

POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea in Ingegneria Aeronautica



Simulazione numerica del flusso nelle cavit  nasali

Relatore: Prof. Maurizio QUADRIO

Co-relatore: Prof. Giovanni FELISATI

Tesi di laurea di:
Stefano CORTI Matr. 730865

Anno Accademico 2010-2011

*Sognai talmente forte
che mi uscì il sangue dal naso
il lampo in un orecchio
nell'altro il paradiso*

(F. De André)

Indice

I	Introduzione	1
1	Descrizione del progetto	3
1.1	Presentazione	3
1.2	Obiettivi	4
1.3	Obiettivo della tesi	6
1.4	Struttura del lavoro	7
2	Anatomia del naso	11
2.1	Descrizione anatomica del naso	11
3	Stato dell'arte	17
3.1	Analisi della letteratura esistente	17
3.2	Articoli di riferimento	21
II	Analisi	25
4	Dalla TAC alla ricostruzione geometrica	27
4.1	Introduzione alla TC	27
4.2	Utilizzo di 3D Slicer	31
5	Verifica della geometria	37
5.1	Il <i>Software</i> MeshLab	37
5.2	Miglioramento della mesh	40
6	Introduzione a OpenFOAM	43
6.1	Presentazione di OpenFOAM 1.7	43
6.2	Struttura di OpenFOAM 1.7	45
7	Cenni di fluidodinamica del flusso nasale	53
7.1	Le equazioni della fluidodinamica	53
7.2	Aerodinamica della cavità nasale	61

8	Dalla geometria alla mesh di volume	65
8.1	Presentazione del problema	65
8.2	L' <i>utility</i> blockMesh	66
8.3	L' <i>utility</i> snappyHexMesh	68
III	Risultati	79
9	Istruzioni per le simulazioni	81
9.1	Elenco delle istruzioni OpenFOAM	81
9.2	CILEA e il <i>cluster</i> Lagrange	90
9.3	Automatizzazione del procedimento	91
10	Analisi del flusso laminare e turbolento	95
10.1	Presentazione simulazione laminare	95
10.2	Presentazione simulazione turbolenta	97
10.3	Analisi dei risultati	100
11	Analisi di sensibilità	113
11.1	Sensibilità della soglia di Slicer	113
11.2	Sensibilità parametri OpenFOAM	124
IV	Conclusioni	131
12	Valutazioni finali	133
12.1	Risultati raggiunti	133
12.2	Obiettivi di breve termine	135
12.3	Obiettivi di medio e lungo termine	136
12.4	Ulteriori sviluppi e studi futuri	137

Elenco delle figure

2.1	Geometria della cavità nasale	12
2.2	Principali seni paranasali	13
2.3	Vista 3D della cavità nasale	14
2.4	Cavità nasale in 3D, tratta da [13]	15
2.5	Sezioni coronali della cavità nasale	16
3.1	Esempio di modello <i>in-vitro</i> , tratto dall'articolo [5]	18
3.2	Modello del tratto respiratorio, tratto dall'articolo [16]	22
3.3	Velocità nel naso, tratto dall'articolo [16]	22
3.4	<i>Contour</i> del modulo della velocità, tratto dall'articolo [17]	23
3.5	Andamento delle linee di flusso, tratto dall'articolo [17]	24
4.1	Immagine assiale di una TC	28
4.2	Immagine sagittale di una TC	29
4.3	Immagine coronale di una TC	30
4.4	Interfaccia di 3D Slicer	32
4.5	Scelta della soglia in 3D Slicer	33
4.6	Passaggio di esclusione della testa in 3D Slicer	34
4.7	Passaggio di esclusione delle cavità auricolari in 3D Slicer	35
4.8	Selezione volume finale in 3D Slicer	36
4.9	Ricostruzione superficie in 3D Slicer	36
5.1	Ricostruzione superficie in MeshLab	37
5.2	Cavità nasale in MeshLab	39
5.3	Cavità nasale in MeshLab, prospettiva	39
5.4	Filtro <i>Fill Hole</i> di MeshLab	40
5.5	Filtro <i>Laplacian Smooth</i> di MeshLab	41
6.1	Struttura di OpenFOAM	44
6.2	Cartella e sottocartelle di un caso di OpenFOAM	46
7.1	Visualizzazioni con inchiostro, tratte dall'articolo [6]	62

7.2	Linee di flusso e velocità esterna, tratto da [6]	63
8.1	Risultato della <i>utility</i> blockMesh	68
8.2	Risultato di snappyHexMesh: <i>castellated mesh</i>	71
8.3	Risultato di snappyHexMesh: fase di <i>snap</i>	72
8.4	Sezione della <i>castellated mesh</i>	75
8.5	Sezione della mesh dopo la fase di <i>snap</i>	75
8.6	Sezione della mesh con in evidenza l'aggiunta dei <i>layers</i>	77
9.1	Sezione della mesh di volume completa	85
9.2	Cluster Lagrange del CILEA	90
9.3	Tempo di calcolo in funzione del numero di processori	94
9.4	Tempo di calcolo in funzione del numero di celle	94
10.1	Foto 3D dei vettori velocità	100
10.2	Foto 3D delle linee di flusso	101
10.3	Simulazione laminare, pressione nelle sezioni sagittali	102
10.4	Simulazione turbolenta, pressione nelle sezioni sagittali	102
10.5	Simul. laminare, modulo velocità nelle sezioni sagittali	102
10.6	Simul. turbolenta, modulo velocità nelle sezioni sagittali	103
10.7	Simulazione laminare, velocità U_y nelle sezioni sagittali	104
10.8	Simulazione turbolenta, velocità U_y nelle sezioni sagittali	104
10.9	Simulazione laminare, velocità U_z nelle sezioni sagittali	104
10.10	Simulazione turbolenta, velocità U_z nelle sezioni sagittali	105
10.11	Simul. laminare, vorticità ω_x in una sezione sagittale	105
10.12	Simul. laminare, pressione e velocità in una sezione assiale	106
10.13	Sezione dei turbinati, pressione casi laminare e turbolento	107
10.14	Sezione dei turbinati, U_x casi laminare e turbolento	107
10.15	Sezione dei turbinati, U_y casi laminare e turbolento	108
10.16	Sezione dei turbinati, U_z casi laminare e turbolento	108
10.17	Sezione dell'atrio, pressione casi laminare e turbolento	108
10.18	Sezione dell'atrio, U_x casi laminare e turbolento	109
10.19	Sezione dell'atrio, U_y casi laminare e turbolento	109
10.20	Sezione dell'atrio, U_z casi laminare e turbolento	110
10.21	Simul. turbolenta, k nelle sezioni sagittali	111
10.22	Simul. turbolenta, ω nelle sezioni sagittali	111
10.23	Simul. turbolenta, ν_t nelle sezioni sagittali	111
10.24	Simul. turbolenta, K e ω nella sezione dei turbinati	112
10.25	Simul. turbolenta, K e ω nella sezione dell'atrio	112
11.1	Vista 3D delle varie ricostruzioni, (a)	114
11.2	Vista 3D delle varie ricostruzioni, (b)	115

11.3	Cavità destra, pressione (a)	117
11.4	Cavità destra, pressione (b)	117
11.5	Cavità destra, modulo della velocità (a)	118
11.6	Cavità destra, modulo della velocità (b)	118
11.7	Cavità destra, velocità U_y (a)	119
11.8	Cavità destra, velocità U_y (b)	119
11.9	Zona dei turbinati, pressione (a)	120
11.10	Zona dei turbinati, pressione (b)	120
11.11	Zona dei turbinati, modulo della velocità	120
11.12	Zona dei turbinati, pressione, analisi laminare (a)	125
11.13	Zona dei turbinati, pressione, analisi laminare (b)	125
11.14	Cavità destra, pressione, analisi laminare	126
11.15	Cavità destra, modulo della velocità, analisi laminare	126
11.16	Cavità destra, energia cinetica turbolenta	127
11.17	Cavità destra, ω	127
11.18	Cavità destra, sensibilità refinementSurfaces	129
11.19	Cavità destra, sensibilità layers	129
11.20	Cavità destra, sensibilità layers (zoom)	129

ELENCO DELLE FIGURE

Elenco delle tabelle

10.1 Portate globali, caso laminare e turbolento	110
11.1 Analisi soglia Slicer	114
11.2 Analisi soglia Slicer, dati in gola	116
11.3 Analisi soglia Slicer, dati nei turbinati	122
11.4 Analisi soglia Slicer, dati nei turbinati (b)	123
11.5 Analisi dimensione mesh	124
11.6 Analisi dimensione mesh: portate globali	125

Sommario

Il presente lavoro di tesi riguarda uno studio fluidodinamico della cavità nasale. Lo spunto iniziale nasce da una collaborazione con il reparto di otorinolaringoiatria dell'azienda ospedaliera - polo universitario - San Paolo di Milano, con l'obiettivo finale di fornire ai medici uno strumento affidabile e basato su software open-source per lo studio specifico di ogni paziente. In tal modo, una volta completata la fase di ricerca, si avranno a disposizione molti dati qualitativi e quantitativi, che, oltre ad aumentare la conoscenza fluidodinamica in questo ambito, forniranno utili indicazioni sia per le operazioni chirurgiche (FESS - Functional Endoscopic Sinus Surgery), sia per una previsione della respirazione post-intervento, con il miglioramento del benessere dei pazienti.

La collaborazione prevede quindi un progetto molto ambizioso, di lunga durata, con numerosi obiettivi da raggiungere. Il presente lavoro si occupa di porre le basi per completare con successo la creazione dello strumento di analisi fluidodinamica. La prima parte affronta il problema del passaggio dai dati della tomografia computerizzata (TC) alla superficie di interesse; in seguito si procede con la costruzione della mesh di volume; da ultimo entra in gioco la simulazione fluidodinamica vera e propria, con la presentazione di alcuni risultati significativi, di esempio e di validazione di quanto ottenuto.

A livello di simulazione fluidodinamica sono state provate numerose geometrie, con flusso laminare o turbolento (modello $k-\omega$ -SST). Più che sul raffinamento della modellistica, demandato a un passo successivo, ci si è concentrati sulla costruzione di una procedura e sulla messa a punto di uno strumento per lo studio del flusso nelle cavità nasali.

Parole chiave: fluidodinamica nasale, 3D Slicer, OpenFOAM, flusso nasale laminare e turbolento, RANS $k-\omega$ -SST, visualizzazioni dati.

Abstract

The present thesis is concerned with a fluid dynamic study on the nasal cavity. The initial idea was inspired by a collaboration between the Department of Aerospace Engineering and the Ear-Nose-Throat Department (ENT) of San Paolo Hospital, in Milan, with the final goal to provide doctors a reliable and open-source tool for the specific study of every patient. In this way, once the research stage is over, doctors will have many qualitative and quantitative data, that, beyond increasing the knowledge on nasal fluid dynamic, will give useful indications for surgical operations (FESS - Functional Endoscopic Sinus Surgery) and for an effective prediction about patient's post-surgery breathing. We are confident that, in a future not too far, patients' health will be greatly enhanced with similar tools.

The collaboration is heavily involved with an ambitious project, with many goals to achieve. The present work is focused on laying down the bases to successfully create the proposed CFD tool. The first part faces the issue of the passage from CT scans (computed tomography) to the definition of the nasal cavity surface. After creating the volume mesh, it is possible to proceed with the CFD simulations and eventually main results are presented, with a particular attention to the validation of the CFD tool.

Fluid dynamic simulations have been performed with several different geometries, with laminar and turbulent flows (RANS, $k-\omega$ -SST model). In the present work we don't focus on the choice of the turbulence model, because our project is to create and to validate an operating procedure for the study of the flow in nasal cavities.

Keywords: nasal fluid dynamics, 3D Slicer, OpenFOAM, laminar and turbulent nasal flow, RANS $k-\omega$ -SST, flow visualization.

Parte I
Introduzione

Capitolo 1

Descrizione del progetto

1.1 Presentazione

Negli ultimi anni l'enorme evoluzione della potenza di calcolo computazionale, anche a basso costo, unita allo sviluppo di *software* gratuiti ed affidabili, ha reso possibile l'analisi fluidodinamica (CFD) del flusso all'interno della cavità nasale. É utile premettere subito che, allo stato attuale della conoscenza e delle possibilità numeriche disponibili, ci si occupa esclusivamente di problemi respiratori legati alla geometria della cavità nasale (ad esempio il setto nasale deviato o ipertrofia dei turbinati), mentre resta per il momento esclusa la possibilità di studiare le altre malattie, come ad esempio quelle causate dalle infiammazioni.

Questo breve accenno, legato alle ipotesi semplificatrici adottate¹, verrà ripreso quando si descriveranno i dettagli del lavoro svolto.

Ci sono quindi alcuni punti critici da risolvere e una serie di limitazioni, di cui si tratterà ampiamente nell'esposizione del presente lavoro, ma si aprono prospettive inedite ed interessanti in campo medico: si avranno a breve (e in parte già adesso) numerosi dati quantitativi per lo studio della respirazione di ogni singolo paziente da operare, che porteranno ad una maggiore conoscenza della fluidodinamica e quindi al benessere fisiologico del paziente stesso.

Infatti attualmente lo stato dell'arte dell'analisi della fisiopatologia respiratoria nasale mostra la possibilità di ottenere solo dati molto grossolani, con esperimenti *in-vivo* o *in-vitro*, che rientrano nella tecnica detta rinomanometria. Si possono quindi ottenere valutazioni esclusivamente globali, quali la portata o la resistenza globale dell'intera cavità

¹tra cui geometria fissata, mucosa non modellata, temperatura non considerata

nasale (o di alcune parti di essa) come rapporto tra la caduta di pressione tra due zone e la portata stessa.

Allo stato dell'arte è impossibile, oggi, disporre di dati utilizzabili clinicamente, che suggeriscano al terapeuta ed al chirurgo quanto le singole varianti anatomiche di ogni specifico caso contribuiscano al determinismo di una ostruzione nasale. Allo stesso modo non è prevedibile, se non in modo empirico, su quali strutture anatomiche alterate sia opportuno esercitare chirurgia e in che misura per ottenere la miglior respirazione possibile con il minor danno anatomico. L'ambizione finale del presente lavoro è quindi quella di fornire ai medici uno strumento gratuito per l'analisi numerica del flusso all'interno della cavità nasale, all'incirca dalle narici fino all'epiglottide.

La presente tesi si può considerare come il primo passo all'interno di questa collaborazione, che prevede quindi un lavoro di lunga durata, con forte interazione tra il gruppo del dipartimento di Ingegneria Aerospaziale che si occupa della parte CFD e gli esperti di otorinolaringoiatria (d'ora in poi abbreviata in ORL). Gli obiettivi, presentati in seguito all'interno di questo capitolo, sono molteplici, di cui solo i primi compiutamente raggiunti in questo lavoro. L'obiettivo primario è infatti quello di mettere a punto una procedura affidabile, il più possibile automatizzata, per lo studio della fluidodinamica nasale a partire dai dati di scansioni TC (tomografia computerizzata). Una volta sistemata questa parte si aprono innumerevoli possibilità di sviluppi futuri, tra cui i più significativi verranno descritti nel capitolo 12.

1.2 Obiettivi

Questo progetto di ricerca è avanzato da un gruppo di lavoro fortemente interdisciplinare. La prima componente del gruppo di lavoro è di tipo medico, con specifiche competenze diagnostiche, cliniche e chirurgiche in campo otorinolaringoiatrico (ORL). In particolare gli ORL del gruppo si caratterizzano per una specifica esperienza nella chirurgia endoscopica naso-sinusale che rappresenta, attualmente, il *gold standard* del trattamento chirurgico di tutta la patologia ostruttiva, malformativa e infiammatoria dei distretti naso-sinusali. La seconda componente è di tipo ingegneristico ed è in grado di aggiungere alle competenze del gruppo ORL competenze avanzate di supercalcolo, di fluidodinamica computazionale e di modellistica numerica di flussi turbolenti (gruppo CFD).

Primo obiettivo del progetto è lo sviluppo e la verifica di una procedura, basata sul calcolo automatico, per la simulazione numerica della flui-

dodinamica nasale, replicando anzitutto e tentando di migliorare quanto finora riportato in letteratura. Un secondo obiettivo, di medio termine, è applicare la procedura a situazioni dinamiche, verificandone l'applicabilità, ad esempio nel confronto fra la situazione pre- e post-chirurgica. Obiettivo di lungo termine è realizzare una procedura sufficientemente raffinata per permettere di valutare, con costi e tempi ragionevoli, diverse opzioni di intervento chirurgico su specifico paziente.

Una possibile suddivisione di tutto il lavoro del progetto di ricerca è la seguente:

Fase preliminare: mesh di volume a partire dalla scansione TC. La letteratura esistente mostra come nelle poche esperienze disponibili questa fase sia quella in assoluto più costosa in termini di ore-uomo, e richieda l'uso combinato di svariati strumenti software, tipicamente commerciali. La nostra ambizione è quella di compiere questa prima fase in maniera rapida ed in gran parte automatizzata, per non precludere la possibilità di studiare una casistica sufficientemente ampia da dare ai risultati il necessario carattere di generalità. Questa prima parte vede coinvolto soprattutto il gruppo CFD, con importanti feedback dal gruppo ORL.

Primo obiettivo: Simulazione fluidodinamica, validazione dei risultati. Occorre mettere a punto sia il modello matematico con cui viene studiato numericamente il flusso, sia la strategia di simulazione. Per quanto riguarda il modello di calcolo, si va dall'estremo più semplice delle equazioni di Navier-Stokes stazionarie per flusso laminare a quello più complesso delle simulazioni LES non stazionarie, attraverso il passaggio intermedio delle equazioni RANS (stazionarie o quasi stazionarie) con opportuni modelli di turbolenza. Per quanto riguarda invece la strategia di simulazione, occorrerà valutare l'applicabilità dell'ipotesi di flusso stazionario, sino ad oggi adottata in maniera relativamente acritica in letteratura, e studiare la strategia migliore di imposizione delle condizioni al contorno. In particolare il ruolo della turbolenza e del suo modello andrà esaminato accuratamente. Ci proponiamo di scendere al livello di validazione nel dettaglio delle principali strutture fluidodinamiche. Questa seconda parte vede coinvolti i due gruppi con impegno equivalente.

Secondo obiettivo: effetti fluidodinamici dei principali interventi. In questo ultimo passo, corrispondente al secondo obiettivo, a partire da dati TC reali si analizzeranno le implicazioni fluidodinamiche e funzionali dei principali interventi di chirurgia. Il confronto, in varie situazioni anatomiche, fra TC pre- e post-operatoria consentirà di verificare se il dato numerico uscito dalla simulazione fluidodinamica riflette, nelle

due situazioni, l'effettivo dato clinico così come individuato da almeno tre esaminatori differenti *in cieco* e dai pazienti. Questa terza parte vede coinvolto soprattutto il gruppo ORL, con importante assistenza dal gruppo CFD.

Il presente lavoro di tesi si occupa di tutta la fase preliminare e arriva fino a metà del primo obiettivo.

1.3 Obiettivo della tesi

Scendendo un po' di più nel dettaglio, il presente lavoro si articola nel seguente modo: una prima parte, come già detto, è dedicata al passaggio dalla TAC (tomografia assiale computerizzata) dei pazienti alla geometria della cavità nasale. L'obiettivo sarà raggiunto grazie al *software open-source* 3D Slicer. Questa parte del lavoro è particolarmente delicata, in quanto richiede un minimo livello di esperienza e conoscenza medica per poter ricostruire correttamente la geometria di interesse. Inoltre non è facilmente automatizzabile per una serie di motivi che saranno più chiari in seguito. Si può però già anticipare che la ricostruzione della geometria (file .stl) avviene tramite un controllo visivo diretto di come potrebbe essere il risultato finale della ricostruzione. Questo, unito alla specificità di ogni TAC, rende pressochè improponibile, anche in futuro, l'automatizzazione completa di questo passaggio.

Una volta ottenuta la superficie della cavità nasale, si procede con il software per la simulazione CFD, OpenFOAM, anch'esso *open-source*. Questa parte è indubbiamente la più consistente del lavoro e lascia spazio ad una lunga serie di possibili sviluppi futuri, a cui si accennerà nel capitolo dedicato. In ogni caso si tratta innanzitutto di generare la mesh di volume dalla superficie ottenuta in precedenza e poi procedere con la simulazione vera e propria. Per quanto riguarda i modelli di calcolo sono stati investigati sia il caso laminare che il caso turbolento, con una simulazione RANS $k - \omega$. Sono stati effettuati calcoli di inspirazione, sia di transitorio che stazionaria. Inoltre sono state effettuate alcune analisi di sensibilità sui parametri di 3D slicer e di OpenFOAM ritenuti più significativi, in modo da validare per quanto possibile la procedura messa a punto.

A questi due *software* si aggiunge MeshLab, anch'esso *open-source*, utilizzato per controllare visivamente ed eventualmente modificare la geometria generata da Slicer.

La presente tesi descrive tutta la procedura e i primi casi analizzati, con le analisi per le validazioni di quanto fatto. Di ciascun pro-

gramma utilizzato verranno spiegati nei dettagli solo i passaggi utili, che ovviamente sono una minima parte delle potenzialità dei *software*.

1.4 Struttura del lavoro

Si delinea di seguito un breve riassunto sulla strutturazione del lavoro svolto. La Parte I è di introduzione a tutta la tesi e contiene i principali riferimenti bibliografici. La Parte II è l'analisi vera e propria del progetto, in cui trovano posto la spiegazione dettagliata di tutto il lavoro svolto e alcuni capitoli dedicati ad una breve presentazione dei software e ad un'introduzione alla fluidodinamica, con particolare attenzione agli aspetti più strettamente legati al caso in esame. La Parte III riguarda i risultati ottenuti, disposti nell'ordine cronologico con cui sono stati svolti i calcoli. In ogni caso tale ordine è comunque legato anche alla difficoltà dei diversi approcci, nel senso che i primi risultati sono quelli più semplici e le complicazioni (di geometria, di solutori) aumentano con lo sviluppo del discorso. Infine la Parte IV ha il compito di trarre le conclusioni ritenute più significative e accennare agli sviluppi futuri.

Parte I

- Capitolo 1: Descrizione del progetto. Contiene una descrizione più approfondita del progetto di collaborazione, dagli spunti iniziali fino agli sviluppi attuali e futuri.
- Capitolo 2: Anatomia del naso. Si descrive l'anatomia delle cavità nasali, in modo da seguire i passaggi successivi.
- Capitolo 3: Stato dell'arte. In questo capitolo sono descritti i risultati principali ottenuti dagli Autori che si stanno occupando da qualche anno della fluidodinamica della cavità nasale. Per ogni lavoro vengono evidenziati pregi e difetti, paragonandoli ai risultati di questo progetto, ricordando sia i risultati attesi che quelli effettivamente ottenuti.

Parte II

- Capitolo 4: Dalla TAC alla ricostruzione geometrica. In questo capitolo si descrive il passaggio che consente di ottenere la superficie nasale di interesse grazie al *3D Slicer*, elencando la procedura definitiva ed i vari tentativi che hanno portato ad essa.

- Capitolo 5: Verifica della geometria. Con il *software* MeshLab si procede ad un controllo visivo di quanto ottenuto dalla procedura del capitolo precedente e si applicano alcuni filtri geometrici per migliorare le caratteristiche della superficie che dovrà essere utilizzata dal CFD.
- Capitolo 6: Introduzione a OpenFOAM. OpenFOAM è il programma principale di questo lavoro di tesi e necessita di un capitolo introduttivo per poter comprendere i meccanismi principali che portano alla soluzione di un problema fluidodinamico. Dato che in realtà OpenFOAM è una libreria scritta in C++ servono alcuni semplici concetti di programmazione, anche se si vuole utilizzare il *software* solo dal punto di vista di un utente. Da ultimo si presenta Paraview, che è lo standard per visualizzare i risultati di molti solutori fluidodinamici.
- Capitolo 7: Cenni di fluidodinamica del flusso nasale. La presente tesi è esclusivamente numerica, tuttavia è fondamentale tenere presente le basi teoriche della fluidodinamica. A queste vanno aggiunte alcune considerazioni relative all'aerodinamica del flusso nasale, che tornano utili quando si tratta di commentare i risultati.
- Capitolo 8: Dalla geometria alla mesh di volume. In questo capitolo si affronta la seconda parte della procedura, che consiste nel passaggio dalla geometria della cavità nasale alla mesh di volume generata dalla libreria snappyHexMesh di OpenFOAM.

Parte III

- Capitolo 9: Istruzioni per le simulazioni. In questo capitolo si descrivono gli ultimi passaggi necessari a far partire le simulazioni fluidodinamiche. Si presenta anche l'utilizzo dei server del CILEA, che è il Consorzio Interuniversitario Lombardo per L'Elaborazione Automatica, a cui sono stati affidati gli ultimi e onerosi calcoli del flusso della cavità nasale.
- Capitolo 10: Analisi del flusso stazionario laminare e turbolento. Si presentano i primi risultati ottenuti, partendo naturalmente dai più semplici, ovvero quelli laminari.
- Capitolo 11: Analisi di sensibilità. Come verrà spiegato ampiamente in seguito, non è possibile una validazione assoluta dei risultati

ottenuti, ma si cerca in questo capitolo di analizzare la sensibilità dei risultati rispetto ad alcuni parametri significativi, in modo tale da mostrare quantomeno la ragionevolezza delle scelte effettuate e dei risultati ottenuti.

Parte IV

- Capitolo 12: Conclusioni. In questo capitolo si riassumono i risultati più significativi del presente lavoro di tesi e si accenna ai numerosi possibili sviluppi futuri.

Capitolo 2

Anatomia del naso

2.1 Descrizione anatomica del naso

Questa tesi si occupa solo della prima parte del progetto di collaborazione con la clinica S. Paolo di Milano, pertanto non sono richieste conoscenze approfondite dell'anatomia del naso, almeno fino alla parte di commento dei risultati.

Tuttavia, per seguire tutti i passaggi futuri con un minimo di chiarezza, può essere utile una breve descrizione anatomica della cavità nasale.

Il naso è il primo tratto delle vie respiratorie e deve svolgere alcune fondamentali funzioni, tra cui il riscaldamento, l'umidificazione e il filtraggio dell'aria inspirata.

Si ritiene che la complessità geometrica delle strutture interne al naso, descritte di seguito, e la mucosa di rivestimento siano dovute principalmente alla necessità di assolvere questi compiti. Altrettanto importante risulta anche la percezione olfattiva, anche se nel presente lavoro è quasi totalmente trascurata.

Ritornebbe in gioco qualora si volesse in futuro effettuare qualche analisi CFD di *particle tracking*, già adesso utilizzate per indagini più specificatamente farmacologiche.

È però interessante notare che il naso assolve funzioni molto diverse, che sono caratterizzate da tempi differenti: il riscaldamento infatti dev'essere il più veloce possibile, mentre i sensori dell'olfatto richiedono tempi caratteristici molto più lunghi.

Infatti sono posizionati in zone di ricircolo in cui la velocità media del flusso è molto più bassa rispetto al flusso principale che dalle narici procede verso la nasofaringe.

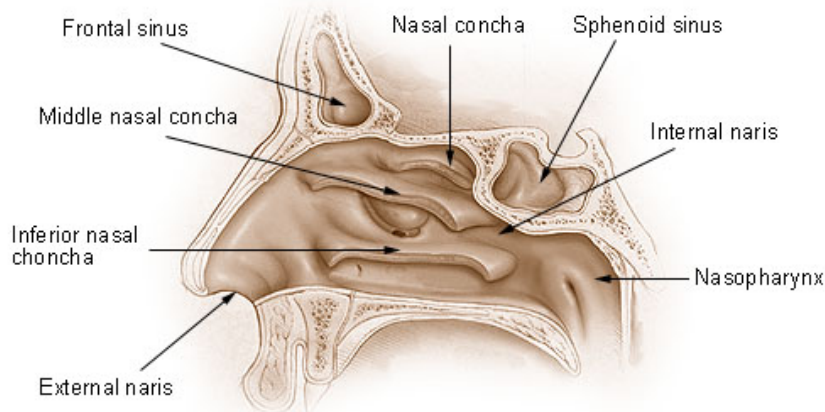
Nose and Nasal Cavities

Figura 2.1: Geometria e strutture principali della cavità nasale, viste in una generica sezione sagittale

In figura 2.1 si riporta un disegno con la geometria della cavità nasale e le principali parti in evidenza, nella vista sagittale. Da sinistra si nota l'ingresso delle narici esterne¹, il passaggio centrale attraverso diverse zone dette “meati” (superiore, medio e inferiore) e il ricongiungimento di questi passaggi nella zona della nasofaringe, attraverso le aperture posteriori dette “coane”.

Il naso è formato da due fosse nasali all'incirca simmetriche separate dal setto nasale e ha uno scheletro a struttura osteocartilaginea. Si identificano per entrambe le cavità 4 pareti:

- la **parete inferiore** (pavimento) della cavità nasale corrisponde alla volta della cavità orale e risulta quindi leggermente inclinato verso l'alto procedendo verso l'interno della cavità;
- la **parete superiore** (volta ossea) è costituita anteriormente dall'osso nasale, dall'osso frontale e posteriormente dall'osso etmoide, che costituisce gran parte della volta della cavità nasale. In questa zona sono anche presenti le fibre del nervo olfattivo. L'etmoide si articola infine con lo sfenoide, in particolare con il pavimento dei seni sfenoidali, che piegano inferiormente e posteriormente, completando la volta del naso;

¹ovviamente in questa sezione si vede solo metà cavità nasale

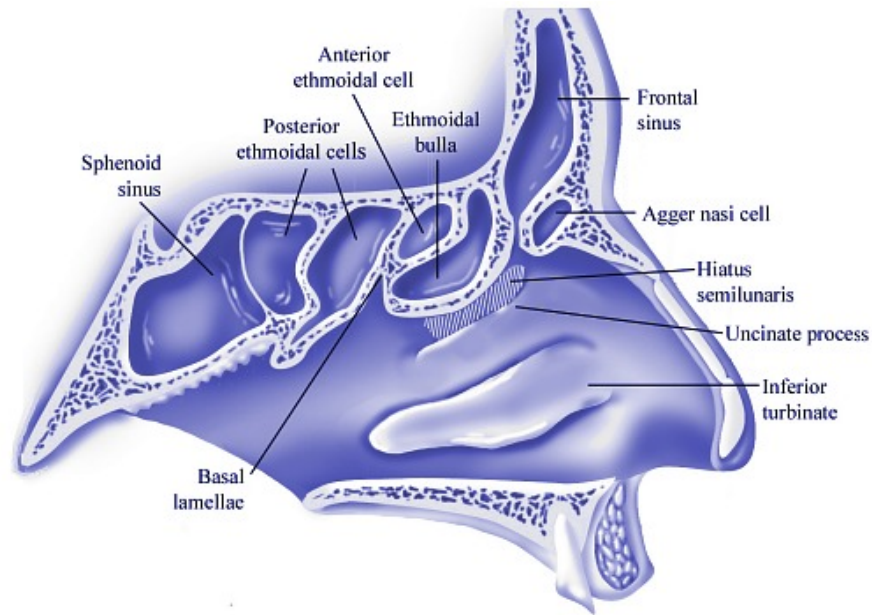


Figura 2.2: Principali seni paranasali, visti in sezione sagittale

- la **parete mediale** è formata dal setto nasale, costituito da una porzione cartilaginea anteriore e da una porzione ossea posteriore. Procedendo dalle narici verso l'interno si trovano quindi la cartilagine quadrangolare del setto, la lamina perpendicolare dell'etmoide, il vomere e in basso le creste nasali del mascellare e del palatino;
- la **parete laterale** della cavità nasale è costituita dai tre turbinati: superiore e medio, parti dell'etmoide, e inferiore, osso autonomo; inoltre sono costituite dal processo frontale del mascellare e dalla lamina verticale del palatino. A volte, in maniera incostante, esiste anche un turbinato supremo del Santorini, al di sopra del turbinato superiore. Le sporgenze ossee dei turbinati decorrono parallele tra loro e formano altrettanti meati comunicanti con le fosse nasali, nei quali si trovano gli orifizi di sbocco dei seni paranasali.

La cavità nasale infatti è anche caratterizzata dalla presenza di numerosi seni paranasali, che sono cavità accessorie scavate all'interno delle ossa del cranio e del massiccio facciale; comunicano con le fosse nasali per mezzo di piccoli osti che ne consentono la ventilazione e il drenaggio delle secrezioni. In figura 2.2 si vede una sezione sagittale con in evidenza i seni paranasali. In particolare nel meato superiore, il più stretto,

drenano le cellule etmoidali posteriori; nel meato medio, il più ampio, drenano le cellule etmoidali anteriori, il seno frontale, il seno mascellare; questi orifizi sono localizzati a livello dello iato semilunare, un solco a concavità dorsale delimitato dal processo uncinato anteriormente e dalla bolla etmoidale posteriormente; nel meato inferiore drena il condotto naso-lacrimale.



Figura 2.3: Vista 3D della cavità nasale. Si notino in particolare in viola e in blu i seni mascellari, in verde e in azzurro i numerosi seni etmoidali, in giallo e lilla i seni sfenoidali. Non sono visualizzati i seni frontali.

Per una visione d'insieme di maggiore chiarezza ci si può aiutare con la figura 2.3, in cui sono in evidenza (in 3D) i principali seni paranasali. Nelle figure 2.4 e 2.5 (tratte dall'articolo di Shi, Kleinstreuer, Zhang [13]) si riporta invece la geometria del naso (solo il condotto dalle narici alla nasofaringe, senza seni paranasali) con due diverse sezioni coronali. In queste sezioni sono molto evidenti i meati e i turbinati descritti in precedenza.

Il seno mascellare, detto Antro di Hignoro, è il più ampio, in genere con una capacità di 10 cm^3 : è una cavità scavata nello spessore del

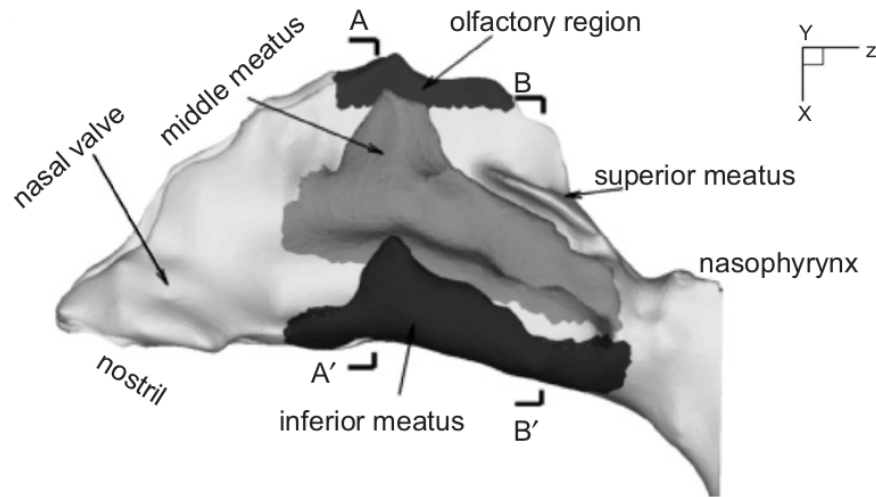


Figura 2.4: Cavità nasale in 3D, dalle narici alla nasofaringe. Si notino i meati, la valvola nasale e la regione olfattiva. Le sezioni A-A' e B-B' sono riportate in figura 2.5.

corpo dell'osso mascellare e si trova, nei due lati, al di sopra dell'arcata dentaria superiore in rapporto con premolari e molari, lateralmente alle fosse nasali.

