

UNIVERSITA' DEGLI STUDI DI GENOVA  
Corso di laurea in Ingegneria Meccanica  
**ANALISI DELLE PERFORMANCE  
DEL GENERATORE DI MESH  
NATIVO DI OpenFOAM®:  
snappyHexMesh**



Relatore: Chiar.mo Prof. Alessandro Bottaro  
Co-Relatore: Dr.Ing. Matteo Bargiacchi  
Candidato: Andrea Concas

**DICEMBRE 2014**

---

# Abstract

The aim of this thesis is to evaluate the influence of snappyHexMesh tool's parameters in the OpenFOAM<sup>®</sup> suite.

After a first introductory chapter regarding OpenFOAM<sup>®</sup>'s potentiality and functionalities, we present the different steps involving the creation of the mesh, with particular regard to the intrinsic quality of the latter, especially in terms of maximum skewness and maximum non-orthogonality.

Furthermore, we present a brief overview of a post-processing tool, ParaView, which has been used to visually analyze data.

The final goal is to determine which parameters provide the highest performances concerning the pre-mentioned intrinsic quality and, in particular, the computing time.

---

# Prefazione

La fase di generazione della mesh (ovvero discretizzazione del dominio di calcolo continuo) è argomento centrale nell'ambito delle simulazioni CFD.

La qualità della mesh è di gran lunga l'aspetto più importante e critico che può influenzare l'accuratezza, la stabilità e l'efficienza di una simulazione numerica.

Un classico refrain afferma " *garbage in - garbage out*", ovvero: dati di input scadenti non possono far altro che generare risultati in output da cestinare senza possibilità di appello. E' quindi chiaro come una discretizzazione attenta e ragionata del dominio di calcolo sia l'unica via che possa consentire il setup di una simulazione affidabile.

Dal momento che, chi possiede e comprende a fondo gli aspetti legati alla generazione della mesh, ha grande controllo sui risultati, è di fondamentale importanza ottimizzarne il processo di creazione. Ottenere una mesh di qualità significa ottenere una discretizzazione del dominio di calcolo che influenzi il meno possibile il risultato finale e che minimizzi l'errore dovuto alla discretizzazione stessa. Quindi la mesh dovrà essere:

1. fine abbastanza da poter catturare (o modellare) tutte le strutture del moto;
2. con caratteristiche intrinseche tali da minimizzare gli errori commessi dalle approssimazioni degli schemi numerici.

Come ogni problema ingegneristico che si rispetti esiste un trade-off: ovvero il numero di celle ed il tempo di creazione. Un buon meshatore è colui che, con il minor numero di celle e nel minor tempo possibile, riesce ad ottenere una mesh abbastanza buona da garantire un'indipendenza dei risultati dalla mesh stessa (*grid dependency*).

A fronte di tutto ciò, l'obiettivo della tesi è stato quello di indagare tutti i parametri rilevanti per la creazione della mesh con uno strumento open-source: snappyHexMesh.

Si è poi corredato il tutto con la visualizzazione grafica, attraverso la fase di post-processing adempiuta dall'utility Paraview.

Il file geometrico .stl importato è stato quello di un'automobile, l'asmoCar, modellizzabile come commistione di quattro regioni principali: il *body*, il *back* e le due ruote, *frontwheel* e *backwheel*. Per la simmetria del problema è infatti stata indagata solo metà della automobile.

Si è ricercato quindi quali fossero i valori di maggiore influenza sulla qualità della mesh con l'obiettivo di individuarne i più performanti, sempre in termini

---

di qualità: in merito a quest'ultima, nella fattispecie, si è trattato della massima skewness e della nonortogonalità massima, le cui rispettive definizioni formali verranno date nei capitoli dedicati.

In particolare si è dato rilievo al tempo di creazione della mesh stessa, ed al numero di celle generate.

La generazione della mesh è stata svolta utilizzando sempre un singolo processore ("Intel Core i5 430M").

