Politecnico di Milano

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
Corso di Laurea Magistrale in Aeronautical Engineering - Ingegneria Aeronautica

Machine Learning techniques for evaluating nasal airflow: preliminary results

Relatore
Prof. Maurizio Quadrio

Correlatori
Prof. Marcello Restelli
Dott. Giacomo Boracchi

Candidato
Gianluca Romani
Matr.875748

Anno Accademico 2017–2018

Ringraziamenti

Sicuramente un primo grazie è per la mia famiglia. In questi anni mi siete sempre stati accanto nel modo migliore: aiutandomi nei momenti di bisogno senza mai chiedere nulla in cambio. Grazie mamma, grazie papà, grazie bocia, non sarei qui oggi senza di voi, spero di avervi resi orgogliosi. E grazie nonne, per non avermi mai lasciato morire di fame.

A conti fatti, devo ringraziare un'altra famiglia: quella che mi sono costruito in questi anni di avventure. Grazie Ale, Sgamby, Toni, Franci, Luca, Ludo e tutti gli altri: con voi Milano è diventata davvero casa. Grazie alla vecchia guardia, gli amici che mi riaccolgono sempre col sorriso: Bule, Marta, Luca, Alice, Ire, Bozzi, Caste, Scap, Gian. È sempre bello ritrovarsi e vedere che non è cambiato nulla. Un grazie anche a Cla e Fil, su cui ho sempre potuto contare, a Michi del team porceddu e ad Au, insostituibile psicologa e personal shopper. Grazie al fratello che oggi non poteva esserci, ma con cui ho condiviso l'anno di convivenza più ignorante della storia e giusto un altro paio di cosette: grazie Belie.

Ringrazio di cuore il prof. Quadrio, che mi ha guidato con pazienza e saggezza nel travagliato percorso di questa tesi.

Ad uno ad uno sarebbe impossibile, ma ringrazio, infine, tutti gli altri che hanno reso questa esperienza universitaria incredibile: vi porterò sempre con me.

Milano, April 2019 G. R.

Contents

List of Figures

Listings

Sommario

Questa tesi si inserisce all'interno di OpenNOSE, un progetto di ricerca multidisciplinare il cui fine è la creazione di uno strumento, open source e patient-specific, che affianchi i medici nelle diagnosi e negli interventi chirurgici alle cavità nasali per ridurre tempi, costi ed effetti collaterali. L'obiettivo specifico è l'esplorazione di un approccio alla valutazione della corretta funzionalità delle cavità nasali mediante tecniche di ML (Machine Learning). Il punto di partenza è stata la creazione di un modello CAD (Computer-Aided Design) semplificato delle cavità nasali. In secondo luogo alcuni parametri geometrici significativi sono stati definiti, insieme agli otorinolaringoiatri dell'Ospedale San Paolo di Milano, e combinati per generare 200 geometrie nasali diverse, metà delle quali affette da ipertrofia dei turbinati in varie forme ed entità. Le simulazioni RANS (Reynolds-Averaged Navier-Stokes) hanno permesso di ricostruire l'andamento del flusso d'aria all'interno dei modelli durante la fase di inspirazione. Dai dati così ottenuti sono state estratte delle features, caratteristiche della corrente che riassumono i campi di velocità e pressione in pochi valori numerici da usare come input per i modelli predittivi. La bontà delle caratteristiche scelte è stata verificata con un processo di selezione delle features, basato su algoritmi di ML. Questa procedura ha inoltre consentito di raccogliere informazioni utili sul legame tutt'altro che banale tra la corrente fluida e la geometria del modello. Infine un modello finale di regressione ha valutato la reale capacità predittiva delle features scelte.

Parole chiave: cavità nasali, modello CAD, CFD, RANS, ML, dataset, feature, ipertrofia dei turbinati

Abstract

This thesis is part of OpenNOSE, a multidisciplinary research project aiming at the development of an open source, patient-specific tool to support medical personnel in the diagnoses and surgical procedures to the nasal cavity, to reduce costs and collateral damage. The specific task concerns the exploration of a novel approach to the evaluation of nasal airflow quality, based on ML (Machine Learning) techniques. The first step is the creation of a simplified CAD (Computer-Aided Design) model of the nasal cavity. Secondly, some meaningful geometrical parameters are defined, in close collaboration with the otolaryngology experts at San Paolo Hospital of Milan, and then combined to generate 200 different nasal geometries, half of which exhibit some version and degree of turbinate hypertrophy. RANS (Reynolds-Averaged Navier-Stokes) simulations allow the reproduction of the airflow inside the models during the inhalation phase. Some flow features are extracted from the raw CFD (Computational Fluid Dynamics) results, summarising the entire flow fields in a few numerical values to employ as the predictive model input. The quality of the chosen features is then tested in a feature selection process, based on ML techniques. Additionally, this procedure allows the gathering of insights about the non-trivial connection between model geometry and the airflow inside it. Lastly, a final regression model assesses the real predictive value of the chosen features.

Keywords: nasal cavity, CAD model, CFD, RANS, ML, dataset, feature, turbinate hypertrophy

Introduction

Almost a decade after it moved its first steps, the OpenNOSE project has grown into a consolidated academic research reality. Among others, it involves teams from the Dipartimento di Scienze e Tecnologie Aerospaziali (DAER) of Politecnico di Milano, the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB) of Politecnico di Milano and the otolaryngology department of San Paolo Hospital of Milan. Their final purpose remains the creation of a reliable, patient-specific and open source tool to help medical personnel come up with more precise diagnoses and perform more effective surgical procedures. Most of the past years' work, including [8], has successfully focused on the development of ever improving methodologies to accurately reproduce the airflow inside the human nasal cavity through Computational Fluid Dynamics (CFD): through fine tuning and continuous validations, today the replication of nasal airflow is close to an established practice, despite some limitations highlighted in [16].

This work represents the first step towards a practical application of these methodologies, exploiting one of the most recent products of innovation in the computing world: Machine Learning (ML). The main idea is to train a ML algorithm to recognise different pathological conditions of the nasal cavity by looking at some features of the simulated airflow. Naturally, the job requires a large dataset, in the order of 10^3-10^4 cases, which poses a number of complications. Most of all, collecting suitable CT scans[1] is a slow process: it requires strict procedures to ensure the patients' privacy protection and is far from a daily procedure. Moreover, each scan must be matched to a SNOT (Sino-Nasal Outcome Test), a patient filled questionnaire asserting the perceived quality of their respiration. Therefore, in the time frame of this thesis, it would be impossible to gather enough scans. In light of these issues, this work aims at testing the feasibility of the idea and at establishing procedures to jumpstart a subsequent attempt

[1] Computed Tomography scan is a diagnosis technique based on X-ray measurements to investigate the interior of an object without opening it. It is mostly used for medical applications

made with more realistic data.

The dataset was built using an artificial model of the air volume inside the human nose, implemented through CAD (Computer-aided Design). The model developed for this thesis takes into account several previous articles, like [1], although it is largely of new design due to flow quality considerations. In agreement with the open source logic of OpenNOSE, the chosen software was FreeCAD. The anatomical modifications to generate a wide variety of cases were suggested by the otolaryngology team of the San Paolo Hospital and adapted to the CAD model. In particular, some aim at modifying the geometry without introducing any pathological condition, while a second group consists of variants of turbinate hypertrophy. This was selected as the pathological state to be identified by the Machine Learning model, due to its wide diffusion and easy modelling. The final dataset amounts to 200 cases. The CFD task consists of steady-state RANS (Reynolds-Averaged Navier-Stokes) simulations of the inspiration phase, carried out on the clusters of CINECA[2]. As previous experience, such as [8], recommends, the chosen turbulence model is $k - \omega\ SST$. The mesh size is roughly optimised as the best compromise between low computational cost and result accuracy and the computation's setup is adjusted to fit the present needs. Finally, the resulting data is automatically extracted on ParaView, an open source visualisation software. A further reduction is necessary to obtain a suitable input for the ML algorithm and it is carried out through the extraction of a few significant, handcrafted flow features, via Python scripting. These undergo a feature selection process, to assess their predictive value and to reduce the ML input size for better performance. This task involves techniques and algorithms from the Machine Learning field. The top ranked features are finally used to train and test the ML model to evaluate turbinate hypertrophy. Despite avoiding some of the issues presented by real life nasal geometries, this thesis enters uncharted territory and lays a solid groundwork for the next steps of the project, developing robust procedures while always aiming for efficiency and automation.

This study is the result of a close collaboration with fellow candidate Luca Butera and his supervising team at DEIB, who led the way in the manipulation of data and in the building of the ML model and helped with insightful ideas and on-point observations throughout the work. Although they tackle the problem from different angles, both this thesis and [3] share the same conclusive discussion.

[2] A High Performance Computing (HPC) centre, built by a consortium of Italian universities for academic research purposes

Outline

The work is structured as follows.

The first chapter introduces the concepts of Machine Learning. After a general definition and classification of the methods, the algorithms of interest are presented, generally explaining how they work and how they are employed.

The second chapter summarises the bibliographical research standing at the base of this thesis. In particular, some fundamental notions of anatomy and Computational Fluid Dynamics are briefly presented, along with the current state of the art concerning nasal cavity modelling and nasal airflow simulation.

The third chapter deals with the creation of the dataset, from the model design to the CFD simulations. First off, it thoroughly describes the standard CAD model, its main features and the anatomical modifications introduced. Then, the modelling choices for the CFD phase are illustrated.

The fourth chapter describes the process leading to the Machine Learning model. It starts by outlining the data extraction from the RANS results, a stage made necessary by their excessive size. Then, the procedures for feature extraction and selection are depicted, along with the final training of the ML algorithm. Finally, the results are presented and discussed.

The conclusive chapter reviews the main steps of the work, including a critical discussion of the accomplished results and suggestions about possible developments.

Appendix A reports a further investigation, aimed at evaluating the error introduced by the RANS approach. It is carried out by comparing the results of a RANS and a LES simulation on the same geometry.

Appendix B presents the most important listings and codes, implemented to automatise post-processing operations.

Chapter 1

Machine Learning

Despite the impressively quick proliferation it is undergoing in an ever larger number of applications and the huge amount of ongoing research connected to it, Machine Learning remains an abstract concept in the mind of many people. A rather formal definition is the one provided by Mitchell in [13]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". In other words, ML is concerned with finding the connection between some input data and a desired output, without explicit instructions.

Among the most common ML tasks is Supervised Learning, which specifically deals with cases where the output set is specified in advance. In mathematical terms, given a dataset of the form $\{(x_i, y_i)|i \in (0, N)\}$, where $x_i \in X$ is an input vector containing M values, called features, and $y_i \in Y$ an output vector to be predicted, Supervised Learning consists in finding a function $g : X \to Y$ mapping the values in the input space X to the ones in the output space Y. The selection criterion is usually embodied by a scoring function $f : X \times Y \to \mathbb{R}$, which assigns a score to every input/output pair. Therefore the objective becomes finding:

$$g(x) = \arg\min_{y} f(x, y) \tag{1.1}$$

Practically speaking, f is usually modelled as a loss function, measuring the distance between the predicted output $\hat{y} = g(x)$ and the actual one y.

Supervised Learning settings can be of two kinds: classification or regression. The main difference concerns the nature of the output: discrete in classification problems and continuous in regression ones. For example, given a set of road vehicles, distinguishing cars from motorcycles and trucks would require a classification algorithm, whereas the prediction of the cost of each vehicle would call for a regression algorithm. In this scenario, an Unsupervised Learning application could consist in figuring out the different types of vehicles in the set, since the algorithm would not be given any information about the expected nature of the output.

Some vital concepts to understand the structure of a ML model and correctly evaluate its performances are bias and variance, i.e. the main sources of error in the predicting process. The former is caused by incorrect assumptions in the algorithm and possibly leads to missing relevant relations between inputs and outputs, in a phenomenon known as underfitting. The latter is due to sensitivity to fluctuations in the input set and can result in the algorithm modelling these random fluctuations rather than the actual input features. This behaviour is labelled as overfitting. Often the design choices in a ML model are driven by a bias-variance trade-off, since generally a reduction in the bias implies an increase in the variance and vice versa.

The problem posed in this thesis clearly falls under the Supervised Learning category and represents a regression problem. The input vector can be generally identified as the output of the CFD simulations. The desired output consists in the degree of severity of turbinate hypertrophy in every nasal model, which is known in advance since the models are artificially generated via CAD.

A fundamental step in the data processing is related to the feature selection process. As suggested by the name, it consists in the reduction of the initial dataset by means of a selection of the features, according to some measurement of their significance. In the general frame of Machine Learning methods, feature selection is an essential step when dealing with small samples, like the one generated for this work. In fact, given a number of instances N, where each includes a number of features M, problems for which $N \not\gg M$ are subject to several issues because the data does not fill densely enough the input space X. A further practical advantage of feature selection is the reduction in computational costs, due to the smaller size of the input. With respect to other dimensionality reduction methods, feature selection does not involve an alteration in the nature of the features, thus maintaining their interpretability. This is a key aspect, especially in the medical field, where features usually carry a physical meaning. The value of a robust feature selection process is further discussed by Saeys in [17].

There are essentially three different strategies to approach a feature selection task: the filter methods, the wrapper methods and the embedded

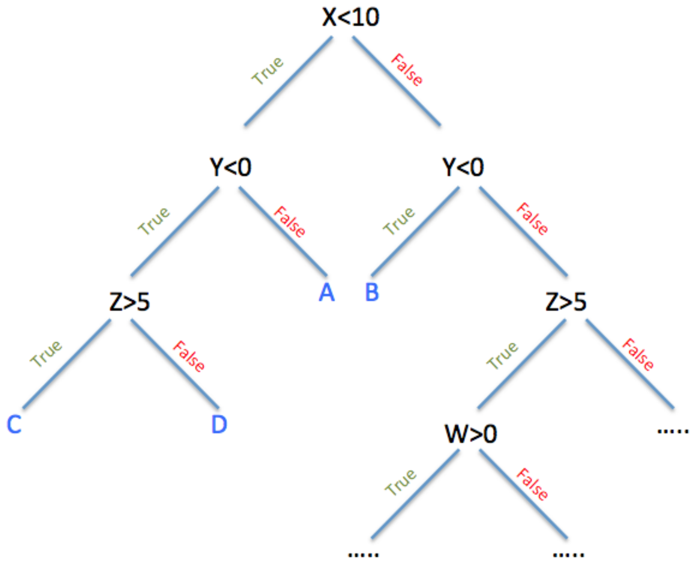


Figure 1.1: Decision tree example. Picture taken from [10]

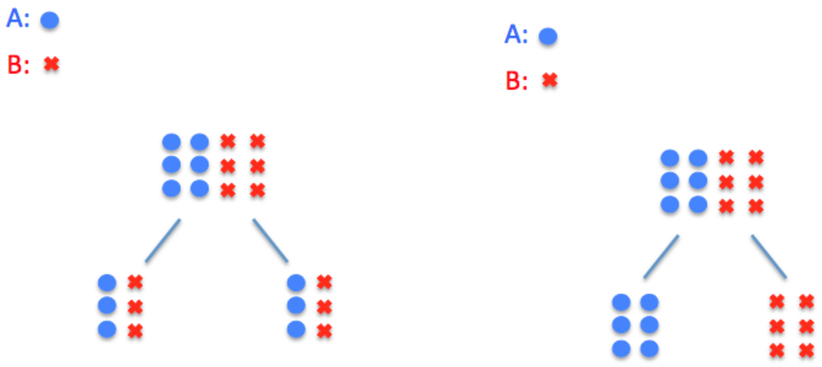
methods. The filter methods are not related to any ML technique and are simply based on ranking the features according to their correlation to the output values. Unlike them, wrapper methods allow to take into account the possible correlation between the features, repeatedly training a ML model with different subsets of features and operating a selection, based on which subset leads to the most accurate prediction. Finally, embedded methods exploit the implicit feature evaluation mechanism that some algorithm possess, as highlighted in the following section.

1.1 Algorithm overview

In the ever growing universe of Machine Learning algorithms, only a few are suitable for the task at hand.

Decision Trees

Decision Trees algorithms are based on the generation of a branched structure, made up of nodes. The main factors determining the shape of the tree are the features, i.e. the values included in each input vector x_i of the sample. At every node, the sample is split, according to the value of a certain feature, in a number of child nodes, each containing a subset of the



(a) Low IG feature **(b)** High IG feature

Figure 1.2: Sample splitting at a decision tree node, according to different features. Pictures taken from [10]

parent node set. The splitting occurs again at every child node, this time according to a different feature, and so on until all features are used or, ideally, every last-generation node only contains instances with the same target output. Those last-generation nodes become the leaves of the tree. Figure 1.1 clarifies the building process of a decision tree: at each node the sample is split according to the value of features W,X, Y, Z. Each branch terminates when all the instances in its subsample are associated to a single output A, B, C, D.

The criterion determining which feature drives the split at every node usually consists in the maximisation of the IG (Information Gain), which measures the correlation of each feature to the target outputs. Figure 1.2 shows two possible sample splits: split (a) is made according to a feature with null IG, resulting in subsamples with the same variance as the original sample. Split (b) considers a feature with maximum IG: the resulting subsamples are associated to a single output value A or B. This way, the most meaningful features are used first, thus offering an ideal strategy for a feature selection task.

One of the main issues presented by the Decision Trees algorithm is the instability in its predictions, especially when dealing with features that are correlated to one another. A widespread solution involves ensemble models. They consist in building many smaller decisional trees, employing different random subsets of features. Then the correct input/output correspondence is determined by mathematical average among them. The ensemble model of choice for this work is Extremely Randomised Trees (also known as Extra-Trees). For further information on it, the reader should refer to [5].

