet S_I is marked with a blue color and it is the inlet of the defroster nozzle. The outlet S_O is marked with red and it is an area close to the trunk of the car. The walls S_W are the remaining surfaces of the cabin.*

Figure 2.1: *Computational domain of the main model simulated. Definition of the inlet and outlet areas.*



(a) *Computational domain Ω and target volume Ω_{tar} in side view.*



(b) *Computational domain Ω and target volume Ω_{tar} in top view.*

Figure 2.2: *The thin target volume Ω_{tar} where the objective function is computed is marked with red color.*

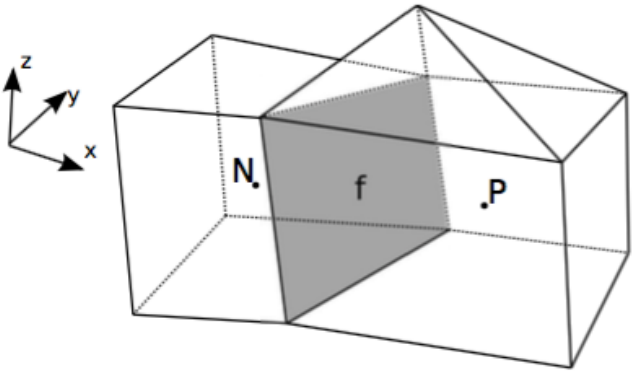


Figure 2.3: *Mesh cell, centered at P and one of its adjacent mesh cells centered at N. The two cells share a single face f [6]*

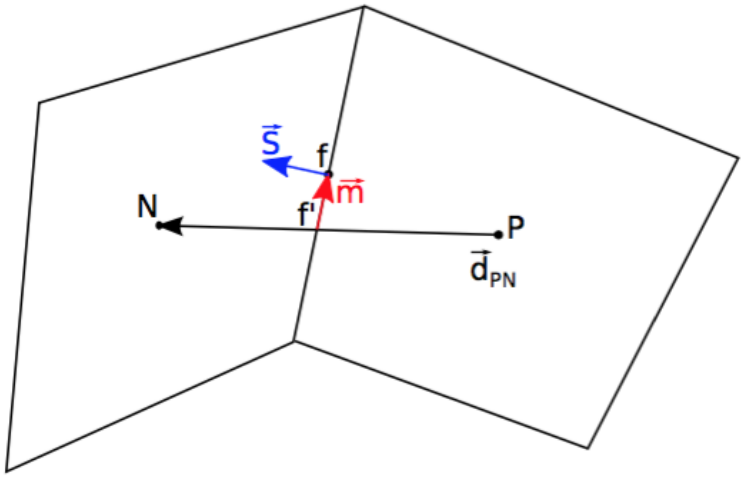


Figure 2.4: *Linear interpolation of the face value based on the adjacent cell values. Point f' is the intersection of the virtual segment $\mathbf{d_{PN}}$ with the face centered at f. Vector $\mathbf{m}$ connects points f' and f. The ratio $|\mathbf{m}|\,/\,|\mathbf{d_{PN}}|$ is defined as the face skewness. The linearly interpolated value of a variable ϕ at f is approximated by computing the value at f' and, then, correcting for face skewness using the last term on the r.h.s. of eq. 2.60. If the face skewness is small, points f' and f practically coincide and the skewness correction can be omitted [6].*

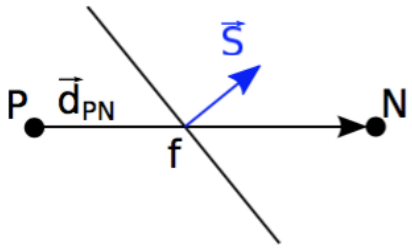


Figure 2.5: *Each internal mesh face belongs to two cells. The cell with the lower cell number is characterized as the "owner" of the face (P) and the other cell (N) becomes its "neighbour". The normal vector $\mathbf{S}$, dimensioned with the face area, points from P to N [6].*

Chapter 3

Shape Morphing

3.1 Introduction to the morpher

As mentioned in chapter 1, a mesh parameterization and movement strategy is needed to perform an automated shape optimization loop. In our case, the shape morpher used is based on volumetric B–splines, which can be seen as a Free Form Deformation (FFD) method. The morpher is coupled with the adjoint solver of PCOpt/NTUA, [20].

Some of its main characteristics are the following:

- A box ($3D$ space) is defined, where a structured grid of control points for volumetric B–splines is generated.
- The CFD mesh points included into the box are parameterized with these control points which are, finally, our design variables.
- The displacement of the control points in each optimization cycle, corresponds to a displacement of the surface and volume nodes of the CFD mesh.

3.2 Mathematical background

The FFD tool as presented in [20] makes use of B–splines defined in $3D$ space, the so–called volumetric B–splines. So as to understand its properties that give so many advantages to the morpher, a few mathematical details are explained, [22], [12], [20].

A B–spline is a generalization of the Bezier curve. It turns out that by adjusting the construction of Bézier curves slightly, we can produce pieces of polynomial curves that automatically tie together smoothly. These piecewise polynomial curves are called spline curves.

In their simplest forms, both methods produce polynomial curves that can be expressed as

$$g(u) = \sum_{i=0}^{d} F_i(u) a_i \tag{3.1}$$

where d is the polynomial degree, a_i are the coefficients and $F_i(t)$ are the basis polynomials.

The difference between the two methods lies in the choice of basis polynomials, or equivalently, how given points relate to the final curve. For Bezier and spline curves, the coefficients are control points with the property that the curve itself lies inside the convex hull of the control points, while the basis polynomials are the Bernstein polynomials and B–splines, respectively. Although both methods are capable of generating any polynomial curve, their differences lead to different polynomial representations.

A Bezier curve or spline curve can conveniently be manipulated interactively by manipulating the curve's control points. It is also quite simple to link several Bezier curves smoothly together. Nevertheless, the disadvantage of Bezier curves is that the smoothness between neighbouring polynomial pieces can only be controlled by choosing the control points appropriately. In other words, the advantage of spline curves over Bezier curves is that smoothness between neighbouring polynomial pieces is built into the basis functions (B–splines) instead of being controlled by constraining control points according to specific rules.

3.2.1 B–spline curves

A B–spline is a parameterized curve $x(u)$ that is defined as a linear combination of $b_i \in [0, n]$ control points and B–spline basis functions $U_{i,p}(u)$ with a degree of p [22]. The curve is described as

$$x(u) = \sum_{i=0}^{n} U_{i,p}(u) b_i \tag{3.2}$$

Equation 3.2 can also be used to obtain the $y(, z)$ coordinates of a monoparametric curve in $2D(, 3D)$. A B–spline is a piecewise polynomial function of degree p. In order to define the basis function $U_{i,p}$, a set of knots which is a non decreasing sequence, known as the knot vector, $\xi_i, i \in [0, m], m = n + p + 1$ must first be defined. The knots $\xi_{p+1}, \ldots, \xi_{m-p-1}$ are called internal knots. A B–spline with no internal knots is a Bezier curve. The uniform knot vector is given as

$$\xi = [\underbrace{0, \ldots, 0}_{p+1}, \frac{1}{N}, \ldots, \frac{N-1}{N}, \underbrace{1, \ldots, 1}_{p+1}] \tag{3.3}$$

where $N = n - p + 1$. This knot vector results to closed curves, this means that we get curves that pass through the first and last control points. The number of control points has to exceed the curve degree by at least one, i.e. $n \geq p$. The basis function is given by

$$U_{i,0}(u) = \begin{cases} 1 & \text{if } \xi_i \leq u < \xi_{i+1} \\ 0 & \text{elsewhere} \end{cases} \tag{3.4}$$

$$U_{i,p}(u) = \frac{u - \xi_i}{\xi_{i+p} - \xi_i} U_{i,p-1}(u) + \frac{\xi_{i+p+1} - u}{\xi_{i+p+1} - \xi_{i+p}} U_{i+1,p-1}(u) \tag{3.5}$$

During the computation of the basis functions values, the quotient $\frac{0}{0}$ may appear and its value is defined to be 0. Two consecutive knots define a knot span.

The degree p determines the extent of the effect of control points. In other words, each basis function (and in consequence, each control point) is affecting only points with a parametric coordinate residing in the $p+1$ knot spans defined by $[\xi, \xi_{i+p+1})$. This gives B–splines curves the desirable property of local support, i.e. a certain part of the curve can be altered by keeping the rest of the curve intact. The range of locality can be controlled by changing the curve degree p, where smaller p values correspond to more localized support. So, control points have a stronger attraction to the curve corresponding to the lower degree basis functions, as it can also be seen in fig. 3.1.

Knots can have a multiplicity greater than one [22], i.e. the representation 3.2 is therefore valid even if some of the knots occur several times. Since B–splines curves are piecewise polynomial functions, they are continuously differentiable in the interior of each knot span. The curve continuity is finite only at the knots. A curve is $p - k$ times differentiable at a point where k duplicate knot values occur. This means that if the knots are all distinct, then a linear spline will be continuous, a quadratic spline will also have a continuous first derivative, while for a cubic spline even the second derivative will be continuous. Predetermining the curve (or surface in $3D$) continuity is a very desirable property for a mesh movement algorithm as well, since the deformed shapes are guaranteed to have a user–defined level of surface continuity, facilitating the manufacturing process.

3.2.2 Volumetric B–splines

The volumes in B–spline form based on B–spline basic functions are now analyzed. It is defined with all attributes as B–spline curve, however, here there are three parameters u, v, w and the definition is similar to the case of curves. The properties of B–spline volumes are similar to the properties of B–spline curves.

The cartesian coordinates $\mathbf{x} = [x_1, x_2, x_3]^T = [x, y, z]^T$ of a CFD mesh point that is chosen to be parameterized, which means that it is residing inside the boundaries

defined by the control grid, are given by

$$x_m(u,v,w) = \sum_{i=0}^{I}\sum_{j=0}^{J}\sum_{k=0}^{K} U_{i,pu}(u)V_{j,pv}(v)W_{k,pw}(w)b_m^{ijk} \tag{3.6}$$

where $\mathbf{u} = [u_1, u_2, u_3]^T = [u, v, w]^T$ are the mesh point parametric coordinates,U, V, W are the B–splines basis functions and pu, pv, pw their respective degrees.$b_m^{ijk}, m \in [1,3], i \in [0,I], j \in [0,J], k \in [0,K]$, signifies the cartesian coordinates of the ijk–th control point of the $3D$ structured control grid, where I, J and K are the number of control points per control grid direction.

As long as the parametric coordinates $\mathbf{u}$ of any parameterized point are known, the computation of its cartesian coordinates is straightforward, at a negligible computational cost. Mesh parametric coordinates can be computed with accuracy, since a mapping from $\Re^3(x,y,z) \rightarrow \Re^3(u,v,w)$ is required. This means that volumetric B–splines can reproduce any geometry to machine accuracy. This is not, for instance, the case when using surface NURBS fitting, where an approximate mapping $\Re^3(x,y,z) \rightarrow \Re^2(u,v)$ is performed.

Given the control points position, the knot vectors and the basis functions degrees, the parametric coordinates (u,v,w) of a point with cartesian coordinates $\mathbf{r} = [x_r, y_r, z_r]^T$ can be computed by solving the system of equations

$$\mathbf{R}(u,v,w) = \begin{bmatrix} x(u,v,w) - x_r = 0 \\ y(u,v,w) - y_r = 0 \\ z(u,v,w) - z_r = 0 \end{bmatrix} \tag{3.7}$$

where $x_m(u,v,w)$ are computed through eq.3.6 based on the given $\mathbf{b}$ values. The 3×3 system of eq.3.7 can be solved independently for each parameterized mesh point numerically, using the Newton–Raphson method, for which is necessary to compute and invert the Jacobian $\frac{\partial x_m}{\partial u_j}, m, j \in [1,3]$. The Jacobian matrix is computed analytically through a closed form expression resulting by differentiating eq. 3.6 with respect to the components of $\mathbf{u}$. Since the evaluation of the parametric coordinates of each point is independent from any other mesh point, this phase may run in parallel.

The aforementioned process has to be done only once and can be seen as the "training phase" of the method. Then, after moving the control points, the cartesian coordinates of each (internal of boundary) mesh point that resides within the control grid can easily be computed through eq. 3.6 at a very low cost. In addition, since x_m depends only on (u,v,w) (which remain unchanged whatever the change in $\mathbf{b}$ might be) and $\mathbf{b}$, the deformed meshes are step–independent. This means that, for a given final control points position, the same mesh quality will be obtained independent of the number of steps taken to reach that position. This is not, for instance, the case

for RBF–based or Laplacian–based mesh movement algorithms.

3.3 Optimization Algorithm

To perform an automated CFD shape optimization loop for the defroster nozzle, as it is mentioned before, the in–house adjoint solver coupled with the in–house morpher was used,[20]. The steps of the shape optimization algorithm are listed below:

1. Define the $3D$ space (box) to enclose the part of the geometry to be optimized. Moreover, define the control points number and the basis functions degree according the logic explained above. A structured control grid is created.

2. Find the CFD mesh points residing within the control grid. These points are to be parameterized and then displaced, according to the control points displacement.

3. Compute the parametric coordinates for each of the points found in step 2. The computational cost of this step increases with the number of control points and the number of the mesh points to be parameterized.

4. Solve the flow equations.

5. Compute the objective function value and apply the termination criterion.

6. Solve the adjoint equations.

7. Compute the objective function gradient w.r.t. the boundary CFD mesh nodes to be displaced, i.e. $\frac{\delta F}{\delta x_m}$ (surface sensitivities).

8. Project the surface sensitivities to control points in order to compute the control points sensitivities,

$$\frac{\delta F}{\delta b_i} = \sum_{j=1}^{n_p} \sum_{m=1}^{3} \frac{\delta F}{\delta x_m^j} \frac{\delta x_m^j}{\delta b_i} \tag{3.8}$$

   where n_p is the number of boundary mesh points to be displaced. In the general case, the control points are allowed to move on all the three directions, however it is possible to confine the displacement in one, or even two directions. As it was previously mentioned, one of the beneficial properties of B–splines is that smoothing is included in the nature of the basis functions. Consequently, no smoothing of the computed sensitivities is required. The quantity $\frac{\delta x_m^j}{\delta b_i}$ is computed analytically by differentiating the linear eq. 3.6 w.r.t. b_i.

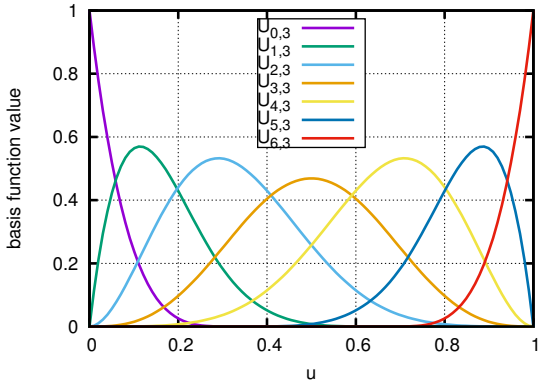
9. Update the control point coordinates. In this thesis, the steepest descent formula is used,

$$b_i^l = b_i^{l-1} - \eta \frac{\delta F}{\delta b_i}\Big|_{l-1} \tag{3.9}$$

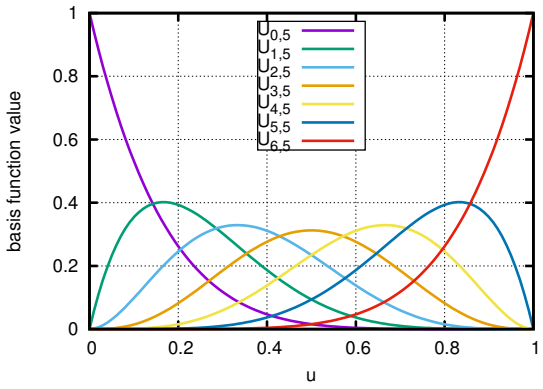
where η is a positive number that defines the step of the descent and l is the current iteration. Apart from the steepest descent method, there are also other algorithms based on the gradient of the objective function or even on the Hessian matrix, such as quasi–Newton methods like BFGS, Newton method, conjugate gradient, etc [24], [29], [13]. The boundary points of the control grid are kept fixed in order to prevent an overlapping between the parameterized and non–parameterized areas of the CFD mesh.

10. Compute the new surface and volume mesh points positions, using the already computed parametric coordinates associated with each one of them.

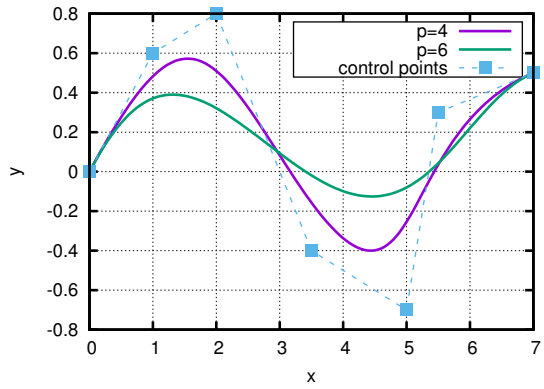
11. Move to step 4.



(a) *Basis functions values of degree* $p = 4$ *for the uniform knot vector* $\xi = \left[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \frac{1}{3} \;\; \frac{2}{3} \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1\right]$.



(b) *Basis functions values of degree* $p = 6$ *for the uniform knot vector* $\xi = \left[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1\right]$.