# Indice

<b>1</b>	<b>Introduzione ad OpenFOAM®</b>	<b>1</b>
1.1	Il post-processing: Paraview . . . . .	2
1.2	OpenFOAM® e C++ . . . . .	3
1.3	Il workflow di una simulazione CFD . . . . .	4
1.3.1	0 . . . . .	4
1.3.2	Constant . . . . .	4
1.3.3	System . . . . .	5
1.3.4	Castellated Mesh, Snapping e Layer addition . . . . .	6
1.4	Qualità intrinseca della mesh . . . . .	7
1.4.1	Skewness Massima . . . . .	7
1.4.2	NonOrtogonalità Massima . . . . .	9
<b>2</b>	<b>L'utility snappyHexMesh</b>	<b>11</b>
2.1	BlockMesh . . . . .	11
2.2	Castellated Mesh . . . . .	15
2.3	Snapping . . . . .	19
2.4	Aggiunta dei layers . . . . .	20
<b>3</b>	<b>Analisi parametrica</b>	<b>25</b>
3.1	Refinement Parameters . . . . .	26
3.1.1	MaxLocalCells . . . . .	26
3.1.2	MaxGlobalCells . . . . .	26
3.1.3	MinRefinementCells . . . . .	27
3.1.4	nCellsBetweenLevels . . . . .	28
3.1.5	MaxLoadUnbalance . . . . .	30
3.2	Mesh selection . . . . .	31
3.2.1	LocationInMesh . . . . .	31
3.2.2	AllowFreeStandingZoneFaces . . . . .	31
3.3	SnapControls . . . . .	32
3.3.1	ResolveFeatureAngle . . . . .	32
3.3.2	NSmoothPatch . . . . .	35
3.3.3	PlanarAngle . . . . .	36
3.3.4	NSolveIter . . . . .	36
3.3.5	NRelaxIter . . . . .	37
3.3.6	Tolerance . . . . .	38
3.4	Feature snapping . . . . .	39

---

3.4.1	NFeatureSnapIter . . . . .	39
3.4.2	ImplicitFeatureSnap . . . . .	40
3.4.3	ExplicitFeatureSnap . . . . .	40
3.5	AddLayersControls . . . . .	41
3.5.1	RelativeSizes . . . . .	41
3.5.2	ExpansionRatio . . . . .	41
3.5.3	FinalLayerThickness . . . . .	41
3.5.4	MinThickness . . . . .	41
3.5.5	Ngrow . . . . .	42
3.6	Static analysis of starting mesh . . . . .	43
3.6.1	FeatureAngle . . . . .	43
3.6.2	MaxFaceThicknessRatio . . . . .	45
3.7	Patch displacement . . . . .	47
3.7.1	NSmoothSurfaceNormals . . . . .	47
3.7.2	NSmoothThickness . . . . .	49
3.8	Medial axis analysis . . . . .	52
3.8.1	MinMedianAxisAngle . . . . .	52
3.8.2	MaxThicknessToMedialRatio . . . . .	55
3.8.3	NSmoothNormals . . . . .	57
3.8.4	NMedialAxisIter . . . . .	59
3.8.5	NSmoothDisplacement . . . . .	61
3.9	Mesh shrinking . . . . .	62
3.9.1	SlipFeatureAngle . . . . .	62
3.9.2	NRelaxIter . . . . .	64
3.9.3	NBufferCellsNoExtrude . . . . .	65
3.9.4	NLayerIter . . . . .	67
3.9.5	NRelaxedIter . . . . .	68
3.10	Mesh Quality Controls . . . . .	69
3.10.1	MaxnonOrthogonality . . . . .	69
3.10.2	MaxInternalSkewness . . . . .	69
3.10.3	MaxBoundarySkewness . . . . .	69
3.10.4	MinDeterminant . . . . .	70
3.10.5	MinTriangleTwist . . . . .	70
<b>4</b>	<b>Conclusioni</b>	<b>73</b>
4.1	Sviluppi futuri . . . . .	76



# Elenco delle figure

1.1	Struttura OpenFOAM® . . . . .	2
1.2	Interfaccia dell'ambiente paraview . . . . .	2
1.3	Caso di OpenFoam . . . . .	4
1.4	Constant . . . . .	5
1.5	System . . . . .	6
1.6	PolyMesh . . . . .	6
1.7	Grafico Skewness . . . . .	8
1.8	Grafico Max NonOrthogonality . . . . .	9
2.1	Grado di refinement . . . . .	17
2.2	CastellatedMesh: Vista dell'intero body e delle ruote . . . . .	18
2.3	CastellatedMesh: Vista lungo il piano sagittale mediano . . . . .	18
2.4	Post-snapping Mesh: Vista dell'intero body e delle ruote . . . . .	19
2.5	Post-snapping Mesh: Vista lungo il piano sagittale mediano . . . . .	20
2.6	Mesh dopo l'aggiunta dei layers: vista lungo il piano sagittale mediano . . . . .	23
2.7	Mesh dopo l'aggiunta dei layers: zoom in corrispondenza del bordo dell'asmoCar . . . . .	23
3.1	Skewness . . . . .	27
3.2	MaxnonOrthogonality . . . . .	27
3.3	Numero di celle . . . . .	27
3.4	Tempo di creazione mesh normalizzato . . . . .	27
3.5	Skewness . . . . .	28
3.6	MaxnonOrthogonality . . . . .	28
3.7	Numero di celle . . . . .	28
3.8	Tempo di creazione mesh normalizzato . . . . .	28
3.9	Tre screen con paraview . . . . .	29
3.10	Skewness . . . . .	30
3.11	MaxnonOrthogonality . . . . .	30
3.12	Numero di celle . . . . .	30
3.13	Tempo di creazione mesh normalizzato . . . . .	30
3.14	AllowFreeStandingZoneFaces [5] . . . . .	32
3.15	Skewness . . . . .	33
3.16	MaxnonOrthogonality . . . . .	33
3.17	Numero di celle . . . . .	33
3.18	Tempo di creazione mesh normalizzato . . . . .	33