Lasso

Lasso is based on the classical linear regression problem:

$$y = X\beta + \varepsilon \tag{1.2}$$

where y is the output vector with N elements, X is the matrix where the N rows are the input vectors and the M columns are the features, β is the unknown vector of parameters determining the weight of each feature and ε represents the noise. For a common ML problem, the number of instances N is much larger than the number of features M, therefore equation 1.2 represents an over-constrained problem, whose solution can be easily found by means of a least square error approach. As suggested by Tibshirani in [21], Lasso consists in introducing a penalisation term, based on the L_1-norm of the parameter vector β. The solution becomes:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{j} \beta_j x_{ij} \right)^2 + \lambda \sum_{j} |\beta_j| \right\} \tag{1.3}$$

The result of the Lasso regularisation is introducing a non-linearity in the process, which directly tends to assign null value to the weights of the less meaningful features, highlighting instead the most significant. A consequence of this behaviour is that Lasso-based methods are easily employable for feature selection.

Variants include Group Lasso and Sparse Group Lasso, which essentially constrain the standard Lasso to employ user-defined groups of features. For example, in a Group Lasso solution $\hat{\beta}$, all the weights connected to a certain group of features are either null or different from zero.

Gaussian processes

As opposed to the methods presented so far, Gaussian processes can not be used in a feature selection process. Instead, they are a valid alternative for the final Supervised Learning task. Among their favourable characteristics are the high degree of interpretability of the results and the fact that, in addition to the predicted value, they output its variance, thus providing some measure of uncertainty of the prediction. The disadvantages include an applicability issue: the covariance matrix grows according to the dataset size and, since the method implementation requires its inversion, a large dataset would entail unsustainable computational costs.

Gaussian processes basically consist in the generalisation of the Gaussian distribution to the infinite dimension space of functions. Hence, the learning process becomes a fitting problem of the sample data on a mean function and a covariance function, which fully describe the Gaussian process. A more thorough description of Gaussian processes and their application to Machine Learning tasks can be found in [11].

Chapter 2

State of the art

2.1 3D modelling of nasal cavity

The fact that only in recent years technological advancement made the CFD simulation of human nasal airflow a concrete possibility says a lot about the complexity of the geometry considered. A key factor is the extreme variability it presents, making a patient-specific approach almost a necessity and frustrating any attempt to use common sense to make up generalised insights on how geometrical features are linked to efficient nasal breathing. In fact, this is the very mission of this project: exploiting the Machine Learning techniques to offer ENT[1] specialists a valid aid to their experience in the clinical choices they make.

Despite the great anatomical variety, there are some features of the nasal cavity which are common to the vast majority of the human species. They are briefly introduced to contextualise some terms and concepts, which are necessary to describe the modelling efforts later on. For further details, the reader should refer to [19].

The anatomic portion of interest is located beneath the skull base and above the oral cavity. For simplicity purposes, it is assumed to be symmetrical with respect to the median plane of the human body, even though this is never the case in the real world. The nasal cavity begins with the nares, two symmetrical openings leading into the nasal vestibules and ends in the laryngopharynx, the portion of the throat where nasal and oral cavity reconnect before the trachea. The nasal septum, a thin structure lying approximately on the median plane, separates the first

[1] Ears, Nose and Throat. It is a branch of the medical field also known as otolaryngology

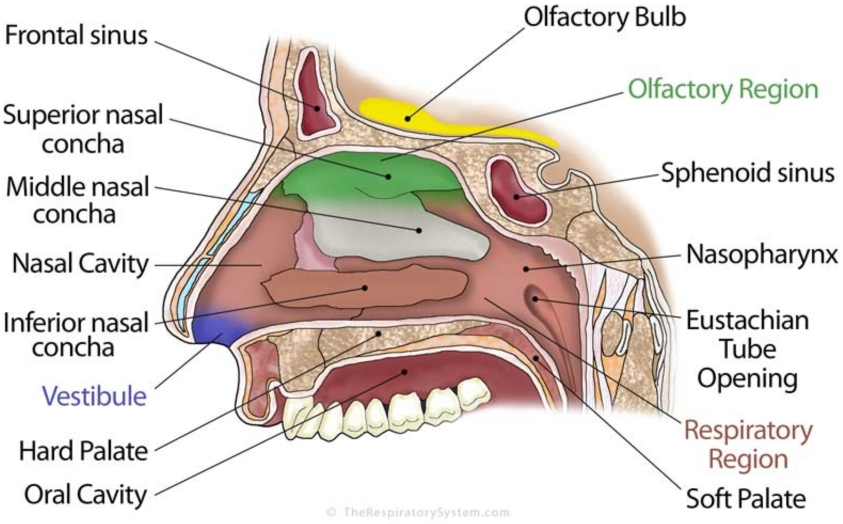


Figure 2.1: Human nasal cavity scheme (1). Picture taken from [20]

section of the airway in two symmetrical volumes, the nasal fossae, before they are rejoined in the nasopharynx.

Each fossa undergoes a relevant section change in the nasal vestibules, developing as a narrow channel enclosed between the septum and the nasal lateral wall. This section includes the nasal valve, a point of constriction along the passageway due to the overlap of the septum and the lateral walls. It is responsible for the recirculatory character of the airflow in the top part of the nasal cavity, which enhances olfactory functions. In fact, this is the area where olfactory sensors are located.

Three long and curled bony structures stand out in the following section of each nasal fossa: the turbinates or conchae, unfolding roughly parallel to the flow and attached to the lateral walls. The inferior turbinate is the largest, running almost the entire way from the vestibules to the nasophariynx. The shorter middle turbinate is located above it and further downstream and the same goes for the superior turbinate with respect to the middle one. The nasal airway develops around the turbinates in thin ducts known as meatuses, with each meatus lying beneath the corresponding turbinate. This way, the mucosal surface is extended and functions of humidification and heating of the incoming air are carried out more efficiently by the hypervascular mucosa covering the turbinates.

As anticipated, the two nasal fossae reconnect in the nasopharynx, following the posterior ends of the septum and turbinates. Subsequently, the airway makes a corner and enters the section called laryngopharynx,

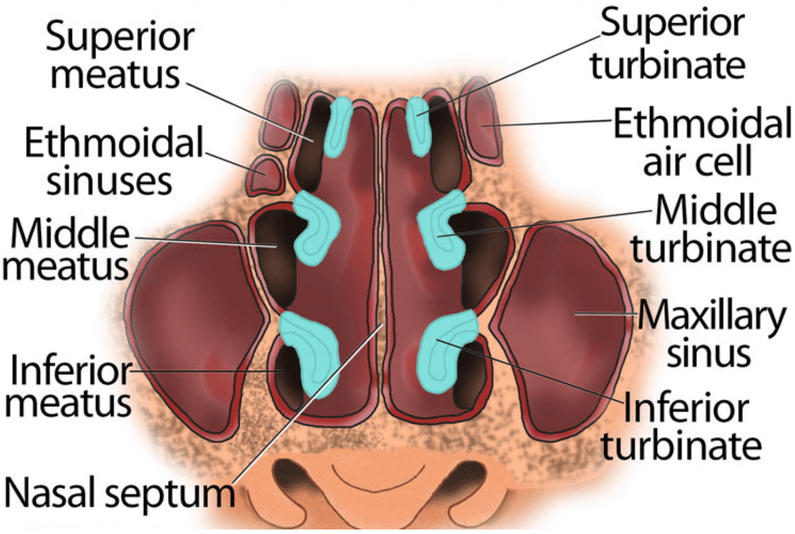


Figure 2.2: Human nasal cavity scheme (2). Picture taken from [20]

which later turns into the trachea and finally leads into the lungs.

The initial part of this work deals with the creation of a database of CAD modelled geometries, replicating the anatomical features described above. Some of these models are modified to simulate a condition called turbinate hypertrophy. As the name suggests, it consists in the swelling of the turbinates, which leads to a constriction of the meatuses, up to the point where the airway can be completely obstructed in the most severe cases. It may have a variety of different causes, from allergic rhinitis to sinus inflammation, or it can present itself paired with other pathological deformations of the nasal cavity, like septum deviation[2]. Turbinate hypertrophy can affect one or more turbinates in either of the nasal fossae and it may involve the full length of the bone or just a portion of it. The treatment of choice is often surgical turbinate reduction.

As of today, the design of a synthetic model of the human nasal cavity is not entirely new: in the past ten years, many attempts were made in that direction. Generally, their objective is the same that drives the modelling effort in this work: to obtain a general, simplified geometry while reproducing, as accurately as possible, the features of a real nasal airflow.

[2] Physical condition involving the displacement of the nasal septum. It is very common, but does not always implicate breathing difficulties

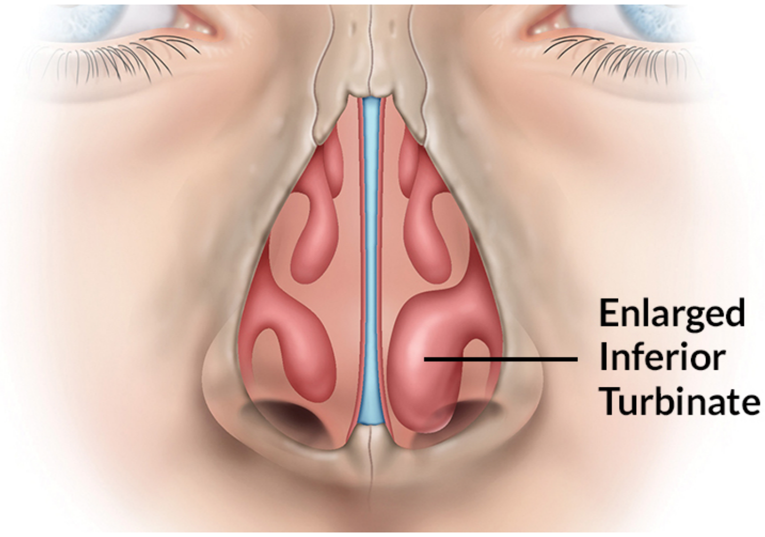


Figure 2.3: Turbinate hypertrophy. Picture taken from [18]

According to a chronological criterion, the first work worth mentioning is the one by Liu et al. [9] from January 2009. This article actually deals with the creation of a standardised nasal model through an averaging process of real nasal geometries obtained from CT-scans. The procedure consists of a number of different steps, including the alignment and scaling of coarse 3D geometries, the extraction of 2D coronal sections from each, the creation of median sections through image processing and the final reconstruction of a median 3D model, in accordance with individual measurements like nasal volume and area. The most interesting part is the one concerning the initial spatial adjustment of the coarse models: in order to compensate for size difference between individuals, two landmarks are defined in each one. One is the tip of the AMS (anterior maxillary spine), which is set as the origin of the reference system, while the other is the choana, the point where the two nasal fossae merge into the nasopharynx. This point lies on the *y-axis* and the last rotational degree of freedom is set by taking the plane minimising the projected area as the *xy-plane*. Finally, the coarse model dimensions are normalised to the average distance between said landmarks. According to their dataset, the average distance is $65.1mm$. The resulting model is known in literature as the Carleton-Civic standardised nasal model.

The work by Javaheri et al. [7] focuses on the creation of an idealised infant nasal geometry for experimental studies on particle deposition. The model is artificially built with CAD and adopts a number of simplifications, justifying each of them. Perhaps, the most relevant one is the approximation of the nasal septum as a flat plate, following the consideration that its typical bumps and deviations are on a dimension scale too small to significantly affect streamline curvature. Then, the effect of sinuses on the airflow is deemed negligible, making their presence in the model redundant. On the other hand, the article highlights the impact of the nasal valve on the flow characteristics downstream from it. It should be noted that these simplifications are considered with respect to their effect on particle deposition, although, in part, they can be extended to a general simulation of the nasal airflow.

In the work by Zhang [23], the modelling of the nasal geometry assumes the same role as in this thesis: reducing computational costs through the use of a simpler geometry. He builds a 3D model for each of the real geometries in his dataset by extracting characteristic dimensions and characteristic points' positions from every instance and applying them to a template model. He then compares CFD results on both the real geometries and the synthetic models, in order to gain insights on the main airflow patterns. Some of the flow features he identifies are consistent with other studies, such as the velocity distribution on the coronal sections, but the terminal portions of the meatuses appear too empty and might even hide signs of backflow due to the abundant presence of sharp edges in the geometry adopted.

The articles mainly taken as reference for the design of the model in this work are the ones by Afiza et al. (2015) [1] [2]. Despite approaching the issue from different angles, experimental and numerical, they both strive towards the creation of a synthetic CAD model of the human nasal geometry and they both achieve their goal by experimenting different solutions and studying how each feature they focus on affects the simulated airflow. In both publications, the authors analyse the effect of the length of the turbinates and the presence of the nasal valve. Their results look consistent between experimental and numerical part and even their simplified model shows several flow features corresponding to those found in the real geometries. Their artificial model is shown in figure 2.4 and constitutes the baseline for the model used in this work, although several modifications are necessary to produce a flow acceptably similar to the real-life inhalation airflow.

A more recent attempt at obtaining an average upper airway geometry was made in 2016 by Nejati et al. [14]. The process is essentially the same

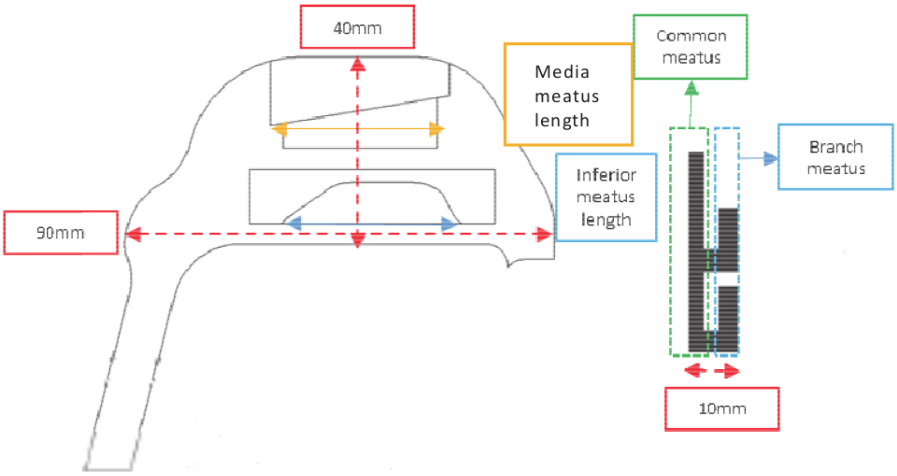


Figure 2.4: Synthetic model of human upper airways. Picture taken from [1]

as in [9], but the methods employed are more sophisticated: 2D coronal sections are "skeletonized" using splines and a template model is deformed to fit them, through control point matching. Finally, the control points undergo the median-finding step, before reconstruction of the 3D median model, based on the resulting sections. The improvements in this approach include an increased robustness to heavier anatomical variations and the possibility to artificially shape the synthetic model to one's needs. For the initial geometry adjustment, the article mostly replicates the procedure described in [9], while using a normalisation length between AMS and choana of $60mm$.

Finally, the article by Xi et al. (2016) [22] eviscerates the effects of nostril orientation on flow properties. The model used is a real nasal geometry reconstructed from MRI (Magnetic Resonance Imaging) and the numerical simulations used are of the LES (Large Eddy Simulation) type. The results suggest that the preservation of the realistic oblique nostril orientation in the modelling process is the best choice. In fact, a downward-directed inlet leads to inaccurate flow distribution in the fossae, while an anterior-directed inlet favours the generation of non-physical vortices in the initial portion of the nasal cavity. Indeed, this turned out to be a critical aspect in the design of the model used for this work.

2.2 Numerical simulation of nasal airflow

The numerical simulation of the airflow in such a complex geometry as the human upper airways is no ordinary deal. The difficulties it presents and the great relevance from the clinical point of view encouraged several studies during the past few years, involving any kind of CFD methods. In the scope of this thesis, RANS simulations are the preferred choice, while LES is marginally employed for validation purposes, as more extensively described in Appendix A.

The incompressible steady-state Reynolds-Averaged Navier-Stokes equations are a set of PDE (Partial Differential Equations) describing the motion of fluids. They are derived from the more general Navier-Stokes equations by applying the Reynolds decomposition.

While the Navier-Stokes equations accurately describe the fluid motion in most cases, a phenomenon known as turbulence often comes into play. It can be defined as the chaotic behaviour displayed by the flow when its Reynolds number is large enough, which is true for almost any relevant real-life situation. Since the fluctuations continuously involve every length scale, from the flow typical one to the microscopic Kolmogorov scales, and since they all need to be computed for accurate results, the DNS (Direct Numerical Simulation) approach is often deemed prohibitive in terms of computational cost. Although it displays wide limitations and it carries along several sources of error, the statistical approach introduced by Osborne Reynolds still is the most widespread way of tackling the turbulence issue. It results in a new set of PDE, the steady RANS:

$$\nabla \cdot \langle \mathbf{u} \rangle = 0 \tag{2.1a}$$

$$\nabla \cdot \langle \mathbf{u} \rangle \langle \mathbf{u} \rangle + \nabla \cdot \langle \mathbf{u}'\mathbf{u}' \rangle + \frac{1}{\rho} \nabla \langle p \rangle = \nu \nabla^2 \langle \mathbf{u} \rangle \tag{2.1b}$$

In the notation used, $\mathbf{u}$ is the velocity vector, p is the pressure and ρ is the density. The obtained set of equations is not a closed one. In fact, there is no mathematical relation tying the mean fields $\langle \mathbf{u} \rangle$ and $\langle p \rangle$ to the term $\langle \mathbf{u}'\mathbf{u}' \rangle$, which is known as Reynolds stresses tensor and includes the effect of the fluctuating motions on the mean flow field. Closure requires the adoption of a model for the Reynolds stresses and the past fifty years of research on the subject produced a wide variety of solutions in that perspective.