Il seno frontale fa parte dell'osso frontale, ed è situato al di sopra dell'orbita e della radice del naso, davanti alla fossa cranica anteriore; per mezzo del canale naso-frontale è in comunicazione con il meato medio. I seni frontali sono molto variabili per forma e dimensioni, potendo risultare ipoplasici o al contrario così sviluppati da raggiungere i seni sfenoidali, oppure possono mancare completamente; spesso sono asimmetrici tra i due lati, essendo separati da un sottilissimo setto mediano che in genere devia da un lato e a volte possono anche formare una cavità unica.

Le cellule etmoidali anteriori sono piccole cavità multiple - in genere 5 o 6 per ogni lato, ma possono variare tra 2 e 8 - poste all'interno della regione anteriore delle masse laterali dell'etmoide, situate medialmente alla cavità orbitaria e lateralmente alla fossa nasale.

Le cellule etmoidali posteriori si trovano nella regione posteriore delle masse laterali dell'etmoide, davanti al seno sfenoidale e sono in genere meno numerose di quelle anteriori, ma più grandi; drenano nel meato superiore, vicino all'ostio del seno sfenoidale.

Il seno sfenoidale è scavato nello spessore del corpo dello sfenoide,

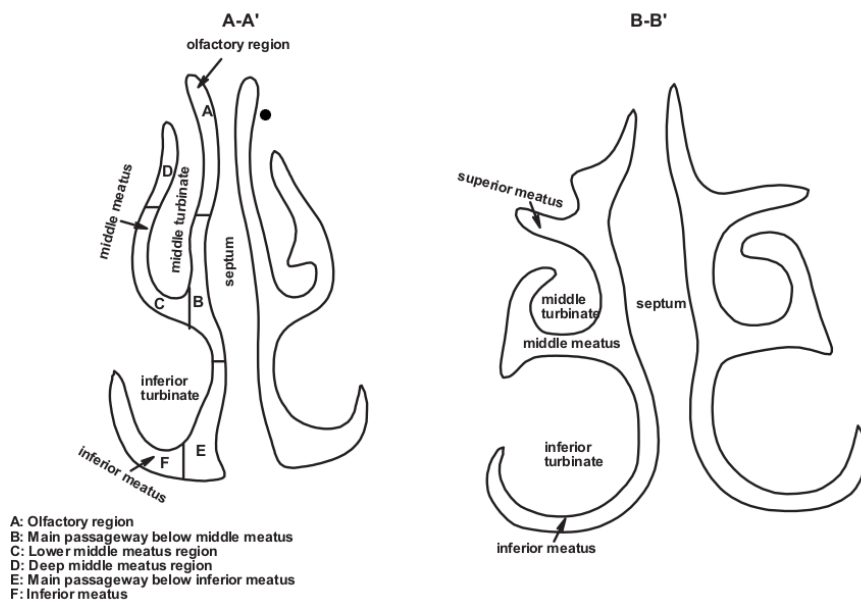


Figura 2.5: Sezioni coronali della cavità nasale, utili per comprendere la forma in sezioni di passaggio dell'aria intermedie tra le narici e la nasofaringe. Le sezioni A-A' e B-B' si riferiscono alla figura 2.4. Si notino le forme complicate dei turbinati e dei meati.

ed è posto subito al di sotto della sella turcica, medialmente al seno cavernoso e superiormente alle coane e al rinofaringe. I seni sfenoidali dei due lati sono praticamente contigui tra loro, ma spesso non uniformi per volume poichè separati da un setto che in genere non è perfettamente mediano. L'orifizio si apre nella volta delle cavità nasali.

I seni paranasali svolgono un ruolo importante nella respirazione: riscaldano e umidificano l'aria inspirata, e durante gli atti della respirazione l'aria in essi racchiusa si mescola con l'aria inspirata e con l'aria espirata, modificandone la composizione. Svolgono un certo ruolo anche nella fonazione, fungendo da cassa di risonanza.

Capitolo 3

Stato dell'arte

3.1 Analisi della letteratura esistente

L'analisi fluidodinamica del flusso all'interno della cavità nasale è un argomento relativamente recente, che si è sviluppato in modo particolare negli ultimi 5-6 anni. È logico aspettarsi che la letteratura a riguardo non sia pertanto estesa, dato che sono pochi i gruppi di lavoro in tutto il mondo che stanno affrontando questi argomenti.

Presentiamo di seguito lo stato attuale dell'arte, ponendo maggiormente la nostra attenzione sugli strumenti informatici utilizzati dagli autori di questi articoli (software per il passaggio dalla TAC alle superfici di interesse, software per il pre-processing, per costruire la mesh, per i calcoli fluidodinamici e da ultimo per il post-processing). Riteniamo inoltre di fondamentale importanza capire la complessità delle simulazioni presentate, sia in termini di modellazione (flusso laminare o turbolento, stazionario o instazionario, geometrie più o meno complicate, numero delle celle di calcolo...) che in termini di impegno richiesto come ore-uomo e ore-macchina.

L'obiettivo infatti del nostro progetto è quello di costruire e mettere a punto una procedura per l'analisi fluidodinamica, che sia la più semplice possibile e allo stesso tempo adeguatamente precisa e affidabile, basata sull'utilizzo di software open-source e con costi di calcolo ragionevoli.

Gli articoli seguenti sono presentati (alcuni solo accennati) in ordine cronologico a partire dai meno recenti, in modo da avere un'idea anche dell'evoluzione nello studio in questo ambito. Infatti i progressi nello studio della fluidodinamica nasale permettono oggi risultati molto più precisi e complicati, rispetto anche solo a pochi anni fa. Per il momento trascuriamo quasi del tutto gli approcci *in-vivo* (ovvero rinomanometria)

e *in-vitro* (ad esempio in figura 3.1), dato che il presente lavoro di tesi si concentra solo sull'analisi CFD.



Figura 3.1: Esempio di modello plastificato per studi *in-vitro*, dalle narici (a destra) fino alla nasofaringe. Tratto dall'articolo di Doorly et al. del 2007 [5].

Tra i primi articoli che presentano analisi CFD della cavità nasale troviamo quelli di Zhao et al. [18],[17] del 2004 e del 2006: sono entrambi focalizzati sullo studio della geometria nasale e sull'effetto che questa ha sul trasporto di particelle. In questi articoli si fa anche riferimento a lavori precedenti (dal 1995 circa), che contenevano i primi tentativi di generare modelli 3D a partire da scansioni TC. L'innovazione di questo lavoro consiste nell'aver messo a punto una tecnica (utilizzando software commerciali) per produrre in pochi giorni una mesh utilizzabile per le simulazioni, con 1.77 milioni di celle di calcolo. Il flusso è ipotizzato stazionario e laminare nel primo articolo, mentre vengono indagati diversi modelli di turbolenza, uniti ad un calcolo non stazionario, nel secondo articolo.

Anche l'articolo del 2005 di Naftali et al. [12] tratta il problema del trasporto di particelle, confrontando risultati ottenuti con 2 diversi modelli 3D: il primo di questi è un'idealizzazione molto semplice della cavità nasale, l'altro è ottenuto dai dati TC; Il numero di celle di calcolo è rispettivamente di 300.000 e 500.000. Viene simulato un flusso laminare

non-stazionario, utilizzando software commerciali (Gambit e Fluent) sia per la costruzione della mesh che per la simulazione.

Concettualmente più vicino al nostro progetto è l'articolo di Wexler et al. [14], anch'esso del 2005. L'obiettivo di questo lavoro è infatti quello di confrontare una geometria ottenuta da scansioni TC con la stessa a cui è stata arbitrariamente alterata una piccola zona del turbinato inferiore sinistro; in tal modo gli autori intendono verificare quali siano le conseguenze di un'eventuale riduzione dei turbinati. La geometria, rispetto alle possibilità attuali, è però abbastanza grossolana, potendo contare su un modello da circa 135.000 celle di calcolo, ottenuto con scansioni spaziate di 3 mm. Il flusso è ipotizzato laminare e stazionario, risolto ancora una volta con Fluent.

L'articolo di Croce et al.[4] si basa invece sul paragone tra esperimenti (con un modello in-vitro) e una simulazione fluidodinamica, ottenuta anche in questo caso con Fluent, per un flusso laminare e stazionario.

L'articolo di Elad et al.[7] del 2006, che utilizza gli stessi modelli già descritti dall'articolo [12], riguarda invece lo studio degli sforzi sulla mucosa nasale durante la respirazione.

Lo studio di Ishikawa et al.[8] è focalizzato sulla visualizzazione delle linee di flusso all'interno della cavità nasale. Il modello geometrico utilizzato, ottenuto da scansioni TC, consta di 1.2 milioni di celle; il flusso, laminare non-stazionario, è risolto ancora una volta con Fluent.

Anche l'articolo di Jin et al.[9] riguarda il trasporto e la deposizione delle particelle inalate, ma a differenza dei primi articoli presentati [18], [17], [12], la geometria è idealizzata ed arriva fino ai bronchi, con un modello da 3 milioni di celle; inoltre le simulazioni numeriche (Fluent) sono state effettuate con un modello LES per la turbolenza, sia in caso di flusso stazionario che non-stazionario. A parte la geometria molto semplificata è indubbiamente il conto più costoso in termini di ore-macchina tra quelli finora presentati.

Abbastanza simile a quest'ultimo è anche l'articolo di Shi et al. [13], che però parte da scansioni TC, pur introducendo alcune semplificazioni. Anche in questo caso sono stati utilizzati software commerciali, per risolvere un flusso stazionario laminare con un modello composto da 1.7 milioni di elementi. Rispetto all'articolo precedente si ha quindi una geometria decisamente più complicata, a fronte però di una modellazione numerica più semplice. La particolarità di questo studio è di considerare anche la rugosità delle pareti (con equazioni semi-empiriche), in modo da simulare la cattura delle particelle inalate da parte della mucosa.

Uno degli articoli che riteniamo più significativi è quello di Doorly et al.[6], del 2008. Risulta particolarmente utile l'inquadramento generale

del problema, con una dettagliata presentazione della geometria nasale e dei principali approcci seguiti per la modellazione del flusso nella cavità nasale. A livello di simulazioni gli autori presentano sia un caso laminare e stazionario che un caso di flusso turbolento, con rispettivamente 3.5 e 29 milioni di celle di calcolo. Sono inoltre presenti molti commenti, riguardanti la validità dei modelli e delle ipotesi adottate. Tra le innovazioni più importanti c'è anche l'idea (ripresa in seguito da altri ricercatori) di simulare il flusso anche all'esterno del volto e non solo nella zona di interesse, contribuendo sicuramente ad un maggiore realismo nella zona delle narici, su cui altrimenti si devono imporre delle condizioni al contorno.

L'articolo di Xiong et al.[15] presenta una simulazione CFD su una geometria ottenuta da scansioni TC; in seguito viene simulato un intervento chirurgico, per studiarne le conseguenze. Risulta quindi vicino a quanto ci proponiamo di affrontare con il nostro progetto. Anche in questo caso, come nei precedenti, sono stati utilizzati software commerciali, con simulazioni laminari e stazionarie. Il modello è formato da circa 1.8 milioni di celle.

L'articolo di Kleinstreuer e Zhang del 2009 [10] torna invece al problema del trasporto di particelle: a livello geometrico risulta piuttosto semplice, in quanto è stato deciso dagli autori di considerare configurazioni idealizzate; tuttavia hanno modellato tutto il tratto respiratorio, dalle narici fino agli alveoli polmonari. Per ogni zona viene quindi presentata una idealizzazione geometrica, vengono descritte le equazioni e risulta una simulazione fluidodinamica.

Tra gli articoli più interessanti risulta sicuramente quello di Zachov et al.[16], che viene presentato con alcuni dettagli nella sezione successiva.

Da ultimo citiamo anche gli articoli di Chen et al. [2], [3], del 2009 e del 2010. Nel primo si confronta una geometria di un naso sano, ottenuta da scansioni TC, con un'altra geometria caratterizzata da un setto deviato. Il flusso è considerato stazionario, ma vengono impiegati diversi modelli di turbolenza per risolvere le RANS (software commerciali). Il modello geometrico è formato da 2 milioni di celle, ma considera solo la cavità nasale. Il secondo articolo è analogo al primo, con uno studio delle conseguenze sulla respirazione a causa di una ipertrofia (simulata virtualmente) dei turbinati inferiori.

Da quanto esposto finora si può concludere che il nostro obiettivo principale potrebbe essere quello di replicare i migliori risultati ottenuti in questi articoli. Pensiamo infatti di poter arrivare con successo a sistemare una procedura in grado innanzitutto di costruire una geometria realistica (dalle scansioni), includendo anche la zona del volto, ester-

namente alla cavità nasale. Da questa geometria dettagliata si dovrà passare ad una accurata mesh di volume (qualche milione di celle) e ad una simulazione CFD di un flusso turbolento non stazionario, il tutto utilizzando software open-source. Gli obiettivi a medio e lungo termine del nostro progetto risultano invece originali, in quanto non ancora indagati in dettaglio da altri autori.

3.2 Articoli di riferimento

In questa sezione si vogliono esaminare con maggiore attenzione due articoli che riteniamo particolarmente utili, per due diversi motivi. Nel caso dell'articolo di Zachov [16] sono interessanti sia l'originalità dell'approccio che la complessità dei risultati. Nell'altro caso invece (Zhao et al. [17]) risulta molto utile la possibilità di qualche confronto qualitativo¹ con i risultati che verranno presentati dal presente lavoro.

L'articolo di Zachov et al. [16], del 2009, presenta un approccio diverso da altri, più orientato alla computer graphic. É infatti focalizzato sulla visualizzazione dei risultati di una simulazione CFD di un modello realistico, sempre attraverso software commerciali, con circa 3.5 milioni di elementi. Il modello di flusso è non stazionario e turbolento (RANS) e la geometria comprende anche la zona del volto oltre alle cavità nasali. Consideriamo questo articolo attualmente lo stato dell'arte della simulazione del flusso all'interno del naso, sia per la complessità della geometria che della modellazione.

In figura 3.2 si riporta il modello ricostruito della cavità nasale. Si noti in particolare l'elevato livello di dettaglio e la scelta da parte degli autori di ricostruire dalle TAC anche il volto esterno. La risoluzione spaziale delle TAC a disposizione era quasi isotropica, pari a $0.37 \times 0.37 \times 0.4$ mm, e la ricostruzione 3D è stata ottenuta con il software Amira. Sono state provate diverse griglie e particolare attenzione è stata posta alla qualità delle *mesh* di volume.

Le simulazioni sono state portate a termine dal *software* Ansys CFX, per flusso laminare e turbolento (modello $k-\omega-SST$), calcolando diversi cicli respiratori. In figura 3.3 si riporta a titolo d'esempio un'immagine con le linee di flusso² nella cavità nasale, vista in prospettiva. La maggior parte del lavoro di Zachov e degli autori si concentra quindi sulle tecniche di visualizzazione dei numerosi dati ottenuti dalle simulazioni, tramite il programma SimVis. Il problema di come presentare i risultati è centrale

¹in particolare di alcune sezioni significative

²il cui colore, codificato, rappresenta il modulo della velocità

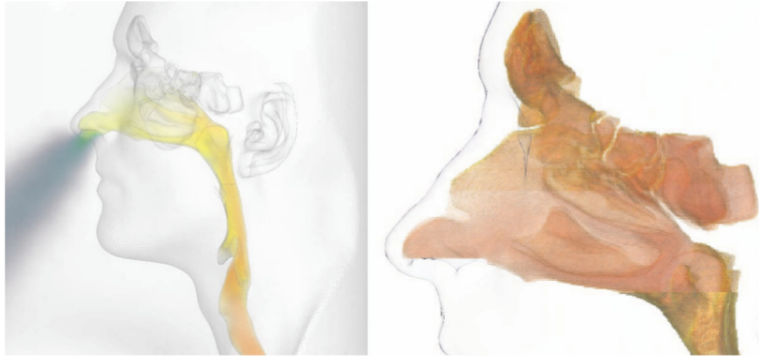


Figura 3.2: Modello del tratto respiratorio, tratto dall'articolo di Zachov et al. [16]. Nella figura sulla destra sono in evidenza i seni paranasali.

in tutte le simulazioni CFD e viene tipicamente risolto con visualizzazioni in prospettiva, in seguito si passa a visualizzazioni in coordinate parallele e da ultimo si applicano opportuni filtri e ingrandimenti nelle zone di interesse. Questo procedimento è quello indicato dal “mantra” di Shneiderman: *Overview first, zoom and filter, then detail-on-demand*.

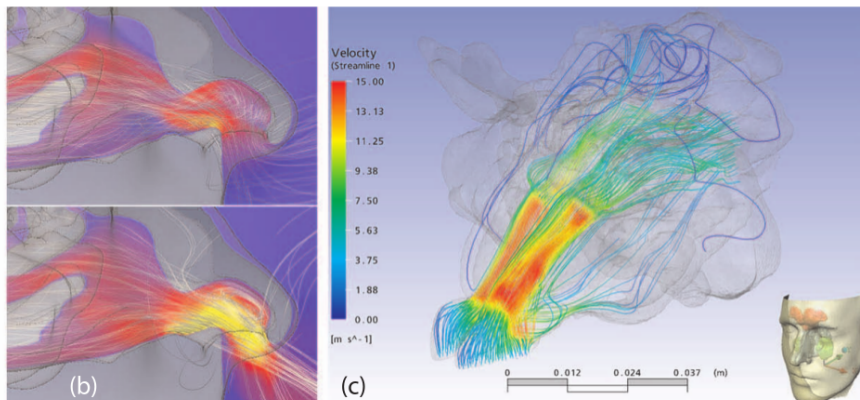


Figura 3.3: Velocità nel naso durante inspirazione (in alto a sinistra) ed espirazione (in basso a sinistra). A destra sono visualizzate le linee di flusso. Tratto da Zachov et al. [16].

L'articolo di Zhao et al. [17] mostra alcuni interessanti confronti tra diverse simulazioni CFD della cavità nasale. Gli autori sono partiti da

TAC di risoluzione $1.00 \times 0.39 \times 0.39$ mm e hanno simulato un'inspirazione stazionaria (imponendo la differenza di pressione tra le narici e la nasofaringe). Per quanto riguarda la modellazione numerica sono stati provati flussi laminari e turbolenti (RANS), in particolare con i modelli Spalart-Allmaras, $k - \epsilon$ e $k - \omega$, tutti risolti da Fluent. Non è specificato il numero di celle di calcolo, tuttavia sono stati provati 4 diversi Δp , pari a 30, 60, 100 e 160 Pa. Quest'ultimo valore corrisponde ad un respiro di media intensità ed è quello imposto anche in alcune simulazioni del presente lavoro di tesi.

L'articolo in esame risulta quindi più semplice del precedente (Zachov), sia per la geometria che per la complessità dei calcoli (qui il flusso è considerato stazionario), ma si possono trarre alcuni spunti.

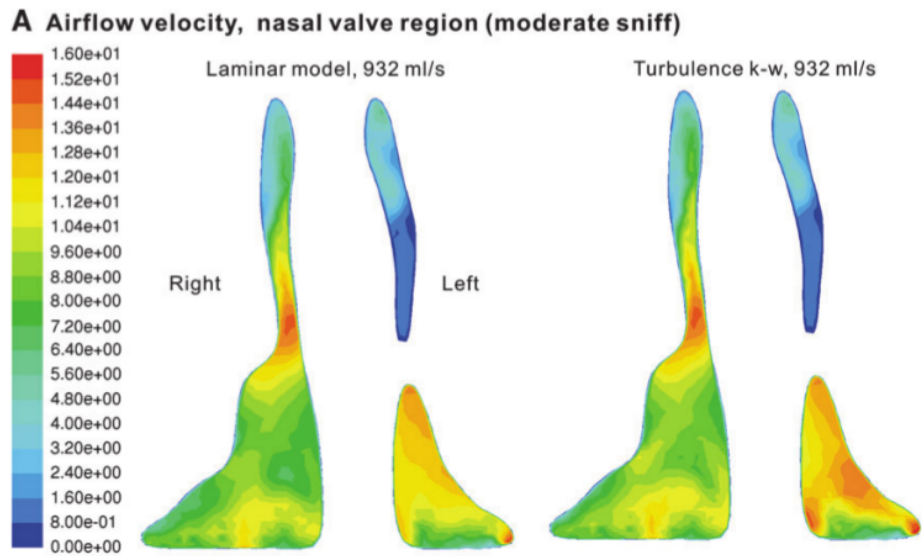


Figura 3.4: Linee di livello del modulo della velocità in una sezione coronale in corrispondenza della valvola nasale. La differenza di pressione è imposta a 160 Pa. Esempio tratto dall'articolo di Zhao [17].

Naturalmente il confronto tra i risultati che presenteremo e quelli di questo articolo di Zhao, a causa delle diverse geometrie, potrà essere solo qualitativo. In ogni caso questo ci permette di fare alcuni interessanti commenti, sulle velocità, le linee di flusso o il confronto tra i vari modelli delle simulazioni fluidodinamiche. Il primo esempio che riportiamo è in figura 3.4 e mostra le linee di livello del modulo della velocità (*contour plot*) in una sezione coronale in corrispondenza della valvola nasale. Il

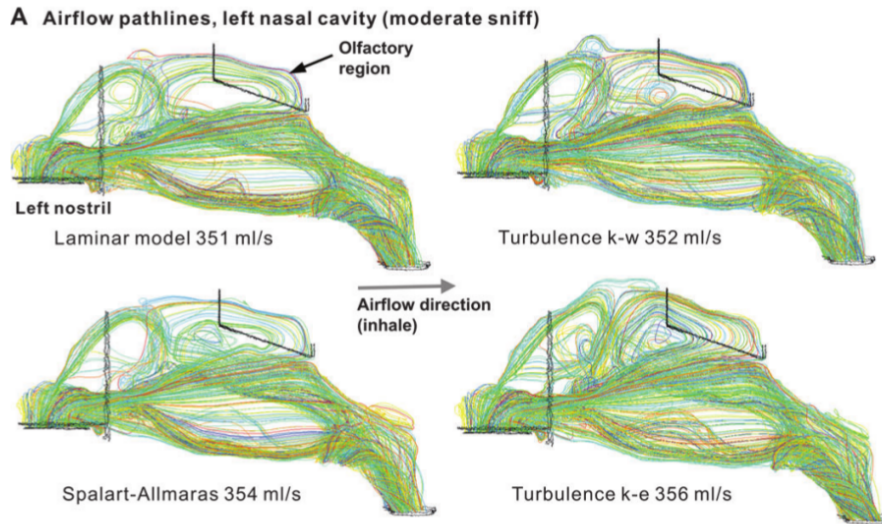


Figura 3.5: Andamento delle linee di flusso nella cavità nasale (in prospettiva) per diversi modelli di simulazione CFD, con differenza di pressione imposta a 160 Pa. Esempio tratto dall'articolo di Zhao [17].

secondo esempio (figura 3.5) mostra invece in prospettiva le linee di flusso, evidenziando in particolare l'andamento nella zona olfattiva a seconda dei diversi modelli di turbolenza scelti.

Parte II

Analisi

Capitolo 4

Dalla TAC alla ricostruzione geometrica

4.1 Introduzione alla TC

La tomografia computerizzata (CT in inglese) è da molti anni uno standard in molti ambiti della medicina. Si tratta di una tecnica di diagnostica per immagini, ottenute tramite radiazioni ionizzanti (raggi X), che riproducono fedelmente l'anatomia del paziente in esame, attraverso sezioni o strati (tomografia). L'aggettivo computerizzata si riferisce al fatto che l'ottenimento delle immagini viene gestito da un computer. Inoltre spesso si parla di TAC (tomografia assiale computerizzata), perchè inizialmente le immagini erano ottenute solo sul piano assiale; nelle TC moderne, grazie alla risoluzione quasi isotropica nelle 3 direzioni e grazie alla potenza di calcolo sempre più elevata, si ha la ricostruzione delle immagini anche nel piano sagittale e coronale.

In questa sezione si vogliono raccontare brevemente solo le informazioni utili per comprendere meglio il resto del progetto. Pertanto non ci si addenterà in eccessivi dettagli a riguardo, considerando che il punto di partenza del progetto sono le ricostruzioni TC già pronte, così come vengono fornite.

Il principio di base dei macchinari per le TC consiste nel generare un fascio di raggi X, che attraversano l'oggetto della scansione. Sul lato opposto alla sorgente dei raggi si trovano i sensori, che misurano l'attenuazione del fascio. L'attenuazione è proporzionale alla distribuzione spaziale degli elettroni nei tessuti attraversati. Una volta completata l'acquisizione si procede con la *slice* successiva. Alla fine di tutte le scansioni si devono generare le immagini (ricostruzione), in modo da visualizzare

il risultato di questo processo.

I *pixels* delle immagini ottenute dalla TC sono visualizzati con una scala di grigi in termini di radiodensità, che è la proprietà di trasparenza relativa di un materiale al passaggio di radiazioni. Un *pixel* corrispondente ad una zona di alta densità (es. le ossa) risulta molto chiaro (bianco, al limite), viceversa risulteranno neri i *pixel* dove le densità sono più basse. La scala di grigi (con 2001 diverse tonalità, dal nero al bianco) è quindi quantitativamente collegata alla densità elettronica, che si misura in HU (unità di Hounsfield). La definizione è:

$$HU = \frac{\mu_x - \mu_{water}}{\mu_{water} - \mu_{air}} \cdot 1000 \quad (4.1)$$

dove μ_{water} e μ_{air} sono i coefficienti di attenuazione dell'acqua e dell'aria, μ_x è l'attenuazione del materiale x che si sta valutando. L'aria corrisponde quindi a -1000 HU, mentre l'acqua distillata a pressione e temperatura standard è a 0 HU. Da questo risulta ad esempio che le ossa si trovano oltre i 400 HU, i muscoli intorno a 40 HU, i tessuti grassi intorno ai -120 HU. È utile ricordare queste soglie perchè rivestiranno un ruolo fondamentale quando si tratterà di ricostruire la geometria superficiale della cavità nasale con il *software* 3D Slicer.

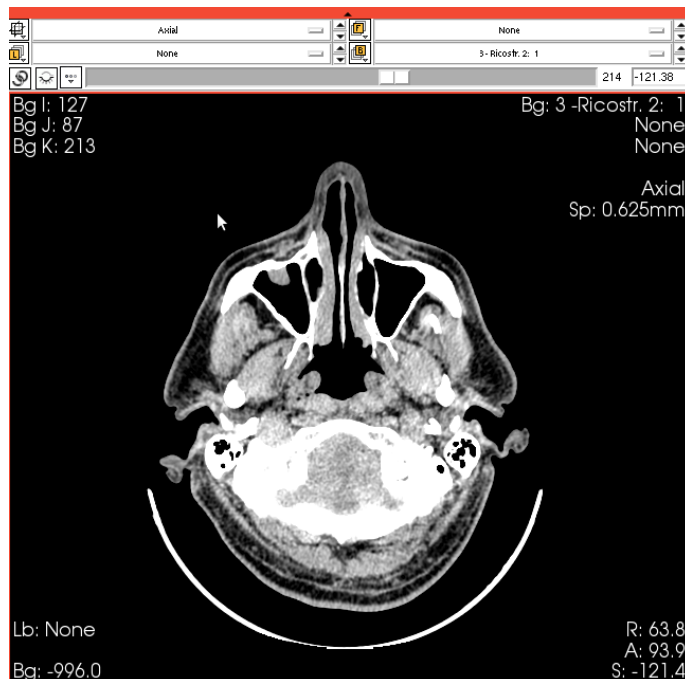


Figura 4.1: Immagine assiale di una TC

In figura 4.1 si riporta un'immagine assiale di ricostruzione TC. In questo caso di nostro interesse la TC riguarda il cranio del paziente, circa dalla zona della laringe (molto più in basso di quello che serve nel nostro studio) fino quasi alla sommità del capo. Sono state ricostruite 349 immagini assiali, spaziate di 0.625 mm, come si legge in alto a destra (Ricostr. 2). Ogni immagine è formata da 512×512 pixels. Questo consente una ricostruzione nel piano sagittale e nel piano coronale con 512 immagini spaziate di 0.494 mm. In basso a destra si leggono le coordinate di dove è posizionato il cursore, mentre in basso a sinistra è riportata la densità elettronica di quel punto: il valore di -996 HU (il cursore è proprio nella zona nera esterna al volto) conferma che stiamo puntando una zona di bassa densità, che corrisponde all'aria.

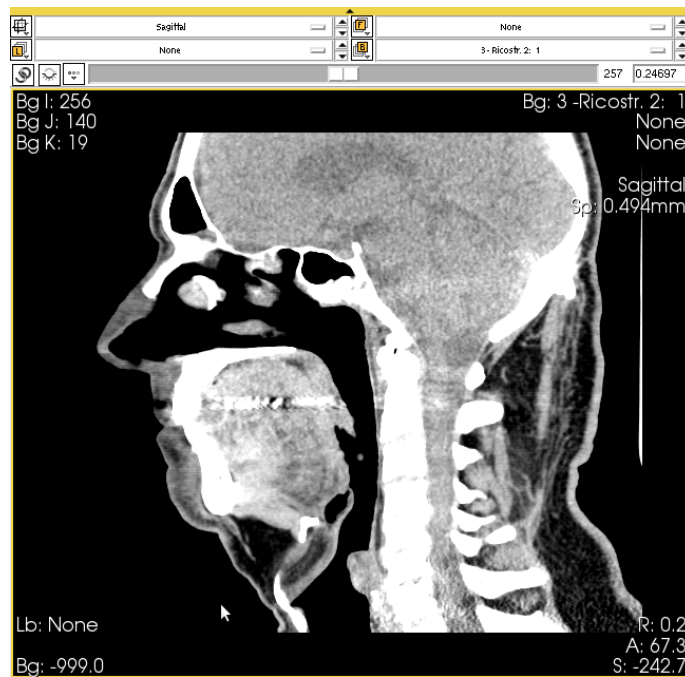


Figura 4.2: Immagine sagittale di una TC

Nelle figure 4.2 e 4.3 sono riportate anche due immagini negli altri due piani, quello sagittale e quello coronale. Valgono le stesse osservazioni già fatte in precedenza, in cui il cursore è ancora posizionato in un punto che corrisponde all'aria.

Concludiamo questa parte ricordando che un esame con TC provoca nel paziente una dose di radiazioni non indifferente, sconsigliabile se l'esame non è strettamente necessario. Una valida alternativa potrebbe essere un esame di risonanza magnetica (MR), che produce risultati

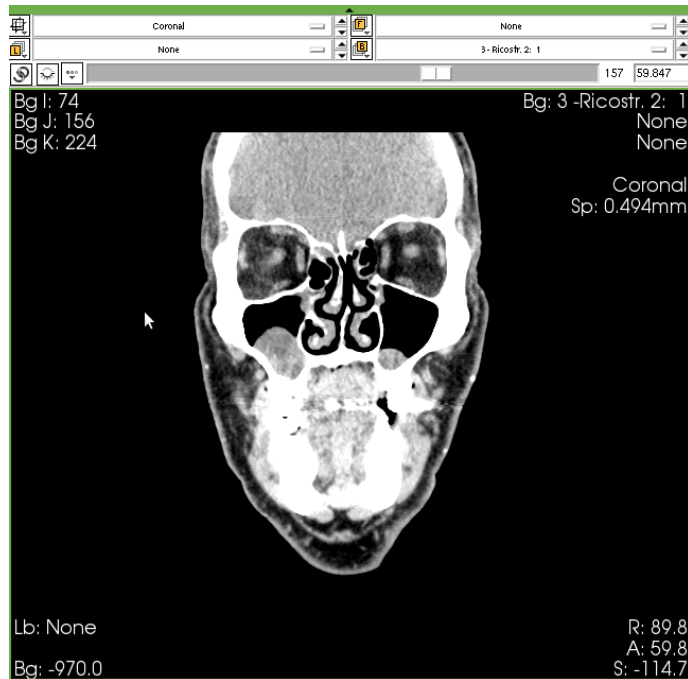


Figura 4.3: Immagine coronale di una TC

simili (in alcuni casi anche migliori, grazie all'elevato contrasto delle immagini ottenibili) attraverso principi fisici diversi, che non coinvolgono l'utilizzo di raggi X. Tuttavia nel nostro caso non interessa un contrasto migliore tra i vari tessuti del corpo e in generale l'esame con TC è più veloce, più economico e non richiede quasi mai di anestetizzare il paziente. Disponiamo anche di scansioni MR, tra gli sviluppi futuri si valuterà un confronto tra le due tecniche diagnostiche.

4.2 Utilizzo di 3D Slicer

3D Slicer è un programma *open-source* di *image processing* e visualizzazione tridimensionale di immagini mediche, ottenute tramite scansioni CT (tomografia computerizzata) o MR (risonanza magnetica). È sviluppato in collaborazione con importanti università e centri di ricerca in campo medico.

In questa sezione si vogliono illustrare i passaggi principali che portano dalle immagini delle scansioni alla ricostruzione della superficie della cavità nasale e del volto.

Questi passaggi vengono presentati solo nella loro forma definitiva, ma sono frutto di un lungo lavoro di continue prove e miglioramenti. Inoltre non rappresentano l'unico modo per raggiungere il risultato voluto, ma nascono dalla valutazione di diverse possibilità operative, con l'obiettivo di ottenere una procedura veloce, affidabile e robusta, che non richieda strumenti esterni.

3D Slicer, così come OpenFOAM, presenta una possibilità di utilizzo molto più ampia rispetto a quello che serve per proseguire con il nostro progetto; chi fosse interessato a più dettagli può visitare il sito www.slicer.org.

Inoltre, data la brevità di questa presentazione, non ci si soffermerà sui dettagli dei passaggi: verranno tutti elencati e discussi, ma per poterli riprodurre autonomamente serve una conoscenza di base dell'utilizzo del programma. La versione di *Slicer* utilizzata nel presente lavoro di tesi è la 3.6.3.

In estrema sintesi i passaggi fondamentali sono i seguenti:

- caricare la serie delle immagini CT;
- selezionare il volume di interesse¹ e ricostruirlo con precisione;
- generare la superficie corrispondente e salvarla.

Una volta aperto il programma si devono eseguire le seguenti operazioni:

1. si caricano i file DICOM della CT: *File* → *Add Volume* → *Apply*; si deve solo stare attenti al fatto che tipicamente ogni CT contiene più serie di dati, ad esempio per diverse ricostruzioni.

¹la cavità nasale e tutta la zona esterna al volto

2. Si cambia il modulo di *Slicer*, scegliendo l'Editor, in modo da poter modificare le immagini caricate: *Modules* → *Editor* → *Apply*. A questo punto ci si dovrebbe trovare in una situazione come quella riportata in figura 4.4.

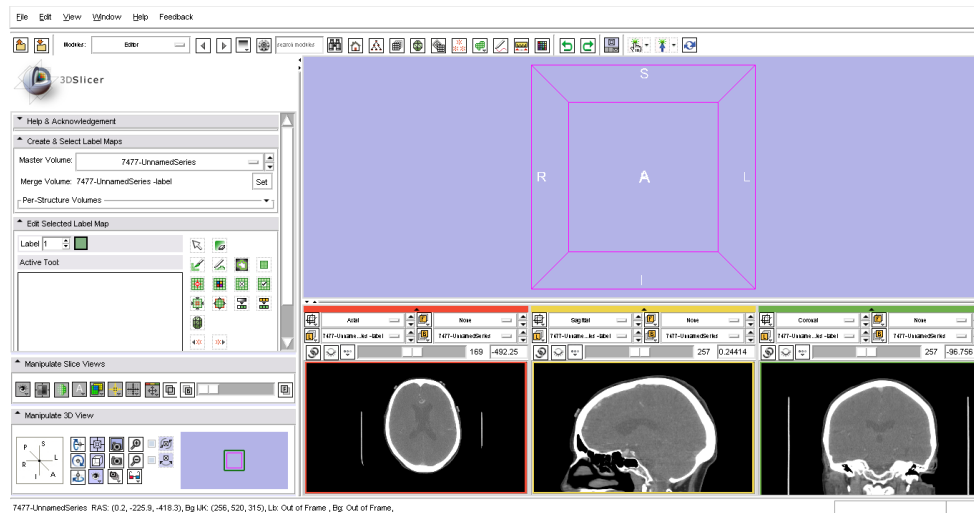


Figura 4.4: Interfaccia di 3D Slicer, dopo aver caricato le immagini CT e aver applicato il modulo Editor. A sinistra ci sono i menu a tendina con i comandi del modulo prescelto, a destra in basso ci sono le tre viste ortogonali (assiale, sagittale e coronale) della scansione e al di sopra la vista 3D, in cui da ultimo si potrà vedere la ricostruzione dei volumi e delle superfici.