---

3.19	Tre screenshots con paraview . . . . .	34
3.20	Skewness . . . . .	35
3.21	MaxnonOrthogonality . . . . .	35
3.22	Numero di celle . . . . .	35
3.23	Tempo di creazione mesh normalizzato . . . . .	35
3.24	Skewness . . . . .	36
3.25	MaxnonOrthogonality . . . . .	36
3.26	Numero di celle . . . . .	36
3.27	Tempo di creazione della Mesh . . . . .	36
3.28	Skewness . . . . .	37
3.29	MaxnonOrthogonality . . . . .	37
3.30	Numero di Celle . . . . .	37
3.31	Tempo di Creazione Mesh normalizzato . . . . .	37
3.32	Skewness . . . . .	39
3.33	MaxnonOrthogonality . . . . .	39
3.34	Numero di Celle . . . . .	39
3.35	Tempo di creazione mesh normalizzato . . . . .	39
3.36	Screen con paraview dopo una slice . . . . .	42
3.37	Skewness . . . . .	43
3.38	MaxnonOrthogonality . . . . .	43
3.39	Numero di Celle . . . . .	43
3.40	Tempo di creazione mesh normalizzato . . . . .	43
3.41	Screen con paraview dopo una slice . . . . .	45
3.42	Skewness . . . . .	46
3.43	MaxnonOrthogonality . . . . .	46
3.44	Numero di Celle . . . . .	46
3.45	Tempo di creazione mesh normalizzato . . . . .	46
3.46	Skewness . . . . .	47
3.47	MaxnonOrthogonality . . . . .	47
3.48	Numero di Celle . . . . .	47
3.49	Tempo di creazione Mesh normalizzato . . . . .	47
3.50	Screen con paraview nel retro dell'automobile . . . . .	48
3.51	Skewness . . . . .	49
3.52	MaxnonOrthogonality . . . . .	49
3.53	Numero di Celle . . . . .	49
3.54	Tempo di Creazione Mesh normalizzato . . . . .	49
3.55	Screen con paraview . . . . .	50
3.56	Screen con paraview . . . . .	51
3.57	Skewness . . . . .	52
3.58	MaxnonOrthogonality . . . . .	52
3.59	Numero di Celle . . . . .	52
3.60	Tempo di creazione Mesh normalizzato . . . . .	52
3.61	MinMedianAxisAngle [5] . . . . .	53
3.62	Screen con paraview . . . . .	54
3.63	Skewness . . . . .	55
3.64	MaxnonOrthogonality . . . . .	55

---

3.65	Numero di Celle . . . . .	55
3.66	Tempo di creazione Mesh normalizzato . . . . .	55
3.67	Screen con paraview . . . . .	56
3.68	Skewness . . . . .	57
3.69	MaxnonOrthogonality . . . . .	57
3.70	NumberofCells . . . . .	57
3.71	Tempo di creazione Mesh normalizzato . . . . .	57
3.72	Screen con paraview . . . . .	58
3.73	Skewness . . . . .	59
3.74	MaxnonOrthogonality . . . . .	59
3.75	Numero di Celle . . . . .	59
3.76	Tempo di Creazione Mesh normalizzato . . . . .	59
3.77	Screen con paraview . . . . .	60
3.78	Skewness . . . . .	61
3.79	MaxnonOrthogonality . . . . .	61
3.80	Numero di Celle . . . . .	61
3.81	Tempo di creazione Mesh normalizzato . . . . .	61
3.82	Skewness . . . . .	63
3.83	MaxnonOrthogonality . . . . .	63
3.84	Numero di Celle . . . . .	63
3.85	Tempo di Creazione Mesh normalizzato . . . . .	63
3.86	Skewness . . . . .	64
3.87	MaxnonOrthogonality . . . . .	64
3.88	Numero di Celle . . . . .	64
3.89	Tempo di Creazione Mesh normalizzato . . . . .	64
3.90	Skewness . . . . .	65
3.91	MaxnonOrthogonality . . . . .	65
3.92	Numero di Celle . . . . .	65
3.93	Tempo di Creazione Mesh normalizzato . . . . .	65
3.94	Post processing sul bordo . . . . .	66
3.95	Skewness . . . . .	67
3.96	MaxnonOrthogonality . . . . .	67
3.97	Numero di Celle . . . . .	67
3.98	Tempo di Creazione Mesh normalizzato . . . . .	67
3.99	Skewness . . . . .	68
3.100	MaxnonOrthogonality . . . . .	68
3.101	Numero di Celle . . . . .	68
3.102	Tempo di Creazione Mesh normalizzato . . . . .	68
3.103	MinTriangleTwist [5] . . . . .	71