(c) *B–splines curves generated by the top row basis functions for the given set of control points*

Figure 3.1: *On the top row graphs, for each u, the sum of the basis functions values equals unity. On the third graph, two B–splines curves generated by multiplying the basis functions in the top graphs with the control points depicted by the control polyline. The cartesian coordinates of the curve are given by eq. 3.2, for* $b^i = [b^i_x, b^i_y]^T, 2D$ *vector of control points. The curve corresponding to lower degree basis functions is more strongly attracted by the control points.*

Chapter 4

Simplified Study with Validation: Isolated Defroster Nozzle

Simulation methods have become more prominent at different stages of vehicle development programs. One of the major concerns with simulations is the accuracy of the results. Experimental validation data is required to build confidence in the simulation results, [9].

The main challenges one has to tackle in order to achieve reliable CFD results is to have a good quality of mesh that suits the physics of the fluid problem and an appropriate flow solver. In order to get validate CFD on a simple case with low computational cost before heading to the main target of the adjoint based optimization of the defroster nozzle, an intermediate step was necessary. For that reason, a pressure measurement of the duct was performed and the equivalent CFD model and simulation were realized. The results were compared and the CFD process was successfully validated.

At this point, it is essential to keep in mind that a steady flow solver is used though, in some parts of the domain, unsteadiness may appear. This can be observed looking at the fluctuations of the residuals of equations. However, the study of an unsteady flow has high computational cost and it is decided herein to stick only with steady flow simulations. The convergence using the steady–state solvers for both the primal and the adjoint problem is more than satisfactory, as shown in the following sections.

4.1 Measurement Setup and Results

The pressure measurement of the duct was performed according to the method described below. The measurement setup is based on the use of a pressurized mini chamber. A mini chamber with dimensions of $1 \times 1 \times 1$ m is made. A smaller box $(0.15 \times 0.2 \times 0.4)$ m is attached to it so as to help the positioning (connection) of

the defroster nozzle, and also facilitate the transition of the flow to the nozzle. The static pressure measurement location is the center point of the upper face of the chamber. This point was chosen as no intense flow phenomena happen there. The static pressure is almost equal to the total pressure at this point of measurement, since very little low velocity air is expected to reach the upper and lower sides of the pressurized chamber[1]. This hypothesis will later be confirmed by CFD. During the measurement, four different airflow volumes have been applied (using a throttle valve to adjust the airflow), and the resulting pressure was measured. The airflow was provided by a fan, connected to a callibrated bellmouth (conical inlet nozzle) so as to have an accurate control of the airflow. The air is trasferred from the fan outlet to the chamber by means of a flexible hose. The airflow is computed using the pressure drop across the setup, from the inlet (atmospheric pressure) to the point of measurement, measured with a manometer. This pressure indicated by the manometer is the pressure used for the comparison of results. The setup can be seen in fig 4.1.

In order to validate the pressure for different conditions, two cases were measured, see fig.4.1. In the first case, the outlets of the duct where all open. In the second one, the side outlets were taped.

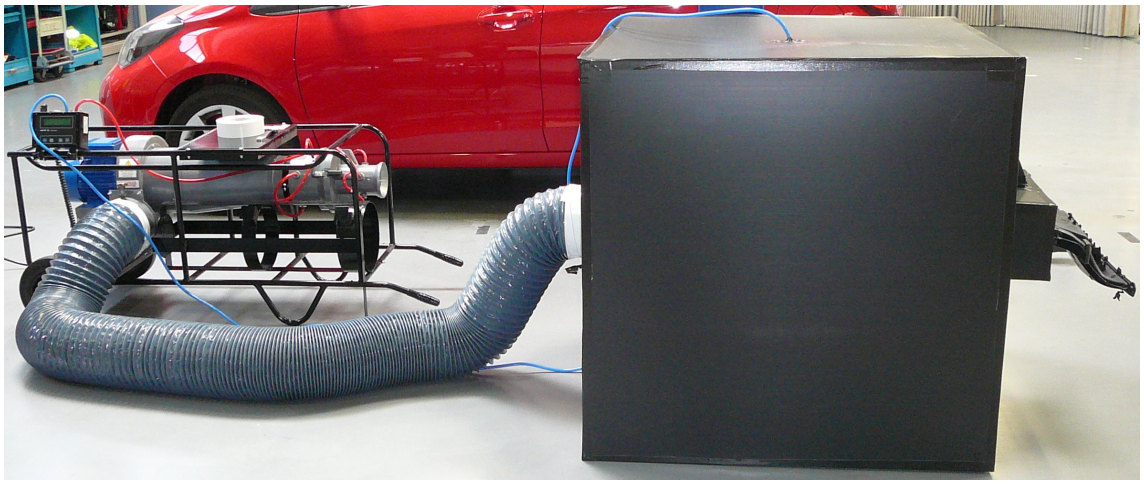


Figure 4.1: *Setup of the pressure measurement. It includes a mini chamber with dimensions of* $1 \times 1 \times 1$ *m. A smaller box* $(0.15 \times 0.2 \times 0.4)$ *m is attached to it, on its right side, in order to facilitate the transition of the flow towards the nozzle itself. The static pressure measurement location is the center point of the upper face of the chamber.*

[1]Bernoulli's principle: $p_t = p_{st} + \frac{1}{2}\rho u^2 + \rho g z \approx p_{st}$ for this case.

	Pressure Measurement Cases	
	1^{st} case: all outlets open	2^{nd} case: side outlets closed
Airflow volume (m^3/h)	100	100
	150	150
	200	200
	250	250

Table 4.1: *Cases for which static pressure was measured.*

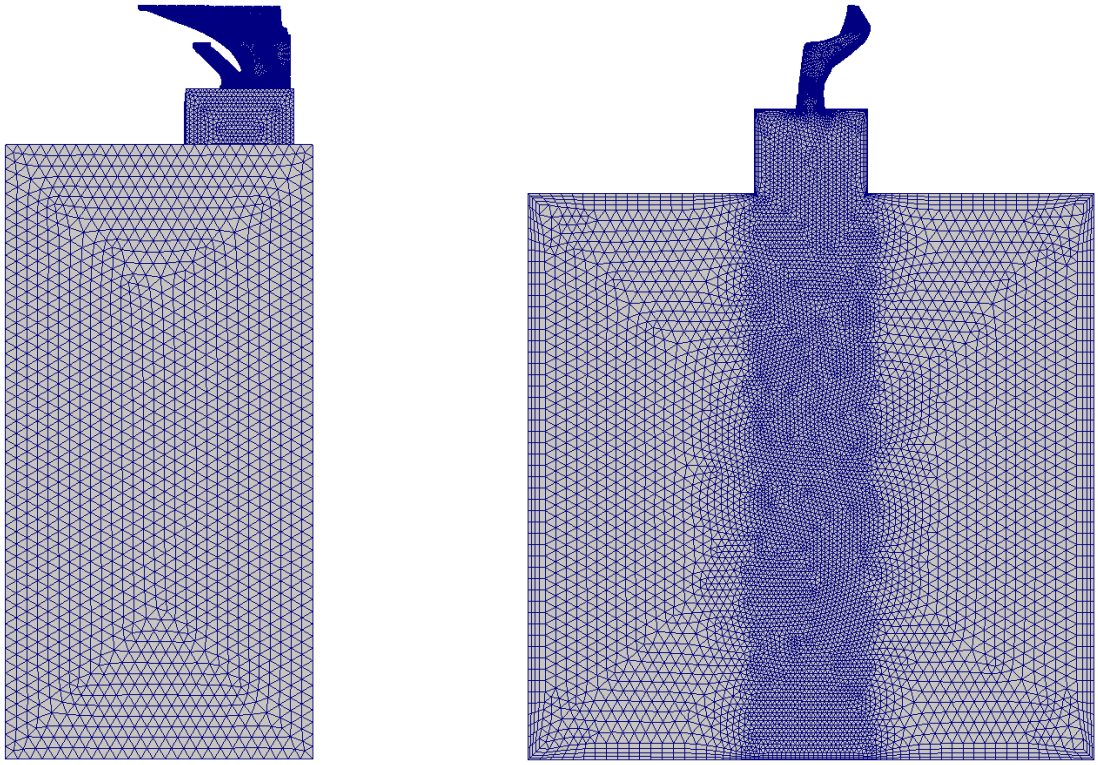
4.2 Meshing and Simulation

So as to validate the CFD method an accurate mesh model of the measurement process had to be produced. For that reason, a mesh that included the chamber, the transition mini box and the inner skin (inside walls of the CAD) of the defroster nozzle was developed. The inlet patch was determined as the circular disk that corresponded to the area of connection between the tube of the fan and the mini chamber. The outlet patches for the first case were the main outlets of defroster nozzle that guide the air to the windshield and its side outlets that guide the air to the side defrosters. The outlet patches for the second case were only the main outlets of the defroster.

In each simulation, the inlet Dirichlet boundary condition for velocity was calculated according to the corresponding value of the airflow volume. Wall boundary conditions were provided by the standard high–Re wall functions of the OpenFOAM CFD toolbox, for k and ϵ.

As the aforementioned geometry of the model is symmetrical, to reduce the computational cost only half of the flow system is modeled, by applying symmetry conditions.. The resulting mesh which is stretched in the areas of the most intense flow phenomena, for the first case shown in fig.4.2 has 2539325 cells, of which 552602 are prisms, 1574 are pyramids and the rest 1985149 are tetrahedra. For the second case, that is slightly different due to the layers along the former side outlets that are now considered to be wall patches, the mesh has 2363873 cells of which 554092 are prisms, 1598 are pyramids and 1808183 are tetrahedra. The mesh was provided by BETA CAE Systems to TME.

The simulation was run until the convergence criteria were satisfied, namely for about 20000 iterations. A solution is considered to be converged when the flow vector and scalar fields are no longer changing, but usually this is not the case for unsteady flows. Most flows in nature and industrial applications are highly unsteady. A good practice to check the convergence is to monitor the residuals. The residual measures how much the approximate solution fails to satisfy the governing differential equation and boundary conditions. However, as in some cases it is possible that the residuals cannot reach full convergence, it is advisable to also monitor a physical quantity



(a) *Half model side view.* **(b)** *Half model volume mesh view.*

Figure 4.2: *Views of the mesh model used to simulate the pressure measurement. The inlet is at the bottom, a circular half disk (with the same diameter as the flexible hose of the test) which is providing the airflow corresponding to each measurement, that flows through the chamber and exits through the outlets of the defroster nozzle. The outlets for the* 1^{st} *case are all the outlets of the duct, while for the* 2^{nd} *the side outlets are sealed.*

representative of the case. If this physical quantity does not significantly change in time (in the case of a steady–state problem) we may say that the solution is converged. In these CFD runs the pressure at the point of the measurements during the tests seems the most appropriate quantity to monitor. The convergence graphs for one of the above mentioned simulations that were run (two cases of four airflows each) are shown in fig. 4.3. The convergence of the physical quantities in this thesis are focused on the last thousands of iterations only, in order to show the small changes that happen as steady–state is approached. Also, the scale is normalized. The physical quantity value is divided by the maximum value monitored along the run and the percentage is presented.

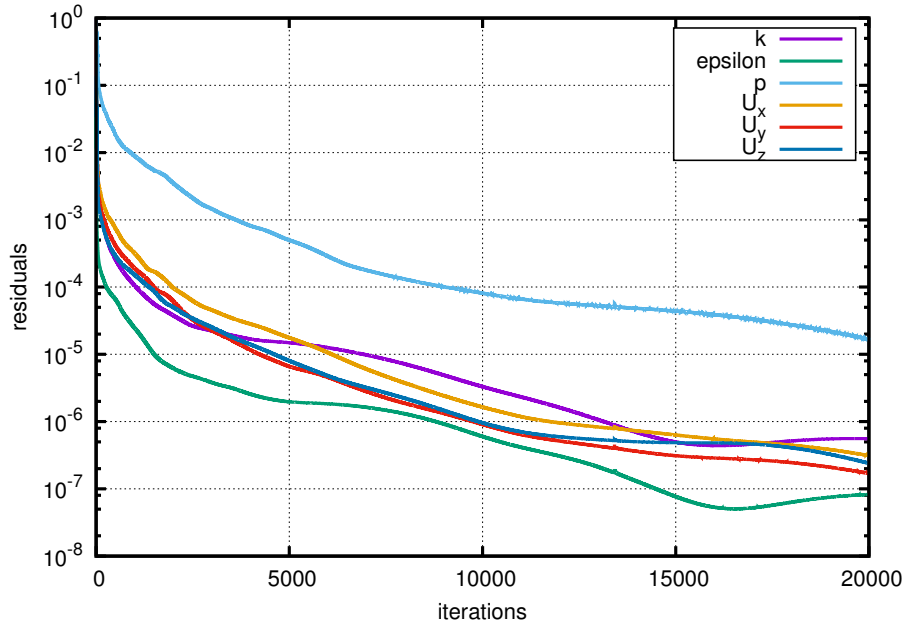
For visualization of the simulations run in OpenFOAM, the ParaView software was used. The post–processing figures in fig. 4.4, show the velocity and pressure distributions for one of the above cases.

4.3 Comparison between CFD and measurement

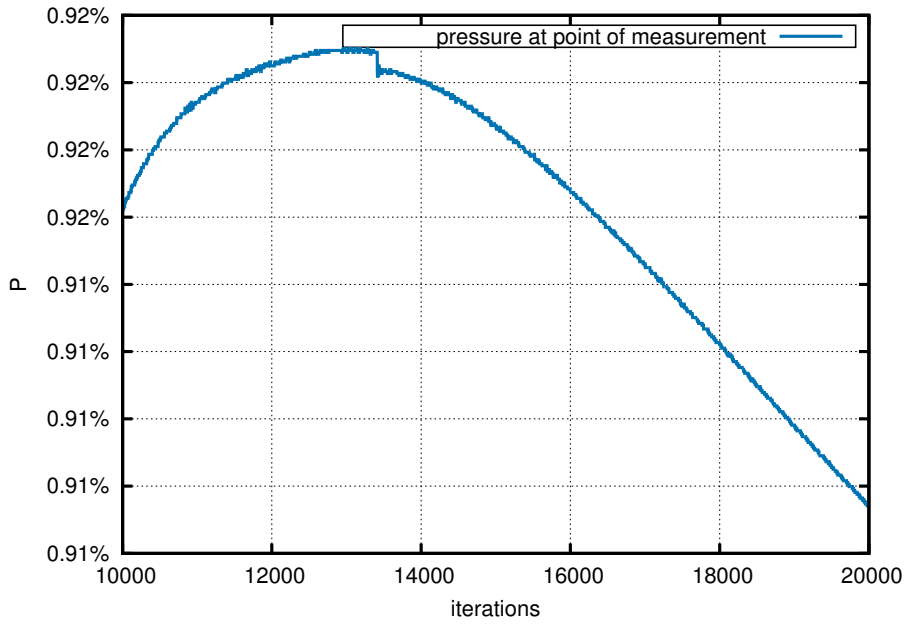
Comparing the CFD results with the measurement the result is a very good agreement as can be seen in tab. 4.2. In the first case, the comparison shows almost identical results, however, in the second case where the side outlets of the nozzle are closed, the pressure measured is constantly smaller than the CFD pressure at the same point. This can easily be justified because the use of tape to seal the outlets does not assure the closure to be watertight, so leakage is highly possible. The higher the airflow, the more the tape is vulnerable resulting to bigger leakage. That is the reason why the deviation percentage is continuously increasing with airflow. As a consequence, the correlation results are considered as very accurate and justifiable. Another way to visualize the good correspondance between CFD and measurement can be seen in fig. 4.5 where the errors of the performed test due to the measuring tools accuracy, are also taken into account.