3. Si tratta ora di affrontare l'unico passaggio delicato, ovvero la scelta della soglia. Questa soglia è la densità elettronica (misurata in HU) che è già stata introdotta nella sezione precedente. Concettualmente il funzionamento di questo passaggio è molto semplice: bisogna dare due soglie (minima e massima) e il programma ricostruisce il volume della CT la cui densità elettronica è nel *range* considerato. In questo modo, conoscendo i valori “ragionevoli” di queste soglie si può ricostruire qualsiasi organo, tessuto od osso all'interno delle scansioni che si sono caricate. Dato che a noi interessa ricostruire la cavità nasale e la zona esterna al volto, siamo interessati in pratica a ricostruire il volume d'aria, quindi tutte le zone a densità minore. Si potrebbe quindi impostare la soglia inferiore al valore minimo possibile e aggiustare l'altra in modo da ottenere il volume cercato, ma per ragioni pratiche risulta più

comodo agire esattamente al contrario: ricostruiamo allora tutto il volume del cranio tranne l'aria e alla fine invertiamo questa selezione (passaggio 8) per riottenere quanto cerchiamo. Perciò impostiamo la soglia inferiore al livello giusto per includere tutti i tessuti e le ossa del cranio (es. -200 HU) e lasciamo la soglia superiore al livello massimo possibile (pulsante *threshold*). Cliccando su *Apply* ci si ritrova in una situazione come in figura 4.5.

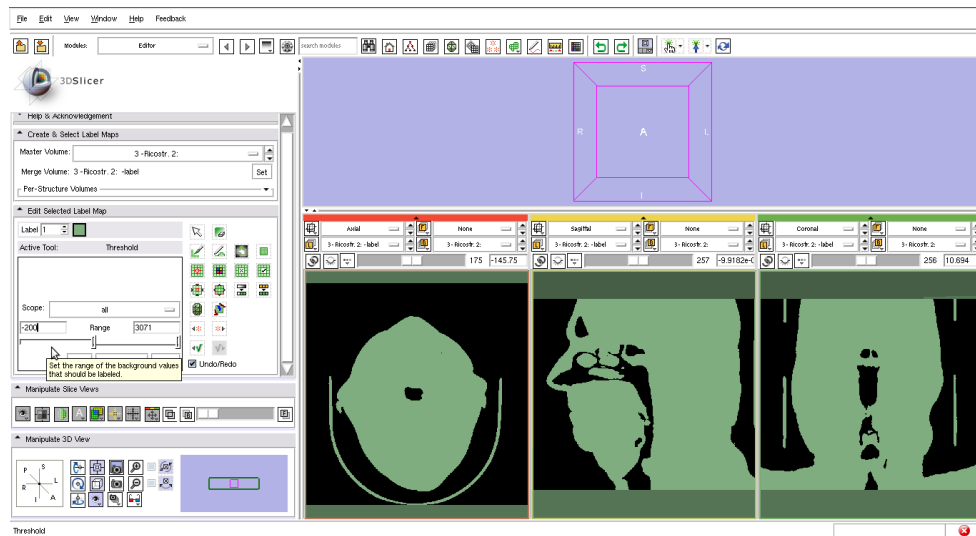


Figura 4.5: Scelta della soglia in 3D Slicer. È in evidenza (in colore verde) la parte selezionata, ovvero tutto il volume del cranio tranne il volume d'aria (cavità nasale e zona esterna). Invertendo questa selezione si otterrà il volume cercato.

4. Per evitare di ricostruire anche il sostegno della testa (l'oggetto semicircolare in sezione assiale, si vede bene in figura 4.5) ci si deve portare (in vista assiale) nella sezione di contatto tra la testa e il sostegno e con lo strumento *Paint* "cancellare" il sostegno (scegliendo un colore diverso dal precedente con il pulsante *Label*). È un passaggio simile al 7, che verrà descritto con più dettagli.
5. Sempre in vista assiale, ci si porta nella *slice* più alta e con lo strumento *LevelTracing* si esclude la testa. Questo passaggio² servirà poi ad avere una superficie (cavità nasale e zona esterna) unica e chiusa in alto. Per maggiore chiarezza si veda la figura 4.6.

²ovviamente non serve se la CT arriva oltre la fine del capo. In tal caso infatti nella *slice* più alta non c'è la testa, è tutto nero.

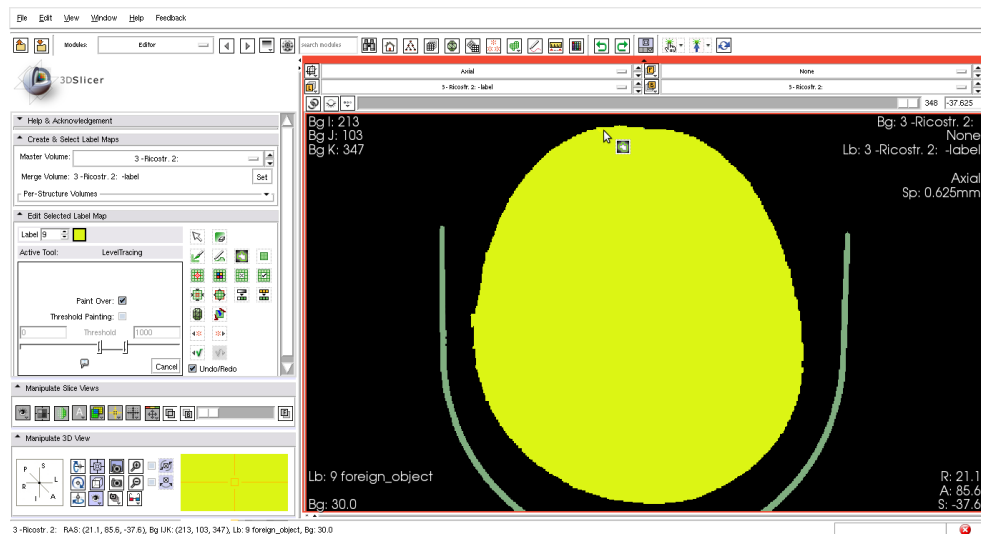


Figura 4.6: Passaggio di esclusione della testa in 3D Slicer. L'area esclusa, con il comando *Threshold*, è quella gialla.

6. In un'altra slice qualsiasi scegliere *SaveIsland* e cliccare su un punto con il primo colore utilizzato (di *default* è il verde). Le armature tornano bianche da sole.
7. Con un procedimento simile ai passaggi 4 e 5 bisogna escludere la zona interna alle orecchie (altrimenti verrebbe ricostruita). Ci si porta quindi nelle 2 sezioni sagittali opportune e si utilizza ancora il comando *LevelTracing*. In figura 4.7 si riporta un'immagine di questo passaggio.
8. Si cambia colore (sempre con il *Label*), si sceglie *ChangeIsland* e si clicca in un punto esterno alla testa (la zona nera, che corrisponde all'aria). Ci si dovrebbe ritrovare come in figura 4.8.
9. Scegliere *SaveIsland* e cliccare su un punto con l'ultimo colore scelto (zona esterna).
10. Cliccare su *MergeAndBuild* e poi salvare la superficie ricostruita (*File* → *Save*). Si scelga come formato *.stl*, lo stesso che viene importato da OpenFOAM per generare la griglia di calcolo. Si dovrebbe anche vedere nella finestra delle immagini 3D la ricostruzione finita, come in figura 4.9.

Concludiamo questa parte con qualche osservazione. Il tempo richiesto da questi passaggi, una volta che si conosce il software, non è quasi

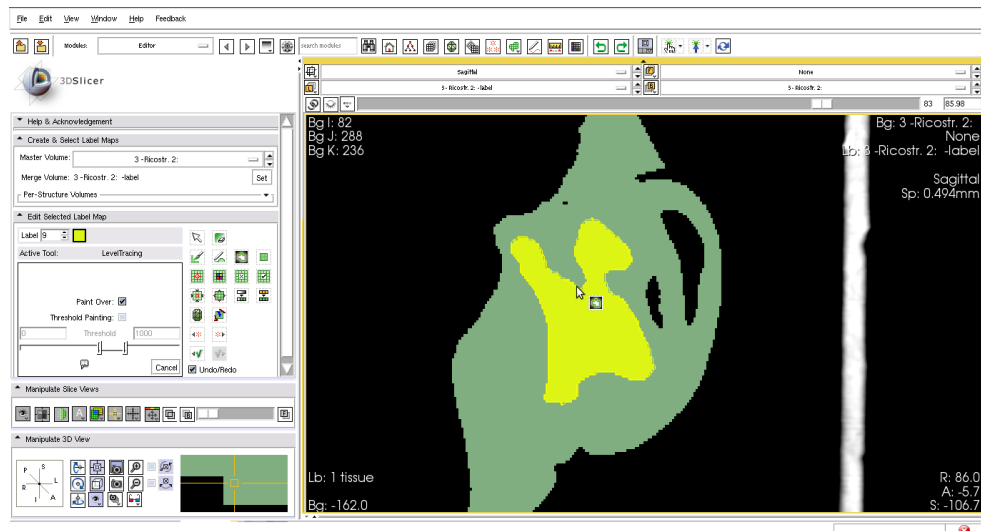


Figura 4.7: Passaggio di esclusione delle cavità auricolari in 3D Slicer. L'area esclusa, con il comando *Threshold*, è quella gialla.

mai superiore ai 10 minuti. È chiaro che questa procedura è difficilmente automatizzabile, anche perchè richiede un controllo visivo e alcune decisioni da prendere, però è accettabile come tempo di lavoro.

L'unico parametro critico resta la soglia inferiore³, che condiziona direttamente la geometria finale della superficie. Tuttavia, come si mostrerà nell'analisi di sensibilità dei risultati, non è troppo influente sui calcoli CFD, se si considera un'incertezza di 10-20 HU. Siamo inoltre convinti che questo problema diventerà sempre meno significativo in futuro, grazie all'aumento della risoluzione delle scansioni CT.

³è un parametro che ha un minimo di arbitrarietà e può essere diverso per diverse CT

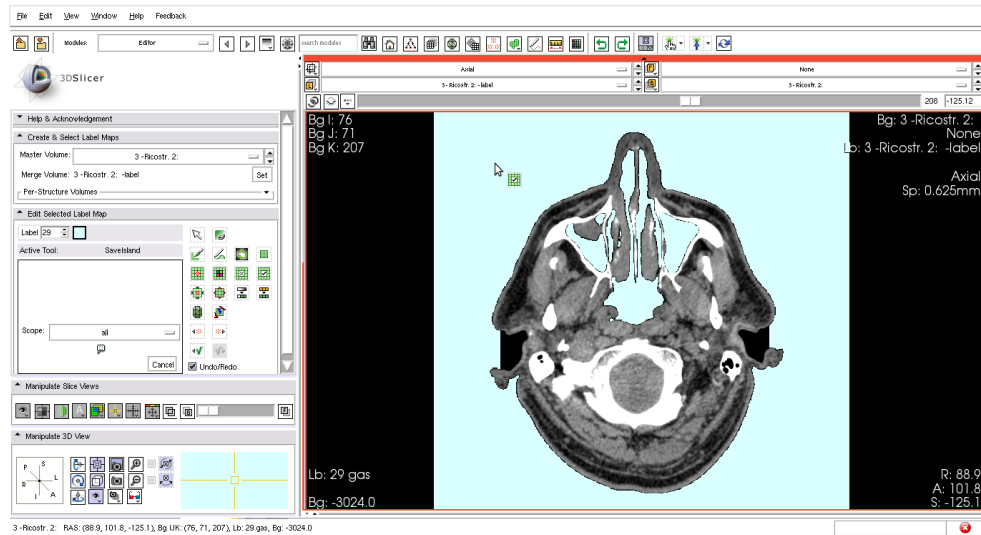


Figura 4.8: Selezione volume finale in 3D Slicer. Il volume scelto, di cui poi si salverà la superficie, è in azzurro.

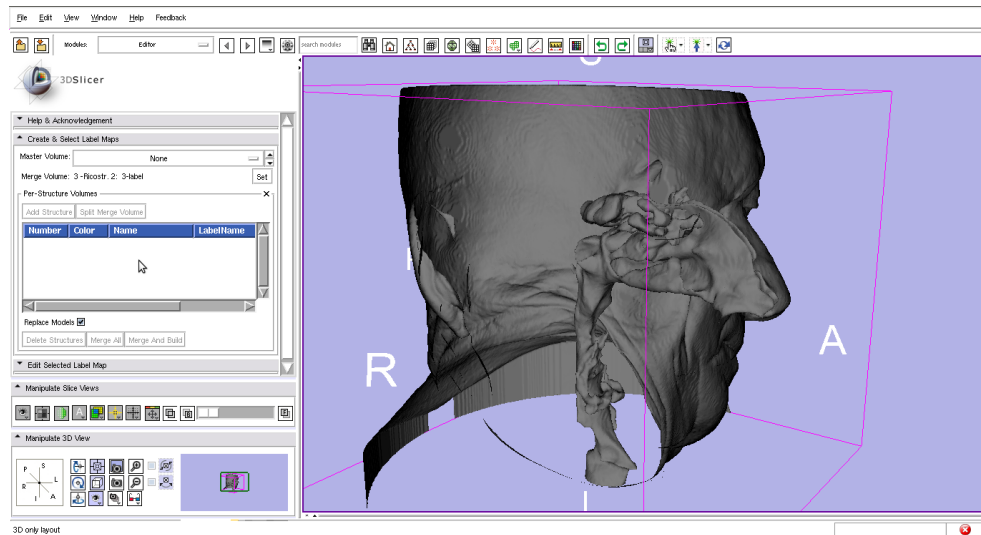


Figura 4.9: Ricostruzione superficie in 3D Slicer.

Capitolo 5

Verifica della geometria

5.1 Il *Software* MeshLab

MeshLab è un programma *open-source* sviluppato dal dipartimento di *Computer Science* dell'Università di Pisa. L'utilizzo di tale programma è dedicato alla modifica (*processing & editing*) di mesh 3D triangolari non-strutturate, proprio quelle di cui si dispone dopo la ricostruzione dalle CT effettuata con 3D Slicer.

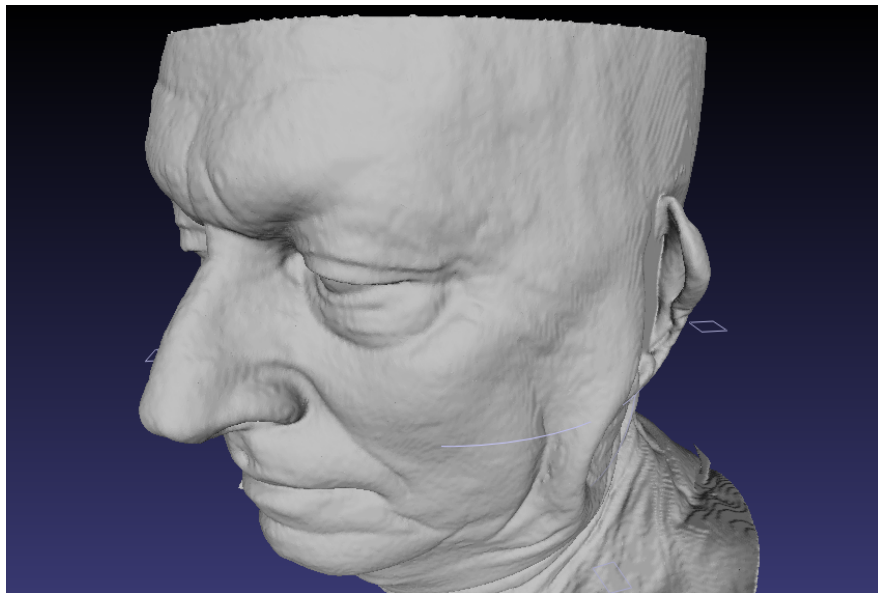


Figura 5.1: Ricostruzione superficie in MeshLab.

In realtà l'utilizzo di MeshLab non è strettamente necessario e la maggior parte delle mesh di volume che sono state create per i calcoli

fluidodinamici sono state ottenute dai file .stl di Slicer senza alcuna modifica. Si può quindi saltare completamente questo passaggio, velocizzando la procedura ed eliminando questo ulteriore lavoro per l'utente.

Riteniamo tuttavia importante dedicare un po' di spazio a MeshLab per due motivi: innanzitutto risulta sempre utile verificare¹, anche solo visivamente, che le mesh superficiali siano corrette. Se questo non fosse verificato si può in seconda battuta procedere con l'applicazione di uno o più filtri di modifica, con l'obiettivo di riparare gli errori o migliorare la geometria esistente (es. smoothing). I filtri presenti in MeshLab sono diverse decine e uno studio più approfondito sull'utilizzo di alcuni di essi nel presente progetto potrebbe rappresentare un possibile sviluppo futuro. Nella sezione successiva ne presentiamo 2, che sono stati provati con successo.

Il secondo motivo è che in futuro si potrà avere bisogno di un programma per il *mesh editing*, quando si dovranno simulare numericamente gli interventi chirurgici.

Per capire meglio come funziona MeshLab presentiamo qualche immagine significativa. In figura 5.1 si vede la stessa ricostruzione del capitolo 4. La testa è superiormente chiusa da un piano, mentre inferiormente è aperta. Si noti che la testa è troncata in alto perchè quella particolare ricostruzione CT finisce in quel punto; se la CT fosse stata effettuata fino alla sommità del capo si avrebbe a questo punto tutta la geometria senza semplificazioni. In ogni caso tutta la cavità nasale è presente e possiamo ipotizzare fin da subito che questa modifica della geometria del capo non influenzerà la fluidodinamica della respirazione.

Da questa prospettiva non si vede la cavità nasale, che quindi riportiamo in figura 5.2. Il piano che chiude in alto la testa non è visualizzato, in modo tale da mostrare appunto l'interno della testa con la cavità nasale. Il fondo della testa, circa nella zona del collo, è invece aperto già dalla ricostruzione di 3D Slicer. In questa vista dall'alto si notano con chiarezza i seni paranasali e si vede che mancano i seni frontali. Questa mancanza è dovuta ad una forte sinusite del paziente in esame. Riportiamo in figura 5.3 un'altra prospettiva della cavità nasale, che permette di apprezzare tutta la complessità della geometria nasale, fino circa alla zona della laringe. Non dovrebbe essere difficile riconoscere le strutture di cui si è parlato sempre nel capitolo 4.

¹con un altro software, diverso da quello che ha creato le mesh superficiali

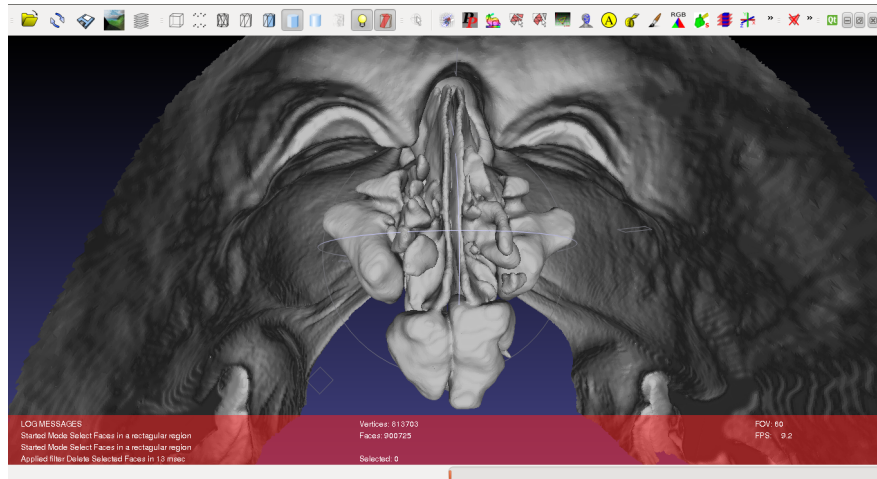


Figura 5.2: Cavità nasale in MeshLab, vista dall'alto.

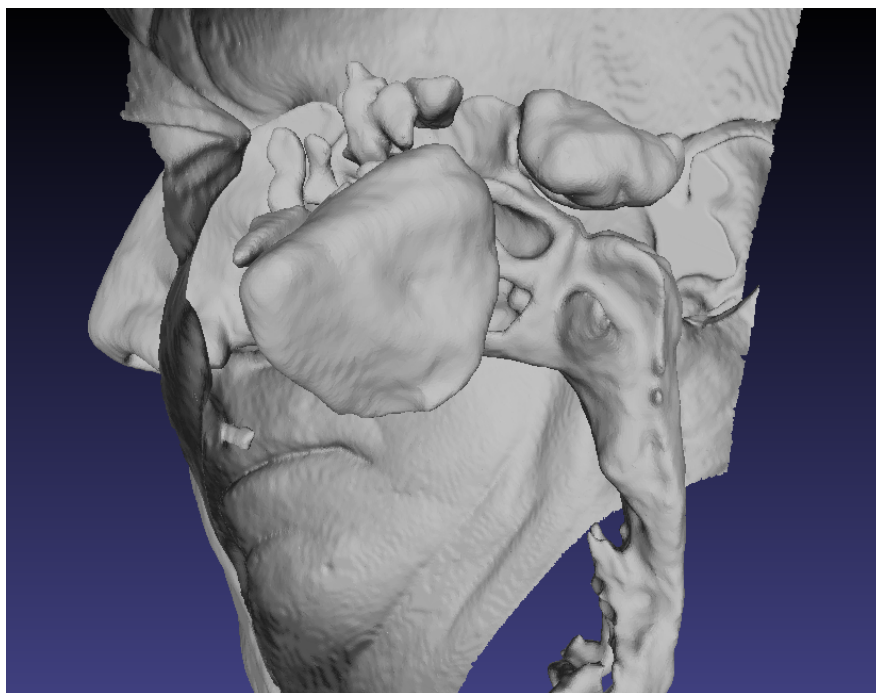


Figura 5.3: Cavità nasale in MeshLab, vista dal lato sinistro circa all'altezza degli occhi. Tutta la parte posteriore del cranio è stata nascosta per poter visualizzare la cavità nasale da questa prospettiva.

5.2 Miglioramento della mesh

Riportiamo, più che altro a titolo d'esempio, due possibili filtri per sistemare e migliorare la mesh. Naturalmente non sono gli unici né tantomeno i migliori, però sono un punto d'inizio per un'indagine più accurata. Anche per MeshLab, così come per Slicer, si procede per tentativi, valutando singolarmente l'utilità di ogni filtro.

Il primo dei due (*Fill Hole*), in figura 5.4, talvolta può rivelarsi davvero necessario. Il suo scopo è quello di controllare se nella mesh superficiale sono presenti dei buchi e quindi si può procedere al riempimento² secondo opportuni criteri. Per motivi che saranno chiari in seguito, è importante che non ci siano buchi, eccetto che nella parte inferiore della geometria ricostruita (il fondo della testa). Questo richiama è dovuta alla logica di funzionamento della libreria di OpenFOAM che si occupa della creazione della mesh di volume a partire dalla superficie .stl. Se ci fossero dei buchi³ di dimensioni non infinitesime sicuramente si avrebbe il fallimento dei passaggi successivi verso la simulazione CFD.

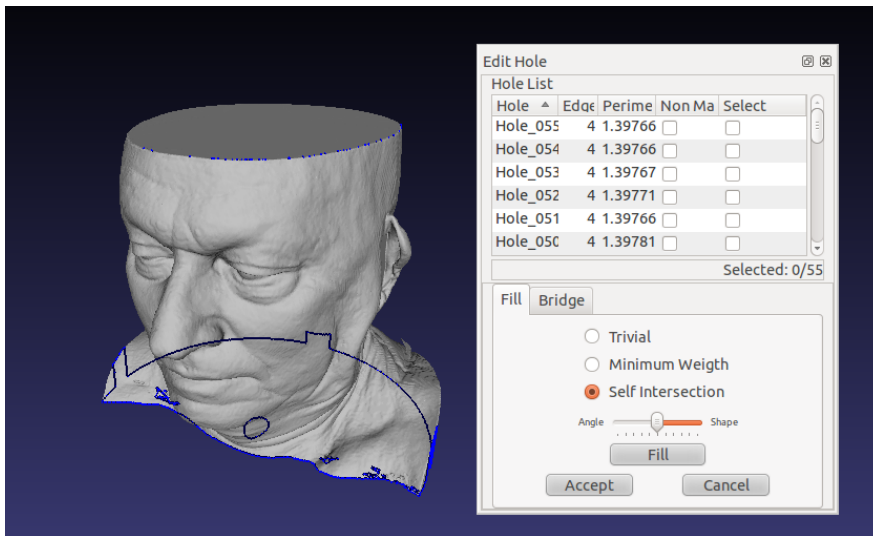


Figura 5.4: Filtro *Fill Hole* di MeshLab.

Il secondo filtro che presentiamo è in figura 5.5. Si tratta di uno degli innumerevoli filtri di *smoothing*, che servono a migliorare la qualità della mesh. Questi filtri non sono assolutamente necessari, però potrebbero anch'essi rivelarsi utili qualora si volesse una geometria di partenza

²di alcuni o di tutti

³che tra l'altro fisicamente non avrebbero senso

più regolare, con una definizione migliore. Questo potrebbe avere un'influenza positiva nel passaggio alla mesh di volume, portando ad una maggiore affidabilità dei risultati numerici, a patto di ridurre l'arbitrarietà dell'applicazione di questi filtri⁴.

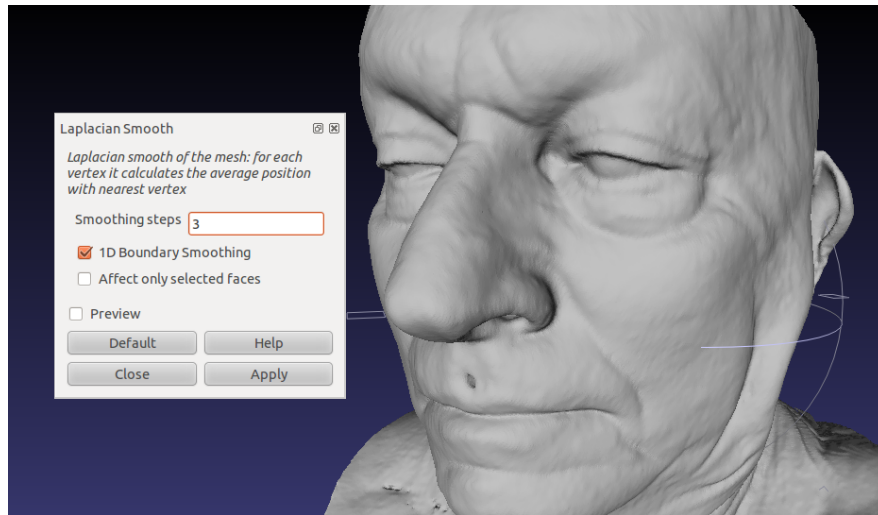


Figura 5.5: Filtro *Laplacian Smooth* di MeshLab.

⁴i filtri di *smoothing* hanno diverse logiche di funzionamento e richiedono la scelta di alcuni parametri

Capitolo 6

Introduzione a OpenFOAM

6.1 Presentazione di OpenFOAM 1.7

OpenFOAM, come suggerisce il nome stesso, è un software open-source, ovvero con licenza pubblica e gratuita. L'acronimo FOAM significa Field Operation And Manipulation, riferendosi alla capacità di gestire strutture di dati, talvolta molto complesse, di campi scalari, vettoriali o tensoriali.

OpenFOAM non è un vero e proprio software, è innanzitutto una libreria scritta in C++ che crea eseguibili chiamati *applications*. Il suo scopo è quello di risolvere equazioni differenziali alle derivate parziali attraverso l'approccio dei volumi finiti; si tratta quindi di un software con potenzialità molto più ampie rispetto alla soluzione di problemi fluido-aerodinamici, che restano comunque l'obiettivo principale della stragrande maggioranza dei solutori già pronti.

Gli eseguibili si dividono in due categorie: i solutori (*solvers*) e le utilità (*utilities*). I solutori servono a risolvere uno specifico problema di meccanica dei continui, mentre le utilità svolgono moltissimi compiti dal pre- al post-processing, alla creazione/manipolazione della mesh e altro ancora. Il punto di forza di OpenFOAM consiste nel mettere a disposizione degli utenti tutto il codice sorgente, con la possibilità di apportare modifiche a proprio piacimento, fino a crearsi dei solutori personalizzati, mantenendo però tutta la struttura già impostata. In questo modo, con alcune basi di programmazione¹ è possibile creare nuovi *solvers* e *utilities* in poco tempo, rispetto a quanto, per assurdo, richiederebbe lo sviluppo da zero di un *software* dedicato.

¹oltre che naturalmente la conoscenza della fisica e delle tecniche numeriche opportune per risolvere il problema che si sta affrontando

OpenFOAM ha una struttura complicata, che però può essere schematizzata in estrema sintesi come in figura 6.1. Si noti che nella fase di post-processing è possibile l'utilizzo di software terzi, tra cui il più utilizzato (e quello scelto anche nella presente tesi) è Paraview, che dialoga con OpenFOAM attraverso la *utility* paraFoam.

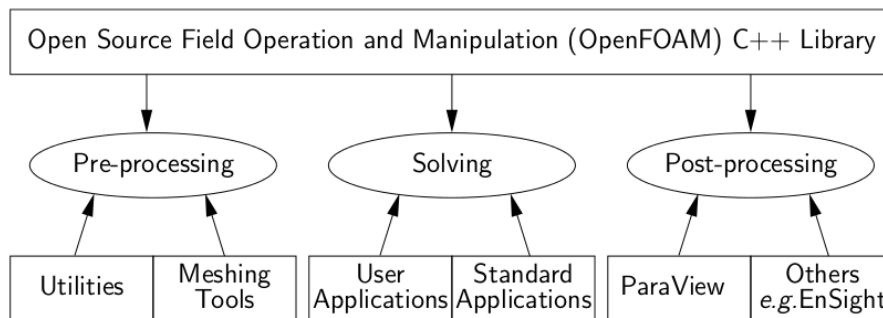


Figura 6.1: Struttura di OpenFOAM

OpenFOAM è totalmente scritto in C++ e questo consente una programmazione definita *object-oriented*, che consente di dichiarare tutte le grandezze fisiche e le relative operazioni in un unico oggetto, chiamato *classe*, in modo tale che la scrittura delle equazioni da risolvere sia analoga al linguaggio verbale e matematico usato per descriverle. Per fare un esempio più concreto si pensi al campo di velocità, indicato con U . La velocità è un campo vettoriale, definito da una classe *vectorField*, mentre il modulo della velocità è $\text{mag}(U)$.

Il fatto di avere oggetti nella programmazione che rappresentano oggetti fisici o entità astratte rende il codice molto più compatto e comprensibile, concentrando la dichiarazione delle classi in una parte limitata del codice ed evitando inutili ripetizioni. Inoltre risulta piuttosto facile modificare o creare nuove classi da quelle già esistenti.

Queste proprietà della programmazione a oggetti possono essere riassunte nel seguente elenco:

- **incapsulamento:** indica il fatto che un oggetto contiene al suo interno tutte le proprietà e le procedure. L'accesso ai dati incapsulati è possibile solo attraverso i metodi accessibili dall'esterno, in modo tale che dal punto di vista di chi utilizza l'oggetto le operazioni interne possano restare completamente nascoste. Perciò di ciascun oggetto, che lavora solo con i propri dati, è noto cosa fa e come interagisce con l'esterno, ma non le operazioni interne.

- Ereditarietà: è un meccanismo che permette di derivare nuove classi a partire da altre classi già definite, dalle quali eredita i metodi e gli attributi. Naturalmente sono anche possibili ulteriori modifiche, sia per aggiungere nuovi metodi che per modificare quelli esistenti. In questo modo molte parti del codice possono essere riutilizzate, portando a numerosi vantaggi tra cui la compattezza del codice e il risparmio di tempo nella programmazione.
- Polimorfismo: in un linguaggio *object-oriented* lo stesso codice può essere utilizzato con classi diverse, portando a risultati diversi a seconda del contesto di utilizzo. In tal modo non serve definire molte funzioni che, pur svolgendo lo stesso compito, differiscono per la definizione degli input/output.

6.2 Struttura di OpenFOAM 1.7

Senza scendere in ulteriori dettagli di programmazione, passiamo ad una breve descrizione del procedimento da seguire per risolvere un problema fluidodinamico e dei principali solutori e utilità utilizzati. Innanzitutto si deve creare una cartella con il caso da analizzare. Tale cartella, come si vede nell'esempio di figura 6.2, contiene al suo interno alcune sottocartelle, con le informazioni sulla griglia, il solutore e i metodi numerici da utilizzare, le condizioni iniziali e al contorno e da ultimo, mentre i calcoli procedono², vengono create le sottocartelle³ con i risultati parziali e finali.

Sempre con riferimento alla figura 6.2, analizziamo un esempio tipico. All'interno della cartella *system* trovano spazio i dizionari che definiscono il problema da risolvere e i metodi numerici. I 3 dizionari fondamentali⁴ sono il *controlDict*, *fvSchemes* e *fvSolution*. Spesso ne sono presenti molti altri, per il *pre-processing* o per altre operazioni intermedie sulla mesh di volume, di cui parleremo ampiamente quando si tratterà di utilizzarli. Per ora limitiamoci a descrivere questi. Il dizionario *controlDict*, in un semplice esempio introduttivo, si presenta come segue:

```
application      icoFoam;

startFrom        startTime;
```

²una volta invocato il solutore da riga di comando

³il cui nome è l'istante di tempo a cui è stata calcolata la soluzione

⁴quelli che non possono mancare

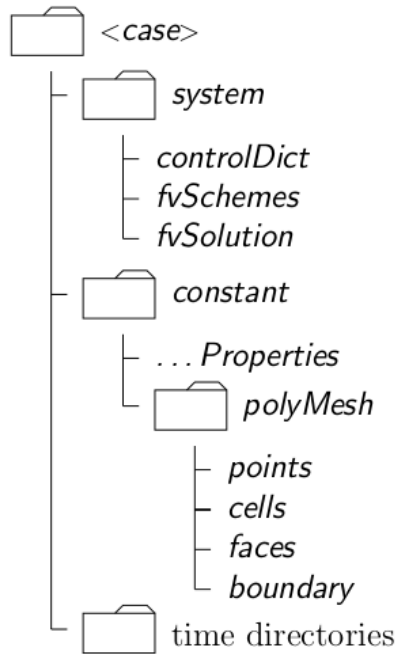


Figura 6.2: Cartella e sottocartelle di un caso di OpenFOAM

```
startTime      0;
stopAt         endTime;
endTime        0.5;

deltaT         0.005;

writeControl   timeStep;
writeInterval  20;
purgeWrite     0;

writeFormat    ascii;
writePrecision 6;
writeCompression uncompressed;

timeFormat     general;
timePrecision  6;
runTimeModifiable yes;
```

Per chi fosse interessato a tutti i dettagli, comprese le numerose alternative, rimandiamo alla guida di OpenFOAM; spieghiamo brevemente

i punti fondamentali. *IcoFoam* è il solutore di questo esempio⁵, relativo ad un calcolo incomprimibile non-stazionario di un flusso laminare per fluidi Newtoniani. Gli ingressi relativi alla parte temporale della simulazione non hanno bisogno di spiegazioni (tempo iniziale e finale, il *deltaT* è quello iniziale, che può essere aggiustato dal solutore in funzione del numero di CFL). Seguono quindi le scelte dei parametri per il salvataggio delle soluzioni intermedie (da *writeControl* a *purgeWrite*) e i dettagli di come devono essere salvati i risultati (formato *binary* o *ascii*, precisione e compressione). Lo stesso per la variabile temporale.