In consistency with previous works, such as [8], the model of choice for the present work is $k - \omega\ SST$, a two equation eddy-viscosity model, first described by Menter in [12]. It originates from $k - \omega$ and $k - \varepsilon$ classic models, blending them according to a suitable weighing function, to capitalise on their respective strengths and mitigate their weaknesses. In particular, $k - \omega$ is dominant in the near-wall area, where its formulation is particularly convenient. In free-stream regions, $k - \varepsilon$ is preferred due to its lower sensitivity to free-stream values of turbulent quantities.

Large Eddy Simulation is based on a spatial filtering operation, which results in the following set of equations:

$$\nabla \cdot \overline{\mathbf{u}} = 0 \tag{2.2a}$$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot \overline{\mathbf{u}\mathbf{u}} + \frac{1}{\rho} \nabla \overline{p} = \nu \nabla^2 \overline{\mathbf{u}} \tag{2.2b}$$

Similarly to the Reynolds stress tensor in the RANS approach, the SGS (Sub-Grid Scale) stress tensor τ_{sgs} is defined as:

$$\tau_{sgs} = \overline{\mathbf{u}\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}} \tag{2.3}$$

and accounts for the effect of the sub-grid scale motions on the filtered quantities. The filtered equations 2.2 present the same closure problem as the RANS equations 2.1, requiring the modelling of τ_{sgs}. Among the most common models for sub-grid scale stresses is the Smagorinsky model, which is also the one adopted in the present work. In analogy with many RANS turbulent models, it is based on a fictitious eddy-viscosity.

One of the main features distinguishing a LES from a RANS simulation is the improved accuracy and increased computational cost: the larger turbulent scales are directly calculated, limiting the modelling process to smaller scales. Furthermore, LES is intrinsically unsteady, requiring an adequate time resolution and deferring a possible average operation to the post-processing phase. For a more in-depth overview on RANS and LES approaches, the reader should refer to [15].

The literature review on nasal airflow simulations starts with the article by Zhao et al. [25]. The study investigates the connection between anatomical variations of the olfactory slit (upper-most part of the nasal fossa) and the olfactory function efficiency. The laminar, steady-state CFD simulations were carried out on approx. 1.8 million element meshes. The objective of the study is, per se, of little relevance to this work, but it remains one of the first attempts at applying CFD to the human upper airways and to do so in a consistent manner, to the point that some of the procedures it establishes are still valid today. In a subsequent work, Zhao et al. (2006) [26] examine both human and rat nasal airflow in a sniffing condition. The increase in the flow rate with respect to the previous study [25] suggests that turbulence might play a role, so the laminar simulations are paired with others employing turbulent models $k-\varepsilon$, $k-\omega$ and Spalart-Allmaras. An interesting aspect in both works concerns the boundary conditions: the flow is driven by a pressure difference imposed between the nostrils and the nasopharynx, a similar solution to the one adopted in this work.

Some interesting considerations about the hypothesis of steady-state flow are included in the article by Hörschler et al. (2010) [6]. Their CFD simulations are run on a simplified model and on a real upper airway geometry, obtained from CT-scans, and further corroborated by rhinomanometry[3] measurements. The numerical methods involve a finite-volume method and a Lattice-Boltzmann method and the flow-driving condition is of the pressure difference type. In order to enhance the unsteady effects, the Strouhal number is set to $St = 0.791$, while the Reynolds number lies in the range $Re \leq 2900$. Results show a significant discrepancy between steady and unsteady conditions at low and increasing mass-flux conditions, while the steady-state hypothesis seems to work adequately in high mass-flux conditions i.e. in the middle of the inspiration or expiration act.

Works by Zubair et al. (2012) [27] and by Quadrio et al. (2014) [16] highlight how the airflow is largely influenced by a number of factors, whose effects are very difficult to account for. In fact, the nasal walls are usually considered rigid and smooth and the flow incompressible. Despite these hypotheses, the geometry of a patient's nasal cavity is far from fixed: the nasal cycle implies an alternate process of congestion and decongestion of the nasal fossae, constantly altering the shape of the airway and postural effects should be taken into account as well. Furthermore, the nasal mucosa is rough and covered with cilia and the heat exchange between the blood vessels in the nasal walls and the airflow suggests that a thermoaerodynamic approach should be adopted. Although of great importance, such issues fall outside the scope of this thesis: they constitute a different line of research within OpenNOSE, focusing on the crucial experimental validation of the numerical results obtained so far and described in article [4]. In any case, plenty of empirical evidence collected over the years corroborates the

[3] Diagnosis technique adopted for nasal cavity and consisting in the direct measurement of pressure and flow

validity of the simulation procedures, making these procedures reasonably well established and robust.

The article by Zhao et al. (2014) [24] finally tries to answer a fundamental question for the potential clinical value of any CFD-related upper airway study: what is normal nasal airflow? In order to find an answer, they ran their investigation on 22 healthy patients, mostly young Caucasian individuals, employing a $k-\omega$ model to account for turbulent effects in their steady-state simulations. Despite the relative uniformity of the sample, results showed great variability in the geometries, with volumes and areas roughly varying by up to a factor 2. Some common features were identified, but most, like flow distribution across the meatuses, appear quite different even between healthy subjects such as these.

Chapter 3

Dataset generation

The initial step of the actual work consists in producing a simplified model of the human upper airways, fulfilling the following specifications: firstly, it needs to be simple, to make the CFD computations efficient. Secondly, despite the limited degree of accuracy required, the computed airflow should be qualitatively comparable to the one obtained in a real nasal geometry. Lastly, the necessity to easily introduce some anatomical variations calls for great versatility. The model is designed with FreeCAD, a popular open source parametric design software.

3.1 Base model

The base model in figure 3.1 satisfies all the aforementioned requirements. It represents the air volume, acting as a negative of the bones and tissues bounding the nasal cavity. As far as possible, the sharp edges are filleted[1] to discourage the formation of recirculation regions and the model is perfectly symmetric, at least in this standard form. The reference system has its origin between the nasal vestibules, for design purposes. The *x-axis* acts as the frontal axis, the *y-axis* runs along the sagittal axis and the *z-axis* represents the vertical axis. About $10 - 15mm$ down the *y-axis* from the origin lies the point corresponding to the AMS, taken as reference in [9], [14]. Also, in compliance to these works, the choana was located on the *y-axis*, 75 mm from the origin and about 60-65 mm from the AMS.

[1] Mechanical engineering term which indicates the rounding of an edge and is commonly used in CAD

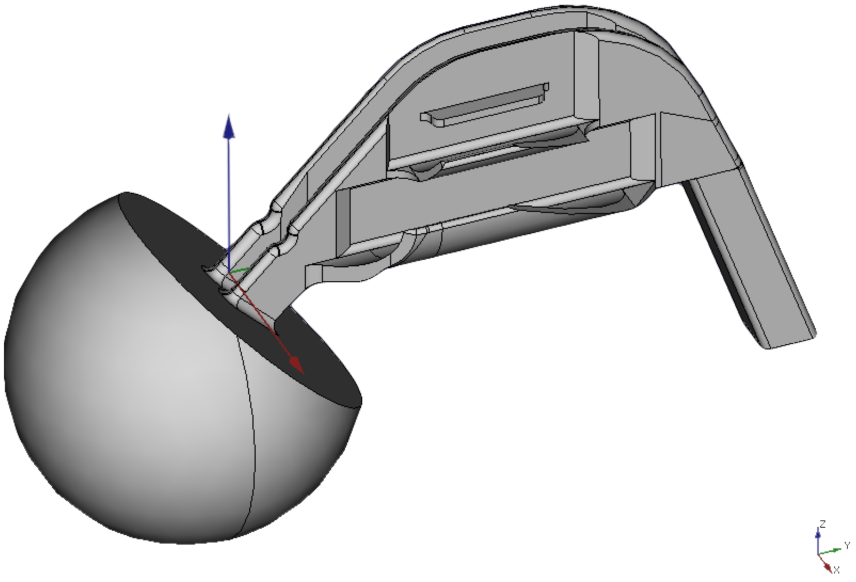


Figure 3.1: Base model

Inlet sphere

The addition to the computational domain of a sphere, centred outside the nostrils and intersecting the external nose surface, achieves accuracy and robustness improvements of the CFD simulations. In fact, imposing boundary conditions away from the nasal vestibules has beneficial effects on the stability of the simulation and allows to already obtain an accurate solution in the beginning portion of the airway. On the other hand, it does not entail an important increase in the computational costs, due to the limited mesh refinement it requires. Only the spherical cap itself acts as an inlet surface, while the cut plane, which the nostrils are connected to, simulates the external nose, where wall boundary conditions are imposed.

Fossa

Figure 3.2 shows the 2D sketch on the *yz-plane* defining the shape of the two identical nasal fossae, from the vestibules (on the left) to the nasopharynx (on the right). As anticipated, the general outline recalls the one described in [1]. Its most interesting aspect is the nostril orientation. Following the findings described in [22], nostril orientation is neither vertical, nor anterior. The value of 45° guarantees a flow distribution qualitatively similar to a real nasal airflow and avoids the presence of backflow in the

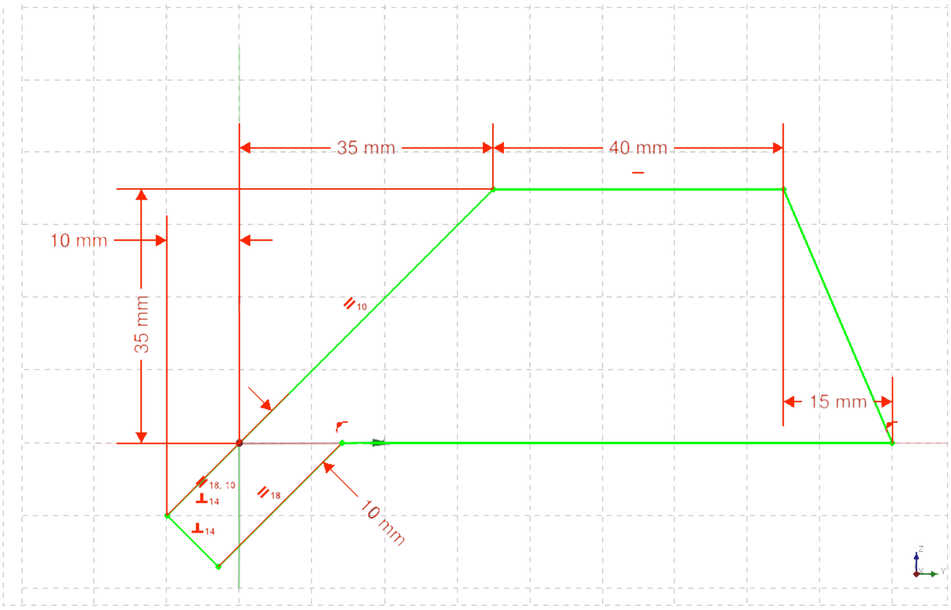


Figure 3.2: 2D sketch of a nasal fossa

terminal portions of the meatuses.

Nasal valve

The nasal valve is designed as a section contraction in the nasal vestibules area, as shown in figure 3.3. The dimensions are defined to optimise the flow downstream: a more pronounced contraction causes a non-physical recirculation zone right behind the valve, while a less evident one implies a high velocity flow on the olfactory region, which is actually a low-velocity area enhancing the olfactory function.

Turbinates

The turbinates are implemented in two steps: firstly, the meatuses are extruded from the sketch in figure 3.4 and suitably smoothed. Then the nasal fossa is carved in their correspondence, to compensate for the increase in CSA (Coronal Section Area). The result can be observed in figure 3.1. The inlet of the inferior meatus deserves a special mention, as it is one of the most critical features for the overall trend of the flow. It is obtained from the union and subtraction of four different blocks, so that the fossa and the meatus are connected as harmonically as possible. The middle meatus inlet and both outlets are less critical and could be obtained via filleting or with the addition of simple volumes. The superior meatus

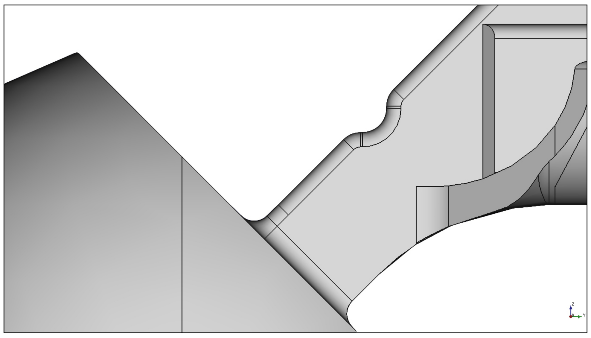


Figure 3.3: CAD model detail: the nasal valve

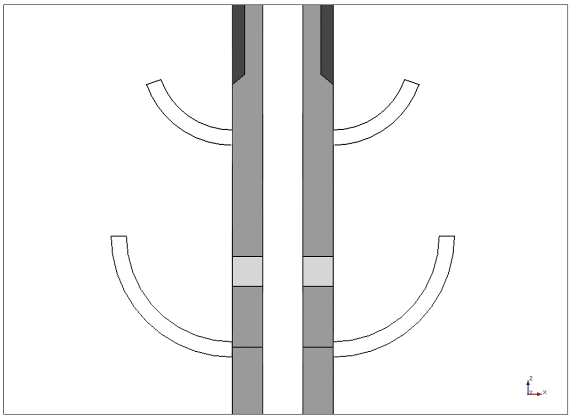
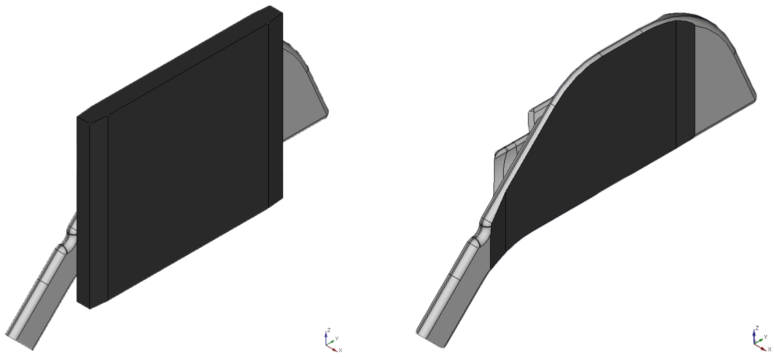


Figure 3.4: 2D sketch of the inferior and middle meatus

is not vital to the general behaviour of the flow, but is mainly introduced to give the model greater versatility: its position can be easily modified without entailing abrupt changes in the airflow.

Central septum-like structure

This peculiar feature mimics the human nasal septum and serves two purposes. One concerns flow quality: reducing the width of the main fossa channel favours a more uniform flow distribution in the meatuses. In addition to this, its position along the *x-axis* becomes the perfect parameter to introduce asymmetry in the model. In figure 3.5 the reader can see the artificial septum besides the main air volume of the fossa and the result of their geometrical subtraction.



(a) **(b)**

Figure 3.5: Nasal septum before (a) and after (b) volume subtraction

Nasopharynx and laryngopharynx

The nasopharynx and laryngopharynx are implemented as separate volumes and fused with the two nasal fossae, as shown in figure 3.6. Apart from the rounded corners, they do not display any notable detail, since the airflow in these regions is not particularly interesting and the geometry adopted does not have a significant impact on the flow upstream. The main reason why they are included in the model is to impose more accurate outlet boundary conditions, replicating the pressure difference generated by the thoracic diaphragm in the lungs.

3.1.1 Airflow comparison on modelled and real geometry

The base model described above is the result of several attempts. The modifications at each step were determined by comparing the CFD-simulated airflow in the model to the one obtained in a real nasal geometry. The final comparison shows the good degree of qualitative accuracy accomplished, while also preserving properties of simplicity and flexibility. The images are obtained on ParaView (5.0.1), an open source visualisation software, which in recent years has become a standard for the post-processing of fluid dynamics simulations.