Airflow volume (m^3/h)	Deviation (%) for 1^{st} case	Deviation (%) for 2^{nd} case
100	+1	+5
150	+0	+7
200	+0	+8
250	+1	+9

Table 4.2: *Deviation of CFD and measurement for static pressure. The point of measurement is the center of the upper side of the chamber. In all the cases the pressure from the CFD simulation was slighly higher than the corresponding pressure measured. The 1^{st} case refers to the measurements taken while having all the outlets of the nozzle open, while the 2^{nd} case refers to the measurements taken while having only the main outlets open and the side outlets closed with tape.*

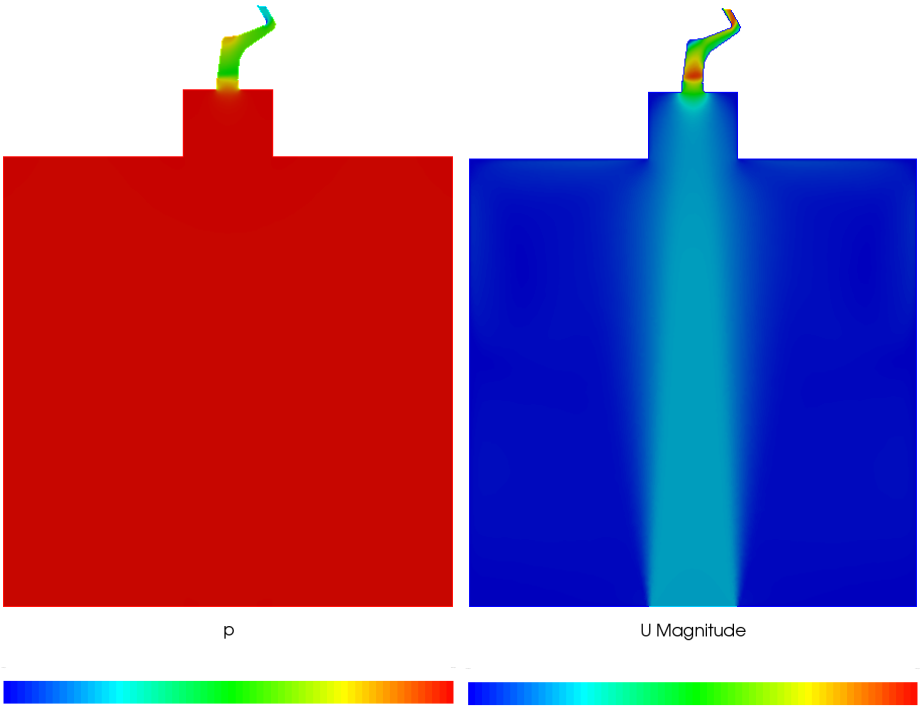


(a) *Non–dimensional residuals are reduced by more than 4 orders of magnitude*

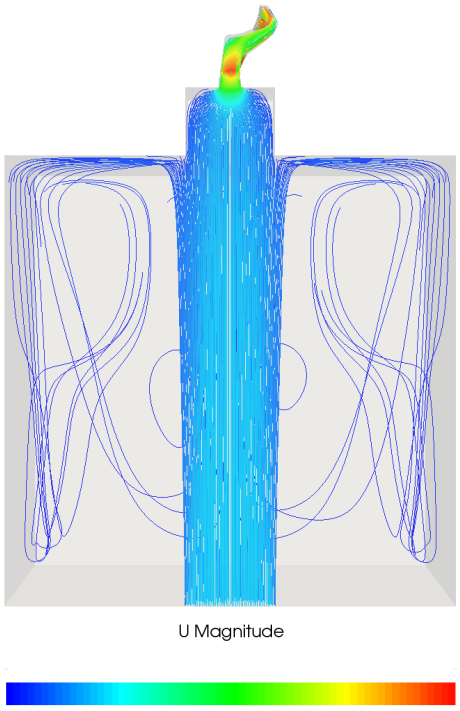


(b) *Convergence of pressure at the point of measurement. The graph is focused on the last thousands of iterations where, the point of interest has reached the steady–state solution. The pressure in the $y-axis$ is normalised, by being divided by its maximum value that appeared during the run.*

Figure 4.3: *The convergence graphs shown are specifically for the first case where all outlets were open and for the first airflow volume of $100m^3/h$. The graphs concerning the other cases were similar to these.*



(a) *Pressure distribution* **(b)** *Velocity distribution*



(c) *Streamlines*

Figure 4.4: *The above figures refer to the first simulation run, that of the* 1^{st} *case, for airflow volume of* $100m^3/h$*. Blue corresponds to lower values while red corresponds to higher. In the first two figures, pressure and velocity distributions accross a plane of the CFD model are shown. The third figure shows the streamlines emitted from seed points at the inlet of the domain to the outlet. There is a recirculation of low velocity air close to the side–walls the pressurized box, though the main airflow is directed towards the nozzle.*

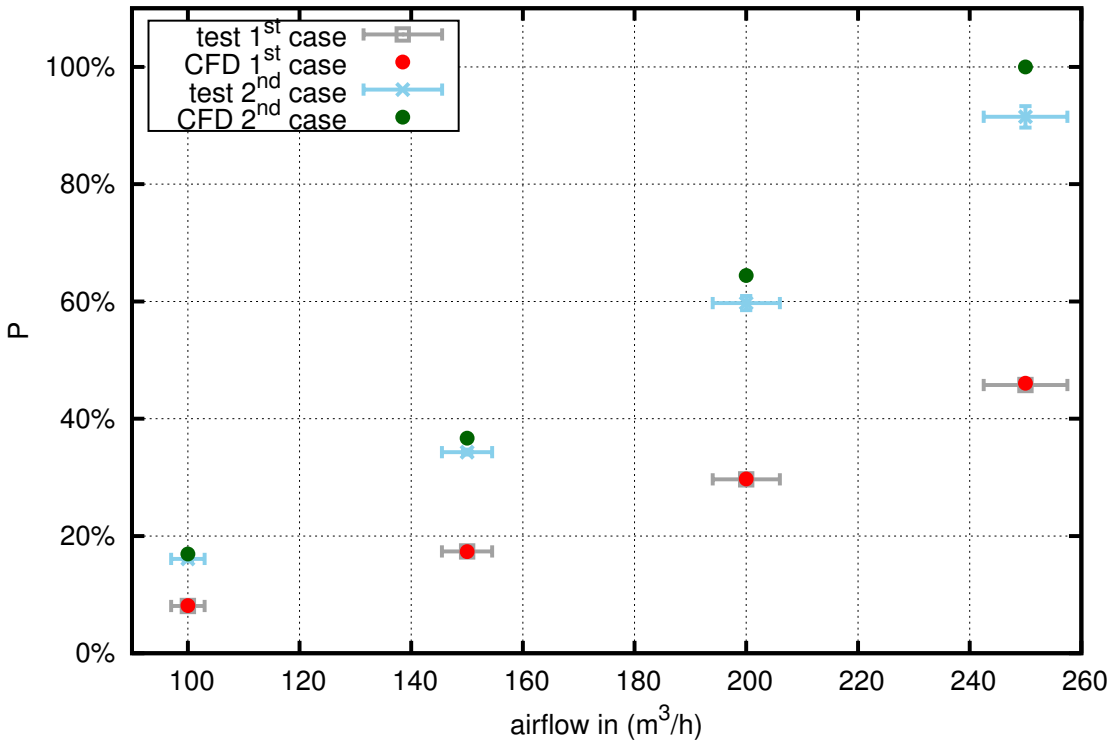


Figure 4.5: *Correlation of CFD and measurement results. Pressure is normalized to the highest pressure measured by CFD. The CFD results on the* 1^{st} *case (all outlets open) are not only within the range of the errors of the test but they almost coincide with the measured static pressure. On the* 2^{nd} *case (side outlets closed), the CFD results are not within the accepted range. The CFD pressure was found to be constantly a bit higher than the measured one. This can be justified due to small leakage from the taped outlets. However, even in this case, the correlation is considered as quite good.*

Chapter 5

Full Model Study with Validation: Defroster Nozzle & Cabin

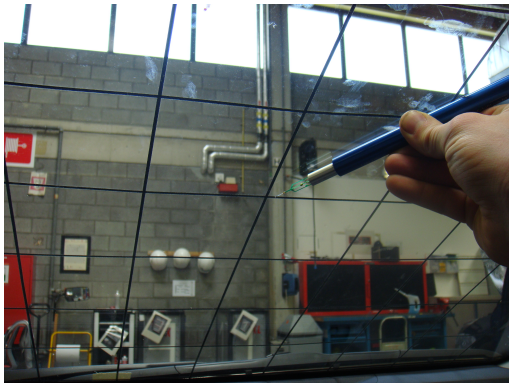
The main target of this diploma thesis is to improve the demisting and defogging performance of a car. To be able to proceed to the optimization process, it is firstly necessary to solve the primal problem so to simulate the flow under consideration. The blower of the HVAC unit provides the defroster nozzle with an airflow which directs hot and high velocity air jets towards the windshield. Afterwards, the air flows to the whole cabin of the car and exits through a flap, located in the rear of the passenger compartment, to the outside. This case will undergo CFD–based simulation.

5.1 Measurement Setup and Results

To begin with, it is important to somehow verify the validity of the CFD results also in the main case of interest. The target is to achieve good accuracy of the computation of the velocity pattern on the windshield. For that reason, a measurement using hot–wire anemometer has been performed, so as to experimentally measure the velocity field in the vicinity of the defroster nozzle jet flow and windshield interior surface [8], [9]. Hot–wire anemometry provides quantitative velocity measurements that are useful for determining defroster and windshield air flow and validating the numerical simulations, according to [9],[5]. To setup the test, a grid with $100mm$ spacing is drawn on the windshield,[8], fig. 5.1a. The HVAC blower is controlled by constant voltage with external power supply so that the defroster nozzle is provided with constant airflow during the experiment. Velocity is measured at each point of the grid using the hot wire anemometer, fig. 5.1b. The velocity cannot be measured on the exact surface of the windshield, since this is zero due to the no–slip condition It will be measured at a number of points located on a surface $7mm$ away. This distance comes from the fact that the hot–wire anemometer is laid on the inner

surface of the windshield and this results having the tip of the measuring tool $7mm$ away from the windshield. It happens, also, that this is the standard depth where the hot–wire measurements are carried out according to [8].

(a) *Measurement grid from the outside.*



(b) *Measurement with hot–wire anemometer.*

(c) *Measurement grid from the inside.*

Figure 5.1: *Setup of the velocity pattern measurement.*

The measured velocity pattern shown in fig 5.8b is not symmetrical because the instrument panel is asymmetrical. On the driver's side, the meter close to the steering wheel is creating this asymmetry that is affecting the flow exiting from the defroster nozzle, towards the windshield. Moreover, on our simulation we consider the HVAC blower to provide uniform flow, which is the ideal case, but this does

not happen in reality. To be more accurate, the whole HVAC unit should have been included in the simulation [27] however this would have big computational and timing cost. Futhermore, another reason why the two compared patterns are not identical is that the possible error of the hot–wire anemometer and the flow disturbance effects caused by it, see fig. 5.1b, also affect the flow. Also, during the measurement, the side outlets of the defroster that provide with hot air the passengers (face outlets) through the side ducts, were taped so leakage is highly possible. Last but not least, in a real car, there is air leakage through the doors and other parts, so the mass flow finally available to the windshield through the defroster is significantly lower than the one provided from the blower. For all these reasons, we expect the velocity pattern provided by the CFD run to be different with the measured one, to have bigger range of velocities and be more symmetrical. However, in general it should give a similar pattern qualitatively.

5.2 Meshing and Simulation

To obtain sufficiently accurate CFD results, it is vital to create a good quality mesh. A good mesh means that it has a good description of the surface of the important, at least, geometries included. Furthermore, it means that it should comply with some quality criteria that vary according to the CFD solver used.

In the current case, to speed up building the model, CAD data was only used for parts where high accuracy was needed (defroster nozzle, windshield, instrument pannel on the outlet of the nozzle, mirror) fig. 5.2. Already available laser scanned surface data is used for the rest of interior. This procedure was followed, because the process of gathering and handling the dozens of the CAD parts included in the cabin of the car can be very time consuming. Furthermore, dealing with the CAD data demands to isolate the inner skin of the parts, before meshing, which is not a trivial task. Last but not least, since according to [4] interior vehicle surface geometry can be simplified to reduce simulation time while keeping acceptable accuracy, the aforementioned procedure is justified.

After geometry clean–up, CAD data was precisely stitched to STL data. Mesh refinement boxes are defined to achieve high accuracy where needed while balancing overall computational cost, [27]. In that way, multiple mesh regions are created in order to accomodate different mesh size in the front cabin, where finer mesh was applied for better accuracy, while in the rear part coarser mesh was chosen. In this model, the inlet patch of the defroster nozzle was set as the inlet patch of the whole cabin flow and as outlet, a rectangular patch at the trunk of the car was created. The resulting mesh has 8399341 cells of which 2435602 are prisms, 7958 are pyramids and the rest 5955781 are tetrahedra fig. 5.3b. The surface mesh consists of 624470 cells. The mesh was provided by BETA CAE Systems to TME.

Figure 5.2: *CAD data of the important parts of the model: defroster nozzle, instrument panel outlet, windshield, mirror.*

The boundary condition for velocity imposed at the inlet is dictated by the corresponding airflow applied during the measurement. The Reynolds number of the flow is approximately 20000, based on inlet hydraulic diameter. Wall boundary conditions were provided by the standard high–Re wall functions of the OpenFOAM CFD toolbox for k and ϵ.

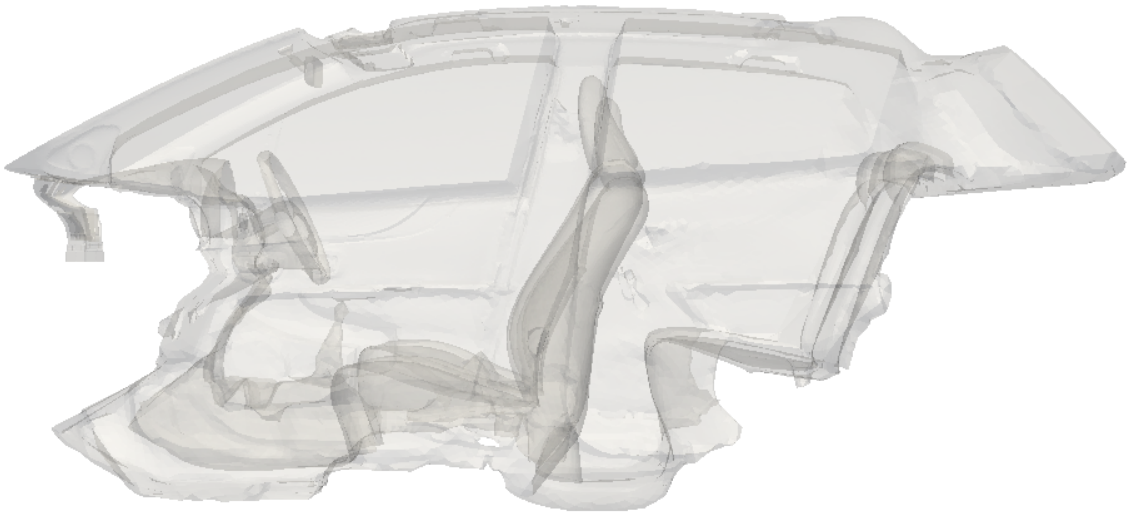
The solution was converged after 40000 iterations when the convergence criteria were satisfied. Apart from the convergence of the residuals, the corresponding mesh node of one point of the grid that was created at the test (shown in fig. 5.1c) was tracked and its velocity magnitude was monitored during the run. The corresponding graphs are presented in fig. 5.4.

The results where post–processed in order to visualize the flow in the cabin and especially close to the area of interest,i.e. the windshield. The streamlines near the windshield can be seen in fig. 5.5 The flow fieds on a cross section can be seen in fig. 5.6. The highest magnitude of velocity and the highest absolute pressure are inside the defroster duct and, then, close to the windshield. In the rest of the cabin there are no intense flow phenomena, as expected.

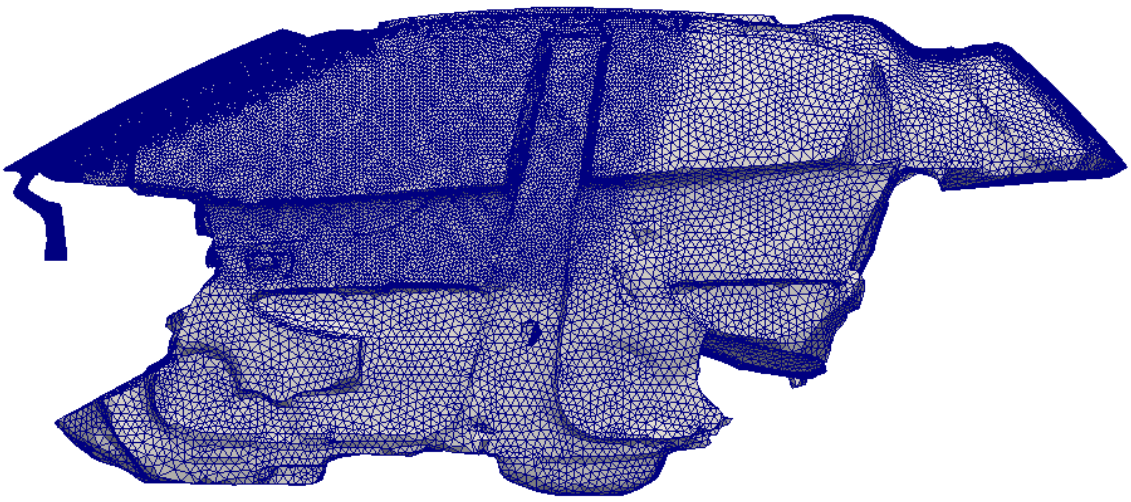
The velocity pattern close to the windshield was also visualised in order to be able to evaluate it and decide the optimization target (objective function) in a way to improve it. As it can be seen in fig. 5.7, the current pattern has low velocity air reaching its upper part, delaying the defrosting process.