Il dizionario *fvSchemes* è invece articolato come segue:

```

ddtSchemes
{
    default          Euler;
}

gradSchemes
{
    default          Gauss linear;
    grad(p)          Gauss linear;
}

divSchemes
{
    default          none;
    div(phi,U)       Gauss linear;
}

laplacianSchemes
{
    default          none;
    laplacian(nu,U)   Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
}

interpolationSchemes
{
    default          linear;
    interpolate(HbyA) linear;
}

```

⁵tratto dal primo tutorial di OpenFOAM

```
}  
  
snGradSchemes  
{  
    default          corrected;  
}  
  
fluxRequired  
{  
    default          no;  
    p                ;  
}
```

Ci troviamo ovviamente in un esempio molto semplice, ma la struttura è sempre questa: cambiano solo i metodi numerici utilizzati. Dal punto di vista di un utente esterno non serve entrare nei dettagli di ogni schema numerico e spesso non serve neanche decidere quali schemi utilizzare, perchè una volta scelto il caso da analizzare (l'*application*, in questo caso è *icoFoam*) basta copiare i dizionari relativi.

Si noti comunque che vanno specificati gli schemi per tutti gli operatori matematici delle equazioni risolte dal caso in esame. Dall'alto si hanno perciò gli schemi dell'integrazione temporale, il calcolo dei gradienti e della divergenza, gli schemi per il laplaciano, quelli per l'interpolazione, lo schema per calcolare la componente del gradiente normale alla faccia di una cella (*snGradSchemes*)⁶ e lo schema per calcolare i flussi. Per ciascuno di essi si può inoltre specificare un diverso schema a seconda della grandezza fisica a cui viene applicato l'operatore.

Il dizionario *fvSolution* si presenta invece come segue:

```
solvers  
{  
    p  
    {  
        solver          PCG;  
        preconditioner  DIC;  
        tolerance       1e-06;  
        relTol          0;  
    }  
  
    U
```

⁶*surface normal Gradient Schemes*


```

    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }
}

PISO
{
    nCorrectors        2;
    nNonOrthogonalCorrectors 0;
    pRefCell           0;
    pRefValue          0;
}

```

Questo dizionario serve a definire il tipo di solutore da usare nelle equazioni. Dato che siamo nel caso `icoFoam`, si devono risolvere le equazioni nelle incognite velocità e pressione. Il solutore per la pressione è un gradiente coniugato pre-condizionato, mentre per la velocità è un gradiente bi-coniugato preconditionato, con le relative tolleranze. Il sub-dizionario PISO (*pressure-implicit split-operator*) è relativo al fatto che le equazioni vengono risolte con tecniche iterative e bisogna specificare il numero dei correttori da utilizzare.

Passiamo quindi a descrivere la cartella *constant*, che a sua volta contiene una sottocartella chiamata *polyMesh* e alcuni dizionari del tipo *...Properties*. Nella cartella *polyMesh* sono contenuti i file di descrizione della mesh di volume (strutturata o meno), generati dalle *utilities* di *pre-processing*. Questi file contengono l'elenco di tutti i punti, le celle, le facce e i contorni del dominio di calcolo. Non trattiamo in questa parte la creazione della mesh perchè è uno dei punti critici del presente lavoro di tesi, a cui dedichiamo per intero il capitolo 8.

I dizionari, nel caso più generale di flusso turbolento, sono invece *transportProperties*, *turbulenceProperties* e *RASProperties*. Nell'esempio che stiamo presentando il flusso è obbligatoriamente laminare e pertanto è presente solo il primo dei tre, che si presenta così:

```
nu          nu [ 0 2 -1 0 0 0 0 ] 1.5e-05;
```

ovvero contiene la definizione delle costanti di cui si ha bisogno, in questo caso solo la viscosità dinamica. Le parentesi quadre servono a stabilire l'unità di misura della grandezza in questione: la prima cifra indica

l'esponente della massa, la seconda indica l'esponente della lunghezza, la terza l'esponente del tempo e così via per le 7 unità di misura fondamentali del sistema internazionale. Perciò stiamo dicendo che la viscosità dinamica è in $\frac{m^2}{s}$. È opportuno sottolineare che i calcoli di OpenFOAM sono svolti con quantità dimensionali.

Il dizionario *turbulentProperties*⁷ risulta invece:

```
simulationType RASModel;
```

in esso si può solo scegliere se la simulazione è una RANS (*Reynolds averaged Navier-Stokes*) o una LES (*Large eddy simulation*), ovviamente per i solutori dei flussi turbolenti. Una breve descrizione delle equazioni della fluidodinamica, con i modelli di turbolenza, verrà presentata nel capitolo 7.

Se siamo nel caso turbolento delle RANS è anche necessario il dizionario *RASProperties*:

```
RASModel      kEpsilon;  
turbulence     on;  
printCoeffs    on;
```

in esso si specifica il modello di turbolenza tra tutti quelli possibili nel caso delle equazioni di Navier-Stokes mediate alla Reynolds⁸, in questo caso è un modello $k - \epsilon$ che aggiunge due equazioni differenziali. Si può anche forzare la soluzione laminare scrivendo *RASModel laminar* e *turbulence off*.

Per poter iniziare la simulazione resta solo da creare la cartella denominata "0", ovvero quella che descrive, al tempo iniziale appunto, le condizioni al contorno (fisse o variabili) e quelle iniziali. All'interno di questa cartella devono essere presenti tanti file quante sono le incognite del problema⁹. Tornando all'esempio semplice che è stato presentato finora, si hanno quindi 2 file, uno per la pressione e uno per la velocità.

Quello della pressione è:

```
dimensions     [0 2 -2 0 0 0 0];
```

⁷che a rigore non è presente nei casi del tipo icoFoam, però lo presentiamo lo stesso per completezza

⁸la maggior parte dei modelli di turbolenza sono già implementati in OpenFOAM, altrimenti si possono aggiungere

⁹le incognite possono essere scalari, vettori o tensori: la velocità quindi, in questo bilancio, è una sola incognita

```

internalField    uniform 0;

boundaryField
{
    movingWall
    {
        type      zeroGradient;
    }

    fixedWalls
    {
        type      zeroGradient;
    }

    frontAndBack
    {
        type      empty;
    }
}

```

L'ingresso *dimensions* è analogo a quanto è già stato spiegato per la viscosità nel dizionario *transportProperties*. In questo caso ci stiamo riferendo alla pressione, che però non è misurata in Pa ($\frac{N}{m^2}$), ma in $\frac{m^2}{s^2}$ (come si evince tra le parentesi quadre) perchè siamo in un caso incomprimibile e quindi la pressione è già stata divisa per la densità. Il campo *internalField* specifica le condizioni iniziali, in questo caso uniformi e nulle. Da ultimo il sub-dizionario *boundaryField* contiene le definizioni delle condizioni al contorno: in questo particolare esempio si ha che il contorno è suddiviso in tre zone (*patches*). Sulle pareti si impone gradiente nullo di pressione, mentre il *type empty* indica che stiamo risolvendo un caso 2D, che in OpenFOAM equivale a generare comunque un dominio 3D, con una sola cella di calcolo nella terza direzione. Sulle *patches* perpendicolari a questa direzione fittizia non si impone nessuna condizione (*empty*).

Il file con le condizioni iniziali e al contorno è invece il seguente:

```

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField

```

```
{
  movingWall
  {
    type          fixedValue;
    value         uniform (1 0 0);
  }

  fixedWalls
  {
    type          fixedValue;
    value         uniform (0 0 0);
  }

  frontAndBack
  {
    type          empty;
  }
}
```

É analogo al precedente, a partire dall'unità di misura, fino alle *patches* (le stesse di prima). Si noti solamente che in questo caso, sia per le condizioni al contorno che per quelle iniziali, si devono dare le tre componenti della velocità (nelle parentesi tonde).

Consigliamo ancora di riferirsi ai tutorial e alle guide di OpenFOAM per maggiori dettagli. In ogni caso questi dizionari verranno poi riportati con altri commenti quando si entrerà nel vivo della simulazione della cavità nasale.

Capitolo 7

Cenni di fluidodinamica del flusso nasale

7.1 Le equazioni della fluidodinamica

In questo capitolo si vogliono richiamare alcuni concetti di base della fluidodinamica, a partire dalle equazioni di Navier-Stokes e dalla media di Reynolds, con particolare attenzione per i flussi incomprimibili. Si affronterà quindi il problema della chiusura, che viene risolto con l'introduzione dei modelli di turbolenza. Da ultimo si vuole trattare un po' più nello specifico il caso del flusso all'interno della cavità nasale.

Naturalmente sono tutti argomenti vastissimi, di cui si può solo accennare qualche formula, senza dimostrazioni. Chi fosse interessato può approfondire questi temi sui numerosi libri a riguardo. Nella seguente trattazione si fa riferimento al libro di Quadrio e Luchini [11], "Aerodinamica", e alle note di Fluidodinamica del Prof. Baron[1].

Le equazioni di Navier-Stokes

Le equazioni che governano il moto dei fluidi derivano direttamente dalle leggi di conservazione della massa, della quantità di moto e dell'energia. Tale sistema di equazioni prende il nome di Equazioni di Navier-Stokes e descrive il moto di qualsiasi fluido Newtoniano, viscoso e comprimibile, nel caso più generico di moto non stazionario e tridimensionale. Costituiscono un sistema di equazioni non-lineari alle derivate parziali, che risulta chiuso se si aggiunge l'equazione di stato, ovvero una relazione funzionale tra le variabili termodinamiche (densità, energia interna e pressione).

La conservazione della massa, espressa per un volume di controllo, risulta:

“la variazione nell’unità di tempo della massa contenuta in un volume di controllo uguaglia il flusso netto di massa attraverso la superficie che delimita il volume di controllo stesso”.

Espressa in formule, in forma conservativa, si ha:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \quad (7.1)$$

in cui ρ è la densità del fluido e \mathbf{V} è il vettore velocità.

La conservazione della quantità di moto ha il seguente enunciato:

“la variazione nell’unità di tempo della quantità di moto del fluido contenuto in un volume di controllo sommata al flusso netto di quantità di moto attraverso la superficie delimitante il volume di controllo stesso uguaglia la risultante delle forze esterne agenti sull’elemento di fluido in esso contenuto”.

$$\frac{\partial (\rho \mathbf{V})}{\partial t} + \nabla \cdot (\rho \mathbf{V} \mathbf{V} + p \mathbf{I} + \mathbf{J}_Q^d) = \mathbf{F} \quad (7.2)$$

in cui p è la pressione, \mathbf{F} è la risultante delle forze esterne e \mathbf{J}_Q^d è il flusso dissipativo di quantità di moto, esprimibile come:

$$\mathbf{J}_Q^d = -\mu \left(\nabla \mathbf{V} + (\nabla \mathbf{V})^T - \frac{2}{3} (\nabla \cdot \mathbf{V}) \mathbf{I} \right) - \lambda (\nabla \cdot \mathbf{V}) \mathbf{I} \quad (7.3)$$

in cui μ è la viscosità dinamica e λ è il secondo coefficiente di viscosità.

Da ultimo l’equazione dell’energia recita:

“la variazione nell’unità di tempo dell’energia totale del fluido contenuto nel volume di controllo sommata al flusso netto di energia totale attraverso la superficie delimitante il volume di controllo stesso uguaglia la somma della potenza delle forze agenti sull’elemento di fluido contenuto nel volume e del flusso netto di energia termica trasmessa all’elemento di fluido, per conduzione, attraverso la superficie del volume di controllo.”.

Esprimendo questo concetto in formule:

$$\frac{\partial}{\partial t} \left[\rho \left(e + \frac{V^2}{2} \right) \right] + \nabla \cdot \left[\rho \left(e + \frac{V^2}{2} \right) \mathbf{V} + p\mathbf{V} + \mathbf{J}_Q^d \cdot \mathbf{V} + \mathbf{J}_E^d \right] = L \quad (7.4)$$

in cui si è indicato con V il modulo della velocità \mathbf{V} , con e l'energia interna, con L la somma della potenza delle forze esterne e con \mathbf{J}_E^d il flusso dissipativo di energia, che risulta:

$$\mathbf{J}_E^d = -k\nabla T \quad (7.5)$$

dove k è il coefficiente di conducibilità termica e T è ovviamente la temperatura.

Nel caso incomprimibile, ipotesi ampiamente soddisfatta nel caso in esame di flusso nella cavità nasale, le equazioni di Navier-Stokes si semplificano notevolmente. Infatti l'equazione di stato si riduce a $\rho = const.$, l'equazione dell'energia si disaccoppia dal resto del sistema e la conservazione della massa si riduce alla ben nota condizione cinematica:

$$\nabla \cdot \mathbf{V} = 0 \quad (7.6)$$

ovvero il campo di velocità è solenoidale. Anche l'equazione della quantità di moto, sotto l'ipotesi di flusso incomprimibile a proprietà costanti, si semplifica e diventa:

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} + \frac{1}{\rho} \nabla p = \frac{\mu}{\rho} \nabla^2 \mathbf{V} + \mathbf{f} \quad (7.7)$$

Equazioni mediate di Reynolds

Le ultime equazioni descrivono quindi compiutamente in modo deterministico qualsiasi flusso, sotto le ipotesi di incomprimibilità e di fluido newtoniano a proprietà costanti, ritenendo sempre soddisfatta l'ipotesi del continuo. Tuttavia le equazioni di Navier-Stokes, per numeri di Reynolds sufficientemente alti esibiscono un comportamento caotico che prende il nome di *turbolenza*. La caratteristica che rende casuali i moti turbolenti è la forte sensibilità alle condizioni iniziali. Un'altra caratteristica peculiare è la totale assenza di separazione di scale. Questo significa che un moto turbolento presenta moti caotici per tutte le scale spaziali e temporali del fenomeno in esame, da quelle microscopiche a quelle macroscopiche, rendendo particolarmente complicato l'approccio statistico, in ogni caso inevitabile. Infatti sarebbe concettualmente

possibile integrare direttamente le equazioni scritte (DNS, *Direct Numerical Simulation*), ma bisognerebbe risolvere accuratamente tutte le scale spaziali e temporali coinvolte, con un costo computazionale ancora oggi impossibile per i tipici problemi fluidodinamici che si affrontano.

Senza entrare nei dettagli di questi argomenti, che meriterebbero molta più enfasi, proseguiamo con la tecnica usuale di mediare le equazioni in modo da risolvere solo il moto medio, che è quello che interessa. Tale tecnica prende il nome di equazioni mediate di Reynolds (RANS) e consiste nel considerare ciascuna variabile (velocità e pressione) come somma di un valor medio e di una fluttuazione temporale, quest'ultima a media nulla. Per le 4 incognite che abbiamo (3 componenti del vettore velocità e pressione):

$$u_i = \bar{u}_i + u'_i \quad (7.8)$$

$$p = \bar{p} + p' \quad (7.9)$$

ed inoltre

$$\overline{u'_i} = 0 \quad (7.10)$$

$$\overline{p'} = 0 \quad (7.11)$$

Sostituendo queste definizioni nelle equazioni precedenti, dopo alcuni passaggi si arriva alle nuove equazioni per le variabili medie $\bar{\mathbf{V}}$ e \bar{p} :

$$\nabla \cdot \bar{\mathbf{V}} = 0 \quad (7.12)$$

$$\rho \frac{\partial \bar{\mathbf{V}}}{\partial t} + \rho \nabla \cdot \overline{\mathbf{V}\mathbf{V}} + \nabla \bar{p} + \nabla \cdot \overline{\mathbf{J}_Q^d} = 0 \quad (7.13)$$

dove si vede che la prima equazione (conservazione della massa) è analoga a quella di partenza, mentre per la seconda resta da esprimere:

$$\overline{\mathbf{V}\mathbf{V}} = \bar{\mathbf{V}}\bar{\mathbf{V}} + \overline{\mathbf{V}'\mathbf{V}'} \quad (7.14)$$

in cui si vede che compare un nuovo termine, che dimensionalmente è uno sforzo, che si chiama tensore degli sforzi di Reynolds:

$$\mathbf{R} = \rho \overline{\mathbf{V}'\mathbf{V}'} \quad (7.15)$$

Definiamo allora un tensore degli sforzi totali \mathbf{T} come:

$$\mathbf{T} = \overline{\mathbf{J}_Q^d} + \mathbf{R} \quad (7.16)$$

e da ultimo si può riscrivere la conservazione della quantità di moto:

$$\rho \frac{\partial \bar{\mathbf{V}}}{\partial t} + \rho \nabla \cdot \bar{\mathbf{V}} \bar{\mathbf{V}} + \nabla \bar{p} + \nabla \cdot \mathbf{T} = 0 \quad (7.17)$$

che è analoga a quella di partenza se non fosse per il tensore degli sforzi di Reynolds, contenuto in \mathbf{T} , che è formato dalle fluttuazioni delle componenti della velocità, proprio quelle che non si volevano calcolare. A questo punto risulta evidente che il sistema di equazioni non è più chiuso: occorre quindi un modello in grado di determinare il tensore \mathbf{R} .

Modelli di turbolenza

La chiusura delle equazioni RANS avviene secondo due strategie distinte:

- modelli a viscosità turbolenta (*eddy viscosity*), ovvero si ipotizza (Boussinesq) che ci sia una relazione tra il tensore di Reynolds e le derivate spaziali delle componenti di velocità media. Questa chiusura può essere algebrica o differenziale, a una o più equazioni;
- modelli per le 6 componenti indipendenti del tensore di Reynolds (RSM, Reynolds stresses model), che aggiungono 6 equazioni differenziali alle Navier-Stokes. Questa soluzione, la più complicata all'interno delle RANS, non viene ulteriormente investigata nel presente lavoro di tesi.

Analizziamo perciò la prima strada. L'ipotesi di Boussinesq può essere scritta come:

$$r_{ij} = -2\nu_t s_{ij} \quad (7.18)$$

dove r_{ij} è una generica componente del tensore di Reynolds, proporzionale attraverso la viscosità turbolenta¹ al tensore di componenti:

$$s_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (7.19)$$

che è determinato dal moto medio. Per chiudere le equazioni basta allora definire o ricavare lo scalare “viscosità turbolenta”, in tutto il campo di moto. Di seguito si elencano i modelli più noti in letteratura, ponendo l'attenzione su quelli utilizzati nel presente lavoro di tesi. Non si entrerà

¹dimensionalmente è una viscosità, ma solo perchè questa legge è in analogia con la relazione tra il tensore degli sforzi viscosi e il tensore della velocità di deformazione. Si ricordi che la viscosità è una proprietà del fluido e non del moto: la viscosità turbolenta è solo un artificio.

nei dettagli di ciascuno, che richiedono tra l'altro la taratura² di alcuni parametri, così come non si discuterà dei limiti intrinseci dell'ipotesi di Boussinesq e dei conseguenti limiti delle RANS.

- Modello di ordine 0: *Mixing length* di Prandtl.

Questo modello (chiusura algebrica) consiste semplicemente nel definire:

$$\nu_t = l_m^2 \frac{\partial u}{\partial y} \quad (7.20)$$

avendo definito la lunghezza di rimescolamento (*mixing length*) come:

$$l_m = ky \quad (7.21)$$

in cui $k = 0.41$ è la costante di Von Karman, y è la distanza dalla parete.

- Modello di ordine 1: Spalart-Allmaras.

La viscosità turbolenta in tutto il campo è ricavata da un'equazione differenziale (con 8 costanti e 3 funzioni empiriche) che ne descrive il campo spaziale. Non scendiamo nel dettaglio di questo modello, ricordiamo solamente che è l'esempio più semplice di modello completo³.

- Modello di ordine 2: Modello standard $k - \epsilon$.

Il modello $k - \epsilon$ si è affermato come uno dei modelli più utilizzati e consiste nell'aggiunta alle RANS di 2 equazioni alle derivate parziali, per le 2 quantità scalari "energia cinetica turbolenta" (k , [m^2/s^2]) e "rateo di dissipazione dell'energia cinetica turbolenta" (ϵ , [m^2/s^3]).

L'equazione di bilancio per k risulta:

$$u_i \frac{\partial k}{\partial x_i} = P_k - \epsilon + \frac{\partial}{\partial x_i} \left[(\nu + \nu_t) \frac{\partial k}{\partial x_i} \right] \quad (7.22)$$

dove P_k è la produzione di energia cinetica turbolenta:

$$P_k = 2\nu_t s_{ij} s_{ij} \quad (7.23)$$

²le tarature dei coefficienti empirici si effettuano con analisi di sensibilità, con considerazioni teoriche (rare), con il paragone con altri metodi o prove sperimentali

³ovvero un modello in cui la viscosità turbolenta non è ricavata da definizioni arbitrarie, ma dalla soluzione del proprio campo

mentre la seconda equazione (bilancio di ϵ) risulta:

$$u_i \frac{\partial \epsilon}{\partial x_i} = \frac{C_{\epsilon 1} P_k - C_{\epsilon 2} \epsilon}{k} \epsilon + \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_i} \right] \quad (7.24)$$

Ricavando quindi i campi scalari k ed ϵ si ha in tutto il campo di moto la viscosità turbolenta:

$$\nu_t = C_\nu \frac{k^2}{\epsilon} \quad (7.25)$$

La maggiore criticità di questo modello risiede nel fatto che k è nulla a parete, rendendo singolare l'equazione per ϵ . Inoltre il risultato è fortemente dipendente dai coefficienti.

- Modello di ordine 2: Modello standard $k - \omega$.

L'altro modello di ordine 2 tipicamente utilizzato deriva direttamente dal precedente: si tiene uguale l'equazione per k e si scrive una seconda equazione per la nuova variabile ω , che è definita come $\frac{\epsilon}{k}$:

$$u_i \frac{\partial \omega}{\partial x_i} = C_{\omega 1} \frac{\omega}{k} P_k - C_{\omega 2} \omega^2 + \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_i} \right] \quad (7.26)$$

ω è la frequenza della turbolenza e può anche essere interpretata come una vorticità R.M.S., oppure si può considerare ω^2 come un'ensrofia. Il modello $k - \omega$ porta tipicamente a risultati più affidabili del $k - \epsilon$, in quanto elimina la singolarità a parete ed è meno sensibile ai (sempre numerosi) coefficienti.

- Modello di ordine 2: Modello $k - \omega - SST$.

Il modello $k - \omega - SST$ è un'ulteriore evoluzione del modello precedente. La formulazione SST (shear stress transport) riesce a combinare il meglio dei modelli precedenti a 2 equazioni differenziali: nello strato limite viene utilizzato un modello $k - \omega$, con i vantaggi già illustrati, mentre altrove si utilizza il $k - \epsilon$, con il vantaggio di rendere tutto il modello meno sensibile alle condizioni al contorno delle quantità turbolente. Questo modello fornisce buoni risultati anche per problemi a bassi numeri di Reynolds, così come nel caso di gradiente avverso di pressione o separazioni nel flusso. Presentiamo questo modello con qualche maggior dettaglio, dato che verrà usato come modello principale per le simulazioni turbolente.

L'equazione per k resta la stessa, mentre l'equazione per ω viene leggermente modificata come segue:

$$\begin{aligned}
 u_i \frac{\partial \omega}{\partial x_i} &= C_{\omega 1} \frac{\omega}{k} P_k - C_{\omega 2} \omega^2 + \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_i} \right] + \dots \\
 &\dots + 2(1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}
 \end{aligned} \tag{7.27}$$

con le seguenti definizioni:

$$\nu_t = \frac{a_1 k}{\max(a_1 \omega, S F_2)}$$

$$a_1 = \frac{5}{9}$$

$$F_2 = \tanh \left[\left[\max \left(\frac{2\sqrt{k}}{\omega \beta^* y}, \frac{500\nu}{y^2 \omega} \right) \right]^2 \right]$$

$$\beta^* = \frac{9}{100}$$

$$C_{\omega 1} = 0.44$$

$$C_{\omega 2} = 0.0828$$

$$\sigma_\omega = 0.5$$

$$\sigma_{\omega 2} = 0.856$$

$$F_1 = \tanh \left[\left[\min \left[\max \left(\frac{\sqrt{k}}{\omega \beta^* y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\sigma_{\omega 2} k}{CD_{k\omega} y^2} \right] \right]^4 \right]$$

$$CD_{k\omega} = \max \left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right)$$

Non servono altri commenti per capire quanto sia empirico e delicato mettere a punto un buon metodo per la chiusura delle equazioni RANS: anche per questi motivi i risultati spesso hanno un'accuratezza insufficiente.

7.2 Aerodinamica della cavità nasale

Si è già discusso in precedenza dell'anatomia della cavità nasale, senza accennare agli aspetti fluidodinamici. In questa sezione si presentano alcuni valori tipici delle grandezze in gioco, in modo da acquisire un po' di sensibilità in vista della preparazione delle simulazioni e dei conseguenti risultati.

Data la specificità di ogni anatomia non è possibile trarre conclusioni assolute, tuttavia si possono fare alcune considerazioni, valide in linea di massima per un naso normale, senza evidenti patologie. Il flusso d'aria, appena entra dalle narici esterne, è diviso in due getti che sono diretti verso l'alto e la parte interna della cavità nasale. In condizioni di riposo, che per un uomo adulto corrisponde ad una differenza di pressione tra l'esterno e la nasofaringe di circa 150 Pa, la velocità del flusso durante l'inspirazione è di circa 2-3 m/s alle narici. Tale velocità aumenta attraverso il vestibolo nasale fino a 12-18 m/s nella zona della valvola nasale, che rappresenta un tratto di forte costrizione per il flusso.

Proseguendo nella cavità nasale propriamente detta (ovvero la zona dei turbinati e dei meati) tipicamente si ha un decremento della velocità media, con la corrente principale che passa nel meato medio, una corrente secondaria nel meato inferiore (zona del pavimento nasale) e modestissime quantità d'aria nella regione olfattiva (meato superiore). A livello sperimentale, prima che iniziassero i lavori di fluidodinamica numerica nella cavità nasale, non si sapeva quasi nulla della portata nei seni paranasali, se non che è trascurabile rispetto al condotto principale. Quando verranno presentati i risultati numerici si potrà investigare più nel dettaglio anche questo aspetto.

Alla fine dei meati i due flussi nella cavità nasale si riuniscono nel rinofaringe, piegandosi verso il basso di quasi 90°. In questa zona si osserva un ulteriore aumento di velocità (si può arrivare mediamente a 4-5 m/s), che da questo punto in poi tende sempre ad aumentare proseguendo verso la faringe, data la progressiva diminuzione della sezione.

Riferendoci alla figura 7.1, tratta dall'articolo di Doorly [6], vediamo di mettere in luce qualche dettaglio significativo. La figura è relativa ad una serie di analisi *in-vitro*, condotte su 3 diverse geometrie e con diverse portate. I tre diversi modelli si chiamano R3, R1 e I1. I primi due si riferiscono a due soggetti diversi, l'ultimo è un'idealizzazione (con opportune tecniche) della geometria R1, che è stata ricostruita con diverse soglie e poi "mediata", secondo criteri spiegati nel dettaglio dall'articolo. In ogni caso per noi non è tanto importante la differenza tra le tre geometrie quanto i risultati qualitativi che si ottengono dalle visualizza-

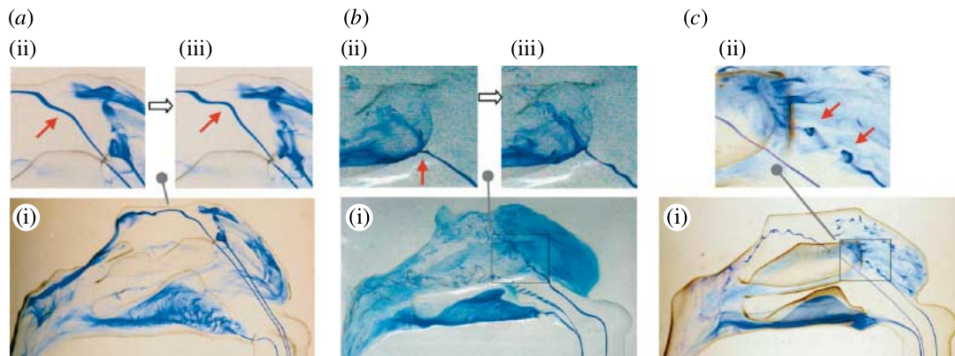


Figura 7.1: Visualizzazioni con inchiostro, tratte dall'articolo [6].

zioni con inchiostro. Nelle figure (a) dalla (i) alla (iii) si ha il modello R3 con una portata, piuttosto modesta, di 150 ml/s. Si vedono chiaramente 3 filamenti di inchiostro che entrano nelle narici e le ricircolazioni e le irregolarità del flusso. Il filamento più in basso passa nel meato inferiore, gli altri due impattano sul turbinato medio e proseguono nella zona più alta della cavità. È interessante vedere la vasta zona di ricircolazione appena dopo il passaggio nella valvola nasale, così come le fluttuazioni nella parte posteriore (verso la nasofaringe), anche se con questa portata il flusso appare laminare nella maggior parte dei condotti.

Nelle figure (b) è presentato il modello R1, con una portata di 170 ml/s. Appare evidente un forte rimescolamento e l'immediata dispersione dei filamenti di inchiostro, ma questo non è sufficiente per concludere che siamo in presenza di un flusso con turbolenza completamente sviluppata. In realtà, come in precedenza, il flusso è laminare fino all'impatto con il turbinato medio, dove il filamento d'inchiostro viene disperso. Nelle figure (c) si ha infine il modello I1, ad una portata di 150 ml/s. Anche questa visualizzazione è molto significativa, in quanto mostra l'instabilità dello *shear-layer*, che si sviluppa appena dopo la valvola nasale, come si vede grazie al filamento più in alto. È interessante seguire anche il percorso del filamento centrale, che entra nel meato medio e in quel punto comincia a mostrare oscillazioni che si protraggono fino alla zona dei recettori olfattivi. Come sempre sono in evidenza le zone di ricircolazione anteriore e posteriore.

Sempre dallo stesso articolo riportiamo la figura 7.2, in cui sono mostrate le linee di flusso e il modulo della velocità esternamente alla cavità nasale, di una simulazione risolta con Fluent. Il flusso è stazionario, con una portata imposta di 100 ml/s per ogni narice. Con questa portata il flusso è prevalentemente laminare, però quest'approssimazione non sa-

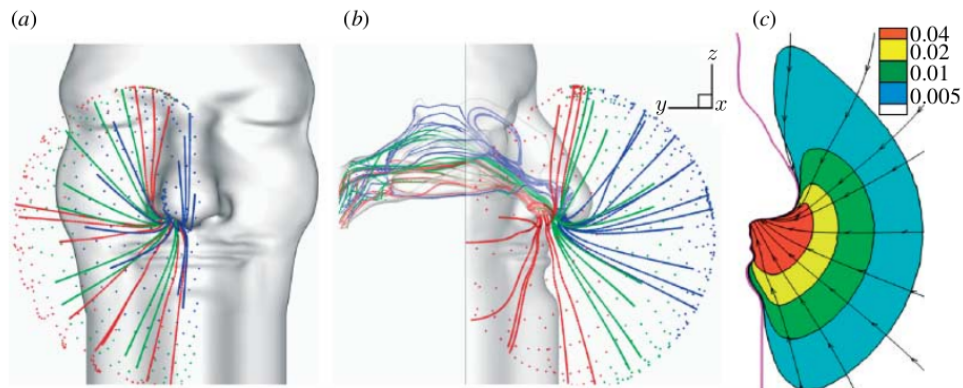


Figura 7.2: Linee di flusso e velocità esterna, tratto da [6]

rebbe più accettabile con portate globali superiori a 500-600 ml/s, che sono una respirazione appena moderata. Il colore delle linee di flusso dipende dalla posizione iniziale di partenza all'esterno delle narici. Le diverse traiettorie restano abbastanza separate e distinguibili, mostrando poco rimescolamento. Come al solito ci sono evidenti ricircoli nella zona dell'atrio, dopo la valvola nasale; è anche interessante notare che il flusso nella regione olfattiva proviene dalla parte anteriore del vestibolo nasale ed esternamente proviene da una zona più o meno centrale di fronte al naso esterno. Da ultimo si notino le isolinee del modulo di velocità all'esterno del naso: ad una distanza superiore di circa 5 cm dalle narici la velocità media è dell'ordine dei millimetri al secondo. Questo fatto indica che, sebbene sia importante risolvere anche la zona esterna al naso, non serve spingersi molto lontano dalle narici per l'imposizione delle condizioni al contorno e in quella zona le celle di calcolo possono anche essere poco raffinate.

Capitolo 8

Dalla geometria alla mesh di volume

8.1 Presentazione del problema

La costruzione della mesh di volume è uno dei passaggi più delicati dell'intero sviluppo dello strumento di analisi fluidodinamica che si è deciso di costruire. Vengono descritte di seguito le due *utilities*, contenute in OpenFOAM, che svolgono questo passaggio.

Come si è già visto, OpenFOAM è una libreria in C++, in cui i solutori e le utilità sono comandate attraverso i relativi dizionari. La creazione della mesh all'interno di OpenFOAM è complicata dai numerosi parametri che regolano questo passaggio e da una limitata letteratura a riguardo. Le maggiori difficoltà sono dovute ad un minimo di arbitrarietà di questi parametri e ad una certa difficoltà nel valutare le conseguenze delle modifiche dei parametri in gioco. Questi problemi, che nella maggior parte dei casi si risolvono per tentativi, verranno analizzati nell'analisi di sensibilità dei risultati, in modo da mostrare che in ogni caso non sono critici per le soluzioni.

Non entreremo nei dettagli di descrizione della mesh in OpenFOAM, rimandiamo come sempre alle relative guide per maggiori informazioni. Presentiamo di seguito solo il procedimento definitivo, che è frutto di molte prove e tentativi. Verranno spiegati i concetti di base ed illustrati i significati dei parametri, relativamente al caso in esame, senza pretese di generalità.

8.2 L'*utility* blockMesh

Il principio di blockMesh è di decomporre il dominio di calcolo in uno o più blocchi tridimensionali esaedrici. I lati di questi esaedri possono essere anche curvi, ma nel caso in esame non servono. Il dizionario contiene semplicemente l'elenco dei vertici¹, con le coordinate di ciascuno, le definizioni dei vari blocchi (*blocks*, definiti da 8 vertici) e le definizioni delle facce esterne, ovvero le *patches*.