Looking at figure 3.7, the reader can notice that the real geometry includes the sinuses, which are missing in the model for simplicity purposes, as suggested by [7]. The plotted quantity is the *y-component* of the velocity

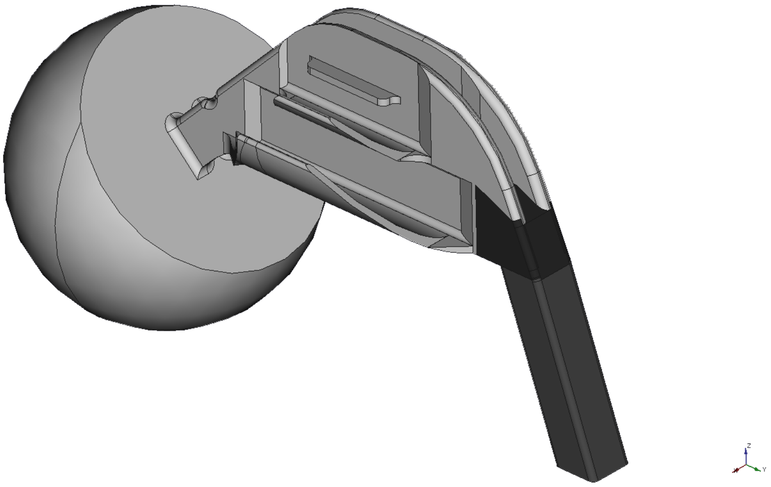
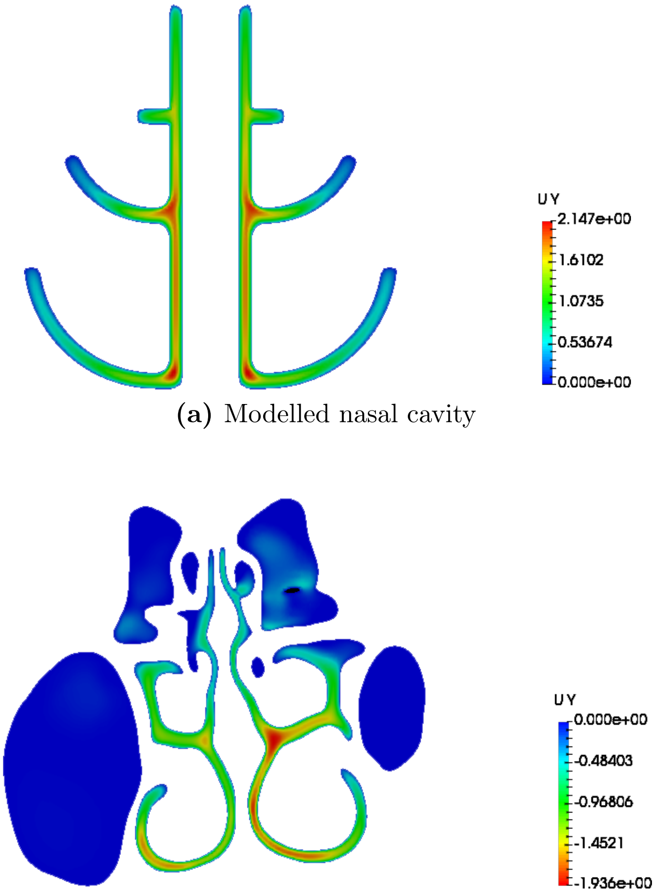


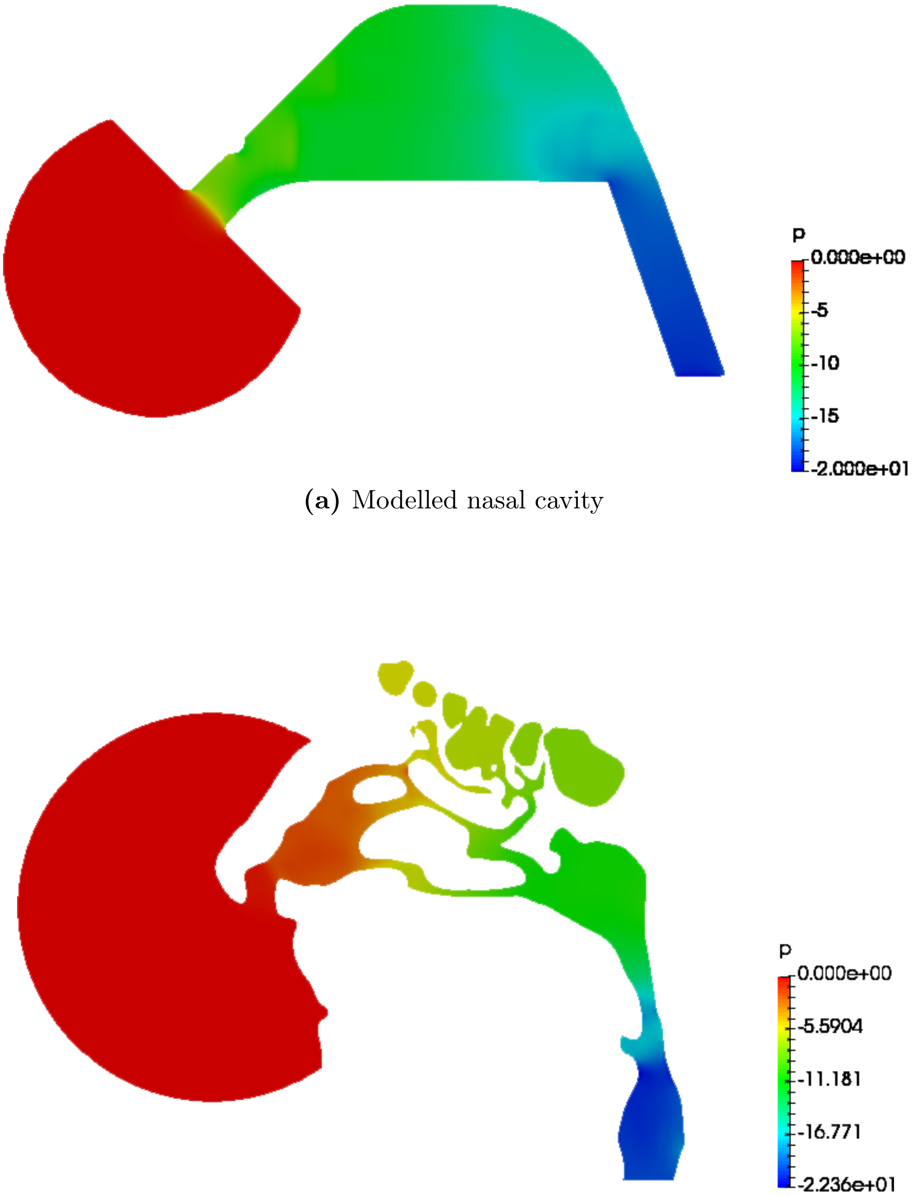
Figure 3.6: CAD model detail: nasopharynx and laryngopharynx



(a) Modelled nasal cavity

(b) Real nasal cavity

Figure 3.7: Coronal section plot of u_y



(a) Modelled nasal cavity

(b) Real nasal cavity

Figure 3.8: Sagittal section plot of p

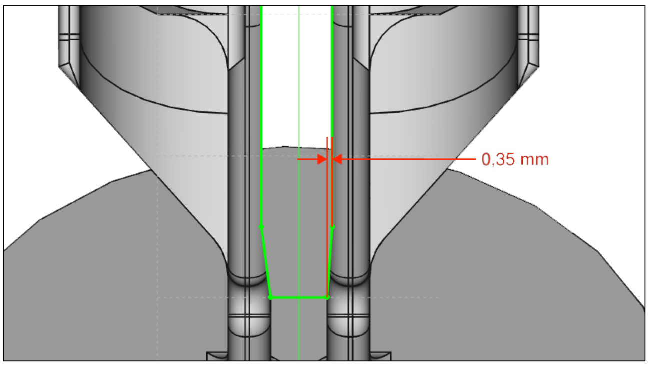


Figure 3.9: Modification 1: septum lateral position

vector, normal to the section plane. The range is slightly different, since the comparison is purely qualitative, but the maximum values in both cases is around $2m/s$. The flow distributions are very similar, with the higher flow-rate regions located in the central part of the fossa, in the middle meatus and in the initial part of the inferior meatus. As for the pressure distribution (figure 3.8), the CFD on the model reports quite different results from the one run on the real nasal geometry. This is, in part, due to the higher size of the modelled nostrils, which causes a high velocity flow in the nasal vestibules. Still, the natural shape proves much more efficient: with similar imposed total pressure-over-density difference ($20m^2/s^2$ in the model, $20.5m^2/s^2$ in the real nasal geometry) the volumetric flow rate in the model is much lower, $10.7l/min$ against $18.3l/min$

3.2 Anatomical modifications

In order to create a wide dataset of diverse geometries and airflows, the following step was applying combinations of anatomical variations to the base model. The variations introduced are eight, five of which behave as harmless anatomical differences between healthy individuals, while the remaining three have a pathological impact on the airflow, replicating the condition known as turbinate hypertrophy described in an earlier chapter. The modifications are defined with the help of the ENT experts of the team.

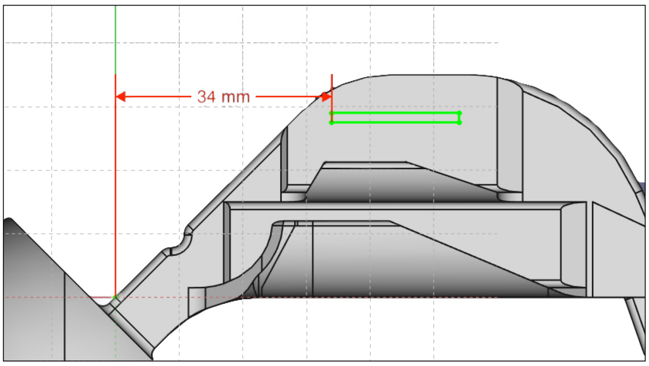


Figure 3.10: Modification 2: longitudinal position of superior meatus

1) Septum lateral position (non-pathological)

The first anatomical variation consists in the translation, along the *x-axis*, of the septum volume described in the previous section. Figure 3.9 shows, from a top view, how the modification is practically implemented by reducing it to a single parameter. In the present dataset, the considered distance varies from $0.3mm$ to $0.7mm$, $0.5mm$ being the neutral position, which produces a symmetrical model.

2) Longitudinal position of superior meatus (non-pahological)

This modification focuses on the superior meatus and consists in translating it along the *y-axis*, without affecting its size, as displayed in figure 3.10. The range adopted for this feature is $30mm$ to $38mm$ from the origin of the reference system.

3) Vertical position of superior meatus (non-pathological)

Similarly, a third variation is represented by the position of the superior meatus on the *z-axis*, ranging between a $20mm$ and a $28mm$ distance from the origin.

4) Section change steepness at inferior meatus inlet (non-pathological)

The fourth non-pathological variation focuses on the section change, which appears on the nasal fossa in correspondence of the beginning of the inferior meatus. In particular, the variation concerns the steepness of such contraction, varying between $1.1mm$ (for volume overlap reasons) and $3.0mm$. It is displayed in figure 3.11.

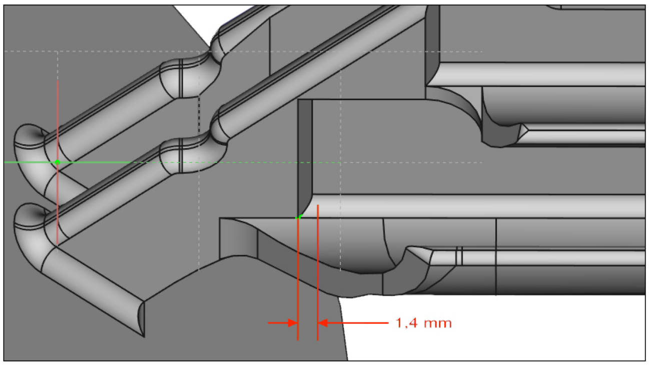


Figure 3.11: Modification 4: section change steepness at inferior meatus inlet

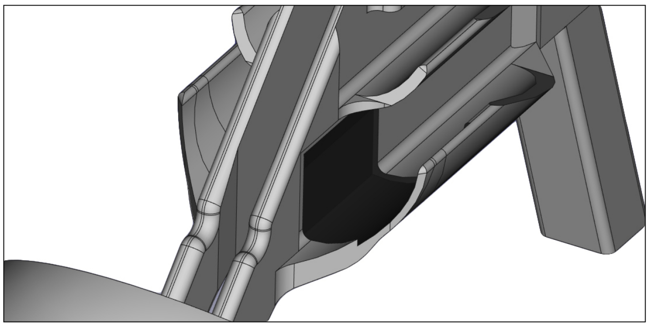


Figure 3.12: Modification 6: head of inferior turbinate hypertrophy

5) Nasal valve position (non-pathological)

As suggested by the title, this alteration involves the location of the nasal valve. The valve is moved along the upper edge of the nasal vestibules, with a vertical distance from the origin of $6mm$ to $8mm$.

6) Head of inferior turbinate hypertrophy (pathological)

As anticipated, the present model acts as a negative of the human nasal cavity: the geometry replicates the air volume of the human upper airways, while the tissues and bones constitute its boundaries. Therefore, the hypertrophy of the inferior turbinate is modelled as a constriction of the airways. The deformation can extend to the entire turbinate or just a portion of it and the sixth modification mimics the hypertrophic condition of the head (anterior portion) of the inferior turbinate. It is implemented as a carve in the inferior meatus and in the corresponding area of the

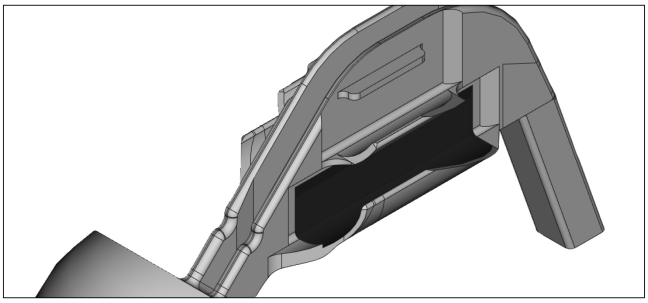


Figure 3.13: Modification 7: overall inferior turbinate hypertrophy

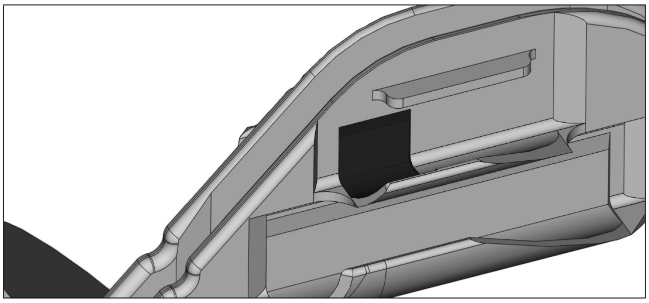


Figure 3.14: Modification 8: head of middle turbinate hypertrophy

vertical channel of the nasal fossa, and has width values ranging from $0.1mm$ to $0.7mm$, which is about half of the original meatus thickness. It is displayed in figure 3.12.

7) Overall inferior turbinate hypertrophy (pathological)

This modification, as shown in figure 3.13, exactly replicates the previous one, although it affects the entire length of the inferior meatus, rather than just its anterior part. Limit values are still $0.1mm$ and $0.7mm$.

8) Head of middle turbinate hypertrophy (pathological)

The last anatomical variation essentially reproduces the sixth one, but it is applied to the middle turbinate instead of the inferior one. It is also reduced in size, proportionately to the lower size of the respective turbinates, varying from $0.1mm$ to $0.4mm$. For greater variability in the dataset, this modification was applied in combination to either of the other two pathological conditions. The result can be seen in figure 3.14.

The eight modifications described are combined to generate 200 geometries, each different from the other. In particular, 100 cases include just the non-pathological variations, while 100 more also include combinations of turbinate hypertrophy. This characteristic composition of the dataset achieves an ideal 1:1 ratio of healthy and pathological cases, to guarantee an optimal training base for the Machine Learning algorithms. The modifications corresponding to turbinate hypertrophy conditions are only applied to the right (positive *x-axis* direction) nasal fossa, since altering the left fossa would mostly produce mirrored images. In case a wider sample was to be obtained, right side hypertrophies could be combined with left side hypertrophies, leading to more geometries, but for the 200 cases of this work it is deemed unnecessary.

3.3 Numerical simulations

The CFD simulations employed are of the RANS type. Because of the huge computational cost they entail, they are executed on the powerful clusters of CINECA, via SSH protocol[2]. The software used is OpenFOAM (v1806+), a widely used open source CFD tool.

The setup of a RANS simulation on OpenFOAM is a delicate operation, especially when dealing with such complex geometries. Apart from the *.stl*[3] files defining the geometry, all the simulations share the same settings. The setup is similar to the one described in a previous thesis on the subject by Lamberti-Manara [8], whose main focus involved the achievement of an accurate numerical simulation of the nasal cavity airflow. Despite this, many adjustments are necessary, mostly due to the different behaviour of the model with respect to the real geometry and to the necessity to make the computations as efficient as possible, while slightly relaxing the accuracy requirements. As mentioned earlier, the turbulent model adopted is $k - \omega\ SST$ and air is considered a Newtonian fluid, with kinematic viscosity $\nu = 1.5 \cdot 10^{-5} m^2/s$.

Mesh generation

The starting point is the creation of an adequate computational grid. With respect to the mesh specifications, the initial block mesh is comprised of

[2] Secure SHell is a standard cryptographic protocol for networks, usually operated via command-line interface

[3] STereo Lithography is a file format used to describe triangulated surfaces. Widely used in CAD applications

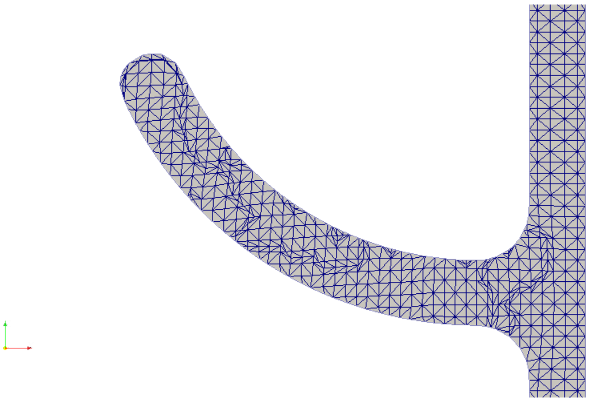


Figure 3.15: Computational mesh detail: middle meatus

cubes of edge length $1mm$. This is then refined and deformed around the model surface by means of the snappyHexMesh[4] utility. Using a higher refinement level for nasal walls and a lower one for the spherical cap boundaries allows to optimise the mesh quality, without excessively increasing its size. A typical final mesh includes 1.0-1.1 million cells.