5.3 Validation of the Velocity Pattern

The results of the CFD simulation are then post–processed to extract the value of the magnitude of velocity at the grid points that we created during the test, [27]. The two resulting $2D$ velocity distributions shown in fig. 5.8 are then, compared



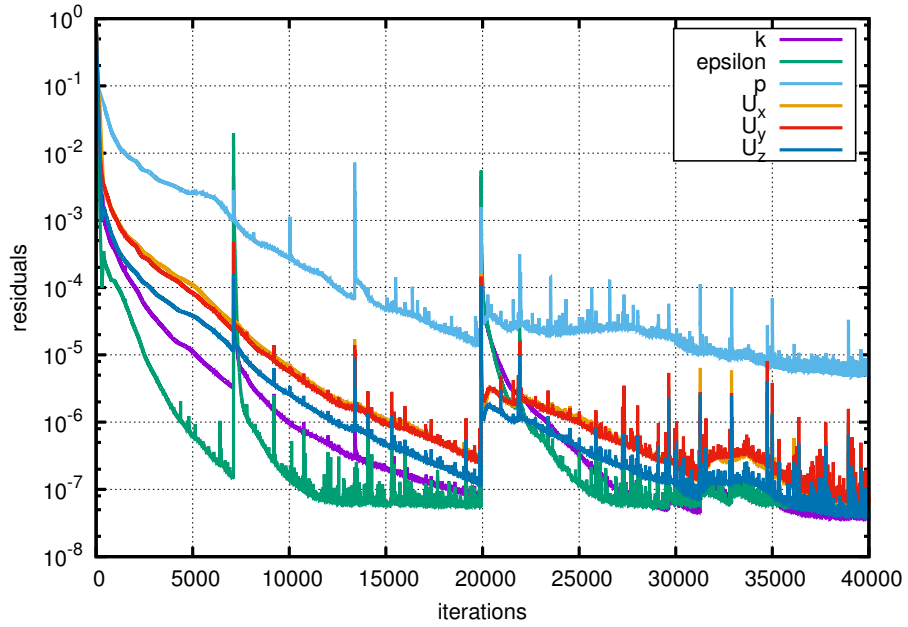
(a) *Watertight model.*



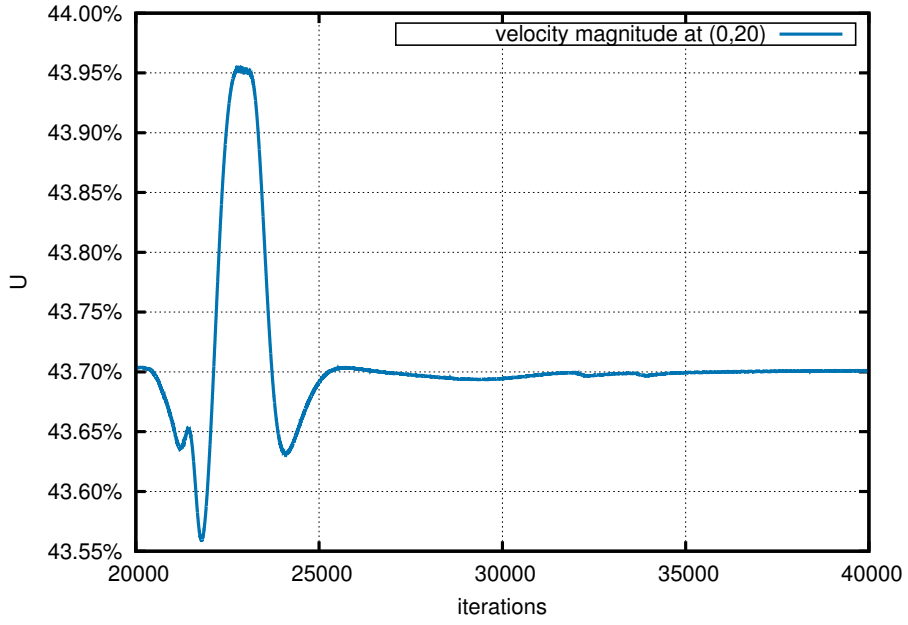
(b) *Computational mesh of the model with almost 9 million cells. The darkest areas correspond to more dense surface and volume mesh, where high accuracy is needed. This was achieved using refinement boxes.*

Figure 5.3: *CFD model and mesh. The inlet to the computational domain is the inlet to the defroster nozzle, while the outlet is a rectangular patch at the rear of the cabin.*

and the comparison between the CFD and test results is judged as good enough, given the above mentioned problems of non–symmetry, the leakage, the errors of the measurement and the flow disturbance effects that come from the holding of the measuring tool by hand. CFD analysis results and measured results are in good agreement and similar pattern can be observed visually. Consequently, the CFD simulation provided a quite accurate solution of the flow and that allows to proceed to the next step, the solution of the adjoint equations.

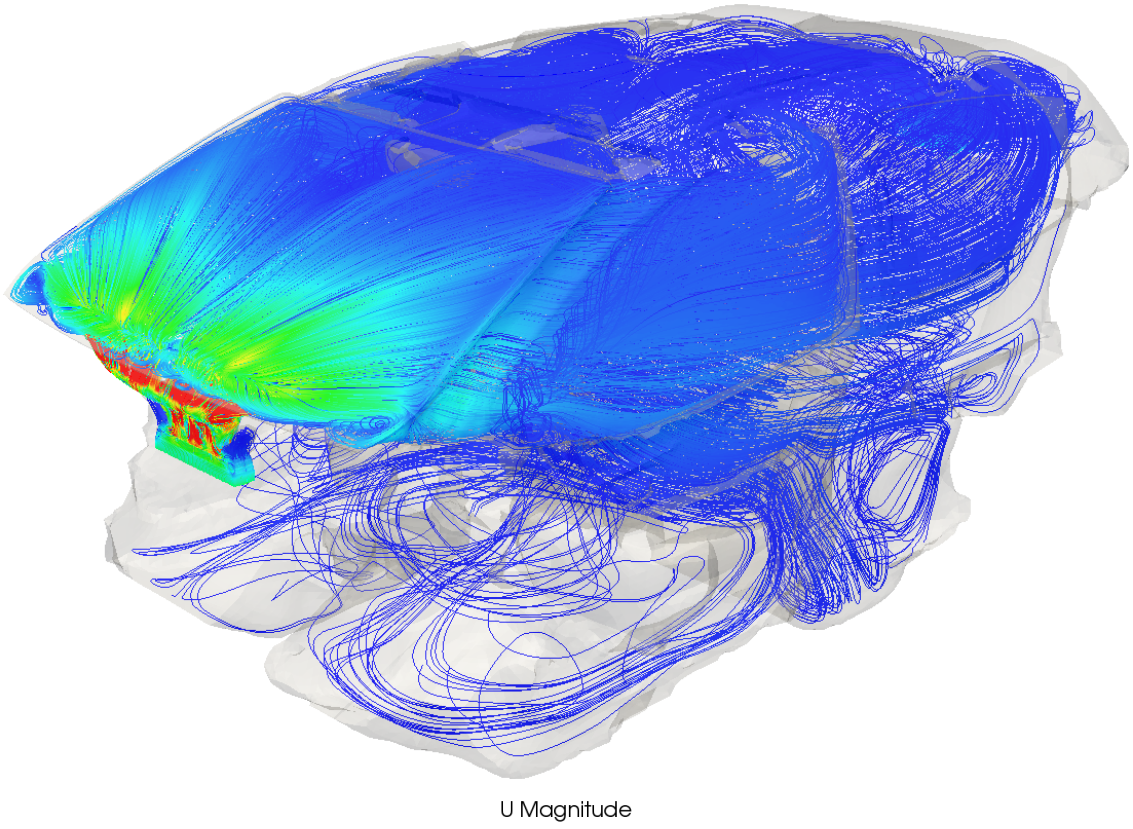


(a) *Non–dimensional residuals are reduced by more than 5 orders of magnitude. The fluctuations of the residuals are due to the use of a steady state solver (for reasons of reducing the computational cost), even though slight unsteadiness appears in the flow.*

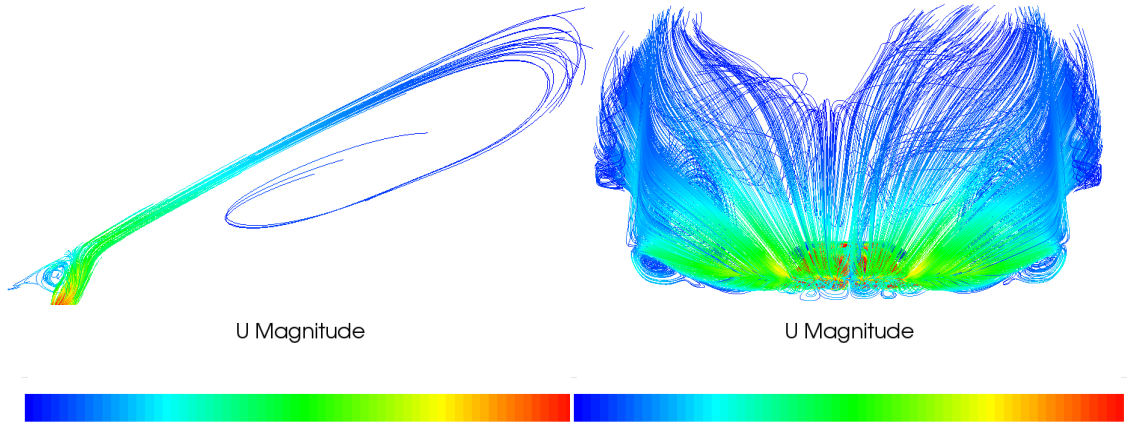


(b) *Velocity magnitude at one corresponding point of the grid created during the measurement, close to the windshield. The graph is focused on the last thousands of iterations where, even though the residuals are fluctuating, the area of interest (close to the windshield) has reached the steady–state solution. The velocity magnitude in the $y-axis$ is normalised, by being divided by its maximum value that appeared during the run.*

Figure 5.4: *Convergence of the primal problem. The residuals and the convergence of the velocity magnitude close to the windshield are satisfactory.*

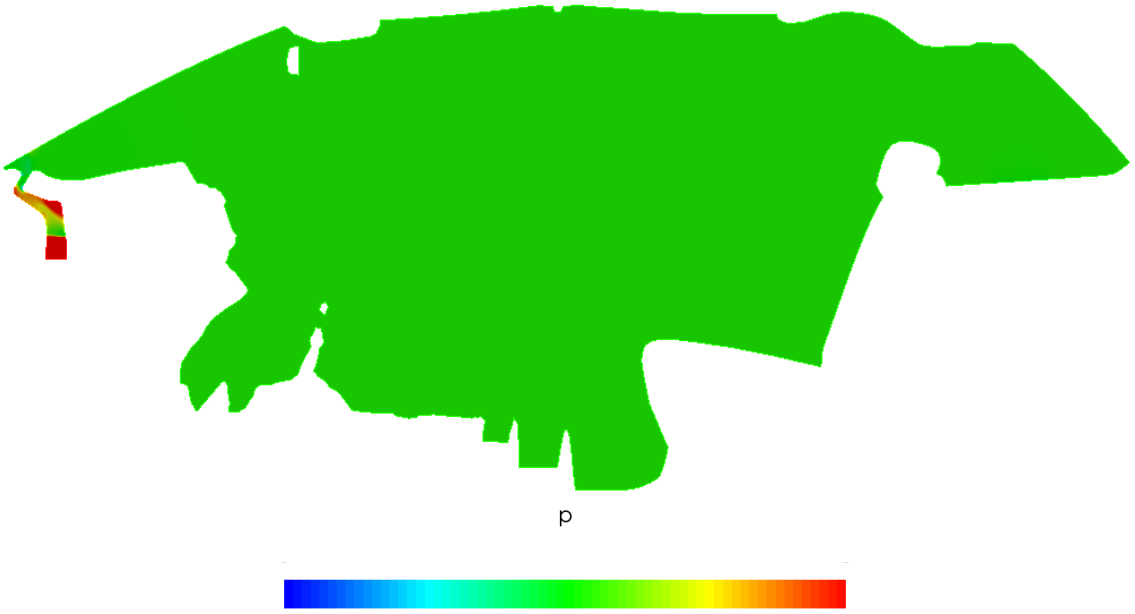


(a) *Streamlines from inlet to outlet.*

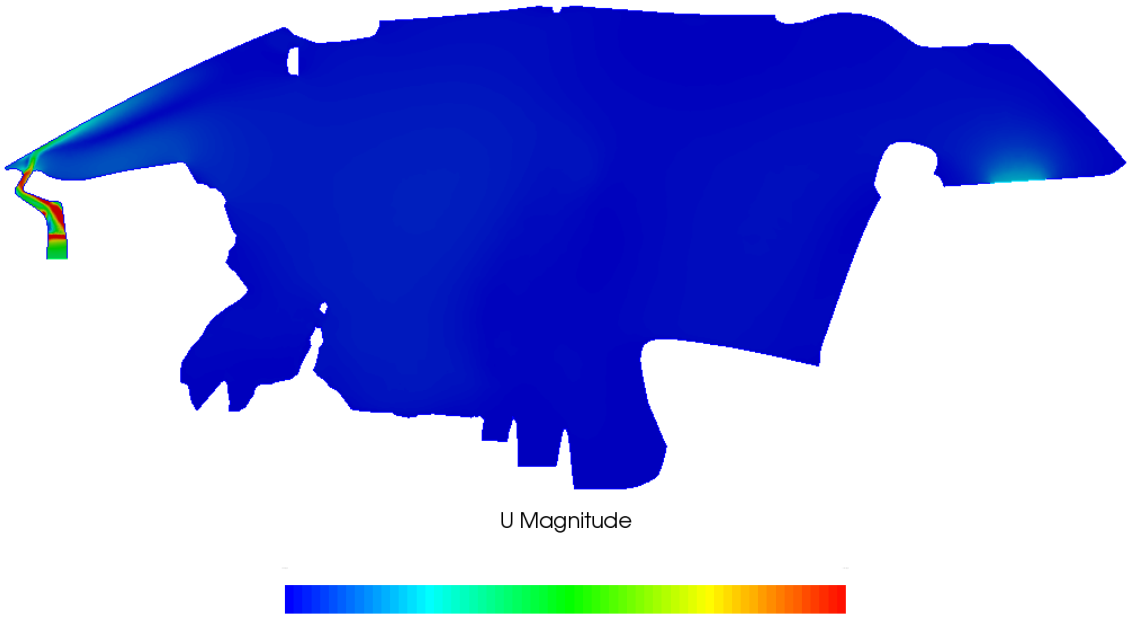


(b) *Streamlines near the windshield.* **(c)** *Streamlines on the windshield.*

Figure 5.5: *Streamlines emitted from seed points at the defroster inlet, indicate the presence of small vortices of low velocity air at the bottom of the windshield, below the level where the jet flow starts to be attached to the windshield. The jet flow stays attached almost up to the level of the rear view mirror where it starts again to recirculate.*



(a) *Pressure distribution accross the symmetry plane. Inside the cabin, pressure is almost zero while in the duct, pressure takes "big negative" values in some areas and "big positive" in some others.*



(b) *Velocity distribution accross the symmetry plane. Inside the cabin, the magnitude of velocity is almost zero while in the duct and close to the windshield, it reaches quite high values.*

Figure 5.6: *Flow fields distributions across the symmetry plane. The color scale used, indicates low values with blue color, medium with green and high with red.*