# Elenco delle tabelle

3.1	Minrefinementcells	27
3.2	Ncellsbetweenlevels	30
3.3	MaxLoadUnbalance	31
3.4	AllowFreeStandingZoneFaces	31
3.5	ResolveFeatureAngle	32
3.6	NSmoothPatch	35
3.7	NSolveIter	37
3.8	NRelaxIter	38
3.9	NFeatureSnapIter	40
3.10	ImplicitFeatureSnap	40
3.11	ExplicitFeatureSnap	40
3.12	NGrow	42
3.13	FeatureAngle	44
3.14	MaxFaceThicknessRatio	45
3.15	NSmoothSurfaceNormals	47
3.16	NSmoothThickness	49
3.17	MinMedianAxisAngle	52
3.18	MaxThicknessToMedialRatio	55
3.19	NSmoothNormals	57
3.20	NMedialAxisIter	59
3.21	NSmoothDisplacement	61
3.22	SlipFeatureAngle	62
3.23	NRelaxIter	64
3.24	NBufferCellsNoExtrude	65
3.25	NLayerIter	67
3.26	NRelaxedIter	69
3.27	MaxNonOrthogonality	69
3.28	MaxInternalSkewness	69
3.29	MaxBoundarySkewness	70
3.30	MinDeterminant	70
3.31	MinTriangleTwist	71
4.1	1	73
4.2	2	74



# Capitolo 1

## Introduzione ad OpenFOAM<sup>®</sup>

OpenFOAM<sup>®</sup> (Field Operation and Manipulation) è un pacchetto software Open-Source, scritto in C++ rilasciato sotto licenza da ESI Group.

Il fatto di essere Open-Source, ovvero con codice sorgente disponibile, lo rende modificabile da parte di chiunque: ciò ne giustifica il largo impiego principalmente in ambito accademico.

Le applicazioni implementate nella versione 2.3.X comprendono, tra le altre, solutori per moti comprimibili e incomprimibili, trasferimento di calore, multifase, trasformazioni chimiche e combustione, elettromagnetismo, moti a potenziale, DNS (direct numerical simulation), stress-analysis, e modelli finanziari. Non è da sorprendersi che OpenFOAM<sup>®</sup> comprenda anche modelli non strettamente legati alla CFD tradizionale. Infatti OpenFOAM<sup>®</sup>, come dice il nome, altro non è che un set di librerie per la manipolazione di campi (scalari, vettoriali, tensoriali) ed è quindi particolarmente adatto alla risoluzione di equazioni differenziali alle derivate parziali su domini arbitrariamente complessi.

Tali applicazioni ricadono principalmente in due categorie:

- solvers, ovvero solutori che implementano loop numerici per la risoluzione di un determinato set di equazioni rappresentanti un modello fisico specifico;
- utilities, sviluppate per risolvere problemi che includono la manipolazione di dati, creazione mesh e alcuni strumenti di post-processing.

OpenFOAM<sup>®</sup> assieme ad altri strumenti di pre e post-processing genera una struttura complessiva che consente all'utente di affrontare il workflow di un'analisi CFD interamente con strumenti open-source (visibile in figura 1.1)

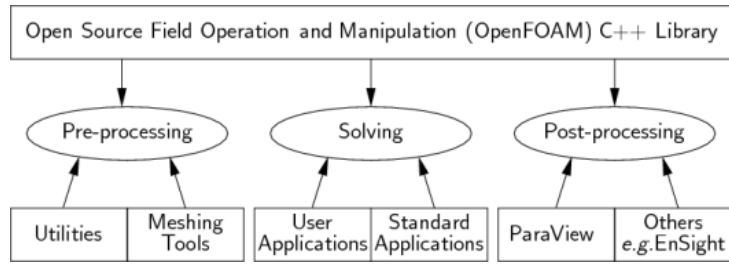


Figura 1.1: Struttura OpenFOAM<sup>®</sup>

## 1.1 Il post-processing: Paraview

Paraview è un'applicazione, anch'essa open-source, creata per la visualizzazione di un grande numero di dati e si presta ad essere eseguita sia su un singolo processore sia in parallelo. Una volta selezionato il caso oggetto di studio e lanciata l'applicazione da terminale, è necessario importare un file VTK (Visualisation Toolkit) contenente la geometria, mesh ed eventuali campi contenenti la soluzione: esso verrà quindi analizzato all'interno della View Area.