Boundary conditions

Since these are steady-state RANS simulations and the turbulence model of choice is a variant of $k-\omega$, the independent variables involved are k, ω, p and $\mathbf{u}$, while eddy viscosity ν_t is computed at every iteration from k and ω. The boundary condition driving the inspiration flow into the nasal cavity is a total pressure-over-density difference of $20m^2/s^2$ between the inlet, represented by the spherical cap, and the outlet, which is the lower face of the laryngopharynx, normal to the *z-axis*. The value was chosen to obtain flow conditions similar to a real nasal cavity airflow. The velocity vector $\mathbf{u}$ has a no-slip condition on the solid boundaries and a null normal gradient condition on the inlet and outlet. Boundary values for k and ω are computed according to the criteria defined by Menter in the original article [12] presenting his $k-\omega\ SST$ model.

Numerical settings

The computational grid is subdivided into 8 parts for parallel computing, specifying equal weights since all processors are expected to have equal

[4] OpenFOAM built-in mesh generation tool. It takes as inputs a uniform block mesh and a surface in *.stl* format and produces a new mesh, deformed around the input surface

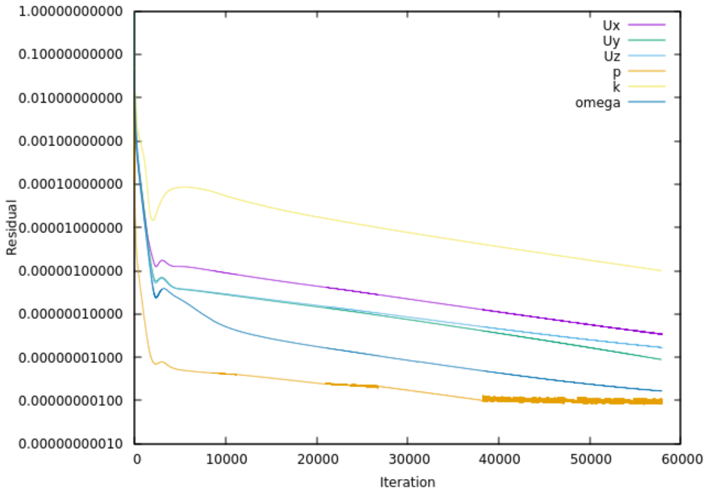


Figure 3.16: Typical residual trend

performances. The numerical schemes adopted are first-order for stability reasons and the convergence criteria are quite strict, since the steady-flow hypothesis is not very sound for this kind of flow, as remarked by [6]. The solver of choice is an OpenFOAM built-in, steady-state algorithm for incompressible flows: simpleFoam.

Convergence results

All the simulations met the convergence criteria in about 20 hours, including the meshing, decomposition and reconstruction processes. This corresponds to a number of iterations between 50000 and 60000. A typical residual plot of the variables is displayed in figure 3.16. The slowest variable to reach convergence is typically k, while the residual of p starts oscillating around a very low value much earlier than the final convergence.

Chapter 4

Learning from data

After the conclusion of the RANS simulations on the CAD generated models, the dataset is almost ready for the ML model training. A preliminary step is necessary and involves a significant reduction in the data size, which in the present form is unsuitable to train an ML algorithm. At first, this is done by switching from huge 3D scalar and vectorial fields to 2D sections and other more manageable data structures. A further reduction is then carried out through the extraction of handcrafted features, which should summarise all the relevant information about the airflow in a few bytes. Lastly, a more objective feature selection process is performed, exploiting some ad hoc techniques from the Machine Learning field, before the final training of the algorithm and the fine tuning of its parameters. In this setting, the word "feature" refers to both the characteristics of the nasal airflow and to the components of the input vectors of the training dataset, as the two concepts end up overlapping. As far as the prediction model is concerned, the data structure requires a Supervised Learning approach. In particular, a regression setting will be adopted: even though the pathological modifications are precisely introduced to 50% of the input geometries, some of them show a moderate degree of turbinate hypertrophy and assigning them a "healthy" or "pathological" label seems too simplistic an approach. Moreover, in a future perspective where the dataset will include more realistic nasal models with an increased level of overall variability, an a priori evaluation of the functionality of a nasal geometry is far from trivial. Therefore, the ability to predict the severity of turbinate hypertrophy is deemed, in prospect, more useful than the one to distinguish a healthy nose from an unhealthy one.

4.1 Data mining

Data reduction represents one of the most important parts of this thesis. On one hand, CFD typically involves huge amounts of data: in fact, for the dataset considered here, a completed simulation output includes the computed airflow fields and the mesh data and weighs on average $210MB$. This is valid for steady-state RANS simulations run on a relatively coarse mesh, since the simplified geometry employed allows to relax the mesh quality constraints and still robustly reach convergence. Of course, in future developments, the need for accuracy will increase: the CFD simulations will become LES or even DNS and the geometrical model used will be more realistic, probably causing the size of CFD results to increase by up to 2-3 orders of magnitude. On the other hand, ML methods generally require very large datasets, comprised of cases with a limited number of features. The dataset used in the present work includes 200 instances, which is close to a lower bound even for a Supervised Learning application. Therefore, the size of every input vector should be at most in the order of some kB, before the feature selection further reduces it to a few numerical values. Clearly, the main obstacle to the application of ML techniques to CFD results resides in the 5 (possibly more) orders of magnitude between what one produces and what the other one needs.

The first step towards solving this issue is achieved through some typical post-processing operations, replacing 3D fields with a more convenient, light and interpretable data structure. The software for this job is ParaView, which allows the user to easily automatise operations through Python scripting. Three kinds of data are extracted from the raw CFD results: 2D sections, streamlines and average pressure values along the stream.

1) 2D sections

Slicing a 3D field to obtain 2D representations of the flow is perhaps the most intuitive way to reduce such a complex information, from an AI (Artificial Intelligence) perspective as much as for a human viewer. In fact, one of the advantages of this choice is the easy interpretation of the obtained images, which is a key factor in the medical field, as stated in [17]. A total of six section planes, shown in figure 4.1, are chosen. They are all roughly at equal distance from each other and are all approximately normal to the main flow direction. Each section file contains the point coordinates of the mesh and the field values for each cell. Since the turbulence variables are not of any interest to the flow general aspect, only $\mathbf{u}$ and p fields are saved, to further optimise data size. The section planes are named a

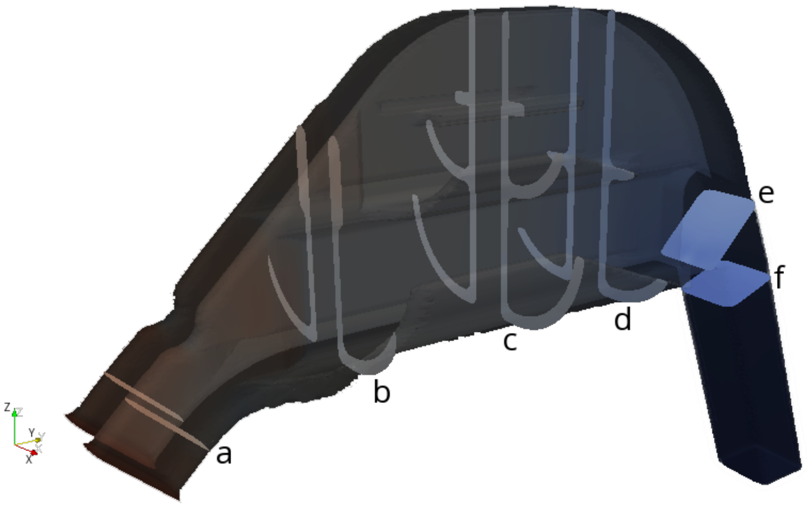


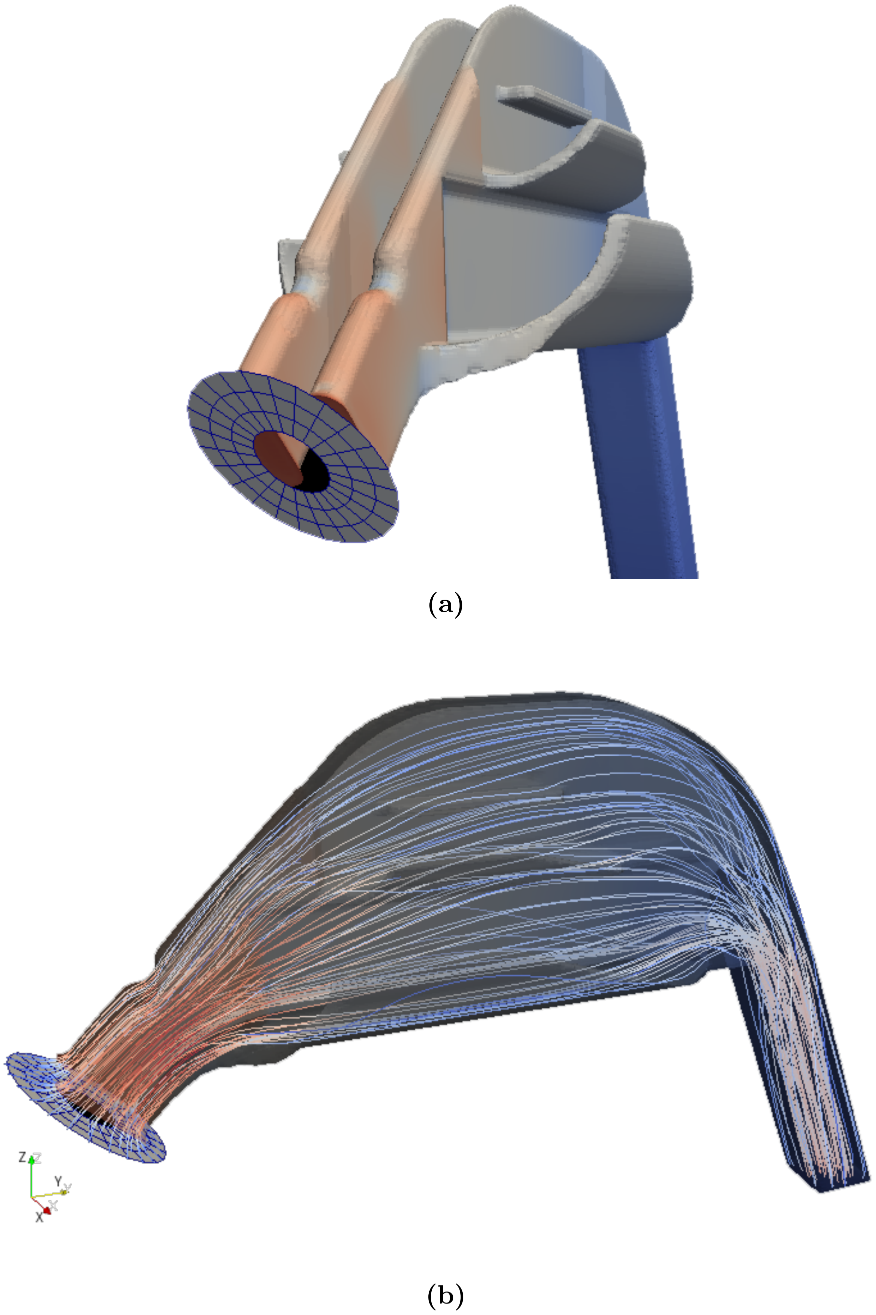
Figure 4.1: Section planes selected for data extraction

through f according to the stream direction: plane a is the one sectioning the nasal vestibules, the three coronal (normal to the *y-axis*) planes in the turbinate region are respectively b, c and d, while the oblique planes across the nasopharynx and laryngopharynx are named e and f.

2) Streamlines

Streamlines are a classical visualisation tool in the CFD world, although their employment as ML input data is far from an established practice. Despite this, they show considerable potential, introducing a time-dependent information in the mix. Naturally, dealing with steady-state RANS simulations, time is just a dummy variable generated by numerical integration, but in the future, with time-dependent simulations, the streamline data will acquire even greater interest.

As displayed in figure 4.2, the streamlines are generated from a roughly uniform grid on a ring placed upstream of the nostrils, for repeatability reasons. The number of streamlines (100), the ring's position and its size are optimised for an efficient description of the airflow. The integration scheme is a Runge-Kutta 4 (RK4) with a fixed step-length of 1.5 cells. The choice of this non-adaptive scheme is due to robustness considerations: because of numerical errors, only about 70% of the streamlines reach the laryngopharynx and RK4 ensures the best success rate. The streamlines data include all field values at each point of every streamline, including the dummy-time produced in the trajectory integration.



(a)

(b)

Figure 4.2: Ring with grid points (a) generating streamlines (b). Streamlines are coloured according to velocity magnitude

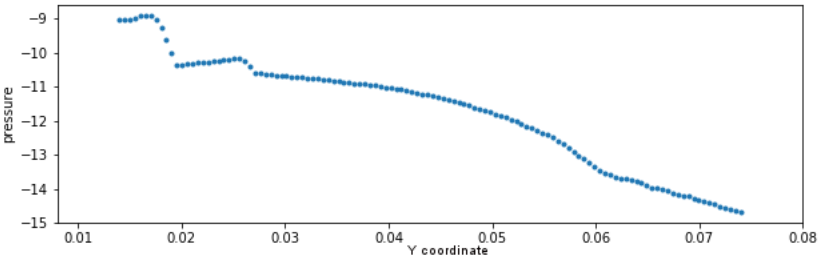


Figure 4.3: Mean pressure plot on *y-axis*

3) Average pressure trend

This idea lies on the consideration that pressure has a typical decreasing trend along the direction of the stream, as imposed by the boundary conditions, while it shows little variability in a generic coronal section. Therefore, each instance is divided in 120 slices by coronal section planes, each thick $0.5mm$. The domain portion covered starts downstream of the nasal valve, to avoid too large an influence on the first few slices, and ends in the choana. In each slice, the integral pressure and volume are computed and saved. The number of slices considered is arbitrarily high: with the integral quantities available, the average pressure can be later computed in any number of slices submultiple of 120. The result is displayed in figure 4.3.

The first step of data mining accomplished a reduction in data size of almost 2 order of magnitudes, leading to post-processed instances approximately as big as $3MB$. In addition, the data is now in a convenient form for human visualisation as well, allowing a better understanding of the general flow behaviour and facilitating a tailored feature extraction to further synthesise the CFD results.

4.2 Handcrafted feature extraction

The issue tackled in this section is the extraction of a limited number of flow features, enclosing the overall aspect of the nasal airflow. In this phase, all the possibly significant features are generated: their actual relevance is evaluated later in the feature selection process. The features should be independent from morphological information and only focus on the flow field: extrapolating geometrical information from geometry-dependent features would result in a trivial problem.

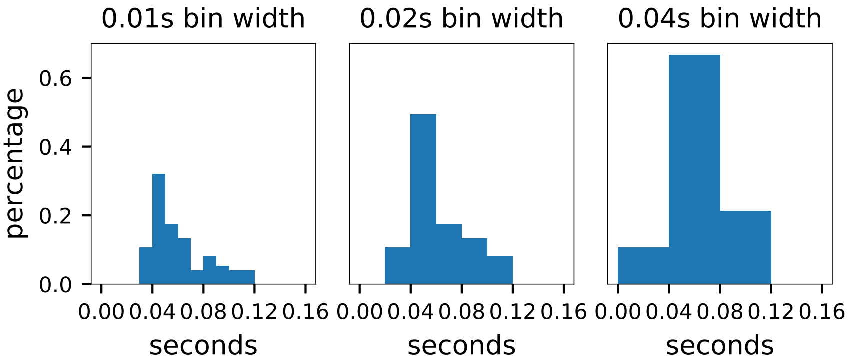


Figure 4.4: Arrival time histograms of streamlines

1) Arrival time histograms

Having plenty of space-dependent information in the 2D sections, some time-dependent data is obtained from the streamlines. In detail, the streamlines are separated from one another according to the dummy time variable, which is reset to zero at the beginning of each line. Then the total time each one takes to get from plane a to plane f is computed and the results are distributed in a number of bins to generate the histograms shown in figure 4.4. This procedure takes place after a selection of the streamlines: the ones not reaching plane f are discarded to avoid contamination of the valid data. The optimal number of bins for the histograms is determined to be four. This allows to preserve some degree of interpretability: the four values generated can be thought of as the ratio of early, mid-early, mid-late and late arriving streamlines. After a suitable normalisation to the total number of streamlines reaching plane f, the time-dependent information carried by the streamlines is thus condensed in a few floating point numbers.