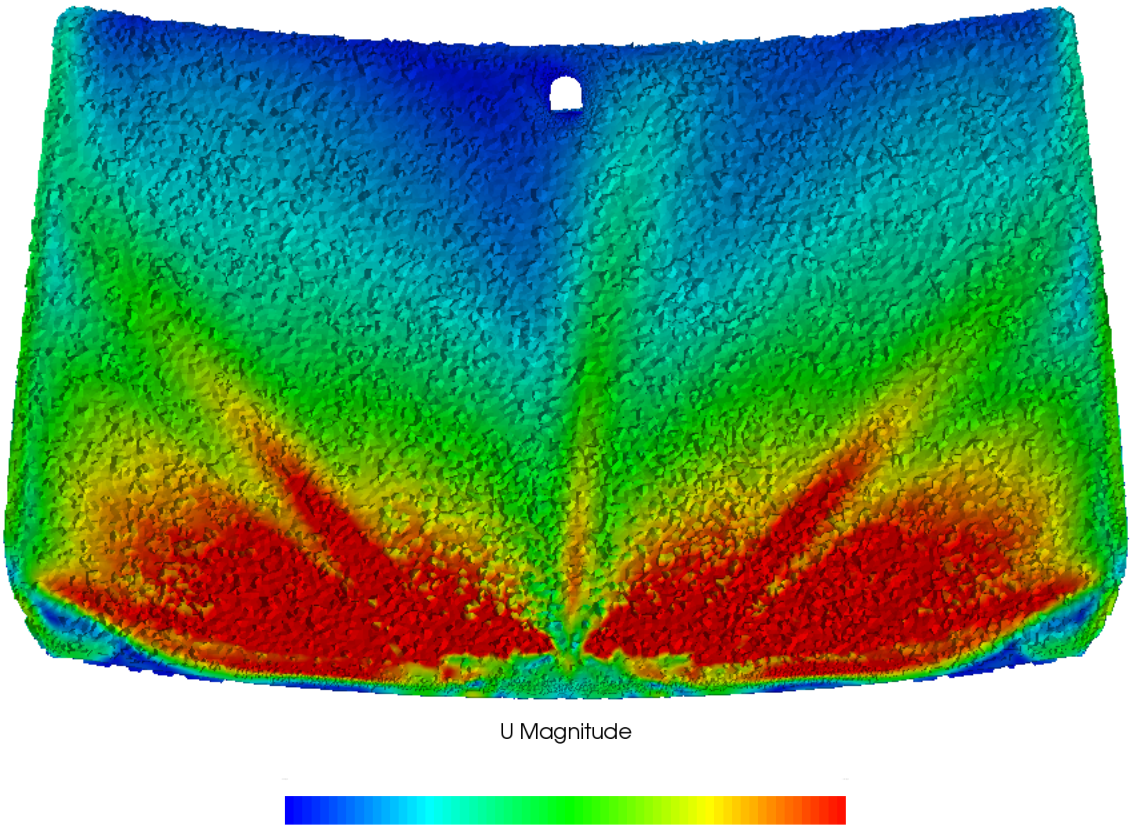
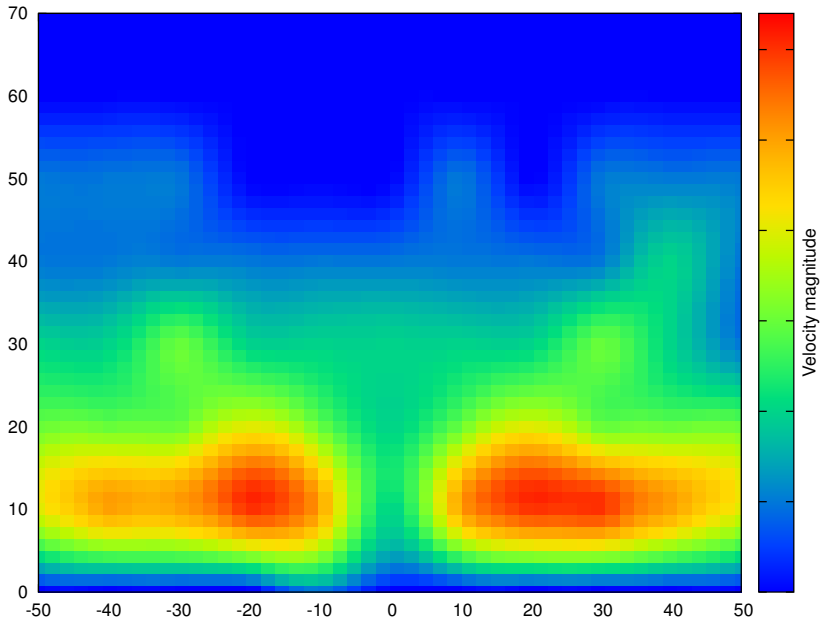
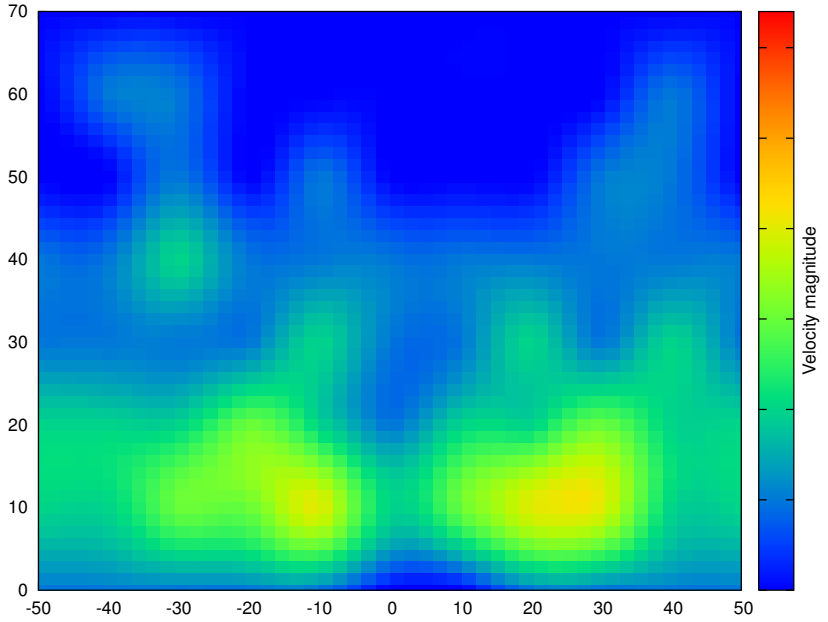


Figure 5.7: *Velocity pattern close to the windshield provided by the CFD run. The velocity magnitude contours shown are on a plane 7mm away from the inner surface of the windshield. The air velocity distribution is colored indicating the low velocity areas with blue color and the high velocity areas with red. It is obvious that the coverage is considerably weak at the upper part of the windshield where low velocity air reaches, while in the lower part, close to the defroster nozzle outlets, the velocity is quite high.*



(a) *2D velocity pattern from CFD, extracting only the velocity magnitude on the same points used in the experimental measurements,*



(b) *2D velocity pattern from measurement.*

Figure 5.8: *Comparison between CFD and measurement for the velocity pattern close to the windshield. The axes are indicating the position on the 2D grid created during the measurement (see fig. 5.1). The values collected for each figure, from both CFD and measurement, are interpolated for better visualization. The two figures have a few differences, however, qualitatively they give the same velocity distribution. In both experimental and CFD study, higher flow velocity is observed on the driver side of the windshield rather than the passenger side, as expected. In general CFD analysis showed the same trend with test data, using visual observation.*

Chapter 6

CFD–based Optimization of Defroster Nozzle

6.1 Solution of the Adjoint Equations and Sensitivity Map

The velocity patten acquired from both the CFD simulation and measurement, as mentioned in the previous chapter, reflects the weaknesses of the current defrost pattern. It is logical and also suggested by papers such as [10] that ideally the airflow velocity distribution close to the surface of the windshield should be homogenous in order to facilitate defrosting performance. Also, spottiness is not desirable from the driver point of view. For these reasons the value of the target velocity is selected to be constant all over the windshield in order to boost uniformity of the optimized velocity pattern. Moreover, the fact that the upper part of the windshield has currently low velocity air reaching it, it is a drawback for defogging performance which has to be tackled through the optimization process.

Consequently, in order to force the pattern to improve, tackling these weaknesses, we first set as target velocity a constant velocity v_{tar} and as the target volume (the volume in which the objective function is computed), an offset windshield $7mm$ away of the inner surface of the real one, that has a thickness of $20mm$. The cells that happen to reside inside the defined target volume are those where the objective function is finally computed. The target volume can be seen in fig. 6.1.

After defining our target velocity and target volume we are ready to proceed with the solution of the adjoint equations and the acquisition of the sensitivity map. The adjoint solver is let to run for a few thousands of iterations until the residuals become quite low (below 10^{-7}) and the adjoint fields are stabilized, see fig.6.3. The maximum values of adjoint velocity and adjoint pressure can also indicate convergence, so they are monitored during the run. The sensitivity derivatives are then calculated for the defroster nozzle, that is the area which will be morphed.

(a) *Target volume side view.* **(b)** *Target volume top view.*

Figure 6.1: *Target volume is an offset windshield* $7mm$ *away from the the real one, with thickness of* $20mm$*, and it is the volume where the objective function is computed. It includes 660104 cells of the computational domain. The target velocity* v_{tar} *is defined as a constant velocity magnitude for all the target volume.*

The resulting sensitivity map can be seen in fig. 6.2. Blue color indicates areas (cells) that should be pulled outwards while red indicates areas to be pushed inwards in order to achieve a better velocity distribution on the windshield.

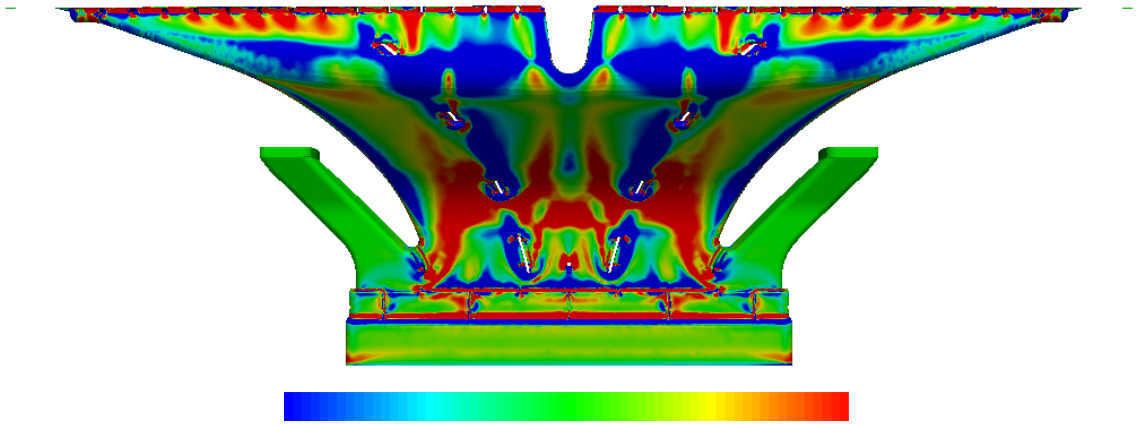
A sensitivity map illustrates the derivatives of the objective function w.r.t. the normal displacement of the wall boundary nodes of the selected shape. Sensitivity maps, in general, are used to highlight the areas where aerodynamic improvement has the greatest potential and are a valuable tool for the designer, even if automatic optimization is not applied.

Consequently, the sensitivity map is used to indicate the areas where a shape change can have great impact in the flow. However, morphing the geometry using raw information given to the sensitivity map is impractical, since it is quite noisy. In order to compute the exact displacement of each point of the mesh (or each control point, in our case) the steepest descent algorithm is used. Moreover, the parameterization of the area to be morphed is performed using B–splines that have the property to also smooth the computed sensitivities.

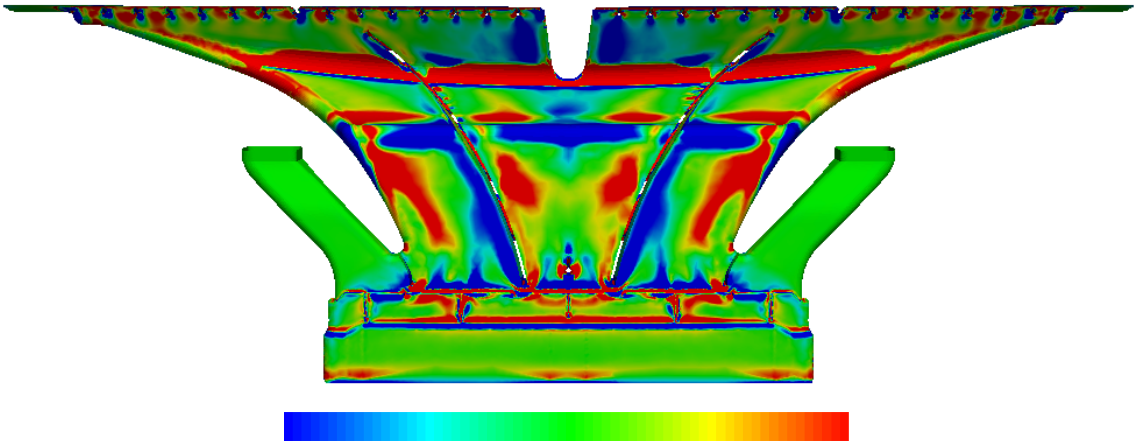
6.2 Automated Runs

For the automated optimization process, a tool developed by PCOpt/NTUA is again used. The in–house adjoint solver developed in OpenFOAM is coupled with a mesh parameterization and movement strategy based on volumetric B–splines, as analyzed in chapter 3. After setting up the optimization and solver parameters, the loop is running until the termination criteria are met (for example maximum number of iterations) or until a non–acceptable mesh (in terms of quality) is produced.

Several optimization runs were performed until the most suitable new shape of the defroster nozzle was produced. The runs differed in their parameterization, the parameterized area and the target volume. The need to perform several optimization



(a) *Shape sensitivities on the front side of the duct.*



(b) *Shape sensitivities on the back side of the duct.*

Figure 6.2: *Shape sensitivities indicate the change of the objective function F due to normal discplacement of cells on the design boundary, in this case of the inside walls of the defroster duct ($\frac{\delta F}{\delta n_i}$). Here, blue highlights areas to be pulled outwards while red areas to be pushed inwards so as to decrease the value of F, in other words to come closer to the target.*

runs, instead of only one, derives from the necessity of selecting the appropriate parameterization setup that gives more optimization potential, but most importantly it is due to the fact that the manufacturing and the topology constraints are not included in the optimization. At the end of each automated run, those are taken into consideration, as the new candidate shape for the defroster nozzle has to be manufacturable and fit inside the assembly of its neighbouring parts.

Run 1 The first run was a first attempt to get familiar with the setup of the automated chain process and to get some initial results. The selection box that includes the area to be parameterized and to be morphed, includes the defroster duct only, excluding the areas of its inlet and outlet. A control grid of $3 \times 7 \times 9$ control points with degrees of $p_u = 2, p_v = 3, p_w = 3$ respectively was used, fig. 6.5.

Only the red control points are allowed to move. The rest are defined as frozen, so as to prevent overlapping between the parameterized and non–parameterized points of the duct and, also, to preserve fixed the inlet and outlet areas. The structured grid (control box) included 497521 mesh points. The displacement of the control points is confined in the Z direction so as to prevent overlapping of the mesh points.

Convergence is seen in fig 6.6. The mesh displacement in the final (14^{th}) shape leads to 55% decrease in the objective function, fig. 6.6. The comparison of the two new shapes, along with the comparison of the $v - v_{tar}$ fields on the target volume are shown in fig. 6.7.

Run 2 The parameterization setup was exactly as in Run 1, fig. 6.8. However, in this loop, it was first tried to change the target volume. In some previous trial loops it happened that the objective function was decreasing during the optimization cycles, however the resulting pattern showed that this was achieved by decreasing the high velocities at the bottom of the volume only, instead of also increasing the low velocities at its top. So one of the two targets, which are uniformity and higher velocities at the top, was not achieved. To strengthen the second and more important target and make it more clear to the optimization, only the upper part of the previous target volume was used.

The convergence of the optimization loop as well as the 17% decrease in the objective function that is given by the 10^{th} shape produced can be seen in fig 6.9. Moreover, the comparison of the two new shapes, along with the comparison of the $v - v_{tar}$ fields on the target volume are shown in the following fig. 6.10.

Compared to the previous run, the objective function drop is less, even though the setup of the control points remains the same. That is due to the fact that using only the upper half of the windshield as target volume, we strictly express the necessity to increase the velocity magnitude there, rather than over the whole windshield, which is more difficult to accomplish.

Run 3 This run is applied by putting an extra restriction to the displacement of the control points. Their allowed displacement is defined by averaging the displacement of all the control points residing in an iso–x, iso–y and iso–z plane. In other words, for example, all the control points laying in an iso–x plane are displaced in the x–direction using the averaged x–component of their sensitivity derivatives. That contributes in producing smoother shapes.

The selection box and the control grid of $3 \times 7 \times 9$ control points with basis function order set to $p_u = 2, p_v = 3, p_w = 3$ respectively can be seen in fig. 6.11. The control grid encloses 526157 mesh points. The displacement of the control points is confined in the Z direction.

The loop convergence graph is shown on fig. 6.12 according to which the last

produced shape, the 12^{th} geometry, achieves 46% drop of the objective function. The produced new shape compared to the initial one, as well as their $v - v_{tar}$ fields on the target volume can be seen in fig. 6.13

Compared to the previous run, the objective function has decreased significantly (even though the averaging of the displacements could imply less optimization potential) due to the fact that the back row of the control points is now active, giving the ability to deform bigger part of the initial geometry. The back row was previously frozen to prevent overlapping of the mesh there. However, in this run, with this setup, this problem does not appear.

Run 4 In this run, averaging is of the displacements per iso–plane is applied too, however compared to the previous run, the differences are that the displacement of the control points is also confined in the Y direction, and that only the very top row of control points is frozen, instead of the two top rows that are frozen in the previous run. The parameterization setup was exactly as in Run 3, fig. 6.14.

The last produced shape, the 2^{nd} geometry, achieves 43% drop of the objective function. The produced new shape compared to the initial one, as well as their $v - v_{tar}$ fields on the target volume can be seen in fig. 6.15.

This optimization run produces only one new shape as the 3^{rd} mesh overlapps. Consequently, the quality check fails and the run is terminated. Nevertheless, even in one optimization cycle, the objective function achieves an impressive drop, due to the fact that only the top row of control points is frozen, letting more nodes close to the outlet of the nozzle to move. This displacement appears to have great impact on the flow near the windshield.

6.3 Selection and manufacture of the new defroster nozzle using rapid prototyping–Validation with defrost test

From the above automated runs and the final new geometries produced, the most suitable new shape, improving significantly the velocity distribution on the windshield, is the one resulting from the 4^{th} run. Its surfaces are very smooth, so it is appropriate for manufacturing, and also the corresponding STL when placed in the position of the initial defroster nozzle, in the assembly of its neighbouring CAD parts, it fits. Consequently, this geometry is appropriate to be manufactured using rapid propotyping and, then, placed in the vehicle to be validated with a defrost test.

Rapid prototyping techniques are common for quickly fabricating models of parts, appropriate for many uses, including testing. In this case, $3D$ printing is

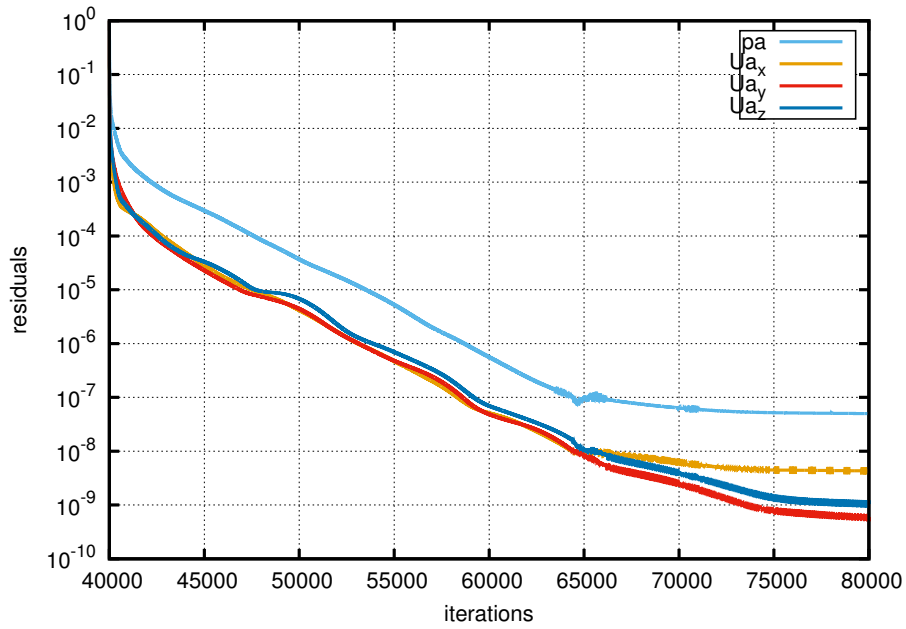
used to fabricate the candidate new defroster nozzle shape, provided in STL format. The accuracy of the process is sufficiently high, giving a prototype suitable for our purpose. The prototype is properly placed in the test vehicle, replacing the initial defroster nozzle.