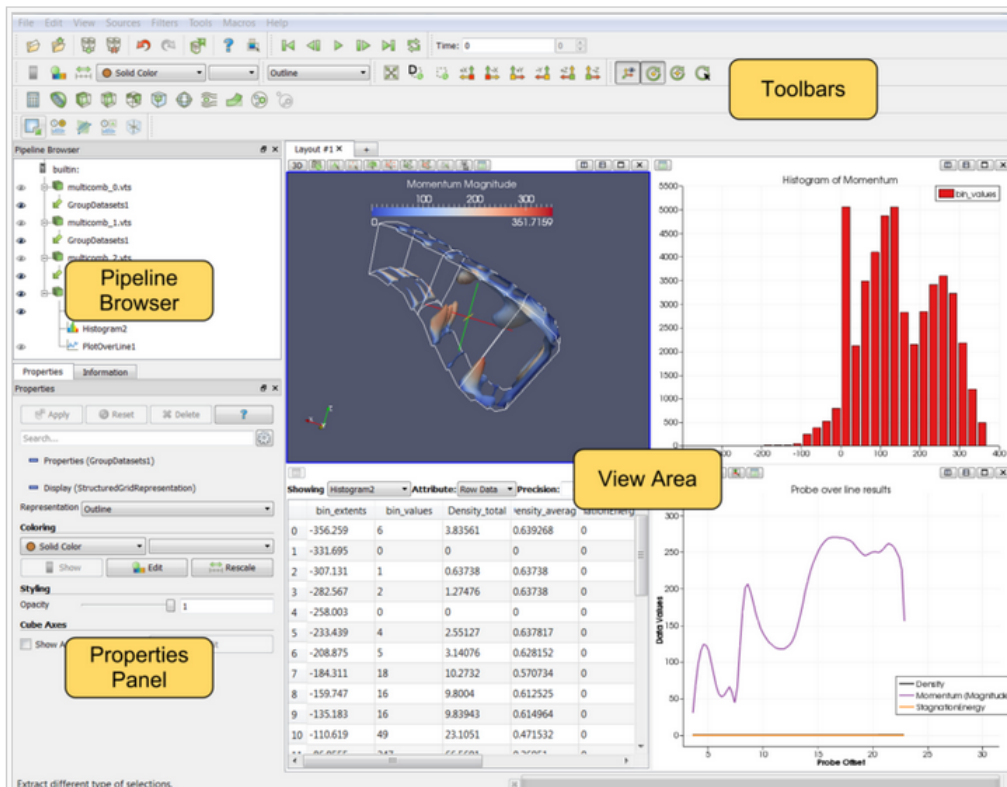


Figura 1.2: Interfaccia dell'ambiente paraview



---

## 1.2 OpenFOAM<sup>®</sup> e C++

OpenFOAM<sup>®</sup> è diviso in una serie di librerie, che sono dinamicamente collegate tra loro durante la compilazione dei solvers e delle utilities.

Trattasi di un linguaggio object-oriented: una programmazione di questo tipo permette di circoscrivere in una zona del codice sorgente, detta classe, la dichiarazione delle strutture dati delle procedure che operano su di esse.

Le classi sono fondamentali al fine di creare gli oggetti e pertanto esse sono l'unità di base per il meccanismo dell'incapsulamento.

Una classe definisce un nuovo tipo di dato che può essere utilizzato per la creazione di oggetti dando vita ad una struttura logica che definisce le relazioni tra i suoi membri; con la definizione di classe vengono definiti sia i dati sia il codice che opera sugli stessi. Tale concetto sarà di largo utilizzo nella definizione dei solvers all'interno di OpenFOAM<sup>®</sup>.

---

## 1.3 Il workflow di una simulazione CFD

Una simulazione in OpenFOAM<sup>®</sup>, come qualsiasi altra simulazione CFD, si articola in tre fasi distinte.