2) Flow rate distribution along *z-axis*

This feature is based on the three coronal sections corresponding to planes b, c and d. Firstly, each one is divided horizontally along the *yz-plane*, in left and right nasal fossa and vertically in a number of bands along the *z-axis*. Then, in every one of the resulting 2D regions, the *y-component* of the velocity is integrated to compute the volume flow rate across that surface. Considering every section plane file includes a matrix U_Y, whose values correspond to the u_y evaluated in each pixel, the flow rate percentage in each region R_k is computed as:

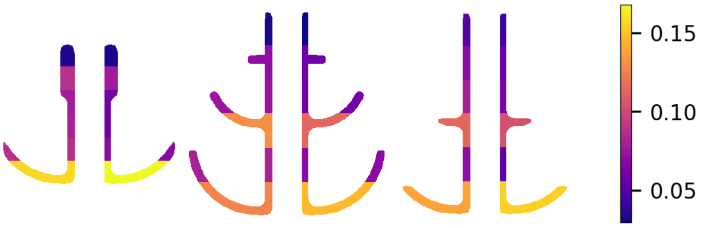


Figure 4.5: Flow distribution across sections b, c, d (left to right)

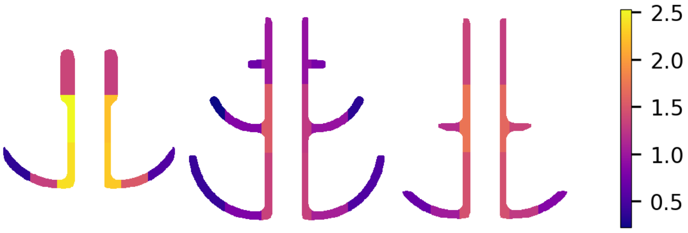


Figure 4.6: Mean values of $|\mathbf{u}|$ over sections b, c, d (left to right)

$$flowrate_k = \frac{\sum_{(i,j)\in R_k} U_Y[i,j]}{\sum_i \sum_j U_Y[i,j]} \tag{4.1}$$

As reported in equation 4.1, the flow rate of every region is normalised to the volume flow rate across the entire section (which naturally is the same for all three planes), yielding the flow distribution along the *z-axis* for each side of every coronal section. The optimal number of horizontal bands is six, as later established by a feature selection process: a bigger number implies too high a correlation among the features, causing the data variance to increase excessively, while a smaller number can not describe adequately the flow distribution. An example of this feature is displayed in figure 4.5, where each region is coloured according to its flow rate share.

3) Average field values in low resolution grids

The idea behind this feature is similar to the previous one. The main differences concern the grid generation strategy, the physical meaning of the features produced and the quantities they involve. First, sections b, c and d are divided vertically in three regions of equal height and horizontally in six regions, three on each nasal fossa, as shown in figure 4.6. The output values

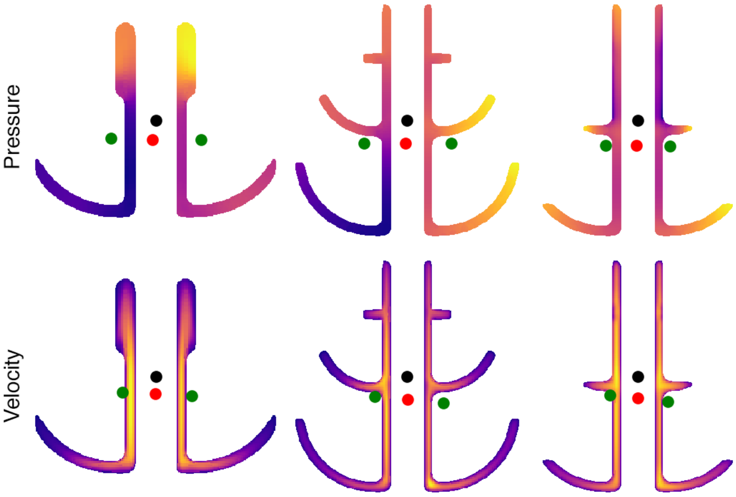


Figure 4.7: Centres of gravity of coronal plots

computed in each region are the mean values of pressure p and velocity magnitude $|\mathbf{u}|$. The number of regions is determined by interpretability reasons: the three vertical bands describe the flow behaviour in the lower, middle and upper portion of the fossae, roughly corresponding to the meatuses, while the three horizontal bands separate the vertical duct along the septum from the closer and farther part of the meatuses.

4) Centres of gravity on coronal sections

This is the most synthetic feature of the lot and is extracted from each of the three coronal sections. A centre of gravity (COG) is located in the section for $|\mathbf{u}|$ and p, computing it as follows:

$$COG_q(S) = \left[\frac{\sum_i \left(i \cdot \sum_j S_q[i,j] \right)}{\sum_i \sum_j S_q[i,j]} \quad , \quad \frac{\sum_j \left(j \cdot \sum_j S_q[i,j] \right)}{\sum_i \sum_j S_q[i,j]} \right] \tag{4.2}$$

where q is the quantity considered and S_q is the matrix containing its values across section S. In figure 4.7 the reader can see the overall COG (in red) and the COGs of each fossa computed separately (in green). Additionally, a black dot represents the fixed centre of the image and serves as reference point.

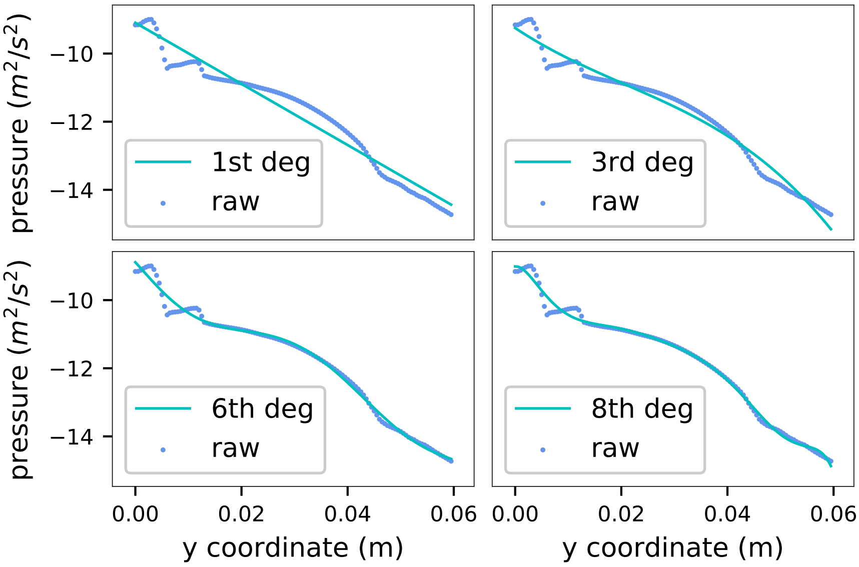


Figure 4.8: Polynomial fitting of various degrees of the pressure trend along *y-axis*

5) Average pressure polynomial coefficients

This feature is based on the average pressure data described in the previous section. Directly using the average pressure values on an arbitrary number of slices would result in highly correlated features, carrying local information. Instead, the solution adopted consists in computing the coefficients of a polynomial, which is fitted on the curve employing a classical least squares method. This way, the data is reduced to a few floating point numbers and each of them is representative of the general trend of the pressure. The degree of the polynomial is empirically chosen to be six, as a compromise between accuracy and feature lightness. Some alternatives are displayed in figure 4.8.

All the 154 values summarising the extracted flow features are concatenated in arrays and become features in the ML sense, i.e. the values forming the input vectors x_i. The 200 resulting input vectors are organised in a matrix, while the values of the three anatomical variations defining the hypertrophic conditions in the turbinates of the initial geometries constitute target output vectors y_i. Even before the final feature selection process, the input dataset has been reduced from the initial weight of $10^8 B$ per instance, to a more manageable size of $10^3 B$ per case.

According to their nature, the features can be organised in the following groups:

- PPT: the coefficients of the polynomial fit of the pressure trend along the *y-axis*;
- FPd: the flow distribution along the *z-axis* on section d;
- FPc: the flow distribution along the *z-axis* on section c;
- FPb: the flow distribution along the *z-axis* on section b;
- APd: the grid averaged values for p on section d;
- AVd: the grid averaged values for $|\mathbf{u}|$ on section d;
- APc: the grid averaged values for p on section c;
- AVc: the grid averaged values for $|\mathbf{u}|$ on section c;
- APb: the grid averaged values for p on section b;
- APb: the grid averaged values for $|\mathbf{u}|$ on section b;
- AI: the arrival-time histograms.

4.3 Feature selection

The objective of the this step is establishing which features are most significant in describing the general flow trend. This process is beneficial in two ways: on one hand, it generates interpretable insights about the effects of turbinate hypertrophy on the nasal airflow and, on the other, it reduces the number of features used in the training process. In fact, as explained in chapter 1, the number of features used for Supervised Learning tasks should be limited, especially when dealing with small datasets.

4.3.1 Feature selection procedures

Practically, three separate regression environments are defined, one for each turbinate hypertrophy version: anterior inferior turbinate hypertrophy, overall inferior turbinate hypertrophy and anterior middle turbinate hypertrophy. This choice is driven by result interpretation reasons. For each target value, a test set including 20% of the cases is set aside, while the remaining 80% is used right away as a training set, in a cross validation (CV) frame. This consists in dividing the training set in L subsets and performing L training tasks on $L-1$ subsets of data each, leaving every time a different one out and using it as a test set. In particular, to avoid generating subsets too small, L is set to 5. This strategy allows to artificially expand the dataset, averaging the results for greater accuracy. Throughout the work, both CV and regression results are evaluated on the base of the R^2-score, defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \tag{4.3}$$

where i is the sample index, y_i is the target value associated to the i-th input vector, $\hat{y}_i$ is the corresponding predicted target value and $\overline{y}$ is the mean over all target outputs. The R^2-score ranges between 0 and 1 and it can be thought of as the amount of variance in the target outputs explained by the regression model. The algorithms used in this phase include Extra-Trees and Lasso and the feature selection process is carried out according to three different strategies.

Recursive Feature Elimination with Cross Validation (RFECV)

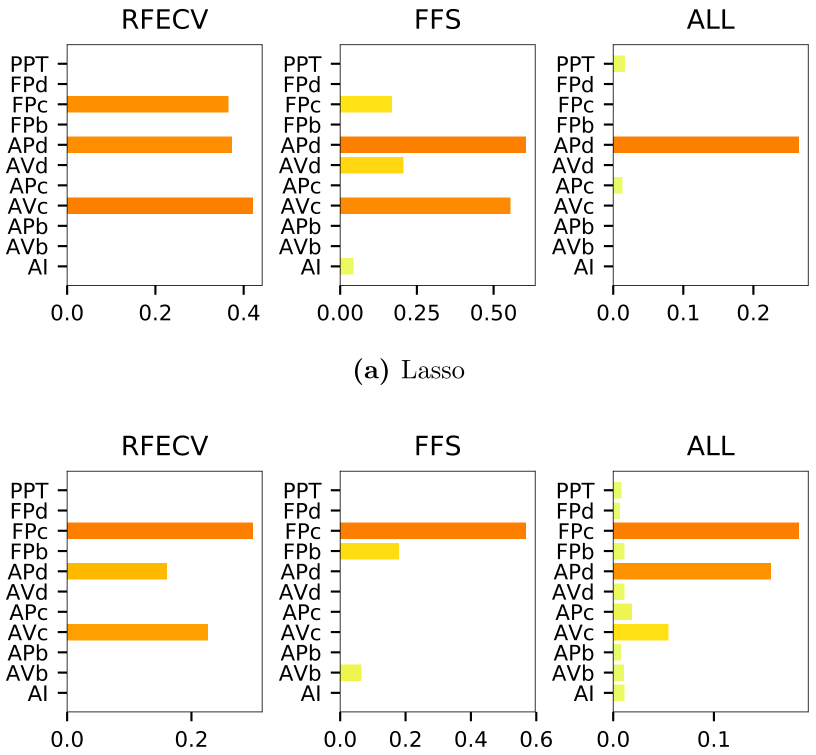
It falls under the embedded methods class and consists in performing recursive training attempts of a chosen algorithm, eliminating at each step from the input set the lowest ranked k features, until a minimum number of features m is reached. In this work, $k = 1$ and $m = 5$.

Forward Feature Selection with Cross Validation (FFSCV)

This one belongs to the wrapper methods class: at every step, the feature guaranteeing the highest score increment is added to an initially empty set. The score is computed through CV by iteratively trying to include in the set every unused feature. The process stops at $M = 100$ features. This way, the score trend can be observed as each feature is added to the set and used to define in retrospect the optimal subset.

Feature selection on all available data

An additional embedded feature selection task is carried out as a validation step, training both Lasso and Extra-Trees algorithms with all the available features and looking at the outcome of their internal feature ranking mechanism.



(a) Lasso

(b) Extra-Trees

Figure 4.9: Bar plot of maximum importance per feature group for each selection strategy. The target output is anterior middle turbinate hypertrophy

4.3.2 Feature selection results

The first relevant outcome of the feature selection process is rather unexpected: one of the groups of extracted features, the COG position of p and $|\mathbf{u}|$, shows a very strong correlation to the target output, i.e. the turbinate hypertrophy severity degree. This makes the problem trivial and leads to the exclusion of this feature group from the input set. The reason is that the COGs are too strongly correlated to the geometry itself, rather than to the airflow trend. Additionally, this spurious correlation is likely to disappear when dealing with more realistic nasal geometries, which show greater geometrical variability than the simplified models employed in this setting.

A further consideration concerns the effectiveness of the two algorithms chosen for the task: due to its intrinsic linearity, the Lasso regularisation proves unable to recognise the correlation among features, thus producing

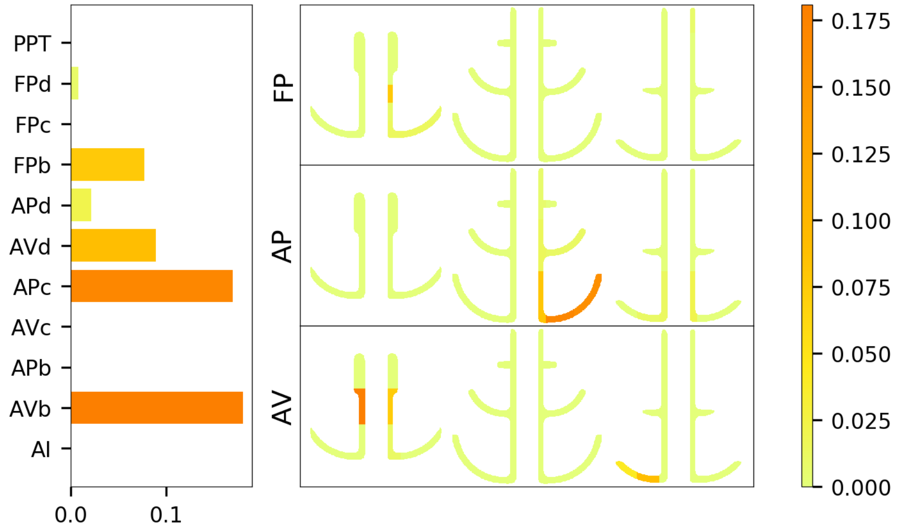


Figure 4.10: Feature importance plots for anterior inferior turbinate hypertrophy, computed with RFECV method on Extra-Trees algorithm

inconsistent results when tested with different feature selection strategies. On the other hand, Extra-Trees shows greater robustness, due to its ensemble nature. Proof of this behaviour is given in figure 4.9: Lasso assigns high importance to different feature types, depending on the method used, while Extra-Trees is more consistent, at least across the RFECV and All-features approaches. As for the feature selection strategy adopted, the Recursive Feature Elimination method is deemed more reliable, as Forward Feature Selection does not show any effect on the R^2-score as the features are added to the input set.

Figures 4.10, 4.11 and 4.12 report the outcome of the feature selection, relative to each version of turbinate hypertrophy. For each, a bar plot shows the maximum importance in each feature group. On the right-hand side of the figures, the nine feature groups generated from coronal sections b, c and d offer insights on which variables and flow regions are most significant for every target output. Both arrival-time histograms and polynomial fit of pressure trend are consistently discarded because of their low predictive value. The results appear anything but trivial: every form of turbinate hypertrophy shows correlation to different features and the areas of interest are located both on the right-hand side, which is the one where pathological deformations are introduced, and on the left-hand side, suggesting that predictive value indeed resides in the simulated airflow, rather than in the geometry.