6.3.1 Defrost Tests

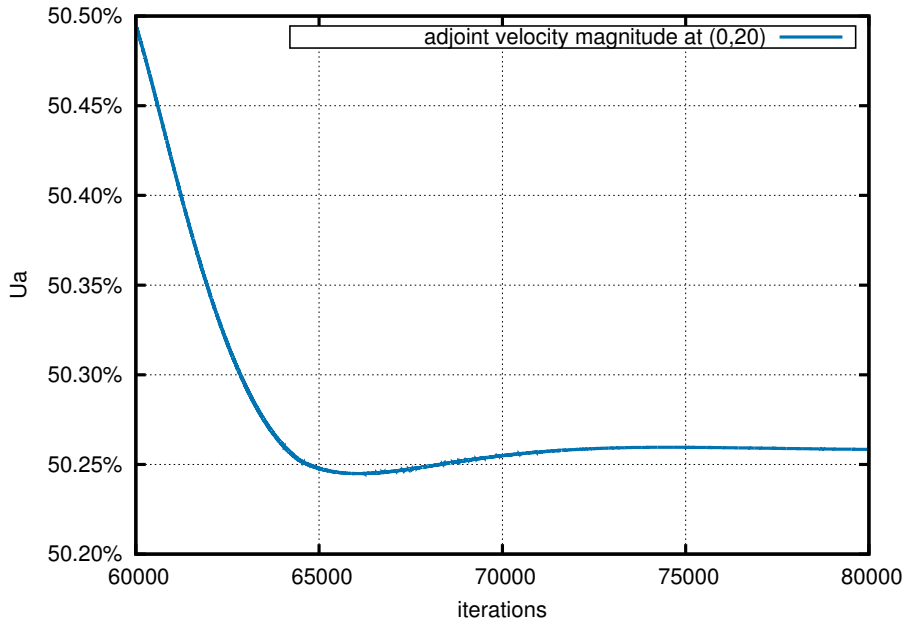
Windshield defrost patterns are obtained from cold room testing [9], [4]. To validate the improved defrosting and demisting efficiency of the new defroster nozzle shape, compared to the initial one, defrost tests for each defroster nozzle are performed.

To reproduce cold start condition, the vehicle is soaked for 6 hours at a temperature of $-20°C$, in TME's climatic chamber. Following the soak, a humiditity generator of capacity $360g/h$ was used to form a frost layer on the windshield for another 30min. Then, the vehicle's engine is started, the defrost option is turned on and the defrost test commenced. The pattern is recorded every few minutes and the windshield is marked from the inside to indicate clearance areas.

The melting pattern for the original and the improved defrosters, for two different time milestones are shown in fig. 6.16. In every instant recorded, the new defroster nozzle geometry gives a bigger clearance zone compared to the initial one. At the end, the new geometry results in clearing the windshield completely, in 15% less time than the initial defroster nozzle.

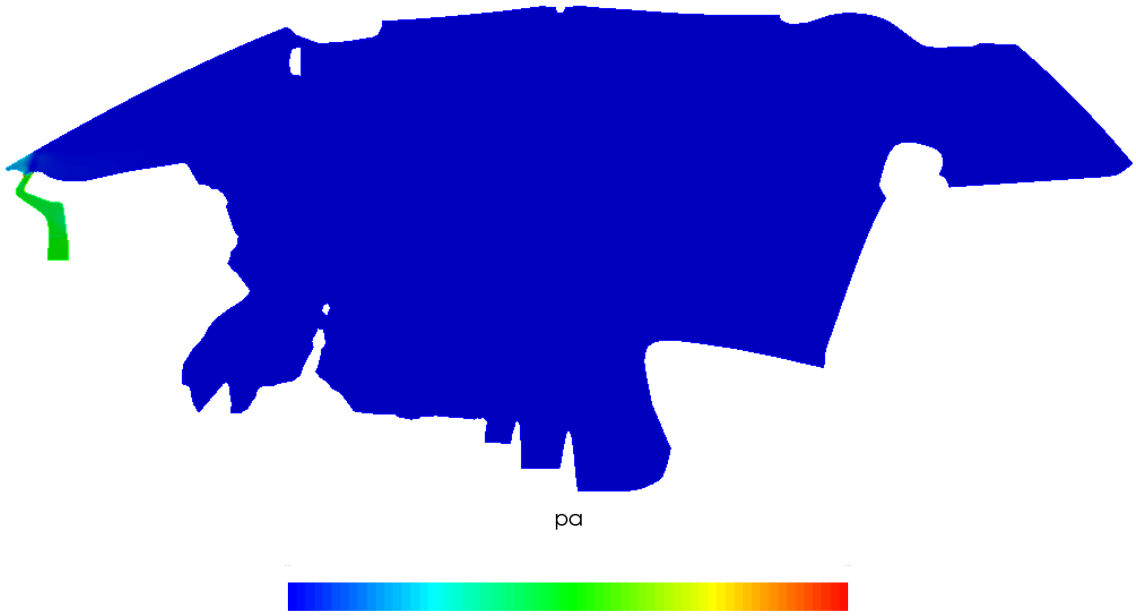


(a) *Non–dimensional residuals are reduced by more than 7 orders of magnitude.*

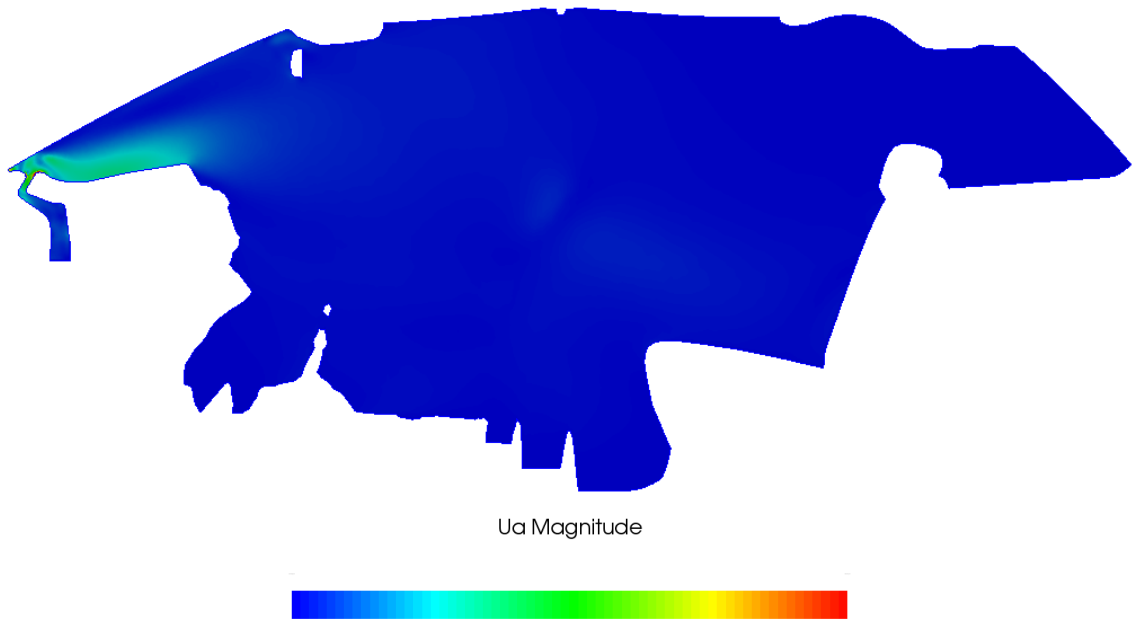


(b) *Adjoint velocity magnitude at one point of the grid created during the measurement, close to the windshield. The graph is focused on the last thousands of iterations where, the area of interest (close to the windshield) has reached the steady–state solution. The adjoint velocity magnitude in the $y-axis$ is normalised, by being divided by its maximum value that appeared during the run.*

Figure 6.3: *Convergence of the adjoint problem. The residuals and the convergence of the velocity magnitude close to the windshield are satisfactory.*



(a) *Adjoint pressure distribution accross the symmetry plane.*



(b) *Adjoint velocity distribution accross the symmetry plane.*

Figure 6.4: *Flow fields distributions accross the symmetry plane. The scale color used, indicates low values with blue color, medium with green and high with red. The adjoint variables are almost zero inside the cabin, but close to the the duct and the windshield they take greater values as these areas have an impact on the flow in the target volume.*

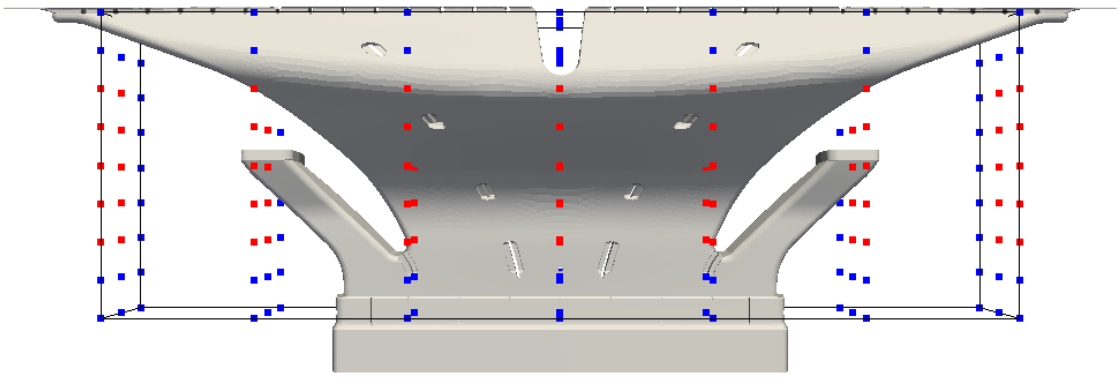


Figure 6.5: ***Run 1****: The boundaries of the control box as well as the active (red) and the frozen (blue) control points define the mesh points to be parameterized and displaced. The frozen control points help to avoid overlapping of the mesh between the parameterized and the non–parameterized areas. In this case, the back row, the two top rows and the two bottom rows of control points remain frozen.*

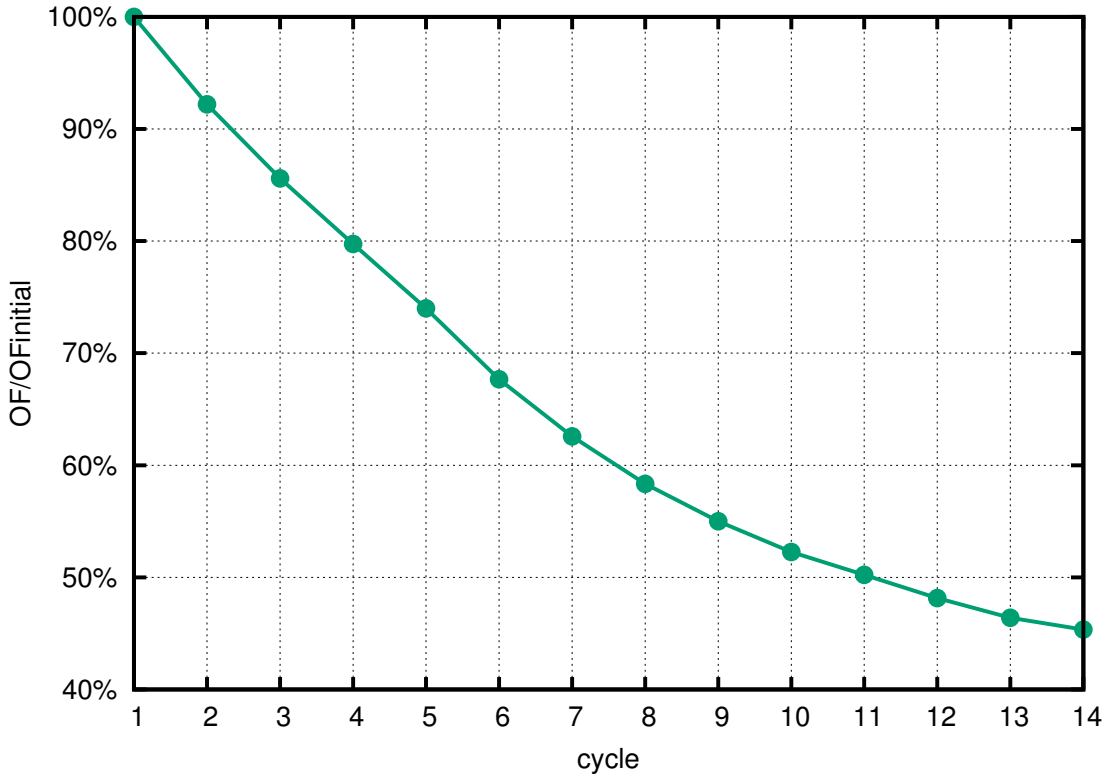
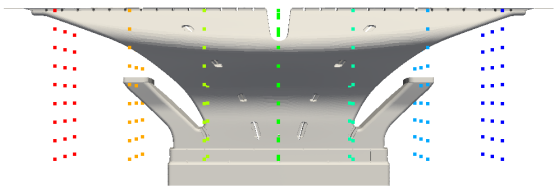
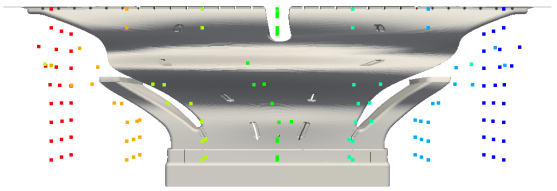


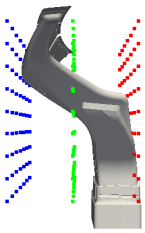
Figure 6.6: ***Run 1****: Evolution of the objective function during the optimization.*



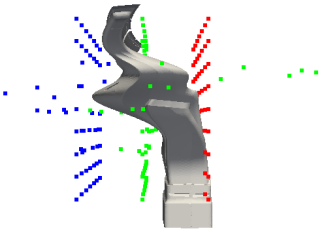
(a) *Original duct, front view.*



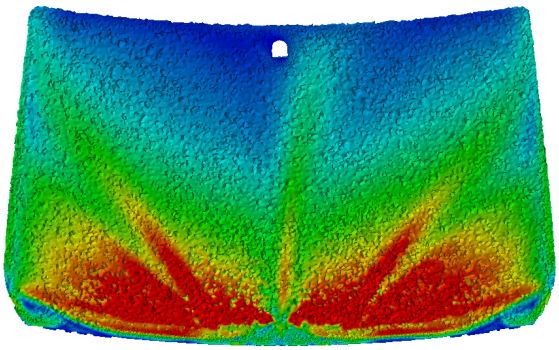
(b) *Final duct, front view.*



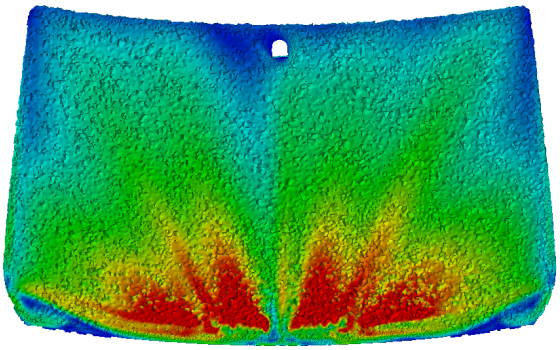
(c) *Original duct, side view.*



(d) *Final duct, side view.*



(e) *Original pattern.*



(f) *Final pattern.*

Figure 6.7: ***Run 1****: In the first row, the control grid nodes are colored based on the v coordinate while in the second row, based on the u. The final shape has many bumps and undercuts so it is not suitable for mass production. However, some of the previous shapes, of previous optimization cycles, could be manifactured as they have smaller displacement compared to the initial shape. The field shown on the target volume is $v - v_{tar}$ in which green areas correspond to areas where the target velocity magnitude v_{tar} was reached, blue to areas with lower air velocity and red to areas with greater air velocity than the target. Consequently, looking at the last row, the comparison between the two patterns shows improvement in the coverage of the upper part of the windshield, as well as increased uniformity.*