- La prima è detta di **pre-processing**: qui, tra le altre cose, avviene la creazione della mesh. A tal fine il programma mette a disposizione due tool: il primo è blockMesh, che consente di creare delle mesh strutturate, mentre il secondo è snappyHexMesh, con cui è possibile generare mesh non strutturate, semplicemente dando in ingresso un file in formato .stl contenente il corpo e una griglia di background;
- Una seconda fase di **risoluzione** in cui, adottando un particolare solutore, vengono risolte le equazioni ad esso associate, specificando le condizioni iniziali e al contorno, la mesh, gli schemi numerici per i vari termini ed alcuni parametri di controllo della simulazione;
- Una fase finale di **post-processing**, dove si vanno a visualizzare (per esempio mediante ParaView) i risultati ottenuti, analizzando i vari campi (vettoriali, scalari e tensoriali) all'interno del dominio.

La struttura di un caso è tipicamente questa:



Figura 1.3: Caso di OpenFoam

### 1.3.1 0

Nella cartella 0 vengono definite le condizioni iniziali e al contorno.

### 1.3.2 Constant

Nella cartella constant vengono definite le proprietà termofisiche del continuo in analisi e la discretizzazione del dominio di calcolo (mesh). Oltre a polyMesh (per la cui definizione si rimanda alla nota entro la cartella system) troviamo extendedFeatureEdgeMesh che contiene informazioni circa le caratteristiche dei bordi (feature edges), includendo la distinzione tra caratteristiche puramente geometriche e regioni di bordo (region edges) generati mediante l'utility surfaceFeatureExtraDict i cui parametri sono settati nel dizionario surfaceFeatureExtraDict (nella cartella system).

Inoltre troviamo la cartella triSurface, in cui è presente il file asmoCar.emesh che contiene la lista degli edges della geometria.

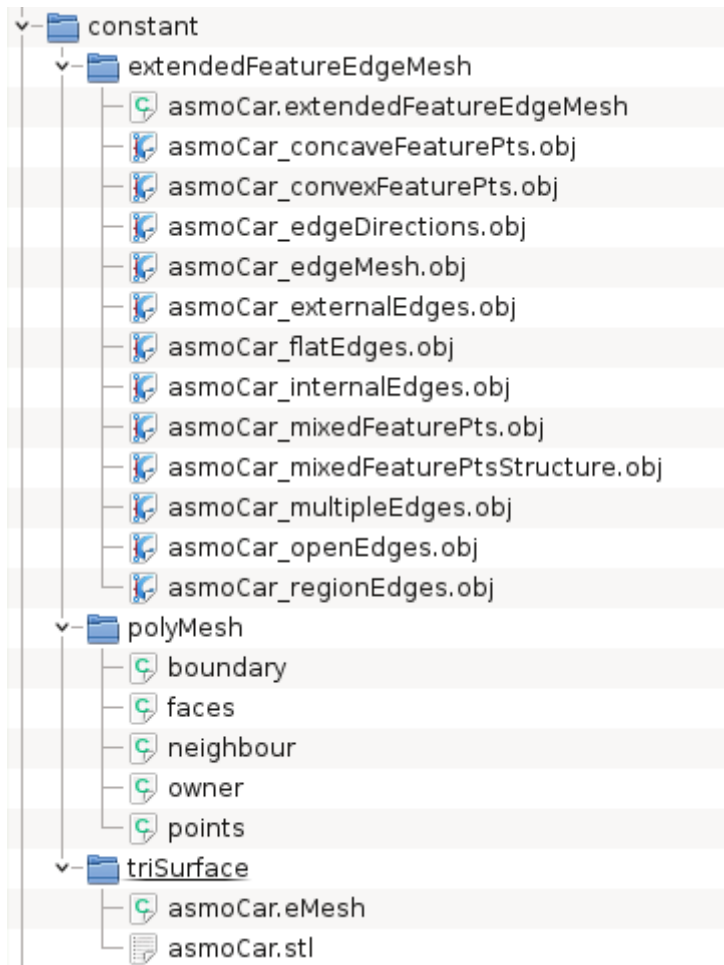


Figura 1.4: Constant

### 1.3.3 System

Nella cartella system troviamo il file controlDict in cui sono presenti tutti i parametri che governano la simulazione, quali ad esempio lo start time, l'end time, il delta t.

Il dizionario fvSchemes contiene indicazioni circa gli schemi numerici utilizzati per ogni singolo termine, mentre in fvsolutions vi sono indicazioni circa i margini di tolleranza.

DecomposeParDict specifica il metodo di decomposizione del dominio, che consiste nel generare un numero di subdomini pari al numero di processori su cui si intende eseguire la simulazione.

Per quanto riguarda blockMesh e snappyHexMesh la trattazione avverrà nei capitoli dedicati.