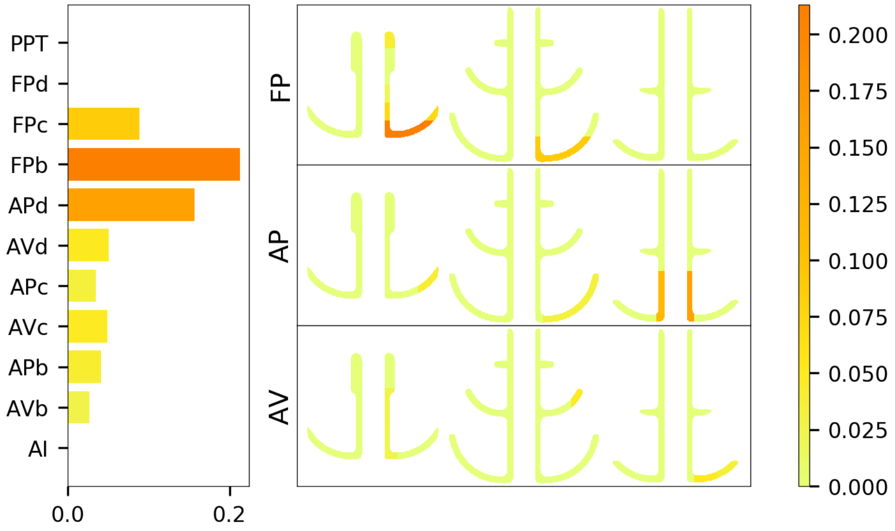


Figure 4.11: Feature importance plots for overall inferior turbinate hypertrophy, computed with RFECV method on Extra-Trees algorithm

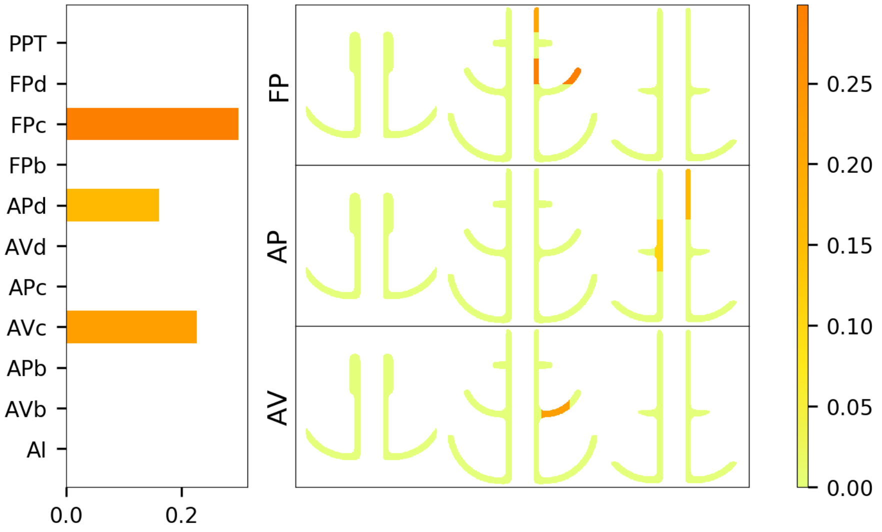
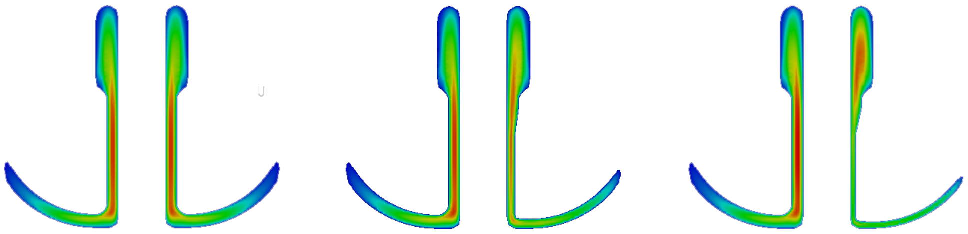


Figure 4.12: Feature importance plots for anterior middle turbinate hypertrophy, computed with RFECV method on Extra-Trees algorithm



(a) Healthy model

(b) Slight anterior inferior turbinate hypertrophy

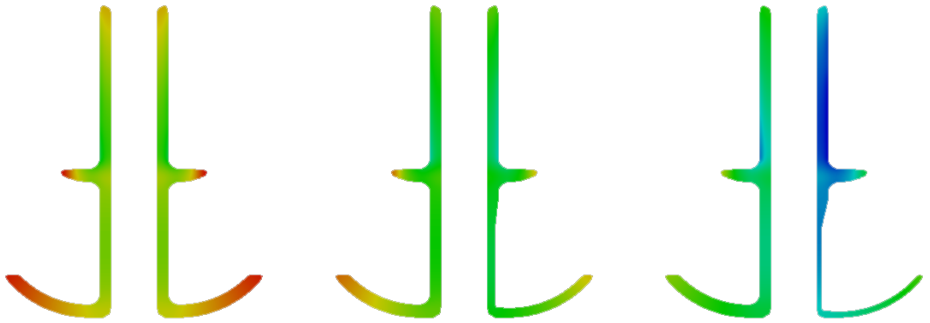
(c) Severe anterior inferior turbinate hypertrophy

Figure 4.13: Velocity magnitude distribution on section b

In detail, a highly relevant feature in the prediction of anterior inferior turbinate hypertrophy is the pressure value in the inferior meatus of section c, which is widely anticipated since it is just downstream of the introduced variation. More surprisingly, the top ranked feature involves the velocity magnitude in section b of both fossae. A simple explanation is that the predictive value is not located separately in the two regions, but in their difference, caused by the pathological shape. This is corroborated by figure 4.13: the flow in the left fossa is similar across all three cases, but its discrepancy with the right fossa increases as turbinate hypertrophy gets more severe.

As far as the overall inferior turbinate hypertrophy is concerned, it is best predicted by the flow rate percentage across the lower part of the inferior meatus on section b and c: this is not surprising, since the geometrical deformation affects that exact region, reducing the passageway width and, thus, the flow rate through it. A more unexpected occurrence among the top features is the average pressure in the lower d section of both fossae. Again, this is probably due to the significance of the difference in the two flow regions. In fact, as displayed in figure 4.14, the difference in pressure distribution between healthy, partially hypertrophic and severely hypertrophic model is remarkable. The lower part of the fossa has probably been chosen over the upper region because it is less heavily impacted by the middle turbinate hypertrophy.

Finally, the anterior middle turbinate hypertrophy turned out to be the most difficult pathological condition to identify: only a few features show decisive predicting value and adding more results in a degradation of the regression score. This small significant group mostly concerns the middle and upper part of the right-hand fossa, where the geometrical modification



(a) Healthy model

(b) Slight overall inferior turbinate hypertrophy

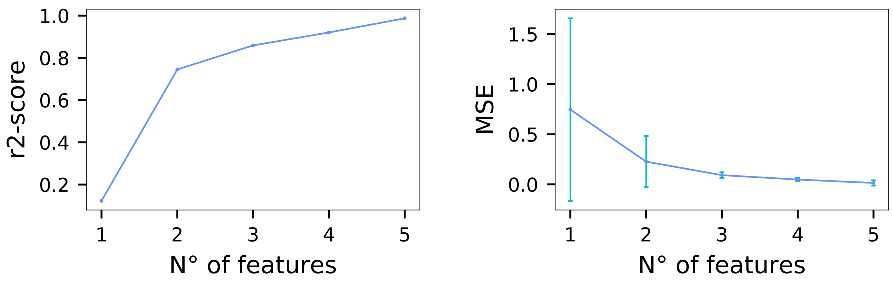
(c) Severe overall inferior turbinate hypertrophy

Figure 4.14: Pressure distribution on section d

occurs. On the other hand, the pressure value in the middle meatus of the left-hand fossa at section d also shows relevance. The additional difficulty in the prediction of this pathological condition is probably due to the fact that it always presents itself paired with either one of the other two hypertrophy variants. This makes it harder for the model to identify how middle turbinate hypertrophy alone affects the airflow.

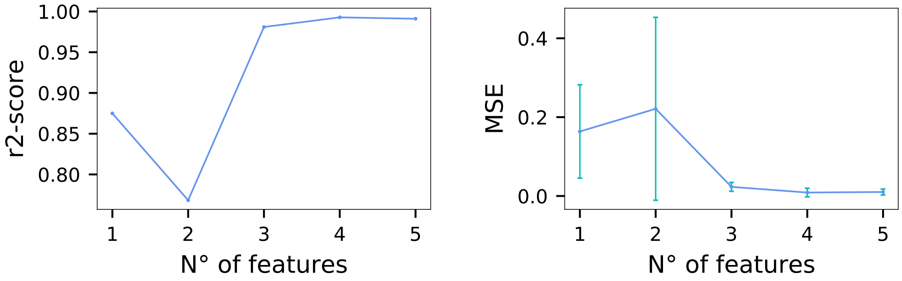
4.4 Predictive model: training and results

The final prediction model is based on Gaussian Processes. This choice is only possible because of the reduced size of the input sample: as anticipated, the covariance matrix inverted in the prediction process grows with the number of instances in the dataset and a larger sample would entail excessive computational costs. Among the advantages of this algorithm is the higher interpretability of the results. Moreover, a mean value and a variance are predicted, thus allowing the evaluation of both the accuracy and the confidence level of the prediction. The regression tasks are carried out separately for each of the three target outputs and several runs are made, in order to carefully tune the algorithm parameters and to investigate how the prediction quality changes by varying the number of features used from 1 to 5. The best features for each regression setting are handpicked, according to the feature selection outcome reported previously. The results are shown in figures 4.15, 4.16 and 4.17, reporting R^2-scores and the related mean square errors (MSE) of the predictions.



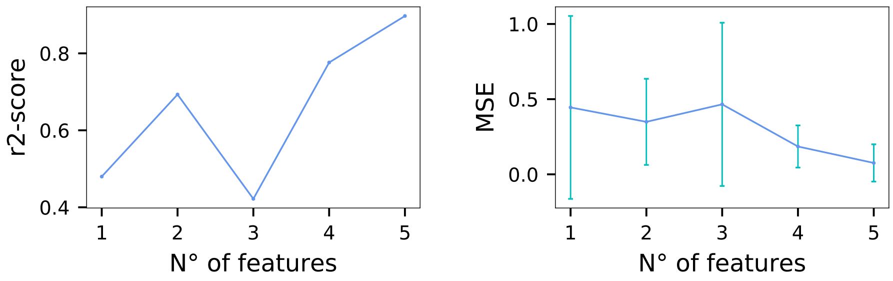
(a) (b)

Figure 4.15: R^2-score and MSE trend for anterior inferior turbinate hypertrophy prediction with number of features varying from 1 to 5



(a) (b)

Figure 4.16: R^2-score and MSE trend for overall inferior turbinate hypertrophy prediction with number of features varying from 1 to 5



(a) (b)

Figure 4.17: R^2-score and MSE trend for anterior middle turbinate hypertrophy prediction with number of features varying from 1 to 5

The outcome looks promising: the R^2-score is close to unity, confirming the strong correlation between the chosen features and the target values. In particular, the hypertrophic conditions of the inferior turbinate show the best results, with a very accurate prediction of the target and very low variance, especially when using all the top five features. On the other hand, middle turbinate hypertrophy proves slightly more difficult to predict: the R^2-score is still very high, but its associated MSE is not negligible as in the other cases. As anticipated, this is probably due to the fact that anterior middle turbinate hypertrophy never occurs as the only hypertrophic condition affecting the nasal geometries, making it tougher for the model to find a clear correlation between flow features and this pathological variation alone. The accomplishment of such high scores, in spite of the fact that pathological modifications are often combined, is quite impressive and constitutes further proof of the robustness and accuracy of the procedures adopted.

Conclusions

This conclusive chapter is shared by this thesis and the one by Luca Butera [3]. This choice highlights the interdisciplinary approach of this work, which involved a collaboration between the CFD team of DAER, the ML team of DEIB and the ENT clinical experts of San Paolo Hospital.

Firstly, the main steps of the work are briefly reviewed: according to the exploratory character of the study, the real nasal geometries are set aside, in favour of a simplified CAD model, with several advantages. From the CFD perspective, a simpler domain facilitates numerical convergence. On the ML side, the variability in the input dataset is reduced because the geometries are obtained from local modifications of a single standard model. This aspect helps the prediction of the pathological condition, since variations in the airflow are imputable to either turbinate hypertrophy or five other harmless geometrical variants. After the input models are generated, a RANS simulations is run on each of them. The choice of RANS is driven by the necessity to keep computational costs to a minimum, to avoid high commitment in a preliminary study such as this. The CFD output undergoes a significant size reduction, mostly through the extraction of hand-tailored features. These are chosen for their high interpretability, since black box approaches are widely distrusted in the medical field. A feature selection task is carried out on the extracted flow characteristics. It represents a key stage: on one hand, it solves the noise issue typical of small samples and, on the other, it allows to gather insights about the significance of the flow features, isolating and ranking the ones with predictive value. Finally, a regression model is trained with the best features, to verify their predictivity and to finally assess whether turbinate hypertrophy in a nasal cavity model can be predicted by looking at the CFD simulated airflow.

Despite the preliminary nature of this study, the outcome is encouraging and the procedures established look effective and robust. The achieved prediction scores are remarkably high, but this does not mean that the problem is trivial. In fact, the results show how even independent pathological variations, when combined, are harder to predict. Moreover, the top ranked features often concern areas not directly influenced by the geometrical modifications, again confirming that the predictive value resides in the airflow, rather than in the geometry itself.

Future perspectives

The exploratory nature of this work puts great emphasis on future development. The results achieved show considerable potential, but the final objective is still quite far: the implementation of a tool that accurately predicts how well a patient's nasal cavity performs and identifies any pathological state affecting it. This will result in more precise and reliable diagnoses and will help validate surgical procedures.

A way to improve the performance of the ML model consists in generating a larger dataset. This would allow to experiment with fully data-driven techniques, exploiting in a deeper way the informative potential hidden in the data. A possible alternative is adopting more advanced Supervised Learning approaches, such as Deep Learning, which need a huge amount of input data, but are able to extrapolate higher level knowledge. A bigger sample size would also allow switching to entirely different paradigms, such as Unsupervised Learning: a large dataset could be clustered to allow a posteriori identification of the important flow features. Moreover, the sample increment will become crucial when dealing with more complex nasal models. In fact, the next suggestions aim at increasing the accuracy of the input data. On the one side, this could be accomplished by developing a more complicated and realistic nasal model, or even by directly working with real geometries, depending on the availability of CT scan data. The result would be a significant step towards real-life application: the predictive models would face the variability and complexity typical of the real human upper airways. On the other side, the nasal airflow simulation could involve a different CFD approach: despite the clear advantages of a RANS approach, the results it produces are inherently wrong. Therefore, switching to LES or DNS would entail higher computational costs, but the reliability of the simulation would increase accordingly.

All things considered, it is proven possible to successfully detect anomalies in a nose-like structure using simulated airflow data. This represents a solid first step towards the creation of a diagnostic tool based on Machine Learning techniques and CFD simulations, and, ultimately, towards understanding what "good" nasal flow is.

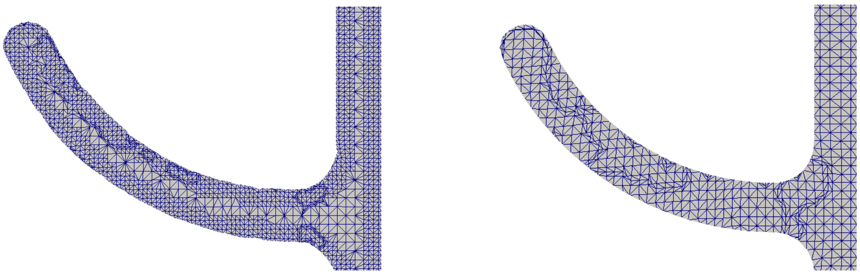
Appendix A

LES and RANS comparison

The necessity to reduce computational costs to a minimum, on one hand, and the limited accuracy required by the use of a simplified geometry with no clinical value, on the other, leads to the selection of RANS as the most suitable CFD technique. Although widely justified by the preliminary nature of this study, the employment of RANS simulations is one of the main sources of inaccuracy in the dataset generation process. In order to quantify this error, a more accurate Large Eddy Simulation is run on the standard CAD model depicted in chapter 3.

The simulation is run on OpenFOAM and the main features of its setup are rapidly reviewed hereby, starting from the computational grid specifications: the mesh is refined around the model surface, starting from a block mesh consisting of cubes of edge length $0.5mm$. The final grid includes 8.6 million elements. The difference with the coarser mesh generated for the RANS can be appreciated in figure A.1.

The model for sub-grid scale stress tensor is the classical Smagorinsky model with Van Driest damping and the solver chosen for the task is



(a) LES

(b) RANS

Figure A.1: Mesh detail comparison: middle meatus

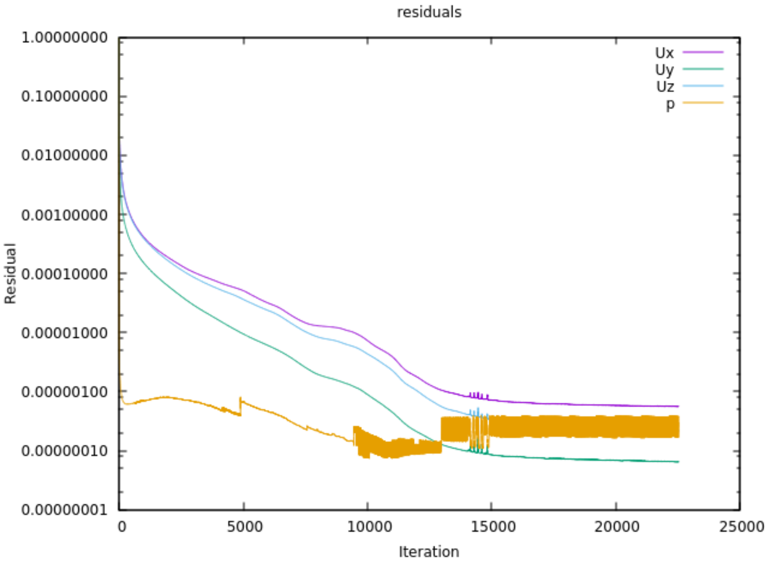


Figure A.2: Residual trend in the initial transient of LES simulation

pimpleFoam. Given the mesh size, the time step is set to $1 \cdot 10^{-5}s$: this way, the maximum Courant number across the mesh throughout the run is around $Co = 0.40$. On the other hand, the time limit depends on the typical time scales of the flow. Based on previous research, such as [8], the total simulation time is $0.75s$. In order to obtain a time-independent solution to employ in the comparison with RANS, the LES solution is averaged between $0.15s$ and $0.75s$, excluding the initial transient.

As for the boundary conditions, they are equivalent to the reference RANS settings, so that the results are comparable: wall conditions on the nasal walls and null gradient at the inlet/outlet for the velocity. The total pressure-over-density difference between spherical cap and laryngopharynx is $10m^2/s^2$ both for the LES and the reference RANS. This value is different from the $20m^2/s^2$ used in RANS throughout the dataset generation process: the reason is that this investigation was conducted in a preliminary phase, where the optimal simulation settings were yet to be determined.