L'utilizzo di cui si ha bisogno è molto semplice, in quanto questa *utility* ci serve solo per creare un parallelepipedo esterno alla testa, che racchiude tutto il dominio di calcolo. Il compito di snappyHexMesh sarà poi quello di mantenere, all'interno di questo dominio iniziale, solo le parti di interesse, specificate dalla superficie .stl salvata in Slicer. Presentiamo di seguito, con alcune sintesi e semplificazioni², l'aspetto del dizionario blockMeshDict:

```
convertToMeters 1000;
vertices
(
    (-0.12 -0.12 -0.195)
    ...
    ( 0.12  0.18  0.005)
);
blocks
(
    hex (0 ... 7) (10 10 10) simpleGrading (1 1 1)
    ...
    hex (19 ... 26) (10 10 10) simpleGrading (1 1 1)
)
patches
(
    patch leftWall
    (
        (0 9 12 3)
        (3 12 15 6)
        (9 18 21 12)
    )
)
```

¹ad ogni vertice è associato un ID, che viene usato poi quando si definiscono spigoli, facce e blocchi

²riportiamo solo lo scheletro del dizionario, senza tutti i valori numerici. Inoltre trascuriamo alcune importanti regole da seguire nella scrittura del dizionario, come ad esempio l'ordine con cui si scrivono i vertici nei *blocks* e nelle *patches*

```

        (12 21 24 15)
    )
    ...
);

```

L'ingresso *convertToMeters* si riferisce al fatto che le coordinate dei vertici sono specificate in metri, mentre vogliamo creare la mesh di volume con i millimetri come unità di misura, dato che la superficie .stl è definita in mm. Appena prima di procedere con la simulazione numerica si riconvertirà tutto in metri, come verrà spiegato in seguito. Il sub-dizionario *vertices* contiene l'elenco dei vertici con le relative coordinate in un sistema cartesiano. Al primo vertice è associato un numero identificativo $ID = 0$, al secondo $ID = 1$ e così via. Nel caso che stiamo presentando bisognerebbe quindi scrivere le coordinate degli 8 vertici dell'esaedro, ma per avere a disposizione più parametri nella scelta delle celle in cui si suddivide il dominio esterno si è deciso di creare 8 esaedri. Perciò sono stati inseriti 27 vertici³ (numerati da 0 a 26), le cui coordinate sono state decise disegnando un dominio che potesse ragionevolmente comprendere la mesh superficiale creata in Slicer. Il sub-dizionario *blocks* contiene quindi gli 8 esaedri (ingressi *hex*). Nelle prime parentesi c'è l'elenco dei vertici che compongono il solido, nelle seconde parentesi il numero di celle, nelle 3 direzioni, in cui viene suddiviso ciascun esaedro. Il *simpleGrading* indica il "rapporto d'espansione delle celle", ovvero la possibilità che le celle non siano tutte uguali all'interno dell'esaedro, ma rifinite secondo il criterio scelto. Ad esempio, nel caso riportato, si hanno 10 celle in ogni direzione, ovvero 1000 celle per ogni esaedro. Con *simpleGrading (1 1 1)* le celle sono tutte uguali. Se invece si mettesse *simpleGrading (2 1 1)* si avrebbero, in direzione x , sempre 10 celle, ma con l'ultima di dimensione doppia (in x) rispetto alla prima. Da ultimo si deve inserire il sub-dizionario *patches*, che contiene le 6 facce esterne del volume di calcolo. Dato che si hanno 8 esaedri è evidente che ogni faccia esterna (ad esempio la *leftWall*) è formata da 4 facce, identificate sempre dai vertici. A questo punto il dizionario è completo e si può eseguire l'*utility*.

Il risultato visivo è in figura 8.1, in cui si vede solo la superficie esterna del volume di calcolo. È stata nascosta la mesh interna, così come la faccia di destra, in modo da far vedere anche l'interno. A parte è stato anche caricato il file .stl con la ricostruzione del volto e della cavità nasale. Si noti che il solido creato circonda la superficie ricostruita, ma

³sono 8 esaedri con 8 vertici ciascuno, ma molti punti sono evidentemente in comune

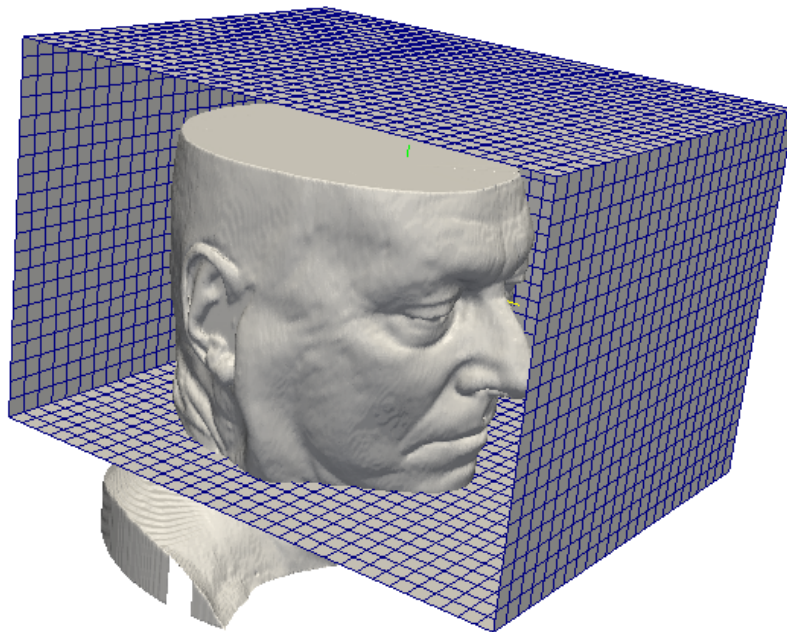


Figura 8.1: Risultato della *utility* blockMesh, con l'aggiunta della superficie .stl. Per chiarezza sono state nascoste la mesh interna e la faccia destra.

la interseca nella zona del collo e della gola, perchè snappyHexMesh non funziona se il volume di partenza circonda completamente una superficie aperta.

8.3 L'*utility* snappyHexMesh

L'*utility* snappyHexMesh genera la mesh di volume 3D, a partire dalla superficie triangolata in formato .stl. Il criterio di funzionamento è quello di partire da una mesh di volume iniziale (*background mesh*, poco raffinata), generata da blockMesh, e iterativamente si procede con un primo raffinamento nelle zone specificate, in seguito si eliminano le celle non raggiungibili. A questo punto si ha una mesh fatta solo di esaedri, con già il volume di calcolo corretto, che viene definita *castellated mesh*. Il secondo passaggio consiste nel deformare la mesh fino a farla aderire alla superficie di ingresso (fase di *snap*); in questa fase non si hanno

più solo esaedri, la deformazione modifica sostanzialmente i solidi (soprattutto quelli sui contorni). Da ultimo si possono inserire uno o più strati di celle vicino alle pareti (*cell layers*), in modo da risolvere bene lo strato limite. Riportiamo di seguito, diviso in più parti, il lungo dizionario di `snappyHexMesh`, con qualche breve spiegazione. Si rimanda come sempre alle guide o ai forum online per le definizioni e le spiegazioni di ciascun ingresso.

```
// Which of the steps to run
castellatedMesh true;
snap           true;
addLayers     true;

// Geometry. Definition of all surfaces.
geometry
{
    nasal_cavity.stl
    {
        type triSurfaceMesh;
        name vcg;
    }
    refinementBox
    {
        type searchableBox;
        min (-38 -10 -145);
        max ( 48  77 -80);
    }
    ...
};
```

Per prima cosa si deve decidere quali passaggi eseguire, in questo caso tutti e tre. Nel sub-dizionario *geometry* bisogna specificare il file `.stl` con la superficie (*nasal-cavity*) e si specificano anche uno o più *refinementBox*, ovvero zone in cui si vogliono inserire più celle, ad esempio nella zona interna alla cavità nasale (definita dalle coordinate minime e massime di un esaedro).

```
castellatedMeshControls
{
    maxLocalCells 2000000;
    maxGlobalCells 8000000;
```

```
minRefinementCells 1;
maxLoadUnbalance 0.05;

nCellsBetweenLevels 4;

// Surface based refinement
refinementSurfaces
{
    vcg
    {
        // Surface-wise min and max refinement level
        level (1 2);
    }
}
resolveFeatureAngle 30;

// Region-wise refinement
refinementRegions
{
    refinementBox
    {
        mode inside;
        levels ((1 4)); /*entry 1 ignored */
    }
    ...
}

// Mesh selection
locationInMesh (0 150 -150);
}
```

Passiamo quindi ai controlli del primo passaggio, la *castellated mesh*. L'ingresso *nCellsBetweenLevels* indica quanti strati di celle vengono inseriti tra due livelli di raffinamento successivi; tali livelli sono specificati nei due sub-dizionari successivi, *refinementSurfaces* e *refinementRegions*, rispettivamente per le superfici e per le regioni. Nel primo dei due si legge che la superficie della cavità nasale ha un livello minimo di raffinamento pari a 1 (ovvero gli esaedri, a meno delle deformazioni successive, restano della dimensione stabilita dal *blockMesh*), mentre il livello massimo (nelle zone specificate dall'ingresso *resolveFeatureAngle*) è pari a 2, il

che significa che ogni esaedro di partenza viene suddiviso in 8 esaedri, dividendo a metà ogni lato. Si capisce allora l'importanza del numero di strati tra due *refinement levels* successivi, in modo da evitare che il passaggio tra zone poco raffinate e zone più fitte sia troppo brusco. Nell'altro sub-dizionario si specifica invece che nella zona definita in precedenza dal *refinementBox* si vuole un livello pari a 4, così da inserire molte celle nella zona della cavità nasale. Il *locationInMesh* serve a definire quale zona della mesh di volume iniziale (il volume esterno generato da *blockMesh*) dev'essere mantenuta. In figura 8.2 si vede il risultato del primo passaggio, che porta alla *castellated mesh*. Per chiarezza si è nascosta la mesh interna e un lato esterno, così da vedere come è stata ricostruita la superficie del volto (quasi irriconoscibile) in questo primo passaggio.

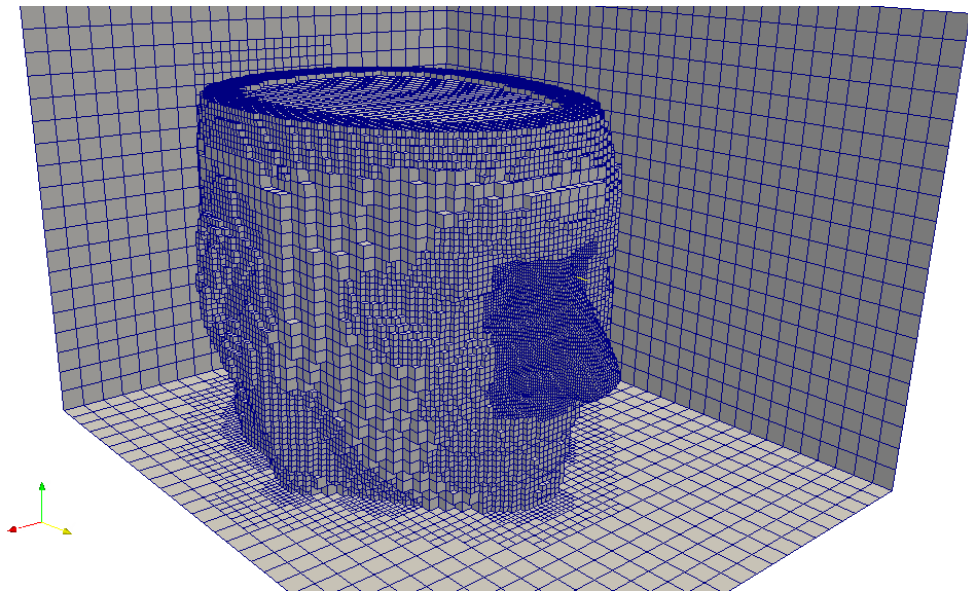


Figura 8.2: Risultato di *snappyHexMesh*: *castellated mesh*.

```

snapControls
{
    nSmoothPatch 5;

    //- Relative distance for points
    // to be attracted by surface.
    tolerance 4;

```

```

    //- Number of mesh displacement relaxation iterations.
    nSolveIter 50;

    //- Maximum number of snapping relaxation iterations.
    nRelaxIter 30;
}

```

Il sub-dizionario dei controlli di *snap* contiene solo una serie di parametri relativi alle iterazioni di questa fase. Ad ogni iterazione (*smoothing iteration*) si proiettano i vertici della *castellated mesh* sulla superficie *.stl* controlla che la nuova mesh soddisfi i requisiti richiesti nel sub-dizionario successivo (*meshQualityControls*). In caso contrario si riproiettano i punti con uno spostamento inferiore e si itera (*relaxation iteration*) fino a che la mesh di volume sia corretta. Il risultato della fase di *snap* è visualizzato (come esempio) in figura 8.3, da confrontare con la 8.2.

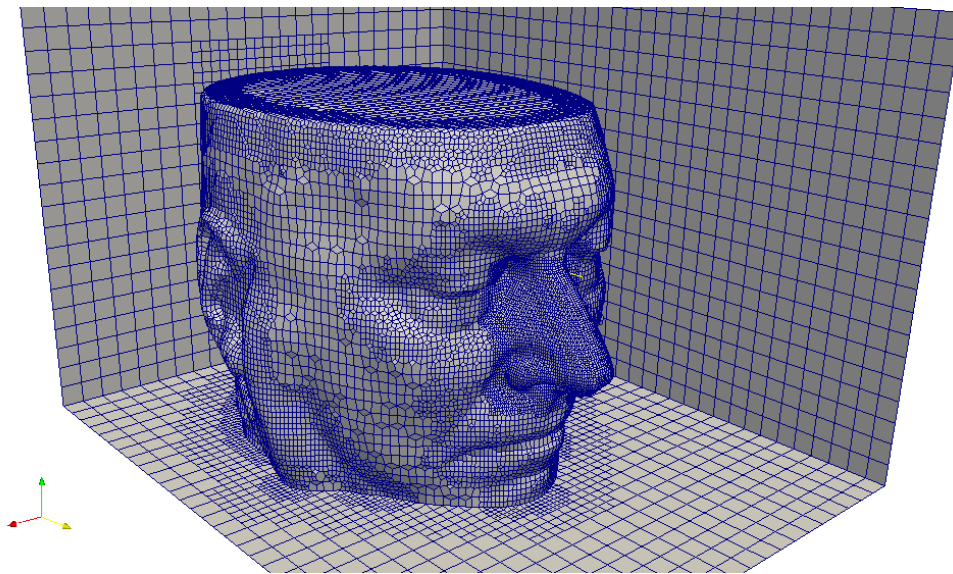


Figura 8.3: Risultato di snappyHexMesh: fase di *snap*.

```

meshQualityControls
{
    //- Maximum non-orthogonality allowed.
    maxNonOrtho 40;

    //- Minimum pyramid volume.
    minVol 1e-6;
}

```



```
//- Max concaveness allowed.
maxConcave 30;

//- Minimum face area.
minArea 1e-3;

//- Max skewness allowed.
maxBoundarySkewness 1.50;
maxInternalSkewness 1.25;

//- minFaceWeight (0 -> 0.5)
minFaceWeight 0.12;

//- minVolRatio (0 -> 1)
minVolRatio 0.04;

//- Minimum face twist.
minTwist 0.15;

//must be >0 for Fluent compatibility
minTriangleTwist 0.10;

//- minimum normalised cell determinant
//- 1 = hex, <= 0 = folded or flattened illegal cell
minDeterminant 0.1;

//- Minimum projected area v.s. actual area.
minFlatness 0.5;

// Advanced
//- Number of error distribution iterations
nSmoothScale 8;
//- amount to scale back displacement at error points
errorReduction 0.8;

// relaxed parameters
relaxed
{
maxNonOrtho 65;
maxConcave 75;
```

```
maxBoundarySkewness 8;  
maxInternalSkewness 4;  
minFaceWeight 0.04;  
minVolRatio 0.01;  
minTwist 0.03;  
minTriangleTwist -1;  
minDeterminant 0.001;  
}  
}
```

Il sub-dizionario *meshQualityControls* è la parte più importante dell'intero *snappyHexMesh*, per questo sono state lasciate le definizioni dei vari parametri. Il loro significato non è sempre evidente, tuttavia regolano la costruzione stessa e la qualità della mesh finale; a seconda dei valori scelti si avrà infatti un risultato buono o scadente. I valori presentati sono frutto di prove e, pur assicurando una mesh di volume accettabile per le simulazioni successive, lasciano spazio ad altri tentativi e miglioramenti. Senza scendere nei dettagli di tutti, si vuole sottolineare che i parametri più critici sono l'ortogonalità delle celle (*maxNonOrtho*) e due parametri relativi alla *skewness* (*maxBoundarySkewness* e *maxInternalSkewness*). Il sub-dizionario *relaxed* si riferisce al fatto che dopo un certo numero di iterazioni si rilassano⁴ i criteri richiesti.

Riportiamo nelle figure 8.4 e 8.5 la stessa sezione sagittale della mesh di volume, con in evidenza la zona della cavità nasale. In queste figure si apprezza meglio il passaggio tra la mesh *castellated* e quella dopo la fase di *snap*.

```
addLayersControls  
{  
    relativeSizes true;  
  
    layers  
    {  
        vcg_patch0  
        {  
            nSurfaceLayers 3;  
        }  
    }  
  
    expansionRatio 1;
```

⁴sia in questo passaggio di *snap* che nel successivo di *addLayers*

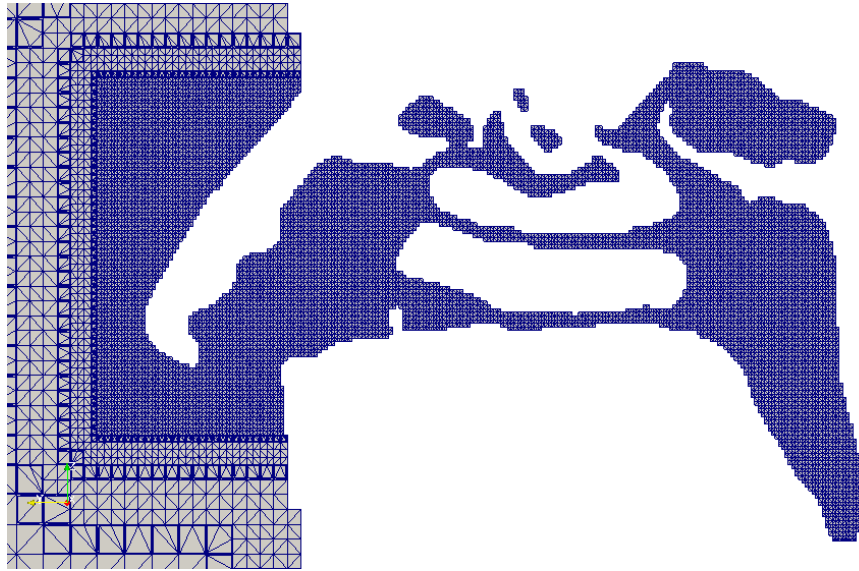


Figura 8.4: Sezione della *castellated mesh*.

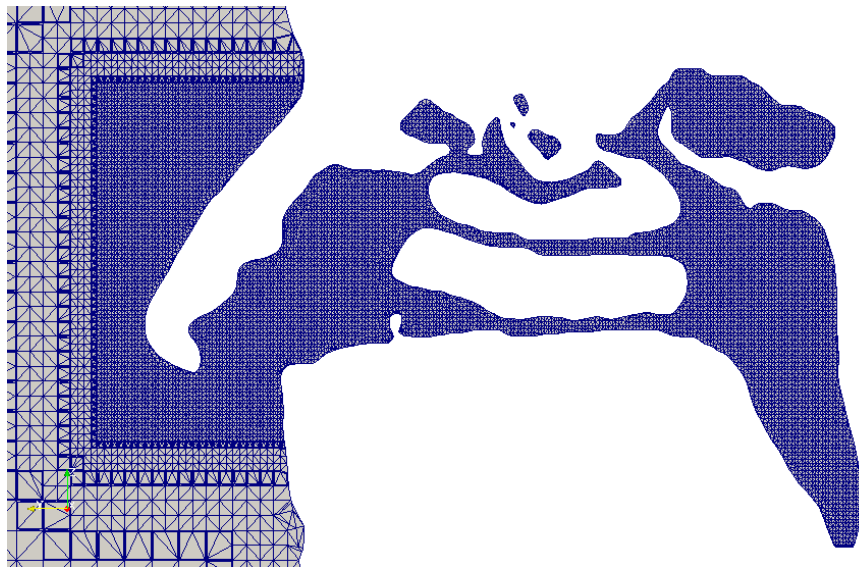


Figura 8.5: Sezione della mesh dopo la fase di *snap*.

```
finalLayerThickness 0.4;

minThickness 0.1;

nGrow 1;

// Advanced settings
featureAngle 80;
nRelaxIter 3;
nSmoothSurfaceNormals 1;
nSmoothNormals 3;
nSmoothThickness 10;
maxFaceThicknessRatio 0.5;
maxThicknessToMedialRatio 0.3;
minMedianAxisAngle 130;
nBufferCellsNoExtrude 0;
nLayerIter 50;
nRelaxedIter 2;
}
```

Anche il sub-dizionario *addLayersControls* contiene numerosi parametri, che andrebbero investigati a lungo. I più importanti sono *nSurfaceLayers*, che indica il numero di strati di celle da aggiungere e i due relativi allo spessore massimo e minimo degli strati in questione (*finalLayerThickness*, *minThickness*). Molto importante anche il *featureAngle*, che analogamente al *resolveFeatureAngle* della fase *castellated* specifica in quali zone dei contorni verranno aggiunte le celle.

In figura 8.6 si vede una sezione sagittale della mesh, centrata nella zona posteriore della cavità nasale (coane), che mette in evidenza l'aggiunta di 3 strati di celle sul contorno.

```
// Flags for optional output
// 0 : only write final meshes
// 1 : write intermediate meshes
// 2 : write volScalarField with cellLevel
// 4 : write current intersections as .obj files
debug 0;

// Merge tolerance.
mergeTolerance 1E-6;
```

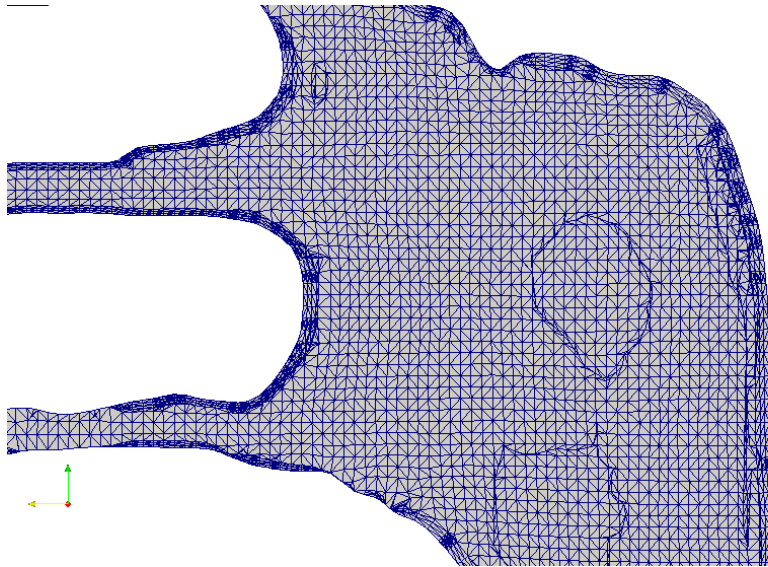


Figura 8.6: Sezione della mesh con in evidenza l'aggiunta dei *layers*.

Da ultimo si può scegliere se visualizzare mesh intermedie (tipicamente per il *debug*) o solo quelle finali. A questo punto è davvero tutto pronto per procedere con le simulazioni e l'analisi dei risultati.

Parte III

Risultati

Capitolo 9

Istruzioni per le simulazioni

9.1 Elenco delle istruzioni OpenFOAM

Prima di entrare nella fase di presentazione dei risultati, restano ancora da descrivere i comandi che permettono di eseguire le simulazioni. Infatti, oltre a generare la mesh di volume come mostrato nel capitolo 8, bisogna sistemare alcuni problemi lasciati in sospeso prima di poter lanciare il solutore opportuno. Tra questi problemi, alcuni banali e altri un po' più delicati, troviamo ad esempio la riconversione da mm a m, il controllo della qualità della mesh, la definizione delle patch di inlet/outlet, le condizioni al contorno. Si elencano di seguito tutti i comandi di OpenFOAM, con le relative spiegazioni, che svolgono quanto detto.

Come primo esempio questi comandi possono essere eseguiti e verificati uno per uno; nelle sezioni successive si mostrerà che possono essere eseguiti in automatico in uno script, data la generalità dei comandi stessi.

Della struttura di un caso di OpenFOAM si è già parlato in maniera molto generale nel capitolo 6, vediamo ora di approfondire un caso funzionante, limitandoci alla descrizione dei dizionari non ancora presentati.

Una cartella per una semplice simulazione laminare di un transitorio che va a regime ha la seguente struttura:

- **constant**
 - RASProperties
 - transportProperties
 - turbulenceProperties
- **polyMesh**

- blockMeshDict
- **triSurface**
 - nasalcavity.stl
- **system**
 - controlDict
 - FVSchemes
 - FVSolution
 - decomposeParDict
 - snappyHexMeshDict
 - cellSet
 - faceSet
 - createPatch
 - p
 - U

Una volta che tutti i dizionari sono preparati¹, si devono eseguire i seguenti comandi in un terminale aperto nella *directory* principale del caso in esame:

1. **blockMesh**
crea una serie di file all'interno della cartella **polyMesh**, con la definizione della mesh di partenza. I dettagli sono già stati discussi nel capitolo 8.
2. **snappyHexMesh**
crea la mesh di volume, utilizzando come punti di partenza la mesh generata all'istruzione precedente e la definizione della superficie (nasalcavity.stl). Anche questo dizionario è stato trattato nel capitolo 8. Si noti inoltre che è possibile eseguire questo comando anche in parallelo, analogamente a quanto verrà presentato per l'esecuzione del solutore vero e proprio. Alla fine produce 3 diverse mesh (quella *castellated*, quella dopo la fase di *snap* e quella con l'aggiunta dei *layers* sui contorni), salvandole in 3 nuove cartelle nella *directory* principale.

¹è importante notare che la fase di preparazione è molto lunga, ma i dizionari sono praticamente identici per tutte le simulazioni, tranne che per qualche dettaglio che sarà specificato. È utile conoscerli e saperli modificare, ma una volta che sono stati preparati non c'è più alcun lavoro per l'utente

3. `checkMesh`

serve a controllare la qualità delle mesh appena generate; nel caso ci fossero degli errori si devono modificare i due dizionari delle istruzioni precedenti. Questo non è (al momento) possibile se tutte le istruzioni stanno girando in automatico, in tal caso questo comando può solo servire come verifica a posteriori, nell'ipotesi che la mesh di volume sia già corretta. Si ricordi però che i due dizionari precedenti (*blockMeshDict* e *snappyHexMeshDict*) non dipendono dal singolo caso in esame, pertanto una volta che sono stati ragionevolmente impostati è abbastanza improbabile che la mesh venga generata con degli errori. L'output di questo comando (per ogni mesh che analizza, nell'esempio è la terza, quella che poi viene utilizzata per le simulazioni) è il seguente:

```
Time = 3e-06

Mesh stats
  points:          1212134
  faces:           3373623
  internal faces:  3255475
  cells:           1091599
  boundary patches: 7
  point zones:    0
  face zones:     0
  cell zones:     0

Overall number of cells of each type:
  hexahedra:      955141
  prisms:         14619
  wedges:         0
  pyramids:       0
  tet wedges:     1253
  tetrahedra:     9
  polyhedra:      120577

Checking topology...
  Boundary definition OK.
  Point usage OK.
  Upper triangular ordering OK.
  Face vertices OK.
  Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces
  Patch           Faces    Points  Surface topology
```

```
leftWall      703      767      ok
rightWall     823      896      ok
frontWall     480      525      ok
backWall      798      874      ok
bottomWall    1697     1953     ok
topWall       768      825      ok
vcg_patch0    112879   133305   multiply connected
<<Writing 2 conflicting points to set nonManifoldPoints
```

Checking geometry...

```
Overall domain bounding box (-120 -120 -195.00018)
                               (120 180 5)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness
    (2.1833044e-16 8.1812639e-17 1.2620801e-15) OK.
Max cell openness = 6.6852618e-16 OK.
Max aspect ratio = 20.36923 OK.
Minimum face area = 0.0075665734.
Maximum face area = 109.42539.
Face area magnitudes OK.
Min volume = 0.0041324411. Max volume = 1054.3459.
Total volume = 10752058. Cell volumes OK.
Mesh non-orthogonality Max: 64.57454 aver.: 7.6187795
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 2.7873394 OK.
```

Mesh OK.

Con questo controllo si hanno tutte le statistiche e si può verificare se i criteri impostati nella creazione della mesh di volume sono stati effettivamente rispettati. In caso contrario si hanno comunque buone indicazioni riguardo ai parametri da modificare. A questo punto, se le istruzioni sono eseguite una per volta, si può anche aprire il *software* Paraview (con il comando `paraFoam`, che legge direttamente i dati di OpenFOAM) per un controllo visivo. Da ultimo, dato che solo l'ultima mesh servirà per la simulazione, si possono cancellare le prime due mesh (la *castellated* e quella di *snap*).

4. `cellSet`

il comando `cellSet` mette da parte le celle dell'ultima zona di gola,

che non si vuole simulare. Non ci sono esigenze particolari, se non che in quella zona la cavità nasale si ricongiunge con quella orale e per il momento si vuole simulare la cavità fino alla rinofaringe (la zona per noi fondamentale è quella del setto nasale e dei turbinati, su cui operano i medici specializzati). Si può chiaramente vedere la zona che si eliminerà in figura 9.1, che mostra una clip della mesh di volume finale, in cui è stata nascosta la mesh interna.

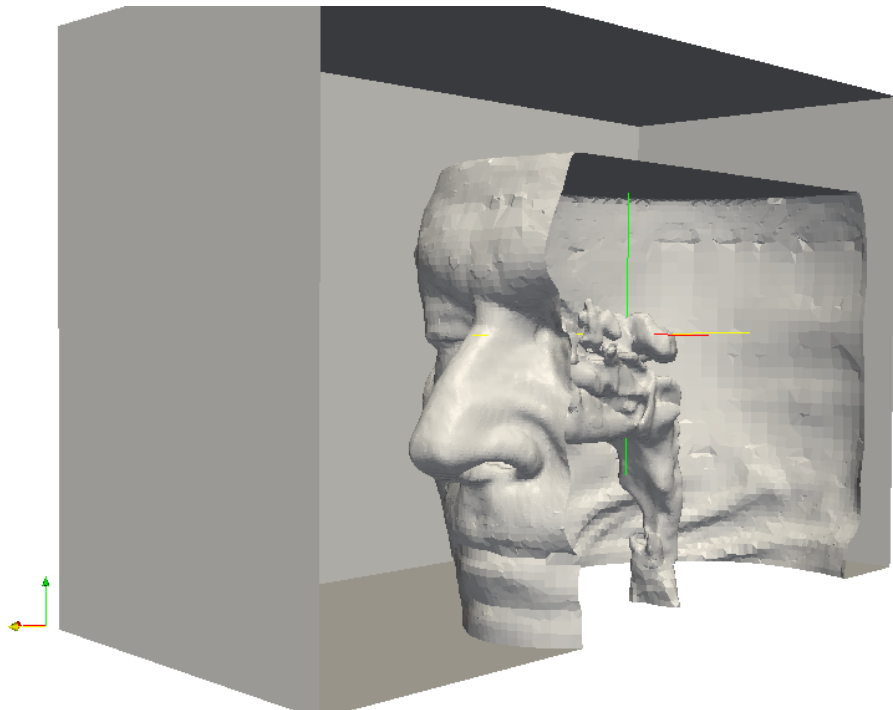


Figura 9.1: Sezione della mesh di volume completa

Si noti che si potrebbe evitare questo passaggio semplicemente definendo un box iniziale (nel *blockMeshDict*) con la faccia inferiore più in alto, ma in tal modo si troverebbe troppo vicina alle narici, che risentirebbero quindi delle condizioni al contorno. Inoltre con questo passaggio siamo in grado di specificare due altezze diverse per la patch di gola e la faccia inferiore, aumentando la flessibilità delle simulazioni. Il dizionario relativo è il seguente:

```
name          throat_cells;
action        new;

topoSetSources
```

```
(
  boxToCell
  {
    box (-0.02 -0.02 -0.1955) (0.02 0.02 -0.165);
  }
);
```

in cui “throat_cells” è il nome dell’insieme delle celle selezionate dal box definito.

5. `setSet`

```
...> cellSet throat_cells invert
...> quit
```

questi 3 comandi riprendono le celle definite al punto precedente e selezionano tutte le altre (comando `invert`), ovvero quelle da tenere per la simulazione.

6. `subSetMesh throat_cells`

crea la nuova mesh di volume con le celle appena definite. A questo punto si possono cancellare le precedenti e tenere solo quest’ultima.

7. `transformPoints -scale '(1e-3 1e-3 1e-3)'`

serve a trasformare la mesh dai mm ai metri.

8. `faceSet`

questo comando serve a mettere da parte le facce della zona di gola, che deve ancora essere definita come una *patch*. Le altre *patches* infatti sono il volto e la cavità nasale ² e le 6 facce del box iniziale definito dal *blockMesh*. Il dizionario si presenta così:

```
name          throat_faces;
action        new;

topoSetSources
(
  patchToFace
  {
    name "oldInternalFaces";
  }
)
```

²formano una *unica* che deriva direttamente dalla superficie .stl fornita in ingresso

in cui `throat_faces` è il nome dell'insieme di facce che sono state salvate, prese dalla `patch oldInternalFaces` che era stata automaticamente generata dal comando `subSetMesh`, quello che aveva definito la nuova mesh di volume³.

9. createPatch

a partire dalle facce definite al punto precedente crea una nuova `patch` nella zona di gola, su cui si dovranno poi imporre le condizioni al contorno. Il dizionario è il seguente:

```
matchTolerance 1E-3;
pointSync true;

patches
(
  {
    name throat;

    dictionary
    {
      type patch;
    }

    constructFrom set;
    set throat_faces;
  }
);
```

Questo comando crea una nuova cartella, con tutte le `patches` corrette. Come al solito si può procedere con l'eliminazione delle precedenti. A questo punto è tutto pronto per far partire le simulazioni vere e proprie, restano solo da copiare nell'ultima cartella creata⁴ i dizionari con le condizioni iniziali e al contorno, uno per ogni variabile del problema. Nel caso incomprimibile laminare questi dizionari sono solo 2, uno per la pressione e uno per la velocità. Li riportiamo sinteticamente per una simulazione stazionaria (in cui però si risolve un transitorio fittizio), in cui è fissata la differenza di pressione di $130m^2/s^2$ tra l'esterno e la gola.

³infatti la nuova mesh ha come facce di bordo alcune facce che in precedenza appartenevano all'interno della mesh, da cui il nome `oldInternalFaces`

⁴che per comodità si può anche rinominare "0"

Dizionario per la pressione:

```
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    leftWall
    {
        type          fixedValue;
        value         uniform 130;
    }
    ...
    topWall
    {
        type          fixedValue;
        value         uniform 130;
    }
    vcg_patch0
    {
        type          zeroGradient;
    }
    throat
    {
        type          fixedValue;
        value         uniform 0;
    }
}
```

in cui i tipi di condizione da imporre (`type`) dovrebbero essere evidenti. Il dizionario della velocità ha una struttura analoga e si presenta come:

```
dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (0 0 0);
boundaryField
{
    leftWall
    {
        type          zeroGradient;
    }
    ...
```



```

topWall
{
    type            zeroGradient;
}
vcg_patch0
{
    type            fixedValue;
    value           uniform (0 0 0);
}
throat
{
    type            zeroGradient;
}
}

```

in cui le condizioni imposte sono ovviamente di velocità nulla a parete (volto e cavità nasale) e gradiente nullo su tutte le *patches* di inlet e outlet.

10. `decomposePar`

nel caso in cui si utilizzino più processori si deve ovviamente suddividere il problema in un certo numero di sottodomini. I metodi di decomposizione sono molteplici⁵, nell'esempio seguente è riportato quello *hierarchical*, che in questo caso particolare suddivide il dominio globale (quello definito dal *blockMeshDict* in 8 sottodomini. Questo comando genera tante nuove cartelle (chiamate *processor..*) quanti sono i processori su cui si ha intenzione di lanciare la simulazione.

```

numberOfSubdomains 8;
method              hierarchical;
hierarchicalCoeffs
{
    n                (2 2 2);
    delta            0.001;
    order            zyx;
}

```

11. `mpirun -np 8 pimpleFoam -parallel`

Questo comando fa partire la simulazione, specificando il solutore e il fatto che si vogliono utilizzare 8 processori (`-np 8`) in parallelo.