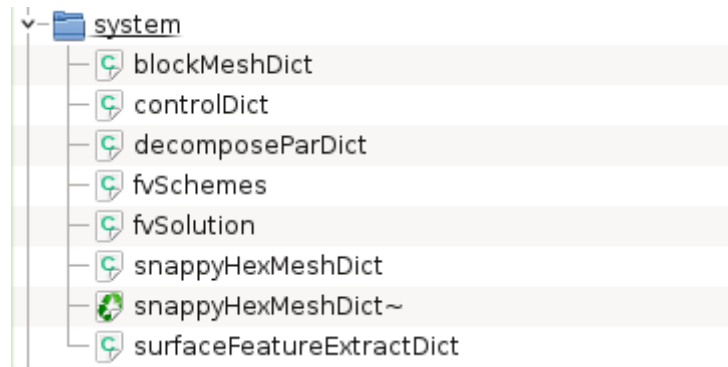


Figura 1.5: System

### 1.3.4 Castellated Mesh, Snapping e Layer addition

Nelle cartelle 1, 2 e 3 corrispondenti alle tre fasi di creazione della mesh di SnappyHexMesh (ovvero CastellatedMesh, Snapping e Layer Addition) troveremo il file polyMesh in cui è presente la descrizione delle mesh ottenute nelle fasi di pre-processing (si veda figura 1.6).

Saranno qui elencate le coordinate dei punti, le celle e le facce, con anche i relativi livelli, nonché i contorni del dominio di calcolo:

*boundary* contiene informazioni circa le celle al contorno;

*faces* contiene l'elenco delle facce;

*neighbour* contiene i confinamenti delle celle;

*owners* contiene l'elenco delle facce appartenenti ai centri cella;

*points* contiene l'elenco dei centri cella.

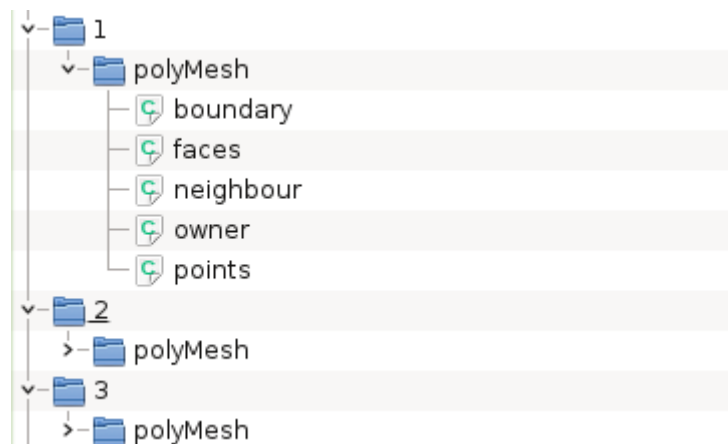


Figura 1.6: PolyMesh

---

## 1.4 Qualità intrinseca della mesh

Possono essere definiti diversi indici di qualità della mesh. Per semplicità, in questa trattazione, si è dato rilievo ai due più cruciali: la massima skewness e la non-ortogonalità massima. E' importante comprendere come, tali indici, sono solo rappresentativi della qualità intrinseca della discretizzazione effettuata in quanto la qualità globale della mesh è strettamente dipendente dalla fisica del problema in analisi. A titolo puramente esemplificativo è chiaro che la stessa mesh adatta a risolvere il moto di un aeromobile a velocità subsonica non è adatta a discretizzare la stessa geometria a velocità transoniche in presenza di onde d'urto nonostante la qualità intrinseca sia invariata.

### 1.4.1 Skewness Massima

Anzitutto si definisce il vettore  $\mathbf{a}$  congiungente i centroidi di due celle aventi una faccia in comune.

In secundis si ricerca il punto di intersezione  $M$  di tale vettore con la faccia  $\Omega$  della cella. La distanza da tale intersezione con il centro della faccia sia  $\mathbf{b}$ . A tal punto è possibile definire la skewness come il rapporto tra il modulo di  $\mathbf{b}$  e il modulo di  $\mathbf{a}$ . Si considera come valore critico, in base all'esperienza, il valore 4. Perché tale grandezza influenza la qualità della discretizzazione? Il motivo risiede nell'approssimazione dei flussi  $\phi$  di massa, quantità di moto, energia exc. attraverso la faccia  $\Omega$ . Tali flussi  $\phi$  vengono approssimati sulla base della distanza  $\mathbf{a}$  dei centri cella e dovrebbero riferirsi all'intera faccia  $\Omega$  in questione ed approssimarne il valore nel suo centro geometrico  $P$ . Un elevato valore di skewness implicherebbe l'approssimazione di  $\phi_p$  a  $\phi_m$  non accettabile in quanto troppo distante dalla fisica del problema.