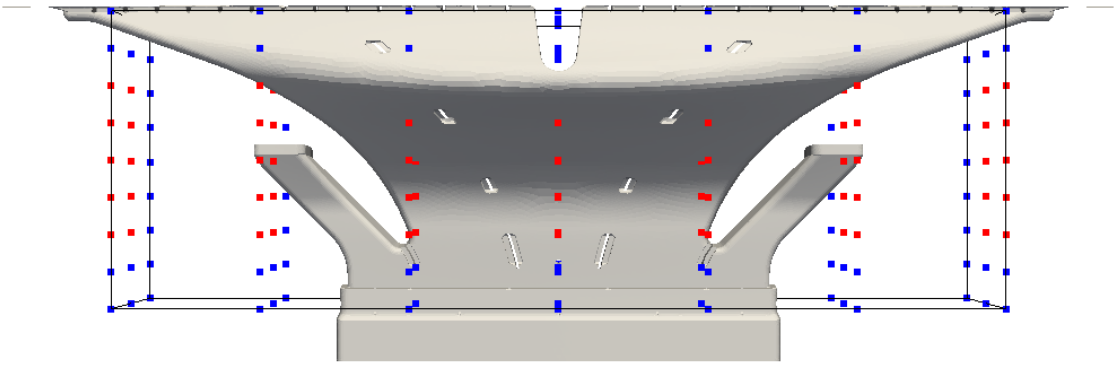


Figure 6.8: ***Run 2****: The boundaries of the control box as well as the active (red) and the frozen (blue) control points define the mesh points to be parameterized and displaced. The frozen control points help to avoid overlapping of the mesh between the parameterized and the non–parameterized areas. In this case, the back row, the two top rows and the two bottom rows of control points remain frozen.*

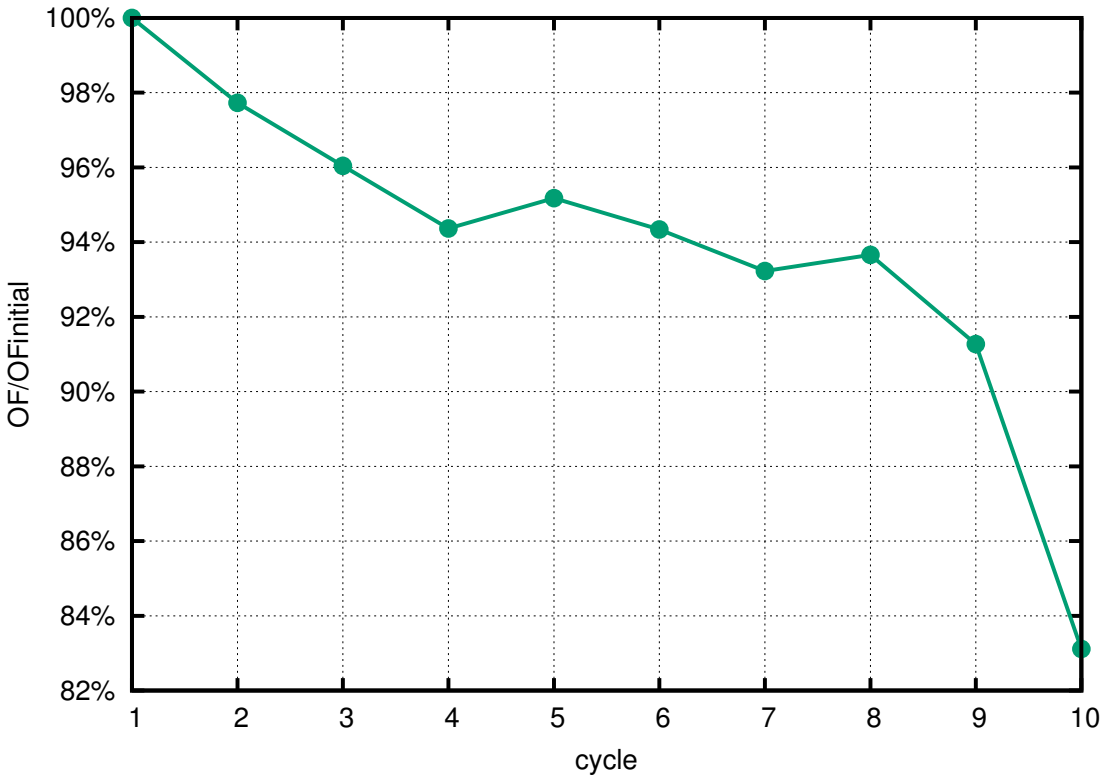
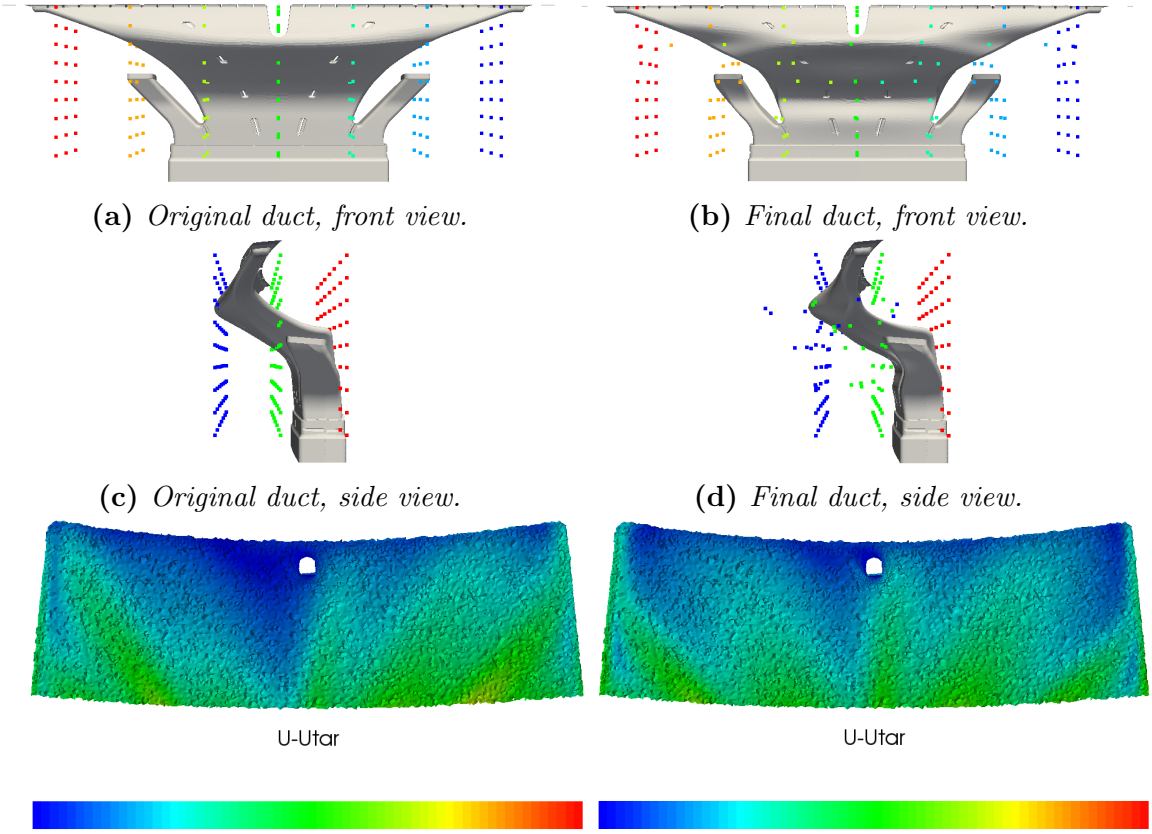


Figure 6.9: ***Run 2****: Evolution of the objective function during the optimization.*



(a) *Original duct, front view.* **(b)** *Final duct, front view.*

(c) *Original duct, side view.* **(d)** *Final duct, side view.*

(e) *Original pattern.* **(f)** *Final pattern.*

Figure 6.10: ***Run 2****: In the first row, the control grid nodes are colored based on the v coordinate while in the second row, based on the u. The final shape produced has many intense curves and undercuts so it seems inappropriate for mass production. The features created could not be molded using a single pull mold so the suggested shape could not be easily manfactured. The field shown on the target volume is $v - v_{tar}$ in which green areas correspond to areas where the target velocity magnitude v_{tar} was reached, blue to areas with lower air velocity and red to areas with greater air velocity than the target. The objective function is not much reduced and this is reflected on the result on the target volume. The coverage on the upper part of the windshield is slighly improved, however there is more potential.*

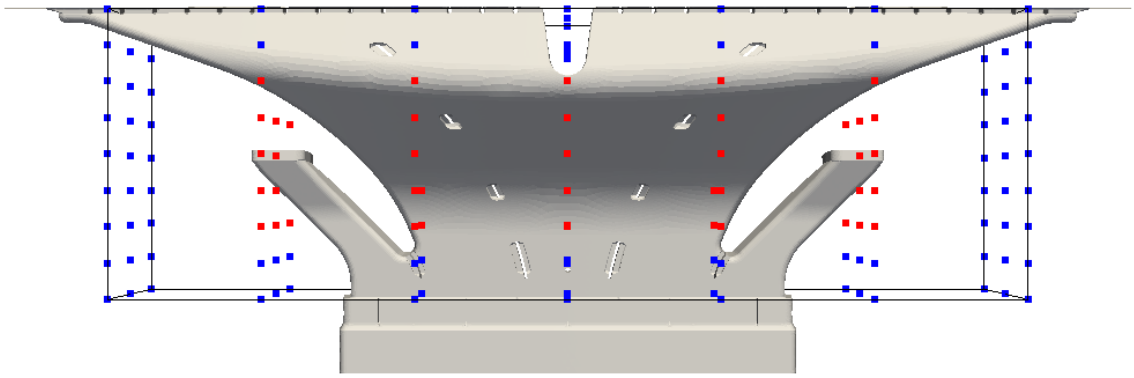


Figure 6.11: ***Run 3****: The boundaries of the control box as well as the active (red) and the frozen (blue) control points define the mesh points to be parameterized and displaced. The frozen control points help to avoid overlapping of the mesh between the parameterized and the non–parameterized areas. In this case, the side rows, the two top rows and the two bottom rows of control points remain frozen.*

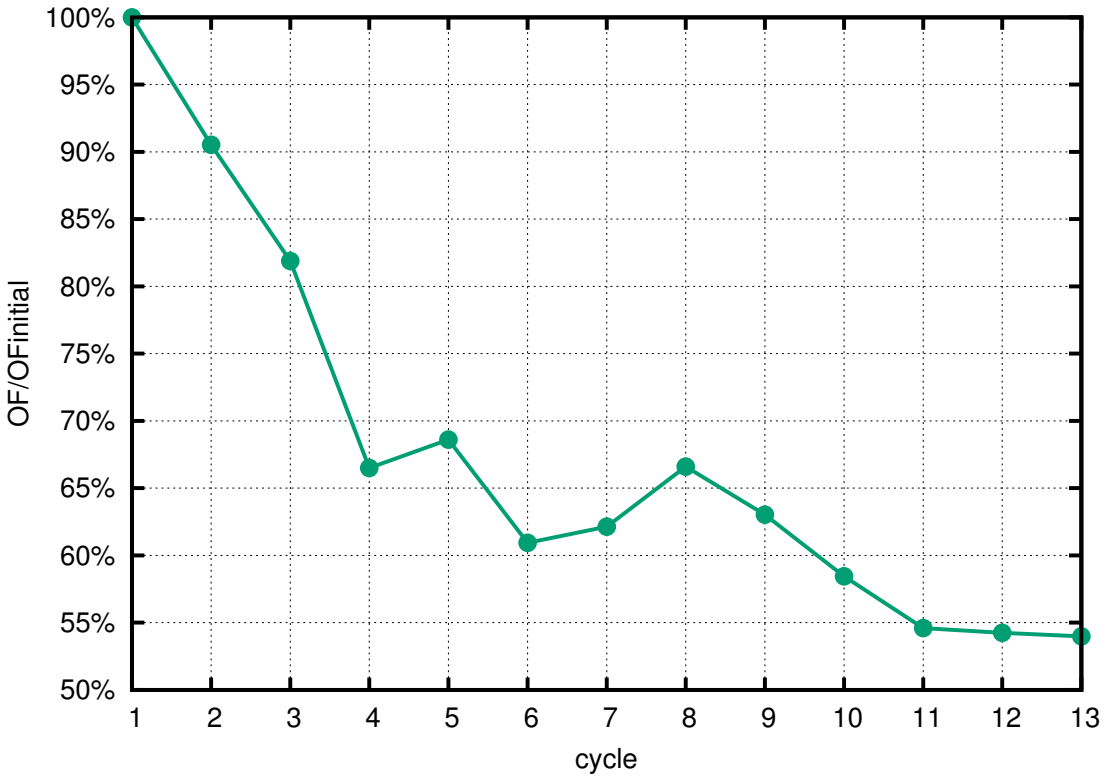
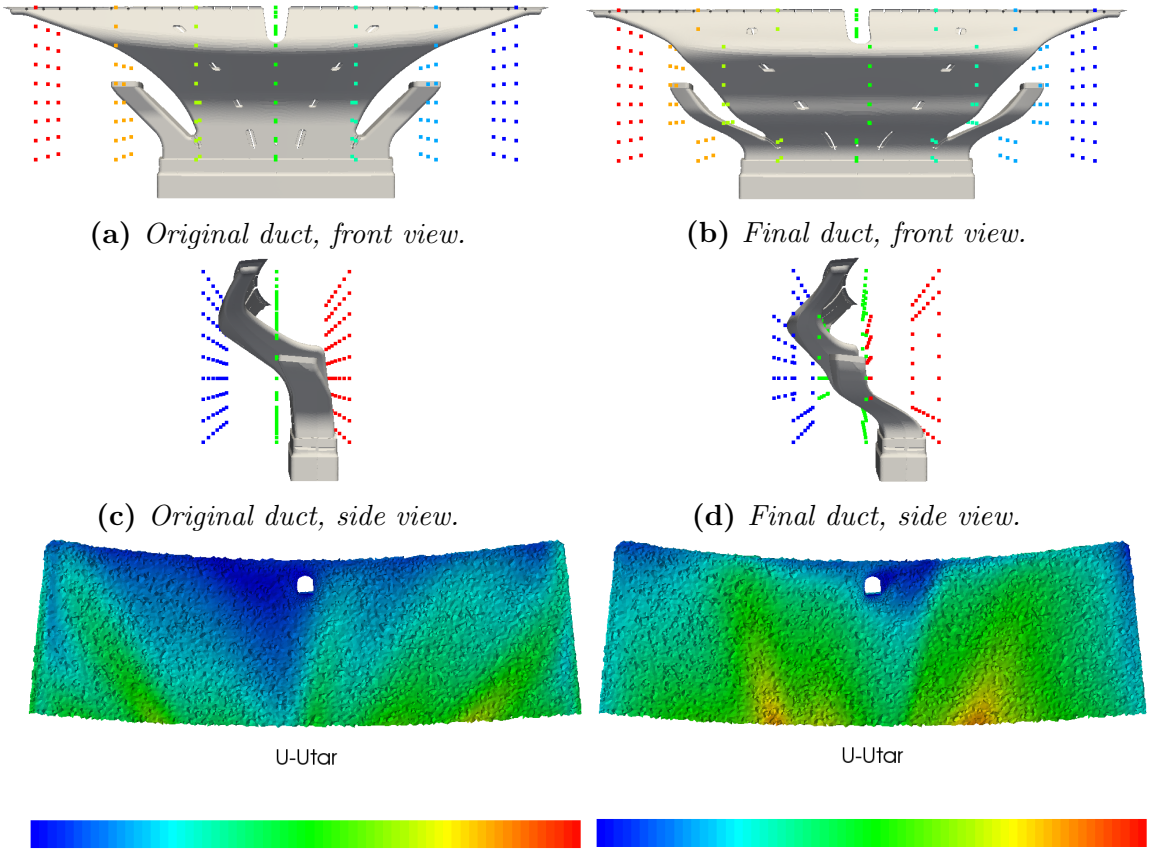


Figure 6.12: ***Run 3****: Evolution of the objective function during the optimization.*



(a) *Original duct, front view.*

(b) *Final duct, front view.*

(c) *Original duct, side view.*

(d) *Final duct, side view.*

(e) *Original pattern.*

(f) *Final pattern.*

Figure 6.13: ***Run 3****: In the first row, the control grid nodes are colored based on the v coordinate while in the second row, based on the u. The final control points position indicates the fact that their displacement per iso–plane is averaged. In other words, the right figures show that the active control points that belong to the same iso–plane have the same displacement in that direction, that resulted from the averaging of the component of the sensitivity derivatives on that plane, producing a smoother shape. The field shown on the target volume is $v - v_{tar}$ in which green areas correspond to areas where the target velocity magnitude v_{tar} was reached, blue to areas with lower air velocity and red to areas with greater air velocity than the target. The resulting pattern is much improved as the target velocity reaches almost the top of the windshield (target volume).*