⁵si rimanda come sempre alla guida di OpenFOAM per i dettagli e le alternative

12. `reconstructPar`

Una volta terminata la simulazione in parallelo bisogna ricostruire i dati dalle cartelle relative alla decomposizione nei vari processori. Questo comando genera nella cartella principale tutte le cartelle con le *time directories*, oppure solo le cartelle con gli istanti di interesse.

Il procedimento è a questo punto terminato e si può procedere con la fase di *post-processing*, affidata al software Paraview.

9.2 CILEA e il *cluster* Lagrange

Il costo computazionale delle simulazioni fluidodinamiche è un parametro fondamentale, che limita di fatto i risultati che si possono ragionevolmente raggiungere. I risultati presentati in questo lavoro di tesi non richiedono eccessiva potenza computazionale, tanto che per il caso laminare sono stati ottenuti su un normale PC portatile *quad-core*, con 4 GB di RAM. Il costo di una simulazione laminare “stazionaria”⁶ è di circa 2 ore, con una mesh da 2.5 milioni di celle. Questo tempo raddoppia circa per un’analogia simulazione nel caso turbolento (RANS), restando comunque pienamente accettabile.



Figura 9.2: Cluster Lagrange del CILEA

È però molto facile, soprattutto in previsione futura, aumentare i costi di calcolo fino a renderli insostenibili: basterebbe simulare un intero ciclo respiratorio (o più cicli) per rendere il costo anche 100 volte superiore. Allo stesso modo si potrebbe decidere di raffinare ulteriormente la geometria oppure di provare una simulazione LES. Si è quindi

⁶che viene sempre eseguita con un transitorio fittizio che va a regime

deciso di affidare i calcoli fluidodinamici al CILEA, che è il Consorzio Interuniversitario Lombardo per l'Elaborazione Automatica.

Il CILEA offre molti servizi, nel nostro caso è sufficiente l'utilizzo dei server dedicati al calcolo intensivo; in particolare è stato utilizzato il *cluster* chiamato "Lagrange" (in figura 9.2), costituito da 208 nodi con due processori *quad-core* ciascuno (Intel Xeon 3.16 GHz) e 16 GB di RAM per ogni nodo. Come curiosità riportiamo che il CILEA occupa attualmente la posizione numero 351 nella lista TOP500 dei supercomputers più performanti del mondo⁷.

9.3 Automatizzazione del procedimento

In questa sezione si vuole sinteticamente riportare la procedura che consente di eseguire automaticamente tutte le istruzioni di OpenFOAM e delle relative *utilities*, come spiegato per esteso nelle sezioni precedenti.

Questa procedura non è nient'altro che uno script con tutti i comandi da eseguire: infatti, per far partire una simulazione sul *cluster* del CILEA, si deve generare uno script con gli stessi comandi utilizzati finora⁸, oltre a qualche linea specifica per l'utilizzo corretto del server. In linea di principio si potrebbe includere in questo script solo l'istruzione relativa al solutore di OpenFOAM da utilizzare, copiando sul server direttamente la mesh di volume preparata in precedenza, ma risulta ovviamente più comodo affidare al CILEA tutti i passaggi, considerando anche che il costo computazionale della creazione della mesh può diventare impegnativo quando si creano geometrie con qualche milione di celle.

Restano in sospeso due questioni da chiarire: la prima è che l'automatizzazione è in realtà un semplice elenco di istruzioni, che include solo la parte relativa ad OpenFOAM. Per il momento non è infatti prevista l'automatizzazione del procedimento da eseguire in Slicer per generare la superficie del volto e della cavità nasale. La seconda questione riguarda invece il fatto che questa automatizzazione difficilmente funziona se si è alla prima simulazione: in tal caso è sicuramente meglio eseguire le istruzioni una per una per poter sistemare i parametri all'interno dei dizionari che regolano tutto il processo. Il punto più critico è infatti la costruzione della mesh di volume (*snappyHexMesh*), che deve portare ad una geome-

⁷ultimi dati aggiornati a Novembre 2010

⁸anche sul *cluster* Lagrange del CILEA è installato un sistema operativo GNU/Linux, con i principali software (tra cui OpenFOAM) per le simulazioni strutturali, fluidodinamiche. . .

tria con certi requisiti. In altre parole conviene automatizzare solo quei casi in cui si sa che la *mesh* si volume sarà generata correttamente, o comunque non ci sono grandi differenze nei due dizionari che costruiscono la mesh (*blockMeshDict* e *snappyHexMeshDict*). L'automatizzazione risulta molto utile ad esempio se si eseguono molte simulazioni con la stessa geometria o molte analisi di sensibilità con minime variazioni dei parametri.

```
#!/bin/sh -f
#
#PBS -N esempio
#PBS -r n
#PBS -j oe
#PBS -q general
#PBS -l select=1:ncpus=8:mpiprocs=8
#PBS -l place=free
cd $PBS_0_WORKDIR

source /data/apps/scripts/xopenmpi_gnu.sh
source /data/apps/scripts/openFoam.sh

NPROCS='wc -l < $PBS_NODEFILE'
echo "allocated on $NPROCS cpus"
blockMesh
snappyHexMesh
checkMesh
rm -r 1e-06
rm -r 2e-06
transformPoints -scale '(1e-3 1e-3 1e-3)'
cellSet
setSet < mesh_data.sh
subsetMesh throat_cells
rm -r VTK
rm -r 3e-06
mv 4e-06 0
faceSet
createPatch
rm -r 0
mv 1e-06 0
cp system/p 0/p
cp system/U 0/U
```

```
checkMesh
decomposePar
mpirun -machinefile $PBS_NODEFILE ...
      ... -np $NPROCS pimpleFoam -parallel
reconstructPar -latestTime
exit
```

Si dovrebbero riconoscere facilmente i comandi di cui si è già discusso nel capitolo 9. Di seguito riportiamo anche lo script `mesh_data.sh`, che permette di completare l'esecuzione della *utility* `setSet`.

```
cellSet throat_cells invert
quit
```

Da ultimo si riportano due grafici in cui si presenta brevemente il problema dei tempi di calcolo in funzione del numero di processori utilizzati e della dimensione della mesh del calcolo.

In figura 9.3 si vede l'andamento del tempo di calcolo, per una simulazione di esempio da 0 a 0.05 s, in funzione del numero di processori utilizzati. La parallelizzazione è evidentemente vantaggiosa, ma è chiaro che raddoppiando il numero dei processori non si possono dimezzare i tempi di calcolo. Questa graduale perdita di efficienza, dovuta in parte all'architettura del sistema *cluster* e in parte all'implementazione di OpenFOAM, rende poco conveniente richiedere più di 8 processori. Inoltre, all'aumentare del numero di processori sui quali viene ripartita l'analisi, aumenta moltissimo il numero di operazioni di I/O, che può portare a problemi di *memory contention*. Il compromesso più conveniente sarebbe quindi richiedere 4 processori, a patto di richiederli nello stesso nodo.

In questo esempio la mesh di volume consta di 2.5M di celle e per simulare 0.05 s di flusso laminare servono circa 5 ore con 8 processori. Se si volesse simulare un ciclo respiratorio laminare (ipotizziamo un ciclo di 2 secondi) si dovrebbero quindi spendere circa 200 ore con questa architettura e con questa mesh, un tempo di calcolo che comincia ad essere considerevole.

In figura 9.4 si riporta il tempo di calcolo in funzione del numero di celle, sempre per un simulazione d'esempio fino a 0.05 secondi. Il numero di celle è uno dei parametri più importanti, che va scelto come compromesso tra la precisione desiderata e le risorse (nonchè il tempo) che si hanno a disposizione.

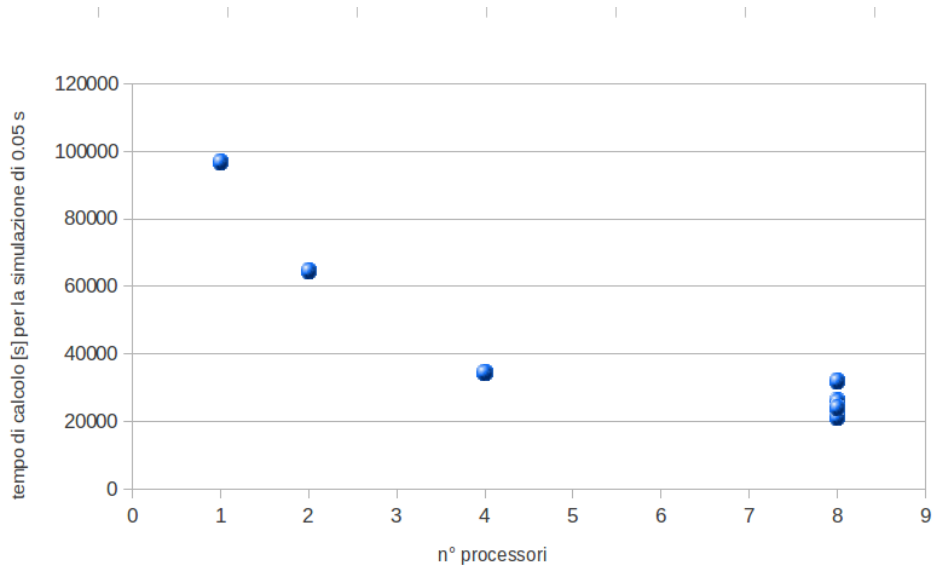


Figura 9.3: Tempo di calcolo in funzione del numero di processori

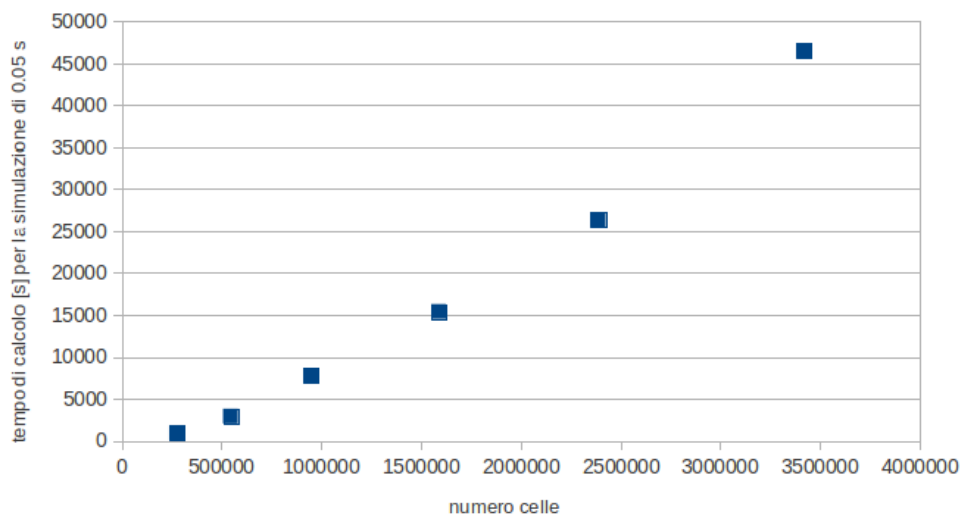


Figura 9.4: Tempo di calcolo in funzione del numero di celle

Capitolo 10

Analisi del flusso laminare e turbolento

10.1 Presentazione simulazione laminare

In questo capitolo si presentano i primi risultati ottenuti. Si tratta in entrambi di un'inspirazione a regime, a cui si arriva con un transitorio fittizio. Si presenta prima la simulazione laminare e poi quella turbolenta (RANS $k-\omega$). Come già detto l'obiettivo di questa tesi è quello di mettere a punto una procedura completa per la simulazione fluidodinamica del flusso nella cavità nasale: i seguenti risultati vogliono più che altro essere un esempio della correttezza di quanto fatto finora, lasciando ad un lavoro futuro uno studio più sistematico di possibili interventi chirurgici.

Si vuole accennare fin da subito al problema della presentazione dei risultati, oltre alla difficoltà di interpretare criticamente gli stessi al fine di ottenere informazioni clinicamente utili. Dalle analisi fluidodinamiche, anche quelle più semplici, si ottiene una notevole mole di risultati, che deve essere sintetizzata in poche informazioni utili. Inoltre, avendo una simulazione tridimensionale, si potranno presentare solo poche sezioni, ritenute più significative. Allo stesso modo, quando si ha una simulazione non stazionaria (transitorio o ciclo completo) bisogna anche pensare a come presentare l'evoluzione temporale della soluzione. A questi problemi si accennerà ancora nel capitolo 12, dedicato agli sviluppi futuri.

Serve quindi una certa esperienza per capire quali risultati possano essere più efficaci: per il momento si presentano alcune sezioni con le principali quantità fisiche (soprattutto pressioni e componenti della velocità). La simulazione dell'inspirazione laminare è stata portata a termine

su un PC portatile *quad-core* con 4 GB di RAM. Come già detto, è stata simulata un'inspirazione che arriva a regime attraverso un transitorio fittizio, avendo imposto una differenza di pressione costante tra l'esterno del volto e la zona di gola (che in questo caso è l'*outlet*). La mesh di volume, di cui si riportano le statistiche, è formata da poco più di 2 milioni di celle.

Mesh stats

```
points:          2370770
faces:           6664711
internal faces:  6477605
cells:           2165157
boundary patches: 8
point zones:    0
face zones:     0
cell zones:     0
```

Overall number of cells of each type:

```
hexahedra:      1913050
prisms:         24325
wedges:         1
pyramids:       0
tet wedges:     1758
tetrahedra:    24
polyhedra:     225999
```

Checking topology...

```
Boundary definition OK.
Point usage OK.
Upper triangular ordering OK.
Face vertices OK.
Number of regions: 1 (OK).
```

Checking patch topology for multiply connected surfaces ...

Patch	Faces	Points	Surface topology
leftWall	954	1022	ok
rightWall	954	1022	ok
frontWall	780	837	ok
backWall	930	1012	ok
bottomWall	1952	2225	ok
topWall	1080	1147	ok

vcg_patch0	179777	214959	ok
throat	679	711	ok

Checking geometry...

```

...
Boundary openness ... OK.
Max cell openness = 4.3414309e-16 OK.
Max aspect ratio = 24.89702 OK.
...
Face area magnitudes OK.
Total volume = 0.010786304. Cell volumes OK.
Mesh non-orthogonality Max: 64.49315 average: 7.1677019
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 2.9529652 OK.

```

Mesh OK.

Non si riportano i dizionari che regolano questa simulazione, perchè sono esattamente gli stessi presentati nel capitolo 9. Ricordiamo solo che la differenza di pressione è imposta a $160 Pa$, che divisi per la densità dell'aria (costante) a pressione e temperatura ambiente, portano ad imporre una differenza di $130 m^2/s^2$.

È utile ricordare che in queste simulazioni, tutte incompressibili, non conta il valore assoluto della pressione, ma solo quello relativo. Pertanto i valori che si leggeranno nelle immagini sono relativi ad un valore arbitrario di riferimento, posto a zero in gola.

Il solutore utilizzato è il *pimpleFoam* e la simulazione è stata effettuata da 0 a 0.75 s, salvando i risultati ogni millisecondo. Questo porta ad oltre 80 GB di dati, di cui però presenteremo solo quelli relativi all'istante finale, in cui la simulazione è a regime.

10.2 Presentazione simulazione turbolenta

La simulazione dell'inspirazione turbolenta è stata completata sfruttando il *cluster* Lagrange del CILEA. Analogamente a prima la simulazione va da 0 a 0.75 s (sempre con il solutore *pimpleFoam*), ma i dati sono stati salvati solo ogni 5 millesimi di secondo. Con una mesh da 2.5 milioni di celle si hanno quindi circa 55 GB di dati. A parità di mesh e di architettura di calcolo la simulazione turbolenta RANS $k - \omega - SST$

costa circa il doppio rispetto a quella laminare e ovviamente tra i risultati da analizzare ci sono anche le quantità turbolente (energia cinetica turbolenta k , ω , e viscosità turbolenta ν_t , che in OpenFOAM è chiamata `nut`).

I dizionari sono analoghi al caso laminare, con le opportune modifiche per impostare la simulazione turbolenta. Si devono solo aggiungere le condizioni iniziali e al contorno per le quantità turbolente.

Per l'energia cinetica turbolenta:

```
dimensions      [ 0 2 -2 0 0 0 0 ];
internalField   uniform 1e-6;
boundaryField
{
    leftWall
    {
        type          turbulentIntensityKineticEnergyInlet;
        intensity     0.01;
        value         uniform 1e-6;
    }
    ...
    vcg_patch0
    {
        type          kqRWallFunction;
        value         uniform 0;
    }
    throat
    {
        type          zeroGradient;
    }
}
```

Per la ω :

```
dimensions      [0 0 -1 0 0 0 0];
internalField   uniform 1;
boundaryField
{
    leftWall
    {
        type          fixedValue;
        value         uniform 1;
    }
}
```

```

    }
    ...
    vcg_patch0
    {
        type            omegaWallFunction;
        value           uniform 1;
    }
    throat
    {
        type            zeroGradient;
    }
}

```

E infine per la viscosità turbolenta:

```

dimensions      [ 0 2 -1 0 0 0 0 ];
internalField   uniform 0;
boundaryField
{
    leftWall
    {
        type            calculated;
        value           uniform 0;
    }
    vcg_patch0
    {
        type            nutWallFunction;
        value           uniform 0;
    }
}

```

I valori impostati sono frutto di esperienze simili¹ e di tentativi, ma uno dei pregi principali delle RANS $k - \omega$ è quello di essere poco sensibile ai dati in ingresso. Nella presente tesi non sono state effettuate sistematiche analisi di sensibilità su questi valori, quindi anche questo potrebbe rientrare in possibili validazioni future.

¹i tutorial e i forum di OpenFOAM risultano molto utili in questi casi

10.3 Analisi dei risultati

Passiamo quindi a presentare i risultati, per lo più come confronto qualitativo tra la soluzione laminare e turbolenta o tra parte destra e la sinistra della cavità nasale. Prima di passare alla visualizzazione per sezioni delle quantità fisiche in gioco, si riportano nelle figure 10.1 e 10.2 tre immagini in prospettiva, poco utili dal punto di vista dei risultati, però interessanti per avere una visione d'insieme del flusso nella cavità nasale. Nella prima si vedono in alcuni punti i vettori della velocità, con colore e lunghezza in funzione del modulo della stessa. Si vede chiaramente che all'esterno del naso la velocità non supera i 2 m/s, mentre raggiunge i valori massimi (circa 13 m/s) in gola. Nelle due figure 10.2 si hanno invece le linee di flusso, anch'esse colorate secondo il modulo della velocità. Valgono gli stessi commenti già fatti e in più si notano alcune linee di flusso che attraversano i seni paranasali, in particolare quelli sfenoidali. Gli altri seni sono pressochè esclusi dal flusso principale dell'aria, giustificando in parte la scelta di alcuni autori di escluderli a priori dalle simulazioni. Nella vista dall'alto si apprezza anche l'evidente asimmetria della cavità nasale e la complessità del flusso.

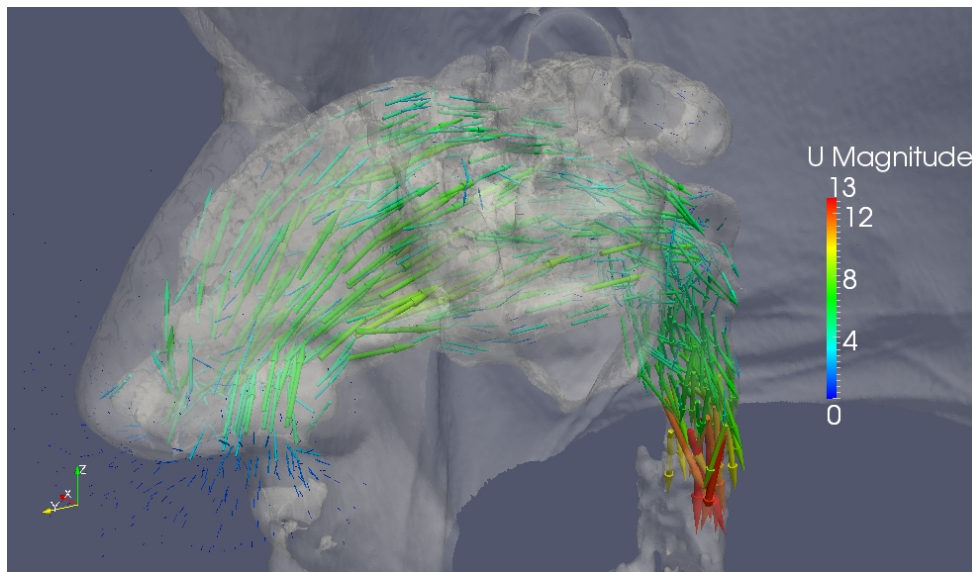


Figura 10.1: Foto 3D dei vettori velocità, per inspirazione laminare. I vettori hanno lunghezza e colore in funzione del modulo della velocità.

La sezione più significativa è probabilmente quella sagittale, che mostra tutta la cavità nasale dalle narici fino alla nasofaringe. Le prossime

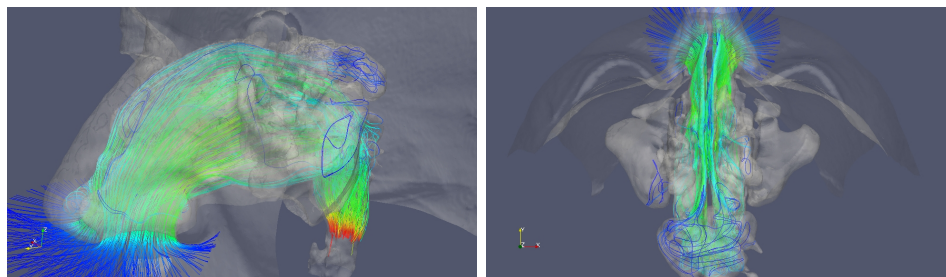


Figura 10.2: Foto 3D delle linee di flusso, viste dall'esterno del volto e dall'alto.

immagini riguardano due sezioni sagittali (piano $y-z$), una per la parte sinistra ($x = -0.001m$, nel sistema di riferimento che già dalla superficie *stl* iniziale si ha in *paraview*) e l'altra per la parte destra ($x = 0.01m$). Naturalmente non esiste un naso perfettamente simmetrico, le due sezioni scelte sono solo indicative, entrambe circa a metà del proprio condotto. Nelle figure 10.3 e 10.4 si vede l'andamento della pressione (in m^2/s^2), rispettivamente per la simulazione laminare e turbolenta. A parte le ovvie differenze geometriche tra le due sezioni destra e sinistra, la pressione non mostra particolari variazioni tra la simulazione laminare e quella turbolenta.

Nelle figure 10.5 e 10.6 sono mostrate le immagini del modulo della velocità, anch'esse divise per le sezioni destra/sinistra e per la simulazione laminare/turbolenta. Le velocità sono tutte paragonabili, ma nel caso turbolento le zone di flusso separato sono meno estese, sia nella zona delle narici che nella parte posteriore (coane). Inoltre, anche se non si nota dalle immagini ma solo dai video, una volta raggiunta la situazione di regime la simulazione laminare presenta molte più fluttuazioni, soprattutto nel tratto finale, dove i turbinati si riuniscono in un unico condotto che piega di 90° verso il basso, verso la nasofaringe. Questo aspetto si riscontra in tutte le immagini, quasi sicuramente a causa della viscosità turbolenta, come verrà mostrato più avanti.

Dalla figura 10.7 fino alla 10.10 si hanno invece i risultati a regime per le componenti della velocità lungo y (direzione longitudinale) e lungo z (direzione verticale), sempre per le due sezioni sagittali. L'altra direzione, trasversale alla cavità nasale, è poco utile in questa vista. Sia nella figura 10.7 che nella 10.8 si nota che nella cavità sinistra le velocità longitudinali sono maggiori in modulo², probabilmente a cau-

²i valori sono negativi perchè il flusso va dalle narici in gola, mentre il sistema di riferimento è nella direzione opposta

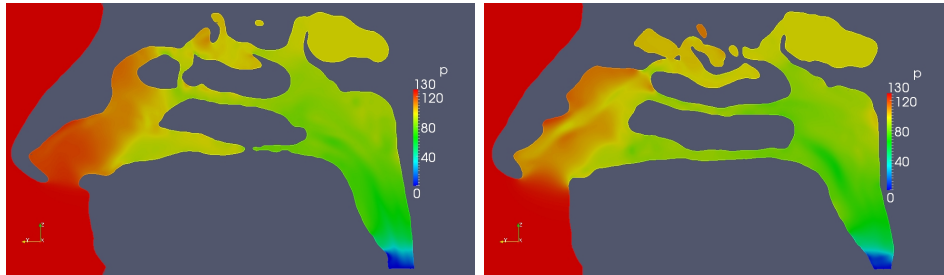


Figura 10.3: Simulazione laminare, pressione $[m^2/s^2]$ nelle sezioni sagittali di sinistra e destra. La pressione è in $[m^2/s^2]$ perché già divisa per la densità.

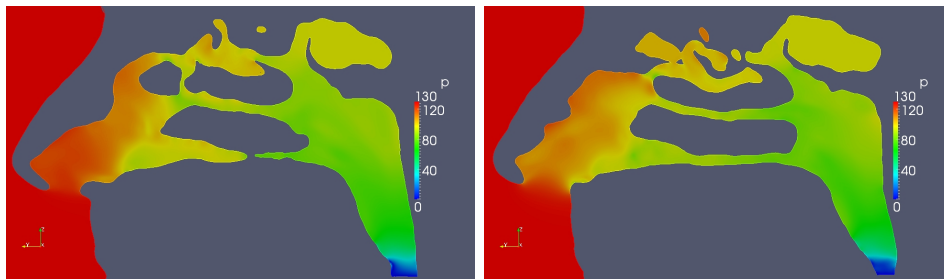


Figura 10.4: Simulazione turbolenta, pressione $[m^2/s^2]$ nelle sezioni sagittali di sinistra e destra.

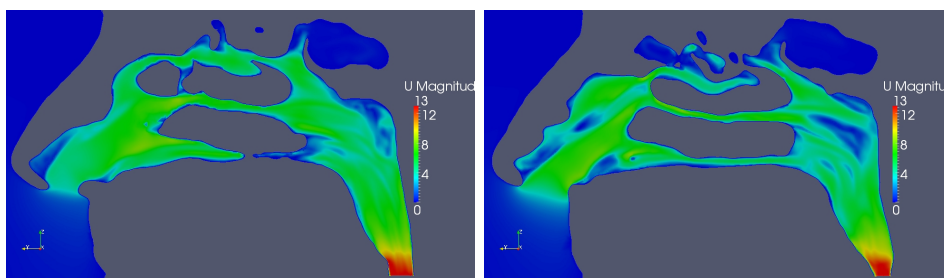


Figura 10.5: Simulazione laminare, modulo della velocità $[m/s]$ nelle sezioni sagittali di sinistra e destra.

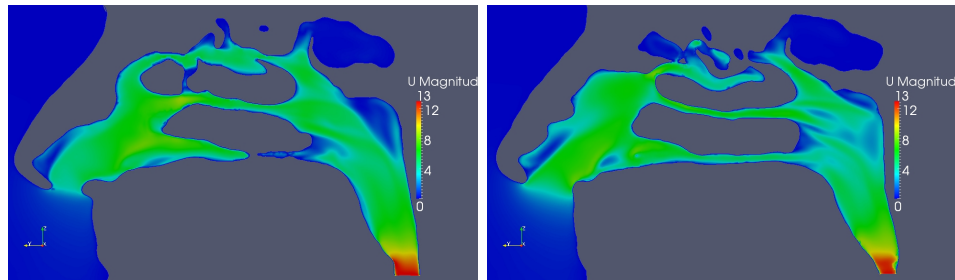


Figura 10.6: Simulazione turbolenta, modulo della velocità [m/s] nelle sezioni sagittali di sinistra e destra.

sa delle maggiori restrizioni. Infatti nella sezione sinistra in esame c'è un'evidente interruzione nel turbinato inferiore, una piccola ostruzione che potrebbe essere la causa delle velocità superiori. Anche in questo caso si nota che la simulazione turbolenta presenta ricircolazioni molto meno accentuate, specialmente nella parte finale del tratto considerato. Questo effetto è un po' meno evidente per la componente verticale della velocità (figure 10.9 e 10.10), che però mostra come le diverse geometrie delle due sezioni portino ad un comportamento nettamente diverso del flusso, che può condizionare negativamente la fisiologia della respirazione. Infatti nella cavità di destra il flusso principale dalle narici è diretto verso il turbinato medio, così come dovrebbe essere in un naso normale, mentre in quella sinistra anche il turbinato superiore presenta velocità paragonabili a quello medio.

L'ultima immagine di questa serie, in figura 10.11, mostra (solo per il caso laminare e per la parte destra della cavità) la componente ω_x della vorticità. Naturalmente i valori maggiori in modulo si hanno a parete, ma è molto interessante perché si distinguono le zone di separazione del flusso (si veda la zona delle narici o la zona posteriore dove si riuniscono i turbinati) e le zone a parete dove ci sono delle ricircolazioni.

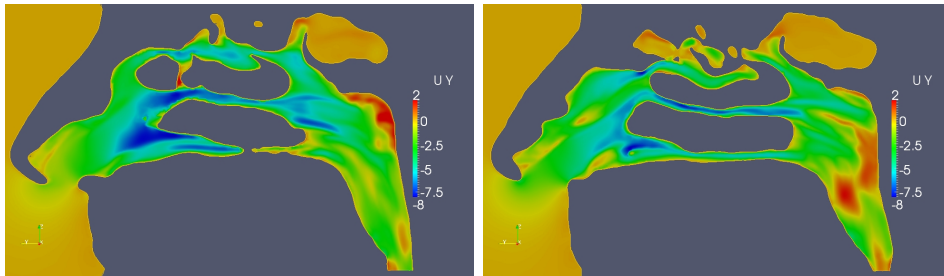


Figura 10.7: Simulazione laminare, velocità longitudinale (lungo y) [m/s] nelle sezioni sagittali di sinistra e destra.

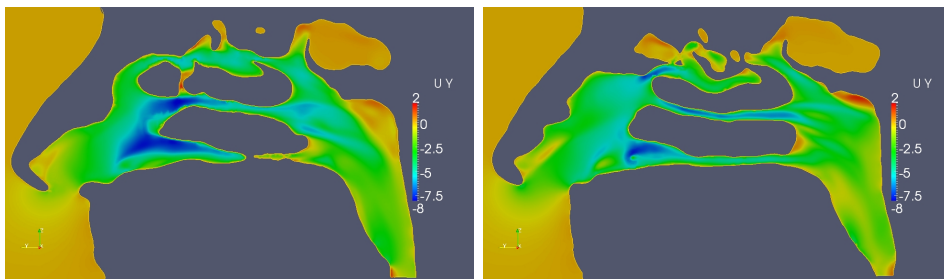


Figura 10.8: Simulazione turbolenta, velocità longitudinale (lungo y) [m/s] nelle sezioni sagittali di sinistra e destra.

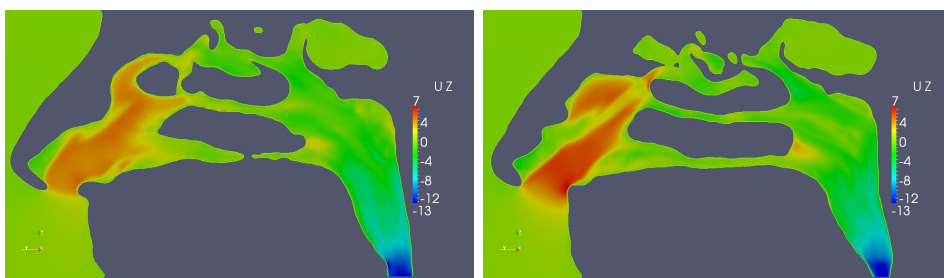


Figura 10.9: Simulazione laminare, velocità verticale (lungo z) [m/s] nelle sezioni sagittali di sinistra e destra.

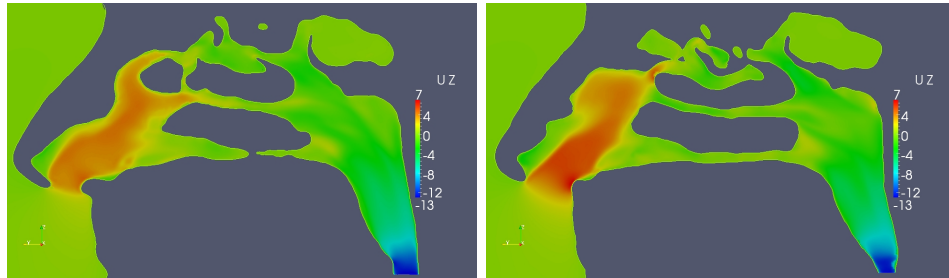


Figura 10.10: Simulazione turbolenta, velocità verticale (lungo z) [m/s] nelle sezioni sagittali di sinistra e destra.

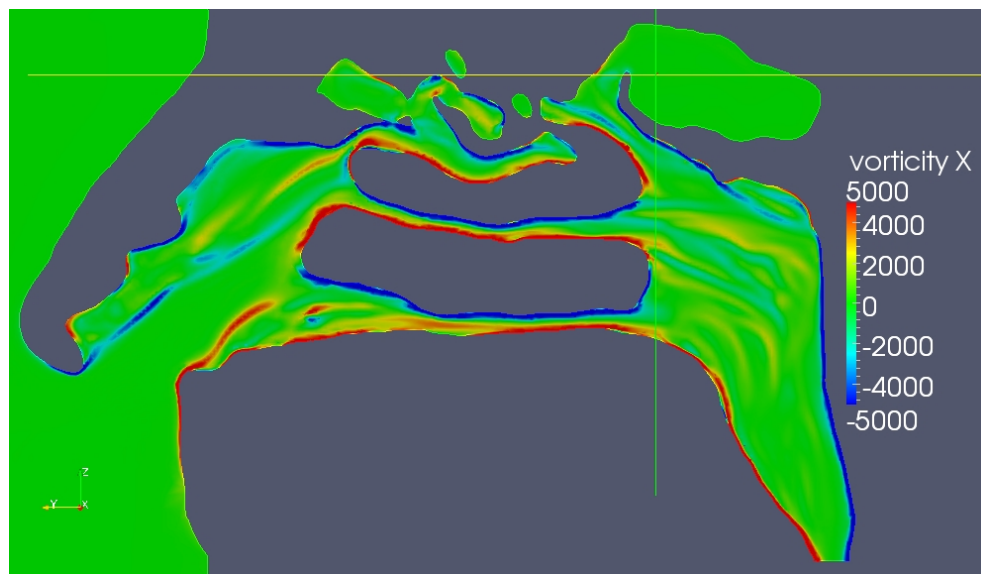


Figura 10.11: Simulazione laminare, vorticità lungo x [$1/s$] nella sezione sagittale destra.

Nella figura 10.12 si riportano le uniche immagini di una sezione assiale, in particolare di una sezione posta a $z = -0.16m$, vicino alla gola. Anche su queste sezioni si potrebbero fare molte considerazioni, ad esempio si possono studiare le ricircolazioni e i vortici in quella zona, ma dal punto di vista clinico siamo maggiormente interessati alle parti antecedenti (dalle narici ai turbinati), perchè le operazioni chirurgiche riguardano solo quelle zone (che sono ovviamente più critiche dal punto di vista della respirazione). Si riportano quindi solo la pressione e il modulo della velocità, per mostrare come il flusso sia tutt'altro che uniforme in questa sezione, vicina all'uscita. Questo fatto dimostra chiaramente che la scelta migliore per simulare il flusso nella cavità nasale sia quella di imporre una differenza di pressione tra l'esterno e la gola. Non è fisico, infatti, imporre un certo profilo di velocità all'efflusso, né tantomeno all'ingresso della cavità.