Prior to the actual run, the computational domain is equally decomposed in 72 subdomains for parallel run. Despite this, the simulation took approximately 80 hours. The residual trend in the transient phase is reported in figure A.2. As anticipated, the transient terminates after 15000 iterations, corresponding to $0.15s$, after which the residuals stabilise around $10^{-7} - 10^{-8}$ and the solution is steady, as confirmed by intermediate results.

A preliminary step is necessary before presenting the comparison of

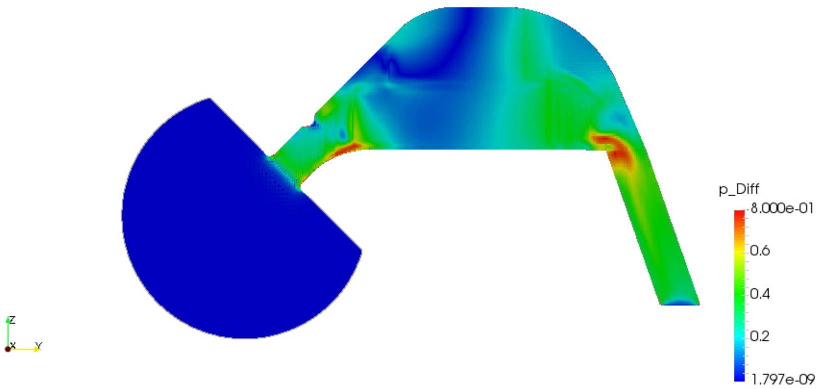


Figure A.3: Difference of pressure fields between RANS and LES solutions in a sagittal section

the results and consists in the mapping of the RANS solution to the finer computational grid used for the LES: this allows to compute difference of velocity and pressure fields, which is the most representative and synthetic way to get a full picture of the magnitude of discrepancy between the two.

The resulting differential fields displayed in figures A.3, A.4 and A.5 are quite far from the ideal null fields which would be obtained if the two simulations produced equal outcomes. The difference in the pressure field actually remains within 10% throughout the domain, with the exception of very local peaks near the wall due to boundary effects, but the velocity magnitude difference is much more significant, especially in the high velocity regions at the inlet of the inferior meatus and in the lower part of the nasal fossae. The coronal section in figure A.5 shows a consistent velocity difference along the vertical channel close to the nasal septum. It is in the order of $0.35m/s$, which roughly corresponds to a $15-20\%$ discrepancy. Additionally, difference of velocity peaks are present at the inferior meatus entrance and are in the order of 100% of the values in the original velocity fields.

Given that LES intrinsically produces a more truthful reconstruction of the flow field, by largely reducing the impact of turbulence modelling, the comparison remarks how the RANS approach limits the accuracy of the results. The main conclusion drawn by this investigations is the potential improvement margin, which can be achieved in the future, by switching to more advanced CFD techniques.

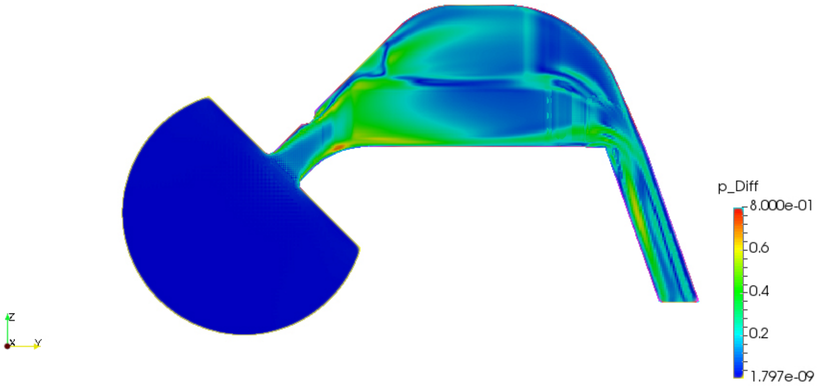


Figure A.4: Difference of velocity magnitude fields between RANS and LES solutions in a sagittal section

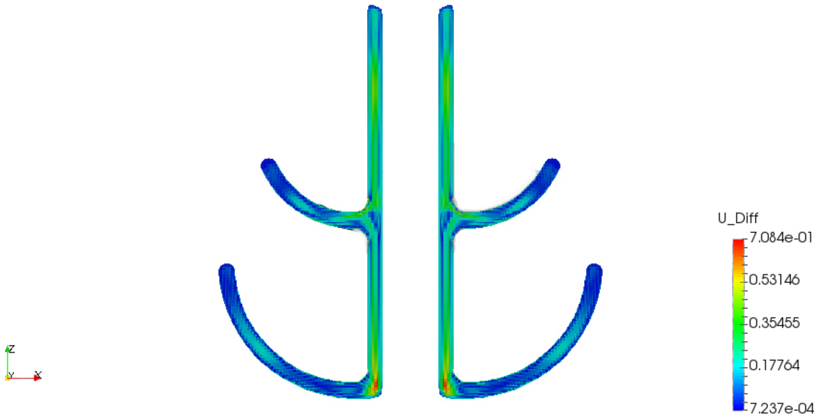


Figure A.5: Difference of velocity magnitude fields between RANS and LES solutions in a coronal section

Appendix B

Codes and listings

This section includes the main codes implemented in the automatisation of the post-processing procedures, for a more thorough overview on the procedures adopted.

Listing B.1: Automatic data download

```
#!/usr/bin/expect

# GRAB PASSWORD
stty -echo
send_user -- "Password for username: "
expect_user -re "(.*)\n"
send_user "\n"
stty echo
set pass $expect_out(1,string)

# DOWNLOAD
spawn scp -r username@login.galileo.cineca.it:/
    path_to_file /path_to_folder
expect "password:"
send "$pass\n"
interact
```

Listing B.1 reports a short *Expect* script that makes the download (and upload) of data on the CINECA servers more efficient. *Expect* is a typical tool to automatise operations on SSH operations. The second part of the code can be copied and pasted to download and upload several objects at a time. The uploading version is simply obtained by exchanging the destination and the source directories.

Listing B.2: Post-processing operations including the extraction of coronal sections and streamlines

```
from paraview.simple import *

for i in range(1,201):

    case = str(i)
    openPath = '/path_to_folder_case_(i)'
    savePath = '/path_to_folder_case_(i)'

    # create a new 'OpenFOAMReader'
    a1foam = OpenFOAMReader(FileName=openPath)
    a1foam.MeshRegions = ['internalMesh']
    a1foam.CellArrays = ['U', 'p']

    # get animation scene
    animationScene1 = GetAnimationScene()
    animationScene1.GoToLast()

    # Slice a
    slice1 = Slice(Input=a1foam)
    slice1.SliceType = 'Plane'
    slice1.SliceOffsetValues = [0.0]
    slice1.SliceType.Origin = [0.0, 0.0, 0.004]
    slice1.SliceType.Normal = [0.0, 1.0, 1.0]

    SaveData(savePath + '/a.vtm', proxy=slice1, Writealltimestepsasfileseries=1,DataMode='Binary')

    # create StreamTracerWithCustomSource
    disk1 = Disk()
    disk1.InnerRadius = 0.004
    disk1.OuterRadius = 0.011
    disk1.RadialResolution = 3
    disk1.CircumferentialResolution = 25

    transform1 = Transform(Input=disk1)
    transform1.Transform = 'Transform'
    transform1.Transform.Translate = [0.0, 0.0005, -0.0065]
    transform1.Transform.Rotate = [-45.0, 0.0, 0.0]
    transform1.Transform.Scale = [0.7, 0.9, 0.0]

    streamTracerWithCustomSource1 = StreamTracerWithCustomSource(Input=a1foam, SeedSource=transform1)
    streamTracerWithCustomSource1.Vectors = ['POINTS', 'U']
    streamTracerWithCustomSource1.IntegrationDirection = 'FORWARD'
    streamTracerWithCustomSource1.IntegratorType = 'Runge-Kutta 4'
    streamTracerWithCustomSource1.InitialStepLength = 1.5
    streamTracerWithCustomSource1.MaximumSteps = 8000
    streamTracerWithCustomSource1.MaximumStreamlineLength = 0.3
    streamTracerWithCustomSource1.ComputeVorticity = 0

    SaveData(savePath + '/st.csv', proxy=streamTracerWithCustomSource1)
```

Listing B.2 on page 59 shows the *Python* code used to automatically run the main post-processing operations on *Paraview*. For synthesis purposes, only the generation of section a is reported: sections b through f share the same commands.

Listing B.3: Post-processing operations including the extraction of pressure integral values in slices along *y-axis*

```
from paraview.simple import *

import csv

for i in range(1,201):

    case = str(i)
    openPath = '/path_to_folder_case_(i)'
    savePath = '/path_to_folder_case_(i)'

    print('Working on case ' + case)

  # create a new 'OpenFOAMReader'
    a1foam = OpenFOAMReader(FileName=openPath)
    a1foam.MeshRegions = ['internalMesh']
    a1foam.CellArrays = ['p']

  # get animation scene
    animationScene1 = GetAnimationScene()
    animationScene1.GoToLast()
    renderView1 = GetActiveViewOrCreate('RenderView')
    a1foamDisplay = Show(a1foam, renderView1)

  # create a new 'Clip'
    clip1 = Clip(Input=a1foam)
    clip1.ClipType = 'Plane'
    clip1.Scalars = ['POINTS', 'p']
    clip1.Value = -10.530899273231626
    clip1.ClipType.Origin = [0.0, 0.0, -0.0035]
    clip1.ClipType.Normal = [0.0, 1.0, 1.0]

  # create a new 'Clip'
    clip2 = Clip(Input=clip1)
    clip2.ClipType = 'Plane'
    clip2.Scalars = ['POINTS', 'p']
    clip2.Value = -11.296679854393005
    clip2.ClipType.Origin = [0.0, 0.014, -0.0]
    clip2.ClipType.Normal = [0.0, 1.0, 0.0]

  # create a new 'Clip'
    clip3 = Clip(Input=clip2)
    clip3.ClipType = 'Plane'
    clip3.Scalars = ['POINTS', 'p']
    clip3.Value = -13.614728450775146
    clip3.ClipType.Origin = [0.0, 0.0145, -0.0]
    clip3.ClipType.Normal = [0.0, -1.0, 0.0]

    IntegrateVariables1 = IntegrateVariables(Input=clip3)

    finalFile = open(savePath + '/clip' + case + '.csv','wb')
    fd = csv.writer(finalFile)

    fd.writerow(['p', 'Volume', 'Cell Type'])

    writer = CreateWriter(savePath + '/clip' + case + '_tmp.csv', IntegrateVariables1)
    writer.FieldAssociation = "Cells"

    list = range(0,120)

    for j in list:

        lb = 0.0140 + 0.0005*j
        ub = 0.0145 + 0.0005*j

    #Properties modified on clip3.ClipType
        clip3.ClipType.Origin = [0.0, ub, 0.0]

    # Properties modified on clip2.ClipType
        clip2.ClipType.Origin = [0.0, lb, 0.0]

    # create a new 'Integrate Variables'
        integrateVariables1 = IntegrateVariables(Input=clip3)

        # writes the integrated variable table to the tmp file
        writer.UpdatePipeline()

        # appends the integrated results of this Clip to the final csv file
        ftmp = open(savePath + '/clip' + case + '_tmp.csv','rb')
        reader = csv.reader(ftmp, delimiter=',')
        row = reader.next()
        row = reader.next()
        fd.writerow(row)
        ftmp.close()

    # destroy integrateVariables1
        Delete(integrateVariables1)
        del integrateVariables1

  # destroy clip3
    Delete(clip3)
    del clip3

    # destroy clip2
    Delete(clip2)
    del clip2

  # destroy clip1
    Delete(clip1)
    del clip1

  # destroy a1foam
    Delete(a1foam)
    del a1foam

    # close case csv file
    finalFile.close()
    del(writer)
```

Listing B.3 on page 61 reports the extraction of the pressure and volume integral values from thin slices distributed along the *y-axis*. They are later manipulated to compute averaged pressure values and, eventually the coefficients of a polynomial fit as described in chapter 4.

Bibliography

[1] E. Afiza et al. "Airflow Patterns within Real and 3D Simplified Models of Nasal Cavities (I. Experimental Study)". In: *Proceedings of the School of Engineering, Tokai University.* E 40 (2015), pp. 97–108 (cit. on pp. 2, 15, 16, 22).

[2] E. Afiza et al. "Airflow Patterns within Real and 3D Simplified Models of Nasal Cavities (II. Numerical Study)". In: *Proceedings of the School of Engineering, Tokai University.* E 40 (2015), pp. 109–119 (cit. on p. 15).

[3] L. Butera. "Nasal pathology assessment through Supervised Learning on Computational Fluid Dynamics data: a preliminary study". MA thesis. Politecnico di Milano (cit. on pp. 2, 53).

[4] F. Cozzi, G. Felisati, and M. Quadrio. "Velocity measurements in nasal cavities by means of stereoscopic PIV - Preliminary tests". In: *Journal of Physics: Conference Series* 882 (Aug. 2017) (cit. on p. 19).

[5] P. Geurts, D. Ernst, and L. Wehenkel. "Extremely Randomized Trees". In: *Machine Learning* 63 (Apr. 2006), pp. 3–42 (cit. on p. 8).

[6] I. Hörschler, W. Schröder, and M. Meinke. "On the assumption of steadiness of nasal cavity flow". In: *Journal of Biomechanics* 43.6 (Apr. 2010), pp. 1081–1085 (cit. on pp. 19, 34).

[7] E. Javaheri, L. Golshahi, and W.H. Finlay. "An idealized geometry that mimics average infant nasal airway deposition". In: *Journal of Aerosol Science* 55 (Jan. 2013), pp. 137–148 (cit. on pp. 14, 25).

[8] G. Lamberti and F. Manara. "Simulazioni RANS e LES della fluidodinamica delle cavità nasali". MA thesis. Politecnico di Milano, 2014 (cit. on pp. 1, 2, 17, 32, 56).

[9] Y. Liu et al. "Creation of a standardized geometry of the human nasal cavity". In: *Journal of Applied Physiology* 106.3 (Mar. 2009), pp. 784–795 (cit. on pp. 14, 16, 21).

[10] *Machine Learning bites.* URL: https://medium.com/machine-learning-bites/machine-learning-decision-tree-classifier-9eb67cad263e (cit. on pp. 7, 8).

[11] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003, pp. 535–548 (cit. on p. 10).

[12] F.R. Menter. “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications”. In: *AIAA Journal* 32.8 (Aug. 1994), pp. 1598–1605 (cit. on pp. 18, 33).

[13] T. M. Mitchell. *Machine Learning.* 1st ed. McGraw-Hill, Inc., 1997 (cit. on p. 5).

[14] A. Nejati et al. “A deformable template method for describing and averaging the anatomical variation of the human nasal cavity”. In: *BMC Medical Imaging* 16 (2016) (cit. on pp. 15, 21).

[15] S.B. Pope. *Turbulent Flows.* Cambridge University Press, 2000 (cit. on p. 18).

[16] M. Quadrio et al. “Review of computational fluid dynamics in the assessment of nasal air flow and analysis of its limitations”. In: *European Archives of Oto-Rhino-Laryngology* 271.9 (Sept. 2014), pp. 2349–2354 (cit. on pp. 1, 19).

[17] Y. Saeys, I. Inza, and P. Larrañaga. “A review of feature selection techniques in bioinformatics”. In: *Bioinformatics* 23.19 (Aug. 2007), pp. 2507–2517 (cit. on pp. 6, 36).

[18] *St. Louis Sinus Center.* URL: https://www.stlsinuscenter.com/turbinate-reduction/ (cit. on p. 14).

[19] S. Standring and N.R. Borley. *Gray’s Anatomy: The Anatomical Basis of Clinical Practice.* ClinicalKey 2012. Churchill Livingstone/Elsevier, 2008 (cit. on p. 11).

[20] *The Respiratory System.* URL: https://www.therespiratorysystem.com/nasal-cavity/ (cit. on pp. 12, 13).

[21] R. Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1996), pp. 267–288 (cit. on p. 9).

[22] J. Xi, J. Kim, and X. April Si. “Effects of nostril orientation on airflow dynamics, heat exchange, and particle depositions in human noses”. In: *European Journal of Mechanics - B/Fluids* 55 (201), pp. 215–228 (cit. on pp. 16, 22).

[23] J. Zhang. “Characteristic size research of human nasal cavity and the respiratory airflow CFD analysis”. In: *Journal of Biosciences and Medicines* 1.2 (2013), pp. 23–27 (cit. on p. 15).

[24] K. Zhao and J. Jianbo. “What is normal nasal airflow? A computational study of 22 healthy adults”. In: *International Forum of Allergy & Rhinology* 4.6 (June 2014), pp. 435–46 (cit. on p. 20).

[25] K. Zhao et al. “Effect of Anatomy on Human Nasal Air Flow and Odorant Transport Patterns: Implications for Olfaction”. In: *Chemical Senses* 29.5 (June 2004), pp. 365–379 (cit. on pp. 18, 19).

[26] K. Zhao et al. “Numerical Modeling of Turbulent and Laminar Airflow and Odorant Transport during Sniffing in the Human and Rat Nose”. In: *Chemical Senses* 31.2 (Feb. 2006), pp. 107–118 (cit. on p. 19).

[27] M. Zubair et al. “Review: A Critical Overview of Limitations of CFD Modeling in Nasal Airflow”. In: *Journal of Medical and Biological Engineering* 32.2 (2012), pp. 77–84 (cit. on p. 19).