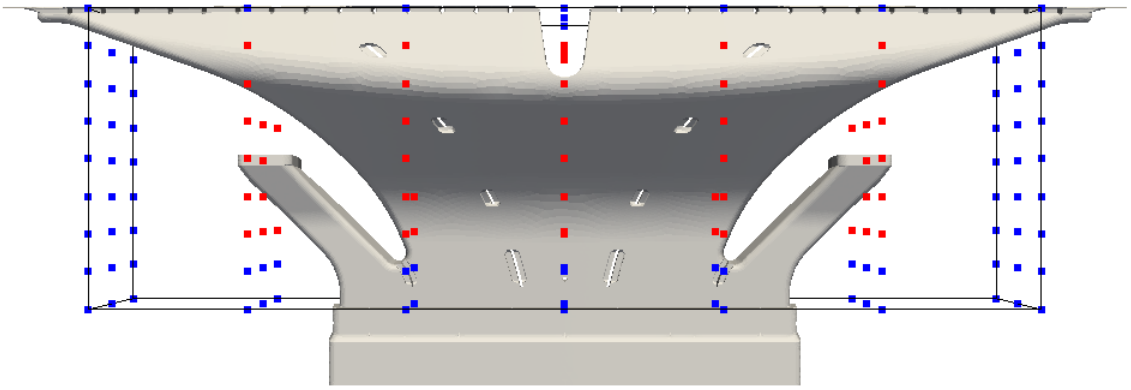
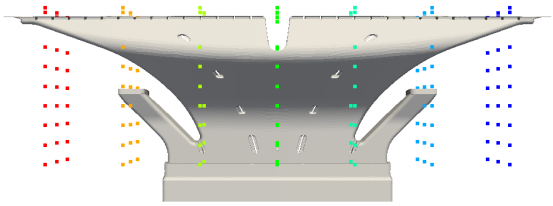
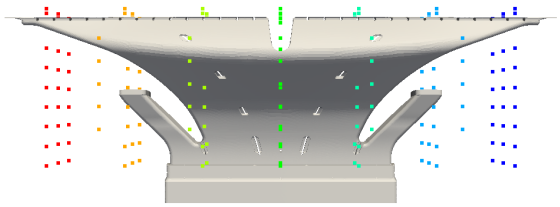


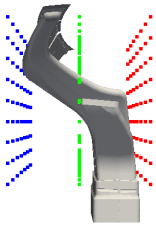
Figure 6.14: ***Run 4****: The boundaries of the control box as well as the active (red) and the frozen (blue) control points define the mesh points to be parameterized and displaced. The frozen control points help to avoid overlapping of the mesh between the parameterized and the non–parameterized areas. In this case, the side rows, two top row and the two bottom rows of control points remain frozen.*



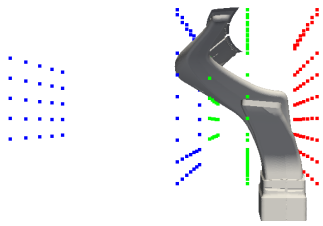
(a) *Original duct, front view.*



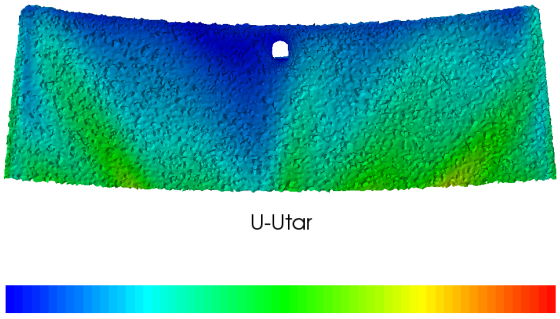
(b) *Final duct, front view.*



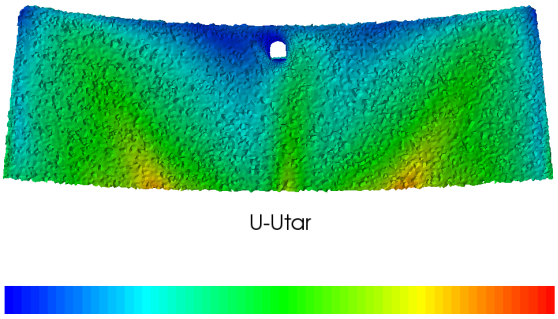
(c) *Original duct, side view.*



(d) *Final duct, side view.*

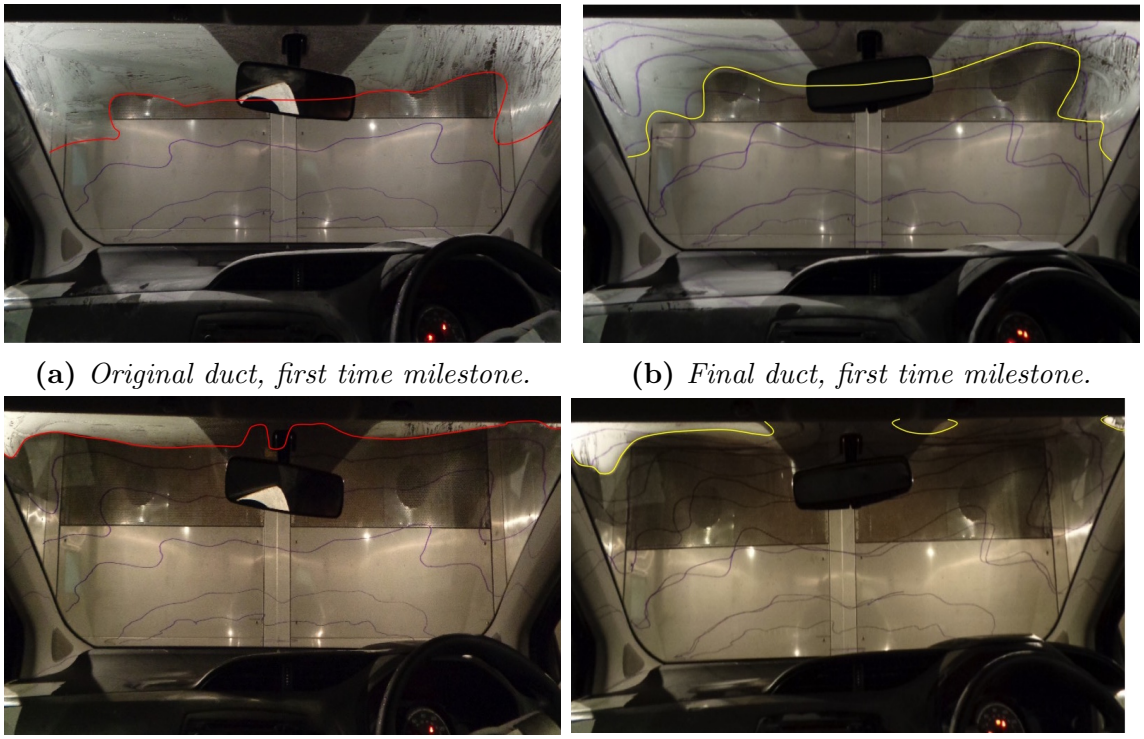


(e) *Original pattern.*



(f) *Final pattern.*

Figure 6.15: ***Run 4****: In the first row, the control grid nodes are colored based on the v coordinate while in the second row, based on the u. The final shape is very smooth ,due to the averaging per iso–plane of the displacements of the control points and it seems to be suitable for mass production. The field shown on the target volume is $v - v_{tar}$ in which green areas correspond to areas where the target velocity magnitude v_{tar} was reached, blue to areas with lower air velocity and red to areas with greater air velocity than the target. Consequently, looking at the last row, the comparison between the two patterns shows improvement in the coverage of the upper part of the windshield, as well as increased uniformity. The target is reached on the majority of the cells of the target volume.*



(a) *Original duct, first time milestone.*

(b) *Final duct, first time milestone.*

(c) *Original duct, second time milestone.*

(d) *Final duct, second time milestone.*

Figure 6.16: *Defrost test of the original and the improved defroster nozzle shapes. The melting pattern is recorded and marked on the inside surface of the windshield every few minutes. The patterns in the above figures are compared for the same time milestones, proving the improved defrosting efficiency of the optimized shape. Finally, the windshield, with the new defroster nozzle shape, is completely clear in 15% less time compared to the original shape.*

Chapter 7

Summary–Conclusions

In this diploma thesis, shape optimization using the continuous adjoint method, developed by the PCOpt Unit of NTUA, was applied to the defroster nozzle, part of the HVAC unit, of a Toyota passenger car. The high importance of the geometry of this nozzle, as far as the demisting and defogging of the windshield is concerned, led to the investigation of its shape optimization potential, improving the efficiency of this vital function of the vehicle.

To begin with, the performance requirements of the HVAC system to dispel condensation or frost on the windshield within a reasonable time and perform uniform defrosting all over its surface, was appropriately expressed in the form of a suitable objective function. This objective function, to be minimized, is the integral of the difference of the air velocity from a target one over a thin control volume, defined close to the windshield inside the car cabin. Moreover, the continuous adjoint method for incompressible, steady–state flow, was mathematically presented in detail, along with the parameterization and shape morhing tool used (developed by PCopt/NTUA), based on volumetric B–splines.

As a first comparison between a measurement and the simulation of the equivalent CFD model, a pressure measurement including the defroster nozzle was performed. The measurement setup included a pressurized cubic chamber, a small box attached to it in order to facilitate the transition of the flow and the defroster nozzle connected to the latter. Four different airflows were applied to the one side of the chamber, with the air exiting from all the outlets in the first case tested, while in the second one, the side outlets are sealed. During the above 2×4 cases performed, static pressure was measured at the center of the top side of the chamber. This setup was modeled with a 2.5 million cells mesh and solved using RANS in conjuction with the k–ϵ turbulence model, in OpenFOAM. The pressure measured and the pressure derived from the CFD analysis on the point of measurement were very close, validating the CFD simulation used.

The main target of this thesis, the shape optimization of the defroster nozzle,

requires to simulate the flow coming from the defroster nozzle towards the interior of the cabin and exiting from its back. For that reason, as a second step, a mesh model of the defroster nozzle connected to the cabin, of almost 8.4 million cells was created. The CFD simulation provided the flow fields necessary for the solution of the adjoint problem but, also, a clear image of the current velocity distribution close to the windshield. This velocity distribution was compared with the one acquired through hot–wire anemometer measurement performed on a 10cm spaced grid, 7mm away from the windshield. The results presented indicate similar velocity patterns, validating the CFD simulation was in good agreement with the experimental study.

The third step was the solution of the adjoint equations and the use of the automated shape optimization process to produce new candidate geometries for the defroster nozzle. Firstly, the use of the adjoint solver developed by PCopt/NTUA in the OpenFOAM environment, gave the sensitivity map, a valuable tool for the designer, indicating how the shape of interest should be morphed to reach the target (minimize the objective function). Afterwards, several automated runs were performed, with their main differences being in the parameterization setup or/and on the volume where the objective function is computed. The need for mass production of the final shape requires more smooth/manufacturable surfaces, which become possible with the technique of averaging the displacements of the control points per iso–plane. This helpful feature was included in the in–house morpher.

The above process resulted in one final geometry, chosen to be the most appropriate for being manufactured using rapid prototyping so as to be tested, to validate its improved defrosting capability, compared to the initial defroster nozzle geometry. The defrost tests performed using the initial and the improved shapes showed significant improvement in the melting pattern on the windshield and decrease in the total time required for the complete clearance of its surface.

More drastic improvement could be accomplished, by modifying the outlet shape, or by applying a different approach, topology optimization using adjoint method, instead of shape morphing optimization. In that case, the study would not begin from the current defroster nozzle shape but from a $3D$ space or, even better, from a draft shape.

Bibliography

[1] M. B. Giles, N. A. Pierce, *An Introduction to the Adjoint Approach to Design.*, 1999.

[2] T. Kohnotou, Y. Iwamoto, K. Hoshiawa, M. Nagataki, *Optimum Design of Defroster Nozzle.*, SAE Technical Paper, 1992.

[3] C. Othmer, *CFD Topology and Shape Optimization with Adjoint Methods.*, 2006.

[4] A. F. Skea, R. D. Harrison, A. J. Baxendale, D. Fletcher, *Comparison of CFD Simulation Methods and Thermal Imaging with Windscreen Defrost Pattern.* Fluids Group, MIRA, SAE Technical Paper, 2001-01-1720, 2001.

[5] K. J. Nasr, B. S. AbdulNour, G. C. Wiklund, *State of Knowledge and Current Challenges in Defrosting Automotive Windshields.* SAE Technical Paper, 980293, 1998.

[6] E.M. Papoutsis–Kiachagias, *Adjoint Methods for Turbulent Flows, Applied to Shape or Topology Optimization and Robust Design.* PhD Thesis, NTUA, 2013.

[7] J. A. Samareh, *Aerodynamic Shape Optimization Based on Free–Form Deformation.* Multidisciplinary Optimization Branch, NASA Langley Research Center, Hampton, VA, 2004.

[8] B. S. AbdulNour, *Hot–Wire Velocity Measurements of Defroster and Windshield Flow.* Ford Motor Company, SAE Technical Paper, 970109, 1997.

[9] B. S. AbdulNour, *CFD Prediction of Automotive Windshield Defrost Pattern.* Ford Motor Company, SAE Technical Paper, 1999-01-1203, 1999.

[10] W. Yang, W. Shi, F. Guo, W. Yang, *Flow Field Simulation and Performance Analysis of HVAC Defrosting Duct.* 2nd International Conference on Electronic & Mechanical Engineering and Information Technology (EMEIT-2012), China, 2012.

[11] J. Park and C. Kim, *Parametric Study on Automotive Windshield Defrost Pattern using CFD.* Hyundai MOBIS, SAE Technical Paper, 2003-01-1078, 2003.

[12] M. Samuelčík *Bézier and B–spline volumes Project of Dissertation.* Comenius University, Bratislava, 2005.

[13] K.C. Giannakoglou, D.I. Papadimitriou, E.M. Papoutsis–Kiachagias, C. Othmer, *Adjoint methods in CFD-based optimization - Gradient computation & beyond.* ECCOMAS 2012–European Congress on Computational Methods in Applied Sciences and Engineering, pp. 8523-8539, 2012.

[14] E.M. Papoutsis–Kiachagias, K.C. Giannakoglou, C. Othmer, *Adjoint wall functions: Validation and application to vehicle aerodynamics Authors of Document.*. 11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014, pp. 7593-7604, 2014.

[15] Giannakoglou, K.C., Papadimitriou, D.I., Papoutsis–Kiachagias, E.M., Kavvadias, I.S., *Adjoint methods for shape optimization and robust design in fluid mechanics.* OPT-i 2014 - 1st International Conference on Engineering and Applied Sciences Optimization, Proceedings, pp. 2252-2265, 2014.

[16] E.M. Papoutsis–Kiachagias, K.C. Giannakoglou, *Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications.* Archives of Computational Methods in Engineering, 2014.

[17] E.M. Papoutsis–Kiachagias, A.S. Zymaris, I.S. Kavvadias, D.I. Papadimitriou, , 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), pp. 370-389, 2015.

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

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

[20] E.M. Papoutsis–Kiachagias, K.C. Giannakoglou, *A parameterization and mesh movement strategy based on volumetric B-splines. Applications to shape optimization.* NTUA/PCOpt/2015/01 REPORT, Athens, Greece, 2015.

[21] J. R. Bull, *Turbulence Models with Adaptive Meshing for Industrial CFD.* PhD Thesis, Imperial College London, 2013.

[22] L. Piegl, W. Tiller, *The NURBS book.* Springer, 1997.

[23] B.E. Launder, D.B. Spalding, *The numerical computation of turbulent flows.* Computer Methods in Applied Mechanics and Engineering 3 (2): 269-289, 1974.

[24] S.G. Nash, J. Nocedal, *A Numerical Study of the Limited Memory BFGS Method and the truncated–Newton Method for Large Scale Optimization.* SIAM Journal on Optimization, 1(3):358–372, 1991.

[25] J. H. Ferziger, M. Peric, *Computational Methods for Fluid Dynamics.* 3rd Edition. Springer, 2001.

[26] Giles, MB., MC. Duta, JD. Muller, and NA. Pierce, *Algorithm developments for discrete adjoint methods.* AIAA Journal, 41(2), 2003.

[27] I. Goldasteh, S. Chang, S. Maaita, G. Mathur, *Numerical Simulation of Airflow Distribution on the Automobile Windshield in Defrost Mode.* SAE Technical Paper 2015-01-0330, doi:10.4271/2015-01-0330, 2015.

[28] H. K. Versteeg, W. Malalasekera, *An introduction to computational fluid dynamics. The finite volume method.* 1995.

[29] J. Nocedal, S. J. Wright, *Numerical Optimization.*, Springer, 1999.

[30] N. Marco, S. Lanteri, *A two–level parallelization strategy for Genetic Algorithms applied to optimum shape design.* Parallel Computing 26 377-397, 2000.

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

[32] K.C. Giannakoglou, D.I. Papadimitriou, I.C. Kampolis, *Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels.* Comput. Methods Appl. Mech. Engrg. 195 6312–6329, 2006.

[33] B. Mohammadi, O. Pironneau, *Applied Shape Optimization for Fluids.*, Oxford University Press, 2001.

[34] O. Pironneau, *Optimal Shape Design for Elliptic Systems.*, Springer–Verlag, New York, 1984.

[35] A. Jameson, *Aerodynamic design via control theory.*, J. Scientific Computing 3 233–260, 1988.

[36] G. Burgreen, O. Baysal, *Three–dimensional aerodynamic shape optimization using discrete sensitivity analysis.*, AIAA J. 34 (9) 1761–1770, 1996.

[37] J. Elliot, J. Peraire, *Aerodynamic design using unstructured meshes.*, AIAA Paper 19 (41), 1996.

[38] M. Giles, N. Pierce, *Adjoint equations in CFD: duality, boundary conditions and solution behaviour.*, AIAA Paper 18 (50), 1997.

[39] W. Anderson, V. Venkatakrishnan, *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation.*, AIAA Paper 06 (43), 1997.

[40] S.Patankar, D.Spalding, *Calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows.*, Int.J.Heat Mass Transfer 15 1787–1806, 1972.