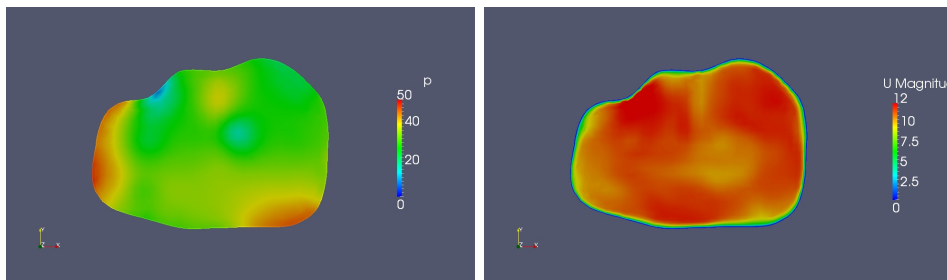


Figura 10.12: Simulazione laminare, pressione $[m^2/s^2]$ e modulo della velocità $[m/s]$ in una sezione assiale vicina alla gola.

Dalla figura 10.13 alla 10.16 si riportano la pressione e le tre componenti della velocità per una generica sezione dei turbinati, posta a $y = 0.06m$. Per ogni figura si vede il risultato della simulazione laminare e turbolenta. Le immagini sono prese dal davanti, quindi la cavità destra è quella che si vede sulla sinistra di ogni immagine e viceversa. Non si notano differenze evidenti tra le due simulazioni, il che conferma, almeno in prima battuta, la correttezza dei risultati. È molto interessante invece questa sezione, che è la parte centrale delle vie aeree superiori, sia per la geometria che per il flusso. Cominciando dalla pressione (figura 10.13) si nota che resta più elevata nella zona dei turbinati superiori, zona in cui ci sono i recettori olfattivi e dove le velocità sono più basse. C'è un'evidente dissimmetria tra la parte destra e la sinistra della cavità, come già notato nelle sezioni sagittali. La pressione è molto più bassa nella cavità destra, mentre nell'altra, nella zona del turbinato inferiore, è addirittura pari a quella che si ha nei seni mascellari. Anche solo da questa figura si

capisce che un possibile intervento chirurgico potrebbe avere come obiettivo quello di riportare un certo equilibrio tra le portate e le pressioni della parte destra e sinistra. Anche la figura 10.15 può fornire spunti interessanti per i medici, perchè risulta evidente che la velocità lungo y (perpendicolare al foglio) presenta un picco esattamente nel meato medio nella parte destra della cavità (sulla sinistra di ogni figura), mentre nell'altra parte il picco è più spostato in basso, verso il meato inferiore, probabilmente a causa delle ostruzioni nella parte sinistra della cavità nasale, di cui si è già parlato. Da ultimo, sempre qualitativamente, si può anche notare nel meato medio sinistro una regione di controcorrente, più evidente nel caso laminare, del tutto assente dall'altra parte.

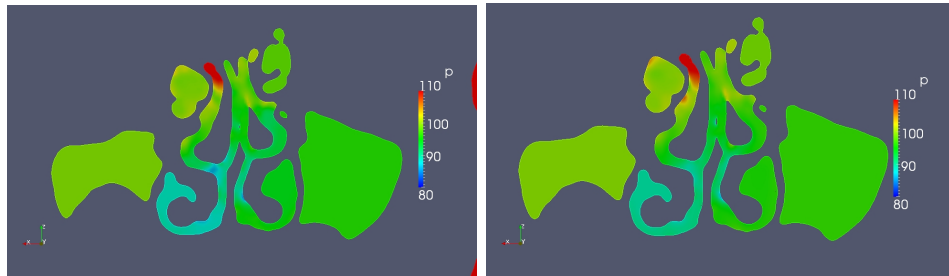


Figura 10.13: Sezione coronale dei turbinati, pressione [m^2/s^2] nel caso laminare (sinistra) e turbolento (destra).

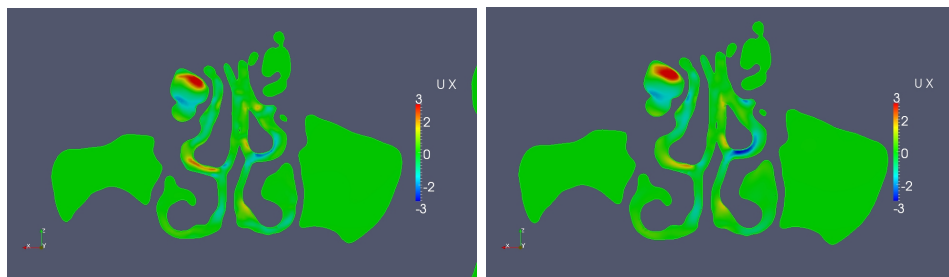


Figura 10.14: Sezione coronale dei turbinati, velocità lungo x [m/s] nel caso laminare (sinistra) e turbolento (destra)

Dalla figura 10.17 alla 10.20 sono riportate la pressione e le componenti della velocità per un'altra sezione coronale, anteriormente a quella dei turbinati, posta a $y = 0.09m$. Questa sezione si trova appena dopo le narici, nella zona detta "atrio nasale". Anche in questo caso sono molto più interessanti le differenze tra la zona destra e la sinistra, piuttosto che tra la simulazione laminare e quella turbolenta. Le asimmetrie sono

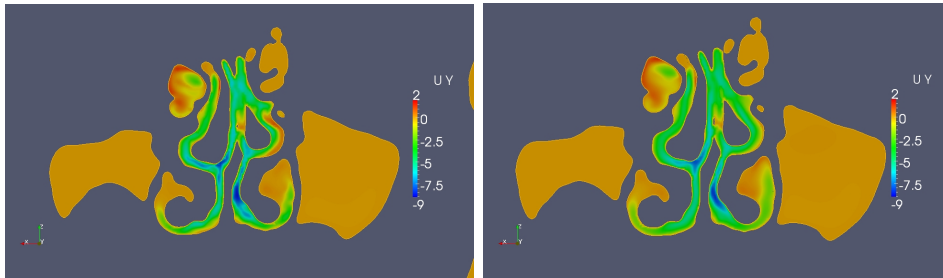


Figura 10.15: Sezione coronale dei turbinati, velocità lungo y [m/s] nel caso laminare (sinistra) e turbolento (destra).

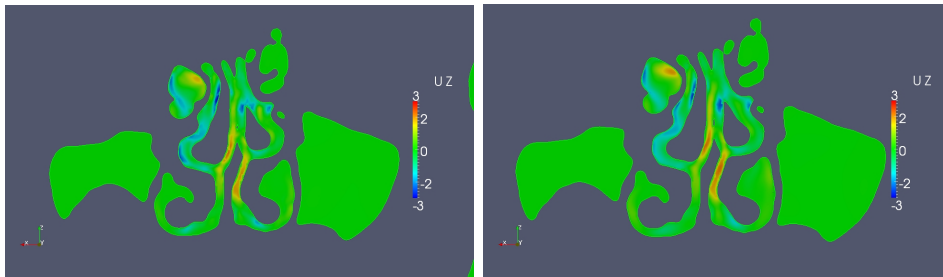


Figura 10.16: Sezione coronale dei turbinati, velocità lungo z [m/s] nel caso laminare (sinistra) e turbolento (destra).

sempre evidenti, in tutte le immagini: in particolare ci concentriamo ancora sulla componente lungo y della velocità (figura 10.19), in cui si vede un'estesa zona di controcorrente nella parte sinistra della cavità nasale.

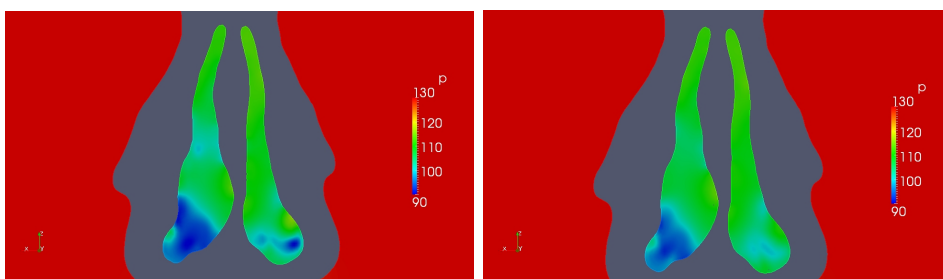


Figura 10.17: Sezione dell'atrio nasale (vicino alle narici), pressione [m^2/s^2] nel caso laminare (sinistra) e turbolento (destra).

Per concludere questo capitolo si presentano ancora le stesse sezioni già viste, con visualizzate le quantità turbolente. In figura 10.21 si legge il valore dell'energia cinetica turbolenta nella sezioni sagittali destra e sinistra. Si nota che la turbolenza non è certo trascurabile nel naso, in

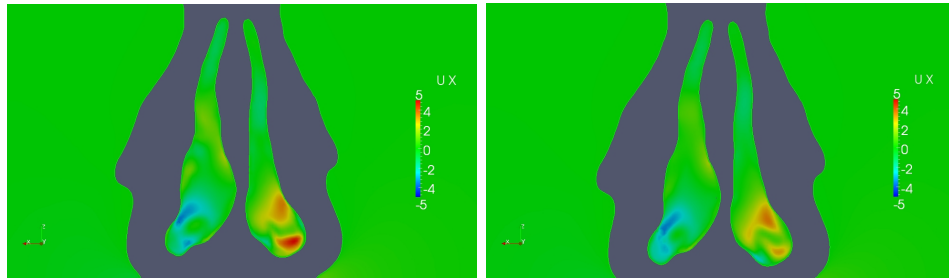


Figura 10.18: Sezione coronale dell'atrio nasale, velocità lungo x [m/s] nel caso laminare (sinistra) e turbolento (destra).

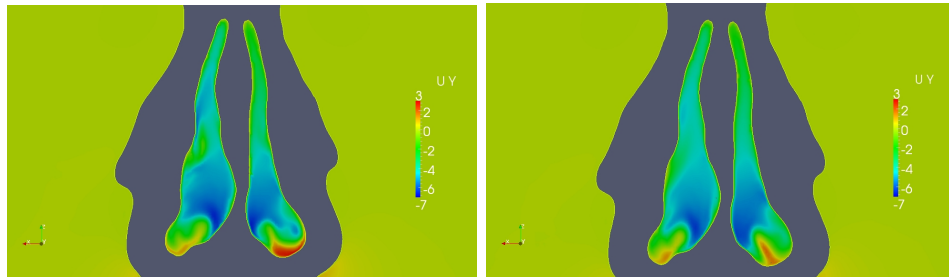


Figura 10.19: Sezione coronale dell'atrio nasale, velocità lungo y [m/s] nel caso laminare (sinistra) e turbolento (destra).

particolare nella sezione di destra c'è una zona di forte intensità nella zona di separazione dietro alla valvola nasale, mentre, al contrario, nella sezione di sinistra l'energia cinetica turbolenta è più elevata nella zona posteriore della cavità. In generale quindi si può affermare che con questa differenza di pressione imposta (e di conseguenza la portata) la simulazione laminare cade un po' in difetto, anche se si è visto che nella maggior parte del campo di moto non si rilevano comportamenti molto diversi per le due strategie di simulazione.

Nelle figure 10.22 e 10.23 si riportano le altre due grandezze fondamentali, ω e ν_t . I valori molto elevati (zone rosse) di ω indicano le zone di forte dissipazione dell'energia cinetica turbolenta, soprattutto vicino a parete. La viscosità turbolenta, che naturalmente ha un andamento analogo all'energia cinetica turbolenta, si concentra soprattutto nella zona posteriore della cavità e giustifica le minori fluttuazioni della simulazione turbolenta rispetto a quella laminare, quando si osservano i video di tutto il transitorio.

Le ultime due figure (10.24 e 10.25) mostrano ancora k e ω nella sezione dei turbinati e dell'atrio nasale. Infine, per concludere questa parte, si riportano in tabella 10.1 alcuni valori di portate nei vari tratti.

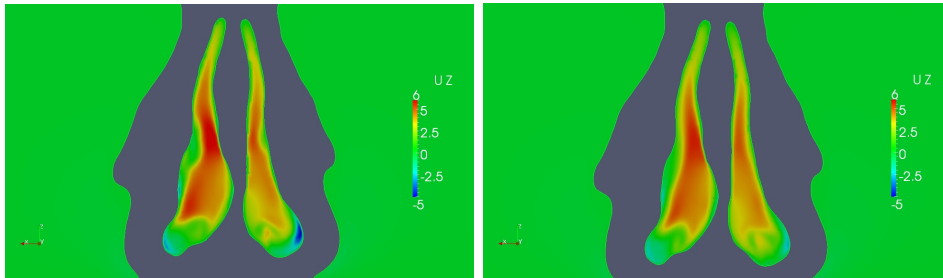


Figura 10.20: Sezione coronale dell'atrio nasale, velocità lungo z [m/s] nel caso laminare (sinistra) e turbolento (destra).

Come già si è notato dalle immagini, il flusso passa in quantità decisamente superiore dalle parte destra della cavità. Inoltre questi valori confermano che la portata maggiore si ha nel turbinato medio, seguito da quello inferiore, così come già noto dalla letteratura. Spesso questi valori globali sono molto utili per decidere come intervenire clinicamente, anche solo per avere un'idea complessiva del comportamento del flusso nella cavità nasale. Uno degli obiettivi futuri sarà sicuramente quello di mettere a punto una procedura completamente automatica per estrarre queste informazioni globali, senza alcun lavoro da parte dell'utente.

Tabella 10.1: Portate globali nei casi laminare e turbolento

	Laminare	Turbolento
Portata totale [lt/s]	1.322	1.330
Portata narice sinistra [lt/s]	0.586	0.563
Portata narice destra [lt/s]	0.786	0.777
Portata turbinati inferiori [lt/s]	0.480	0.498
Portata turbinati medi [lt/s]	0.667	0.667
Portata turbinati superiori [lt/s]	0.216	0.242

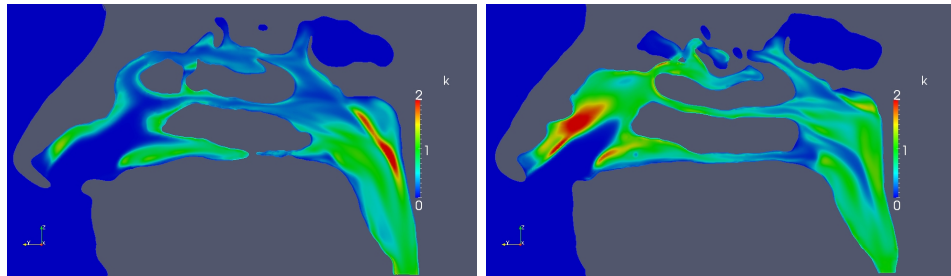


Figura 10.21: Simulazione turbolenta, energia cinetica turbolenta k [m^2/s^2] nelle sezioni sagittali di sinistra e destra.

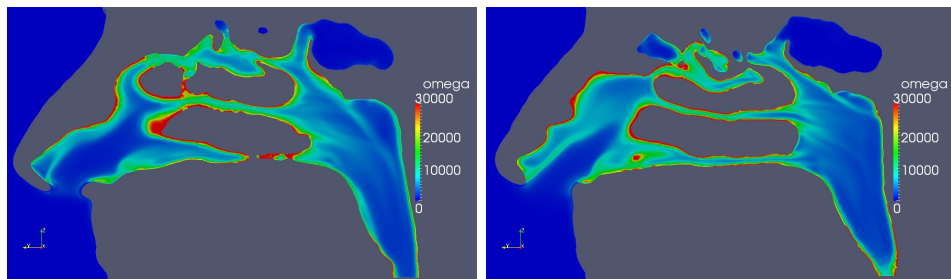


Figura 10.22: Simulazione turbolenta, frequenza della turbolenza ω [$1/s$] nelle sezioni sagittali di sinistra e destra.

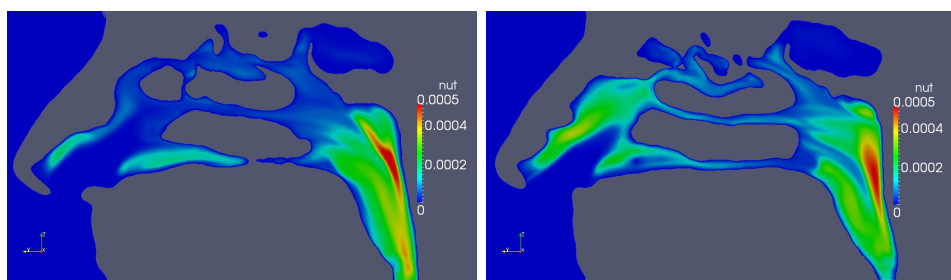


Figura 10.23: Simulazione turbolenta, viscosità turbolenta ν_t [m^2/s] nelle sezioni sagittali di sinistra e destra.

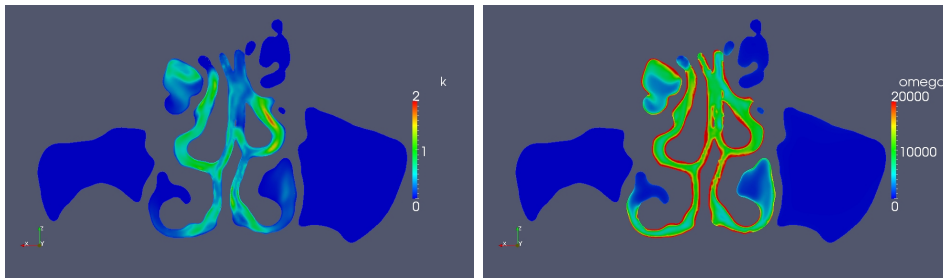


Figura 10.24: Simulazione turbolenta, $K [m^2/s^2]$ e $\omega [1/s]$ nella sezione coronale dei turbinati.

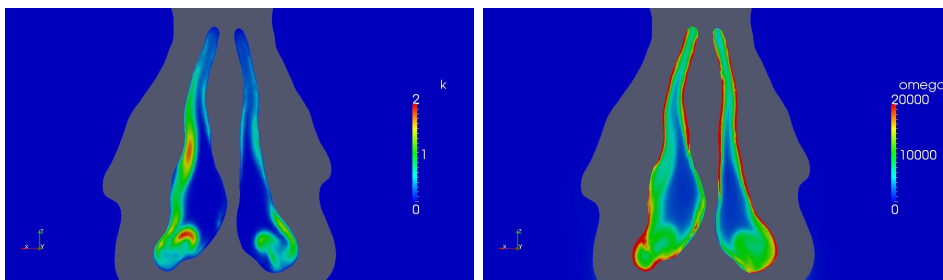


Figura 10.25: Simulazione turbolenta, $K [m^2/s^2]$ e $\omega [1/s]$ nella sezione coronale dell'atrio nasale.

Capitolo 11

Analisi di sensibilità

11.1 Sensibilità della soglia di Slicer

Le analisi di sensibilità sono fondamentali per capire la risposta della simulazione ai numerosi parametri in gioco. L'obiettivo è quello di diminuire le incertezze e le arbitrarietà nella procedura messa a punto, per dimostrare la ragionevolezza dei risultati. Non si tratta di una vera e propria validazione, però queste analisi forniscono tante indicazioni utili. I parametri su cui si può intervenire sono moltissimi e non è possibile variarli tutti, tantomeno provare molte combinazioni.

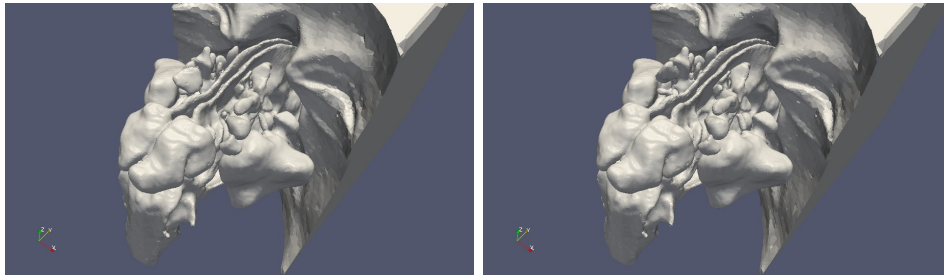
Anche per queste analisi si ripresenta il problema della visualizzazione dei risultati più significativi, dato che si hanno a disposizione più di 20 GB di dati. Verranno presentati confronti e dati delle portate, analogamente al capitolo precedente. L'unico parametro critico della prima parte della procedura, quella che porta dalle TAC alla superficie *stl*, è la soglia di Slicer. Questa soglia, collegata alla densità elettronica (come spiegato nel capitolo 4), regola la costruzione della superficie triangolata. Per l'analisi sono state indagate 7 diverse soglie, più un'altra con una ricostruzione della TAC meno raffinata (spaziatura assiale di $1.25mm$ invece che $0.625mm$). Nel dettaglio le analisi svolte sono riportate in tabella 11.1, naturalmente verranno presentati solo alcuni risultati di esempio.

Per ciascuna di queste ricostruzioni è stata portata a termine un'inspirazione laminare fino a 0.05 s, situazione già a regime. Le immagini mostrate si riferiscono a questo istante temporale. Prima di visualizzare i risultati è utile vedere, nelle figure 11.1 e 11.2 come le diverse soglie possano portare a geometrie differenti: una soglia più elevata (in modulo) tende a rifinire maggiormente la geometria, fino ad escludere le cavità

Tabella 11.1: Analisi di sensibilità della soglia di Slicer

	Spaziatura assiale [mm]	Soglia [HU]
Analisi 1: HQ 120	0.625	-120
Analisi 2: HQ 160	0.625	-160
Analisi 3: HQ 200	0.625	-200
Analisi 4: HQ 220	0.625	-220
Analisi 5: HQ 230	0.625	-230
Analisi 6: HQ 240	0.625	-240
Analisi 7: HQ 280	0.625	-280
Analisi 8: LQ 230	1.250	-230

nasali. A livello visivo le differenze sono importanti¹, ma si mostrerà che il flusso non cambia radicalmente a seconda della soglia scelta. Questo fatto è molto positivo, tenendo conto che le soglie analizzate hanno un range di 160 HU, mentre l'arbitrarietà della scelta (da parte di un utente esperto) può al massimo essere poche decine di HU.

**Figura 11.1:** Vista 3D delle varie ricostruzioni, soglia -200 HU (sinistra), -230 HU (destra)

Prima di visualizzare le immagini di confronto, riportiamo nella tabella 11.2 i principali dati relativi alle prime 7 simulazioni² per una sezione assiale vicino alla nasofaringe, analoga alla figura 10.12, per la precisione a $z = -0.15m$. Nella prima colonna si legge l'area della sezione in esame: come si è già detto, all'aumentare della soglia (in modulo) vengono escluse parti maggiori (la soglia in fondo è come un filtro), con una differenza di circa il 6% tra l'area più rifinita (HQ 280) e quella con la soglia più bassa (HQ 120). Sicuramente più interessante la colonna con le portate, che mostra una differenza del 15% tra la portata massima e

¹tipicamente aumentando troppo la soglia (in modulo) si rischia di escludere dei seni che nella realtà sono collegati, mentre i condotti principali risultano rimpiccioliti

²quelle con la risoluzione assiale più elevata

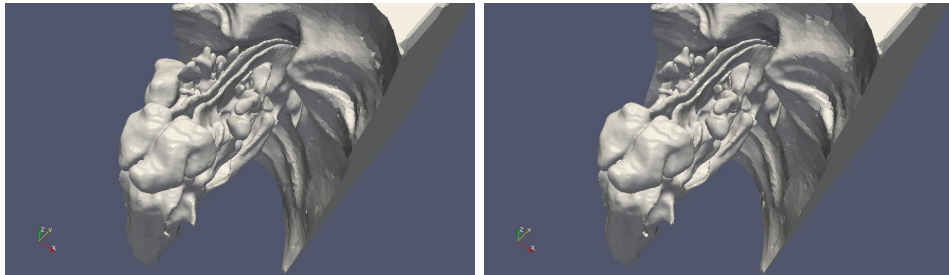


Figura 11.2: Vista 3D delle varie ricostruzioni, soglia -240 HU (sinitra), -280 HU (destra)

minima. Questa differenza non è certo trascurabile, ma è utile ricordare che sono state analizzate geometrie in un *range* di soglie molto più esteso rispetto ai valori ragionevoli su cui ci può essere qualche incertezza: ad esempio, per la TAC in esame, i valori corretti per la soglia possono oscillare tra i -200 e -230 HU.

Con questi dati sarebbe anche possibile procedere con una prima validazione: basterebbe infatti un'analisi *in-vivo* (rinomanometria) per avere il valore esatto della portata, da confrontare con quelli ricavati numericamente. Anche la pressione e il modulo della velocità ricalcano l'andamento dell'area e della portata, mentre le medie delle componenti della velocità (soprattutto U_x e U_y , nel piano in esame) dipendono dalla singola realizzazione.

Tabella 11.2: Analisi di sensibilità della soglia di Slicer, principali dati in una sezione assiale vicino alla nasofaringe, $z = -0.15m$. La pressione e le velocità sono le medie nella sezione.

	Area [cm^2]	Portata [lt/s]	Pressione [Pa]	U [m/s]	Ux [m/s]	Uy [m/s]	Uz [m/s]
HQ 120	2,673	1,485	68,786	5,854	-0,287	-1,820	-5,556
HQ 160	2,629	1,429	67,786	5,738	-0,323	-1,807	-5,436
HQ 200	2,583	1,380	65,513	5,644	-0,303	-1,789	-5,344
HQ 220	2,562	1,350	65,323	5,560	-0,180	-1,770	-5,268
HQ 230	2,552	1,345	64,946	5,549	-0,187	-1,727	-5,270
HQ 240	2,544	1,331	64,964	5,554	-0,178	-1,858	-5,231
HQ 280	2,518	1,266	63,238	5,338	-0,337	-1,765	-5,026

Nelle figure 11.3 e 11.4 si riportano 4 immagini relative alla pressione in una sezione sagittale in corrispondenza della cavità destra. Le immagini si riferiscono, nell'ordine, alle analisi HQ 160, HQ 200, HQ 240 e HQ 280. Si dovrebbero vedere le piccole differenze geometriche (la sezione rimpicciolisce impercettibilmente all'aumentare della soglia), evidenti soprattutto nella zona dei seni etmoidali, ma al contempo la soluzione è qualitativamente la stessa. Pertanto, se l'analisi fluidodinamica ha lo scopo di ottenere informazioni globali che possano essere utili per un intervento, si può affermare che anche senza ulteriori valutazioni la soglia di Slicer non è un parametro critico.

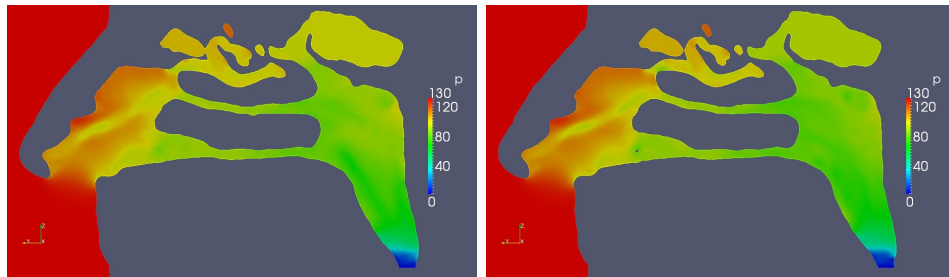


Figura 11.3: Sezione sagittale della cavità destra, pressione $[m^2/s^2]$ per le analisi HQ 160 (a sinistra) e HQ 200 (a destra).

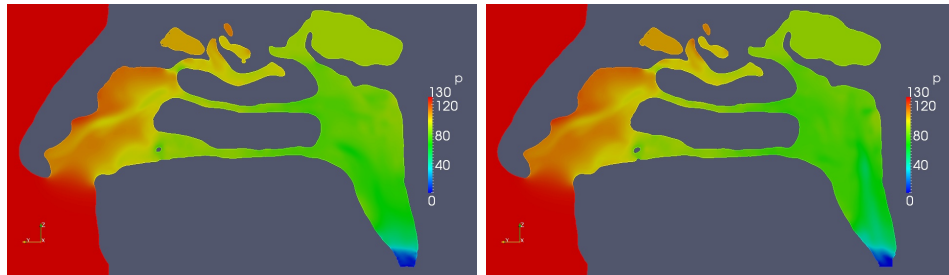


Figura 11.4: Sezione sagittale della cavità destra, pressione $[m^2/s^2]$ per le analisi HQ 240 (a sinistra) e HQ 280 (a destra).

Dalla figura 11.5 alla 11.8 si riportano le immagini della stessa sezione e delle stesse analisi (soglia a 160, 200, 240, 280) per il modulo della velocità e per la componente u_y della stessa. Dalle prime 4 immagini si nota che, a parte la zona posteriore in cui la simulazione ha delle fluttuazioni³, per il resto la soluzione non dipende sensibilmente dalla

³la simulazione è a regime, ma non è stazionaria e ci sono evidenti fluttuazioni soprattutto nella zona posteriore

soglia scelta. Anche in questo caso le maggiori differenze si vedono nella zona dei seni etmoidali e sfenoidale: il flusso al loro interno è infatti molto influenzato da piccole variazioni del flusso nei condotti principali, mentre non è vero il contrario. Gli stessi commenti valgono per le immagini relative alla componente y della velocità.

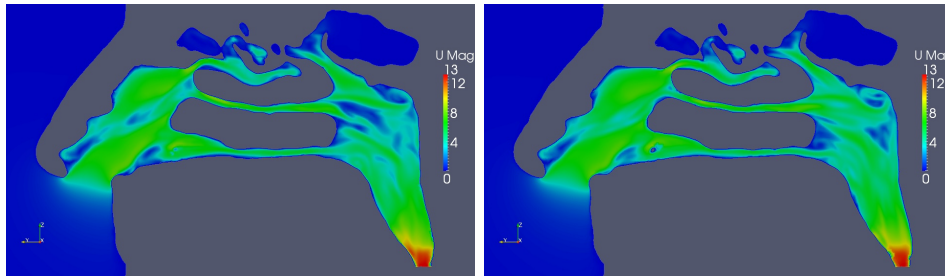


Figura 11.5: Sezione sagittale della cavità destra, modulo della velocità $[m/s]$ per le analisi HQ 160 (a sinistra) e HQ 200 (a destra).

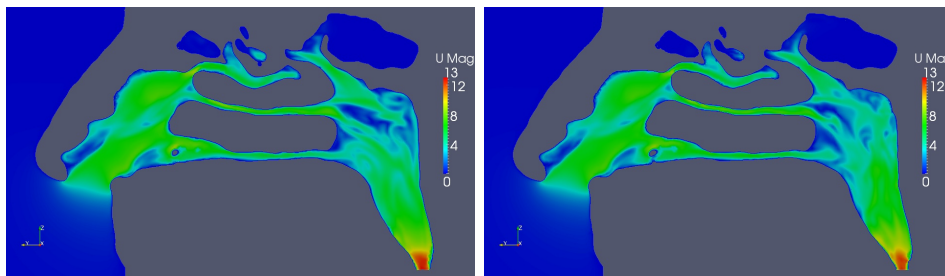


Figura 11.6: Sezione sagittale della cavità destra, modulo della velocità $[m/s]$ per le analisi HQ 240 (a sinistra) e HQ 280 (a destra).

Nelle figure 11.9, 11.10 e 11.11 si vede invece la soluzione in una sezione coronale in corrispondenza dei turbinati. Si riporta nelle prime due figure la pressione per le solite 4 analisi (soglia a 160, 200, 240, 280), mentre si riporta il modulo della velocità solo per le analisi HQ 160 e HQ 200. Per la velocità non si notano sostanziali differenze, mentre la pressione mostra notevoli variazioni tra le varie analisi. A parte il fatto che per la geometria con soglia a -240 HU viene escluso il seno mascellare destro e vengono esclusi entrambi con soglia a -280 HU, anche solo osservando il condotto principale si vede che la pressione è inferiore con le soglie più elevate in modulo. In particolare si vede un minimo molto più pronunciato nel meato inferiore sinistro, vicino ad un punto in cui invece la pressione ha un massimo causato da un'ostruzione nel condotto. Ov-

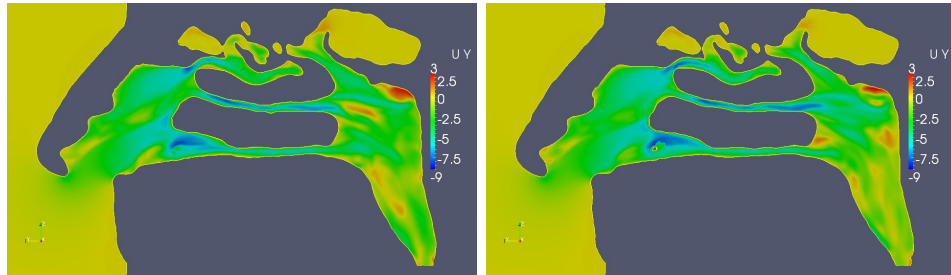


Figura 11.7: Sezione sagittale della cavità destra, velocità lungo y (U_y) [m/s] per le analisi HQ 160 (a sinistra) e HQ 200 (a destra).

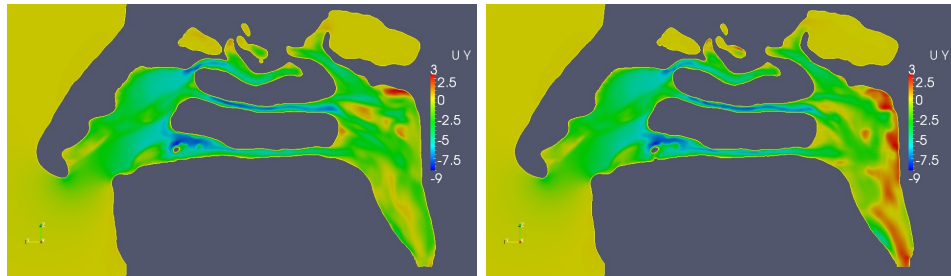


Figura 11.8: Sezione sagittale della cavità destra, velocità lungo y (U_y) [m/s] per le analisi HQ 240 (a sinistra) e HQ 280 (a destra).

viamente anche i seni (nelle figure si vedono quelli etmoidali) risentono della diminuzione della pressione nel condotto principale.

Per completare questa parte con qualche dato, si riportano nelle tabelle 11.3 e 11.4 pressioni, velocità e portate in questa sezione dei turbinati, divise per le varie analisi. Nella prima tabella le ultime 3 righe sono poco significative, perchè mancando uno o entrambi i seni mascellari l'area è drasticamente ridotta (diventa un terzo di quella iniziale) e di conseguenza anche le medie sono falsate. Questo problema non si ha con la seconda tabella, quella delle portate, in quanto i seni mascellari non portano alcun contributo alle portate.

Concentrandosi quindi sulle prime 5 analisi (dalla soglia di -120 alla -230), si ha ancora la conferma della diminuzione dell'area e della pressione all'aumentare (in modulo) della soglia, ma è importante notare che per le soglie più probabili (tra -200 e -230 HU) le differenze sono minime, sia per la pressione che per le velocità medie.

Nell'altra tabella (11.4), per ogni analisi, si legge il valore della portata divisa tra cavità destra e sinistra e tra le tre zone (inferiore, media e superiore) dei turbinati. Si noti che mentre non ci sono dubbi tra la cavità destra e la sinistra, la distinzione tra le varie zone dei turbinati e

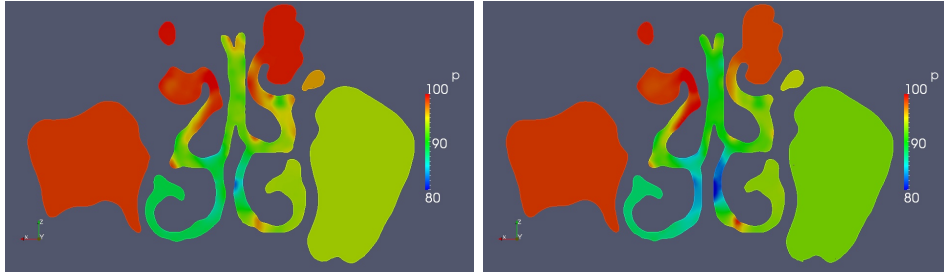


Figura 11.9: Sezione coronale dei turbinati, pressione [m^2/s^2] per le analisi HQ 160 (a sinistra) e HQ 200 (a destra).

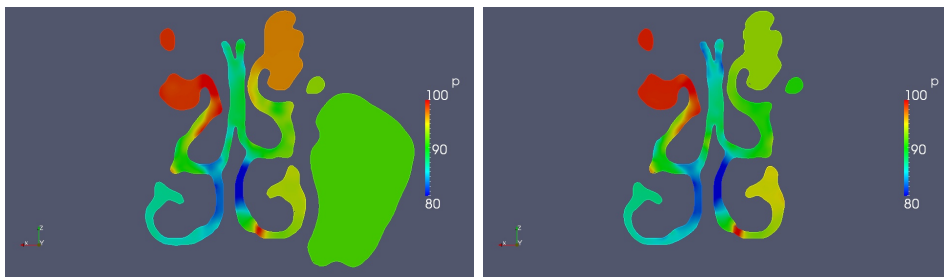


Figura 11.10: Sezione coronale dei turbinati, pressione [m^2/s^2] per le analisi HQ 240 (a sinistra) e HQ 280 (a destra).

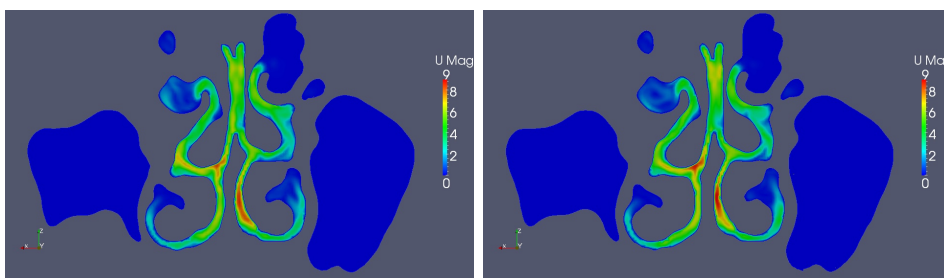


Figura 11.11: Sezione coronale dei turbinati, modulo della velocità [m/s] per le analisi HQ 160 (a sinistra) e HQ 200 (a destra).

dei meati è in parte arbitraria, complicata dal fatto che nessuna persona ha una cavità nasale perfettamente simmetrica. Per avere un'idea della suddivisione ci si è riferiti alla figura 2.5.

I dati confermano quanto già si sapeva dal capitolo precedente, ovvero c'è un forte squilibrio tra la portata nella parte destra e sinistra e il flusso passa principalmente nella zona del turbinato medio. I dati più interessanti sono però nell'ultima colonna, relativa alla portata nel turbinato superiore: all'aumentare della soglia (in modulo) si osserva una drastica diminuzione della portata, mentre nelle altre zone rimane più o meno costante. Quindi tutta la portata in meno è da imputarsi principalmente alla zona superiore della cavità.

Una possibile spiegazione potrebbe risiedere nel fatto che in questa zona i condotti sono molto più piccoli e complicati, rivelandosi così più sensibili alla variazione della soglia. Si può concludere allora che la fluidodinamica globale non è molto influenzata dal valore di soglia scelto, ma localmente le differenze possono essere significative. Questa informazione può essere utile se si tiene conto delle molteplici funzioni del naso oltre alla respirazione, tra cui anche l'olfatto⁴, l'umidificazione e il riscaldamento dell'aria inalata.

⁴i recettori olfattivi sono proprio nella zona superiore della cavità

Tabella 11.3: Analisi di sensibilità della soglia di Slicer, principali dati in una sezione coronale nella zona dei turbinati, $y = 0.055m$. La pressione e le velocità sono le medie nella sezione.

	Area [cm^2]	Pressione [Pa]	U [m/s]	Ux [m/s]	Uy [m/s]	Uz [m/s]
HQ 120	16,372	97,208	0,905	0,007	-0,903	-0,059
HQ 160	15,897	95,404	0,898	0,022	-0,897	-0,037
HQ 200	15,506	94,562	0,891	0,023	-0,890	-0,037
HQ 220	15,296	95,684	0,883	0,023	-0,882	-0,035
HQ 230	15,187	95,066	0,886	0,018	-0,885	-0,035
HQ 240	10,888	91,961	1,222	0,035	-1,221	-0,048
HQ 280	5,166	91,089	2,450	0,061	-2,449	-0,050
LQ 230	9,656	93,028	1,384	0,041	-1,383	-0,053

Tabella 11.4: Analisi di sensibilità della soglia di Slicer, portate in una sezione coronale nella zona dei turbinati, $y = 0.055m$. Sono riportate le portate (tutte in lt/s) per la cavità sinistra/destra, per le zone inferiore, media e superiore dei turbinati.

	Port. sx	Port. dx	Port. turb. inf.	Port. turb. medio	Port. turb. sup.
HQ 120	0,649	0,725	0,434	0,503	0,435
HQ 160	0,601	0,693	0,398	0,497	0,397
HQ 200	0,593	0,656	0,394	0,478	0,375
HQ 220	0,559	0,632	0,378	0,487	0,325
HQ 230	0,565	0,631	0,397	0,479	0,318
HQ 240	0,551	0,631	0,392	0,482	0,308
HQ 280	0,519	0,588	0,386	0,456	0,264
LQ 230	0,551	0,644	0,387	0,497	0,310

11.2 Sensibilità parametri OpenFOAM

Nella procedura che porta alla simulazione fluidodinamica della cavità nasale ci sono molti altri parametri di cui può risultare utile studiarne la sensibilità. Non è pensabile ovviamente indagarli tutti, ci limiteremo a presentare qualche risultato relativo alla dimensione della mesh di volume e da ultimo qualche prova variando due parametri di OpenFOAM.

Sono state provate 4 diverse mesh di volume, elencate nella tabella 11.5 dalla più grossolana alla più raffinata, variando i parametri del dizionario che regola l'utility *blockMesh*. Per ciascuna è stata eseguita un'inspirazione di transitorio che va a regime (da 0 a 0.05 s), sia nel caso laminare che nel caso turbolento (RANS, $k-\omega-SST$). Non si riportano i dettagli del *blockMeshDict* per le diverse analisi, in quanto inessenziali nella presente trattazione.

Tabella 11.5: Analisi di sensibilità della dimensione della mesh di volume

	Numero totale celle
mesh 1	950 000
mesh 2	1 600 000
mesh 3	2 500 000
mesh 4	3 500 000

Dimensione mesh, caso laminare

Partendo dal caso laminare, si riportano nelle figure 11.12 e 11.13 4 immagini con la sezione coronale dei turbinati. La mesh più grossolana, oltre ad avere già visivamente una qualità poco soddisfacente, non ha i seni mascellari. Infatti, partendo dalla mesh strutturata generata dal *blockMesh*, se questa è troppo grossolana non si ha nessuna cella nel punto di passaggio e quindi vengono esclusi. Però è diverso da prima, in questo caso è *snappyHexMesh* ad escludere talvolta i seni paranasali, mentre in precedenza (quando si è trattata la soglia di Slicer) quando mancavano era perchè erano già assenti nel file *stl* generato da Slicer.

Se si guarda il condotto principale (meati e turbinati) la soluzione è qualitativamente la stessa, le differenze più significative si riscontrano nella pressione all'interno dei seni mascellari e delle cellule etmoidali, che non influenzano il flusso principale.

Nelle figure 11.14 e 11.15 si riportano pressioni e modulo della velocità per la mesh più grossolana e per quella più raffinata, nella solita

sezione sagittale della cavità destra. Anche in questo caso si conferma, seppur solo qualitativamente, la robustezza dei risultati al variare del numero di celle. Nella tabella 11.6 si riportano le portate per le 4 mesh analizzate. I valori sono ragionevolmente vicini.

Tabella 11.6: Analisi di sensibilità della dimensione della mesh di volume, portate globali

	Portata [lt/s]
mesh 1	1.326
mesh 2	1.324
mesh 3	1.331
mesh 4	1.330

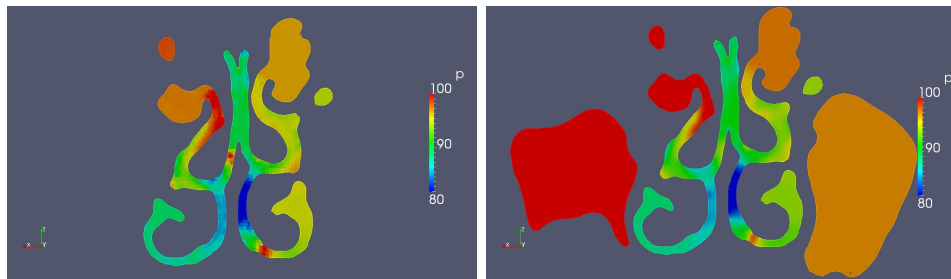


Figura 11.12: Sezione coronale dei turbinati, pressione [m^2/s^2] per le analisi laminari, mesh 1 (a sinistra), mesh 2 (a destra).

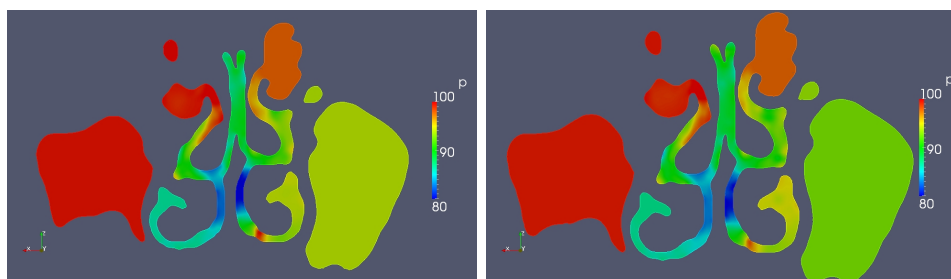


Figura 11.13: Sezione coronale dei turbinati, pressione [m^2/s^2] per le analisi laminari, mesh 3 (a sinistra), mesh 4 (a destra).

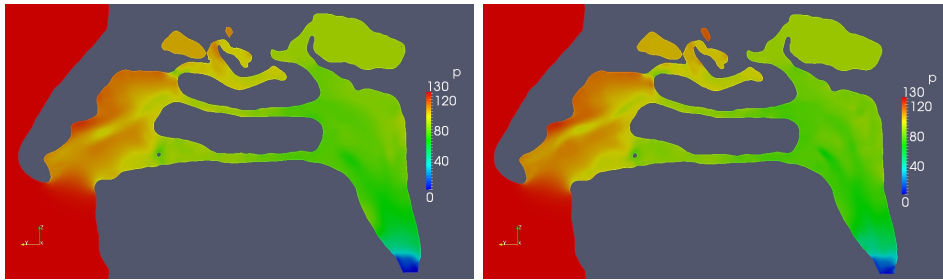


Figura 11.14: Sezione sagittale della cavità destra, pressione [m^2/s^2] per le analisi laminari, mesh 1 (a sinistra), mesh 4 (a destra).

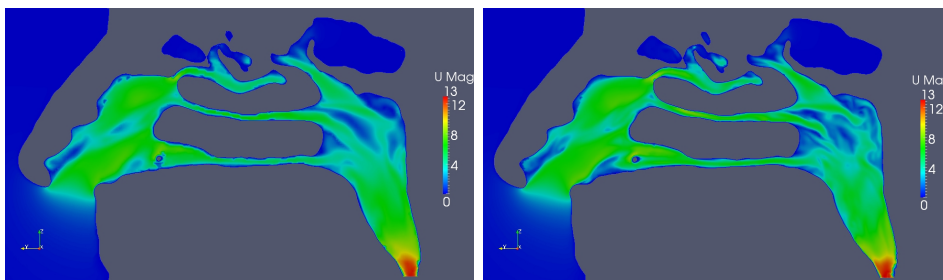


Figura 11.15: Sezione sagittale della cavità destra, modulo della velocità [m/s] per le analisi laminari, mesh 1 (a sinistra), mesh 4 (a destra).

Dimensione mesh, caso turbolento

Per quanto riguarda il caso turbolento si riportano solo 4 immagini, nelle figure 11.16 e 11.17, con i valori di k e ω per la mesh 1 (quella meno raffinata) e la mesh 3 (che è la mesh di base, utilizzata nella maggior parte delle analisi precedenti). Anche in questo caso le immagini mostrano, almeno qualitativamente, un buon accordo tra le due soluzioni.

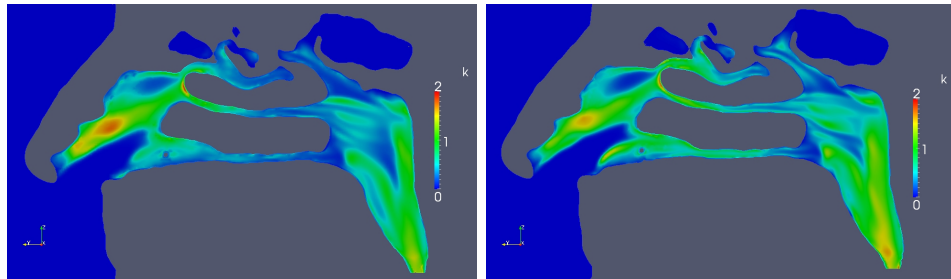


Figura 11.16: Sezione sagittale della cavità destra, energia cinetica turbolenta (k), mesh 1 (a sinistra), mesh 3 (a destra).

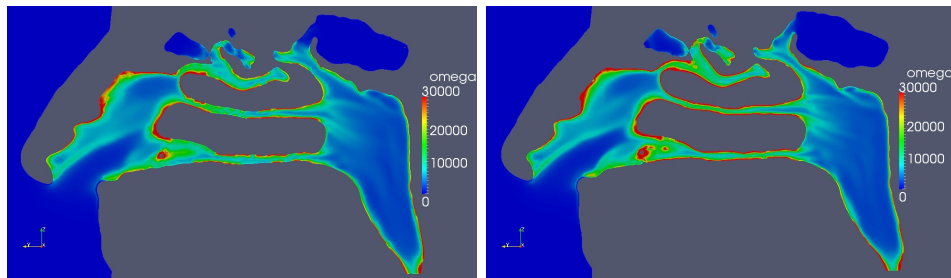


Figura 11.17: Sezione sagittale della cavità destra, ω [s^{-1}], mesh 1 (a sinistra), mesh 3 (a destra).

Parametri di snappyHexMesh

Da ultimo si riportano 6 immagini, più che altro di esempio, della sensibilità ai parametri di *snappyHexMesh*. I parametri, di cui si è già parlato nel capitolo 8, sono molteplici, tra cui quelli più significativi sono:

1. *nCellsBetweenLevels*, layers di celle tra due diversi *refinement levels*;
2. *refinementSurfaces*, se si aumenta il livello si ricostruisce meglio il volto esterno;

3. *refinementRegions*, ha l'effetto di cambiare (notevolmente) il numero di celle, effetto già analizzato in precedenza;
4. *maxNonOrtho*, massima non-ortogonalità permessa;
5. *maxConcave*, massima concavità permessa;
6. *maxBoundarySkewness* e *maxInternalSkewness*;
7. *minVolRatio*, minimo rapporto di volumi di due celle adiacenti;
8. *minDeterminant*, determinante delle celle (per un esaedro vale 1);
9. *layers*, numero di layer da aggiungere;
10. *expansionRatio*, è di quanto si muove la mesh intermedia per creare lo spazio per i layers;
11. *minThickness*, minimo spessore del layer più vicino a parete.

Non sono tutti importanti allo stesso modo ed è difficile valutare gli effetti della variazione di solo uno di questi parametri⁵. Pertanto, come studio di sensibilità, si è deciso per il momento di eseguire solo 4 analisi, due variando il livello di *refinementSurfaces* (default è 2, sono stati provati i livelli 1 e 3), due variando i *layers* (di default erano 3).

La modifica del livello di *refinementSurfaces* non modifica sensibilmente la mesh, tende solo a raffinarla nella zona del volto, lasciando praticamente invariato il resto del volume, come si vede in figura 11.18. Dato che la zona esterna alla cavità nasale, per quanto importante per le condizioni al contorno, non necessita di particolari infittimenti, la modifica è ininfluenza per la soluzione.

Nelle figure 11.19 e 11.20 si riportano invece 2 immagini relative all'analisi con un diverso numero di layers, con due ingrandimenti per visualizzare anche la modifica alla mesh di volume. Anche in questo caso non si rilevano sostanziali differenze tra le mesh, però è importante scegliere il numero corretto di layers da inserire vicino a parete, soprattutto nel caso di analisi RANS. Ad esempio se tra gli sviluppi futuri si proveranno altri modelli di turbolenza potrebbe essere necessario cambiare il numero di layers.

⁵si pensi ad esempio a come si possa valutare l'effetto di modificare *maxConcave* o *maxNonOrtho* senza toccare il resto

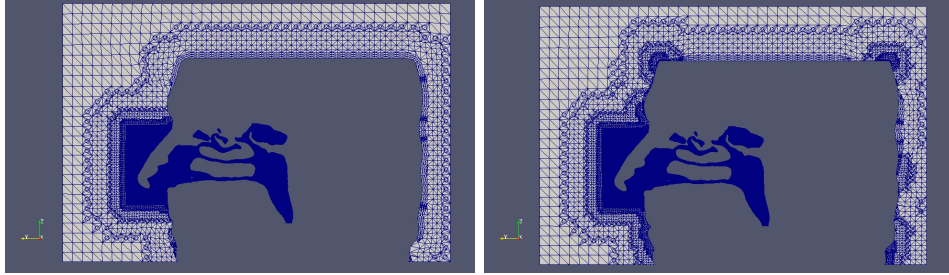


Figura 11.18: Sezione sagittale della cavità destra, sensibilità della mesh alla variazione del refinementSurfaces, refinement 1 (a sinistra), refinement 3 (a destra).

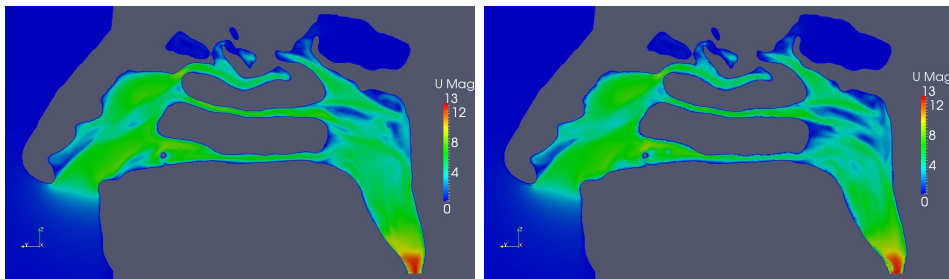


Figura 11.19: Sezione sagittale della cavità destra, modulo della velocità $[m/s]$ con 1 layer (a sinistra), 5 layers (a destra). Dal punto di vista qualitativo non ci sono differenze sostanziali.

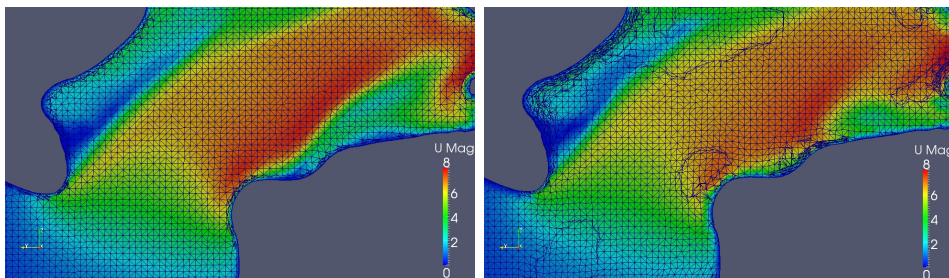


Figura 11.20: Sezione sagittale della cavità destra, zoom della figura 11.19 (zona delle narici), con in evidenza la mesh.

Parte IV
Conclusioni

Capitolo 12

Valutazioni finali

12.1 Risultati raggiunti

Il presente lavoro di tesi conclude il primo passo della collaborazione con la clinica San Paolo di Milano. I primi obiettivi del progetto sono stati raggiunti, ovvero è stata sviluppata e verificata, per quanto possibile, una procedura in grado di portare a termine con successo una realistica simulazione fluidodinamica della cavità nasale.

Scendendo un po' più nel dettaglio, il primo punto affrontato è stato il passaggio dalle scansioni TAC alla definizione della superficie di interesse. A tale scopo è stato utilizzato il software open-source 3D Slicer (capitolo 4), che si è rivelato molto utile ed efficace. Infatti, a differenza di altri casi visti in letteratura, consente in breve tempo (non più di 15 minuti), di arrivare alla superficie *stl* della cavità nasale e del volto, già pronta per essere utilizzata in OpenFOAM. Si pensi che solo nel 2004, come riportato nell'articolo [18], i tempi tipici per ottenere e raffinare una mesh di volume erano dell'ordine dei mesi.

Questa parte della procedura non è stata automatizzata perchè, come è stato evidenziato, resta una decisione arbitraria della soglia di Slicer (il valore della densità elettronica) che genera la superficie; tale decisione non è critica, ma richiede comunque un utente con un minimo di esperienza. In ogni caso il tempo richiesto è davvero ridotto e consente di analizzare una casistica molto ampia. Quanto detto finora non preclude la possibilità di un'automatizzazione futura, accettando una soglia che possa ragionevolmente andare bene per tutte le scansioni TC (ad esempio -200 HU), verificando a posteriori la correttezza della ricostruzione.

Il secondo punto affrontato è stata la verifica visiva della superficie ricostruita, con l'applicazione di eventuali filtri (di correzione o di

smoothing). Il software utilizzato è MeshLab, sempre open-source. Questa parte della procedura, anch'essa poco costosa in termini di tempo (5 minuti), non è in realtà necessaria e può essere eliminata.

Tra gli sviluppi futuri si può invece considerare la possibilità di studiare più a fondo le potenzialità di questi software, ad esempio nel caso si volesse aumentare il numero di celle di calcolo in OpenFOAM e servisse quindi una qualità più elevata della superficie *stl*. MeshLab si potrebbe rivelare molto utile in futuro per modificare la geometria, al fine di simulare numericamente gli interventi chirurgici.

Da ultimo si è messa a punto la simulazione fluidodinamica vera e propria, che è la parte centrale del presente lavoro di tesi, utilizzando il software OpenFOAM. Sono state analizzate simulazioni laminari e turbolente (RANS modello $k-\omega-SST$), per inspirazioni di transitorio che arrivano a regime. Sono state provate anche simulazioni stazionarie, che portano alle stesse soluzioni. Tutta questa parte, grazie alla semplicità e generalità dei comandi di OpenFOAM, è stata completamente automatizzata. Inoltre tutte le simulazioni sono state portate a termine con calcolo in parallelo, sfruttando il server Lagrange del CILEA. Questa parte, se non servono altre modifiche, non richiede tempo all'utente, ma è ovviamente molto costosa in ore-macchina. Si può stimare che con una mesh di volume abbastanza dettagliata (2 milioni di celle¹), una simulazione stazionaria² richiede circa 2 ore con 4 processori. Questo tempo circa raddoppia nel caso di simulazione stazionaria turbolenta e può diventare anche 100 volte per ogni ciclo respiratorio. I costi cominciano quindi a diventare importanti, il che giustifica l'utilizzo di potenti server esterni, soprattutto se considera che per ogni paziente si dovranno eseguire più analisi (situazione pre- e post-operatoria, simulazione di diversi interventi...).

In conclusione si può affermare che la procedura è pronta e già utilizzabile in casi di interesse clinico. Inoltre si è dimostrata sufficientemente robusta, come si è verificato nel capitolo 11 dedicato alle analisi di sensibilità.

¹2 milioni di celle permettono ancora di gestire la creazione della mesh e la fase di post-processing su un normale PC portatile

²che in realtà è sempre un'inspirazione di transitorio che arriva a regime

12.2 Obiettivi di breve termine

Senza bisogno di altre modifiche, quanto presentato in questo lavoro di tesi è sufficiente per portare a termine con successo altri obiettivi importanti, tra cui ad esempio la simulazione di un'espirazione o di un ciclo respiratorio completo.

Per quanto riguarda l'espirazione non servono altre informazioni, se non la modifica del dizionario delle condizioni iniziali e al contorno della pressione, riportato alla fine del capitolo 9. Anche per il ciclo respiratorio completo è già tutto pronto, basta imporre una legge temporale nel dizionario delle condizioni al contorno. Come primo tentativo si può anche solo imporre una pressione in gola con andamento sinusoidale, compreso tra 130 e -130 m^2/s^2 e per un periodo di 2 secondi per ogni ciclo.

Per essere più precisi riportiamo di seguito qualche dettaglio delle modifiche. Il dizionario della pressione (analogo a quello già descritto) diventa:

```

dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    leftWall
    {
        type      fixedValue;
        value     uniform 0;
    }
    ...
    topWall
    {
        type      fixedValue;
        value     uniform 0;
    }
    vcg_patch0
    {
        type      zeroGradient;
    }
    throat
    {
        type timeVaryingUniformFixedValue;
        fileName "0/inlet.dat";
        outOfBounds clamp;
    }
}

```

```
    }  
}
```

in cui il *type timeVaryingUniformFixedValue* legge i dati del file `inlet.dat` e li usa come condizioni al contorno. Questo file è organizzato, ad esempio, nel seguente modo:

```
(  
( 0      0 )  
( 0.1  30 )  
( 0.2  60 )  
( 0.3  90 )  
( 0.4  60 )  
( 0.5  30 )  
)
```

ovvero è composto da due colonne, la prima con gli istanti temporali e la seconda con i valori corrispondenti (in questo caso pressioni). Negli istanti intermedi il valore viene interpolato.

Restano quindi da presentare solo gli obiettivi di lungo termine (studio completo di un caso clinico) e i molteplici sviluppi futuri, dato che questo ambito di studi è vastissimo e relativamente recente. La prossima sezione contiene un elenco parziale di questi argomenti ancora da trattare.

12.3 Obiettivi di medio e lungo termine

L'obiettivo finale di tutto questo lavoro di simulazione del flusso nella cavità nasale è quello di mettere a punto una procedura sicura e affidabile per poter studiare realisticamente gli interventi chirurgici di pazienti affetti da problemi respiratori. La simulazione della situazione pre- e post-operatoria è esattamente quella descritta nel presente lavoro di tesi, mentre resta ancora da decidere la strategia e la procedura per simulare numericamente diversi interventi, da cui sia possibile trarre sufficienti informazioni per valutarne l'efficacia.

A livello di software si potrebbe utilizzare ancora MeshLab, che nasce proprio con l'obiettivo di modificare mesh superficiali anche di grosse dimensioni. La simulazione di un intervento chirurgico si articola quindi nel modo seguente:

1. si parte da una TC pre-operatoria di un paziente e con Slicer si ricostruisce la superficie;

2. in MeshLab si modifica manualmente la mesh nelle zone desiderate, simulando un vero intervento chirurgico, avendo cura che la superficie resti semplicemente connessa;
3. si procede in OpenFOAM, come già visto.

Tra gli obiettivi che sono stati raggiunti solo in parte, il più importante è probabilmente quello della validazione. Nel capitolo dedicato alle analisi di sensibilità della soluzione ai vari parametri si è mostrato che i risultati ottenuti sono ragionevoli e affidabili, ma ancora non si può affermare che tutta la procedura sia validata. Le possibilità sono molteplici, tra cui quella più seguita in letteratura è senza dubbio cercare di riprodurre sperimentalmente (*in-vivo* o *in-vitro*) i risultati numerici. Ad esempio sarebbe molto interessante avere, oltre alla scansione TC, i dati rinomanometrici relativi ad un paziente, ovvero la pressione e la portata, misurate sperimentalmente, di un ciclo respiratorio completo. A questo punto si potrebbe simulare numericamente il flusso imponendo la variazione di pressione misurata e verificare che una quantità globale come la portata venga riprodotta con precisione.

Se si volesse una validazione anche a livello locale si potrebbero condurre esperimenti con visualizzazioni di modelli *in-vitro*, oppure si possono utilizzare altri software fluidodinamici, magari già utilizzati da altri autori in questo ambito. Come ultima alternativa si potrebbero riprodurre esattamente delle simulazioni già presenti in letteratura, ma ovviamente servono le stesse geometrie e i risultati, con numerosi problemi pratici tra cui la compatibilità tra i vari software.

Concludiamo questa sezione ricordando che risolvere il problema della validazione potrebbe anche aiutare a scegliere la soglia “corretta” in Slicer, o per lo meno quella che fornisce i risultati più vicini all’analisi sperimentale.

12.4 Ulteriori sviluppi e studi futuri

Oltre agli obiettivi finali, ci sono molti possibili sviluppi, più o meno importanti, di cui si può tenere conto nelle prossime fasi di questa collaborazione. Senza un particolare ordine, i punti da sviluppare sono principalmente i seguenti:

- confronto tra scansioni TC e dati MR;
- simulazione di più cicli respiratori completi;

- sensibilità degli altri parametri di blockMesh e snappyHexMesh;
- studio dei parametri di OpenFOAM, ad esempio rivedere gli schemi numerici;
- altri modelli di turbolenza, RANS e LES;
- studio dei coefficienti dei modelli RANS;
- sensibilità ai dati iniziali e al contorno (soprattutto nel caso turbolento);
- utilizzo di MeshLab, per modifiche, correzioni, *smoothing*;
- ottenimento automatico dei risultati significativi, come le portate o le medie delle grandezze in una certa sezione;
- presentazione compatta dei risultati, come nell'articolo [16].

Concludiamo questo capitolo e il presente lavoro di tesi accennando a possibili studi futuri, più impegnativi rispetto agli sviluppi appena elencati. Uno studio su cui si trovano già alcuni articoli in letteratura riguarda la simulazione di inspirazione di particelle. C'è molto interesse in queste simulazioni, sia dal punto di vista farmacologico (per scoprire in che zone si depositano i farmaci inalati) che per questioni legate ad esempio all'inquinamento atmosferico.

Un altro possibile studio (che non dovrebbe richiedere eccessivo lavoro) potrebbe essere una simulazione che tenga conto della temperatura (del flusso e delle pareti della cavità nasale), oppure dell'umidità all'interno del naso. In questi ultimi casi si tratta di cambiare solutore in OpenFOAM, scegliendo quello più opportuno. Naturalmente ogni solutore ha le sue particolarità, ma la maggior parte di quello che serve è già implementato in OpenFOAM.

Decisamente più complicata potrebbe invece essere una simulazione con uno studio più approfondito della mucosa nasale, che nel presente lavoro è stata pressochè ignorata, e questo discorso può essere collegato all'inspirazione di particelle. Per un primo approfondimento ci si può riferire all'articolo [13].

Ringraziamenti

A conclusione di questi anni di università sono molte le persone che vorrei ringraziare, spero di ricordarle tutte.

Innanzitutto devo ringraziare il Professor Quadrio, per questo lavoro di Tesi e per quello che mi ha insegnato, non solo dal punto di vista accademico, ma anche dal punto di vista personale.

Ringrazio il Professor Felisati e i suoi assistenti, che sono i destinatari di questo lavoro di simulazione del flusso nella cavità nasale. É grazie al loro desiderio di innovazione e ricerca se in un futuro non troppo lontano i pazienti avranno dei benefici anche da studi come questo.

Un caloroso saluto va al mio tutor Samuele, che mi ha seguito con competenza e amicizia in questo studio.

Ringrazio i miei familiari, i miei genitori e mia sorella, che mi hanno sempre sostenuto e incoraggiato nei momenti più difficili: mi hanno sempre assecondato nelle mie ambizioni e nei miei sogni, aiutandomi a raggiungerli.

Da ultimo un saluto speciale a tutti gli amici dell'università e in particolare (con i soprannomi!): Brambi e Cappelletto (i colleghi aerodinamici), Betti, Berna, Mike, Bari, Sergio, Tia, Marco, Teo, Ale, Carletto. Questi anni di studio sono passati più felicemente con le risate che ci siamo fatti.

Ringrazio infine tutti gli amici e tutti quelli che sicuramente avrò dimenticato.

Bibliografia

- [1] A. Baron. *Fluidodinamica*. 2005.
- [2] X.B. Chen, H.P. Lee, V.F.H. Chong, and D.Y. Wang. Assessment of septal deviation effects on nasal air flow: A computational fluid dynamics model. *American Laryngological Rhinological and Otological Society*, 119:1730–1736, 2009.
- [3] X.B. Chen, H.P. Lee, V.F.H. Chong, and D.Y. Wang. Impact of inferior turbinate hypertrophy on the aerodynamic pattern and physiological functions of the turbulent airflow. *J. of Rhinology*, 48:163–168, 2010.
- [4] C. Croce, R. Fodil, M. Durand, and G. Sbrilea-Apiou. In vitro experiments and numerical simulations of airflow in realistic nasal airway geometry. *Annals of Biomedical Engineering*, 34(6):997–1007, 2006.
- [5] D. Doorly, D.J. Taylor, P. Franke, and R.C. Schroter. Experimental investigation of nasal airflow. *J. Engineering in Medicine*, 222:439–453, 2007.
- [6] D. Doorly, D.J. Taylor, A.M. Gambaruto, R.C. Schroter, and N. Tolley. Nasal architecture: form and flow. *Philosophical Transaction of the Royal Society*, 366:3225–3246, 2008.
- [7] D. Elad, S. Naftali, M. Rosenfeld, and M. Wolf. Physical stresses at the air-wall interface of the human nasal cavity during breathing. *J. of Applied Physiology*, 100:1003–1010, 2006.
- [8] S. Ishikawa, T. Nakayama, M. Watanabe, and T. Matsuzawa. Visualization of flow resistance in physiological nasal respiration. *Arch Otolaryngol Head Neck Surg.*, 132:1203–1209, 2006.
- [9] H.H. Jin, J.R. Fan, M.J. Zeng, and K.F. Cen. Large eddy simulation of inhaled particle deposition within the human upper respiratory tract. *J. of Aerosol Science*, 38:257–268, 2006.

- [10] C. Kleinstreuer and Z. Zhang. Airflow and particle transport in the human respiratory system. *Annual Review of Fluid Mechanics*, 42:301–334, 2009.
- [11] P. Luchini and M. Quadrio. Aerodinamica. 2005. <http://pc-quadrio.aero.polimi.it>.
- [12] S. Naftali, M. Rosenfeld, M. Wolf, and D. Elad. The air-conditioning capacity of the human nose. *Annals of Biomedical Engineering*, 33(4):545–553, 2005.
- [13] H. Shi, C. Kleinstreuer, and Z. Zhang. Modeling of inertial particle transport and deposition in human nasal cavities with wall roughness. *J. of Aerosol Science*, 38:398–419, 2007.
- [14] D. Wexler, R. Segal, and J. Kimbell. Aerodynamic effects of inferior turbinate reduction. *Arch Otolaryngol Head Neck Surg.*, 131:1102–1107, 2005.
- [15] G. Xiong, J. Zhan, K. Zuo, J. Li, L. Rong, and G. Xu. Numerical flow simulation in the post-endoscopic sinus surgery nasal cavity. *Medical and Biological Engineering*, 46:1161–1167, 2008.
- [16] S. Zachov, P. Muigg, T. Hildebrandt, H. Doleisch, and H.C. Hege. Visual exploration of nasal airflow. *IEEE Transaction on Visualization and Computer Graphics*, 15(6):1407–1414, 2009.
- [17] K. Zhao, P. Dalton, G.C. Yang, and P.W. Scherer. Numerical modeling of turbulent and laminar airflow and odorant transport during sniffing in the human and rat nose. *Chem. Senses*, 31:107–118, 2006.
- [18] K. Zhao, P.W. Scherer, S.A. Hajiloo, and P. Dalton. Effect of anatomy on human nasal air flow and odorant transport patterns: Implications for olfaction. *Chem. Senses*, 29:365–379, 2004.