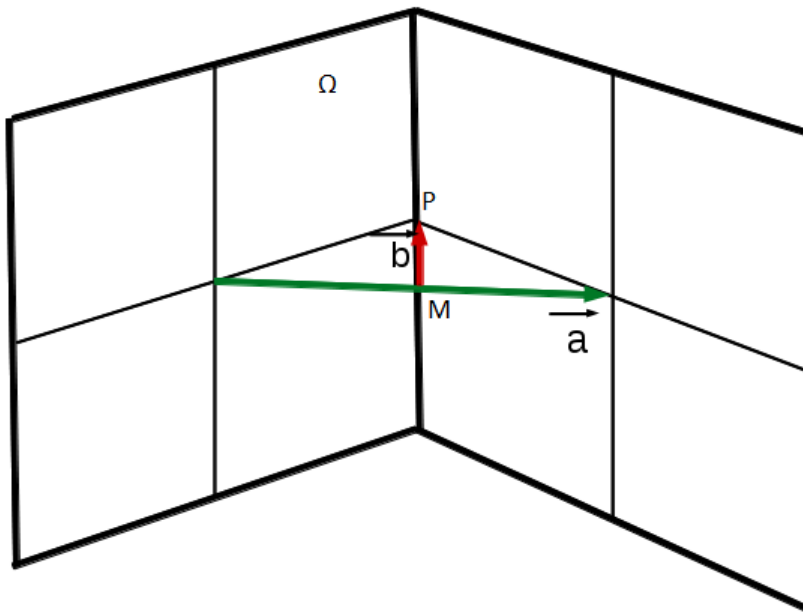


Figura 1.7: Grafico Skewness

---

### 1.4.2 NonOrtogonalità Massima

Tale parametro è rappresentato dall'angolo formato dalla congiungente i due centroidi con il vettore normale alla faccia tra due celle adiacenti. Una mesh è detta ortogonale se la normale alla faccia è parallela al vettore che congiunge i centri delle due celle. Tale angolo deve mantenersi minore di 90 gradi affinché la stabilità della simulazione non sia compromessa. In analogia con gli effetti della skewness, valori diversi da zero per la non-ortogonalità implicano l'approssimazione del flusso  $\phi$  attraverso la faccia  $\Omega$  non perpendicolare al vettore  $\mathbf{s}$  come se  $\Omega$  fosse perpendicolare. Nonostante correzioni sulla non-ortogonalità possono essere previste in fase di risoluzione, queste non sono in grado di eliminare completamente l'errore commesso in fase di discretizzazione e aumentano sensibilmente il tempo di calcolo.

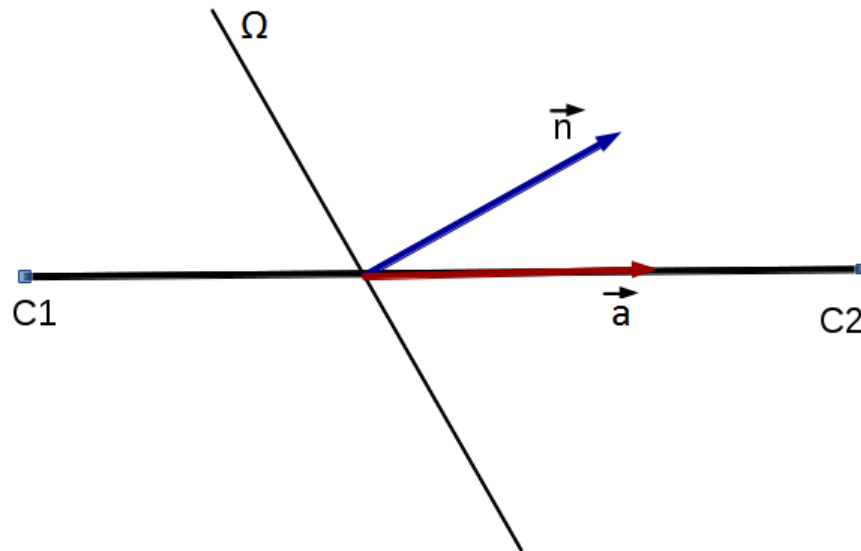


Figura 1.8: Grafico Max NonOrthogonality

---



# Capitolo 2

## L'utility snappyHexMesh

### 2.1 BlockMesh

Il primo passo nella creazione della mesh mediante il meshatore nativo di OpenFOAM<sup>®</sup> è quello di creare una griglia cartesiana che circonda l'oggetto geometrico importato in file .stl.

Si crea quindi una mesh di background attraverso l'utility blockMesh, in cui viene definito il dominio computazionale nonché il grado di refinement di base. Il principio di blockMesh è quello di decomporre il dominio geometrico in blocchi esaedrici: verrà generato un parallelepipedo che circonda l'asmoCar, all'interno del quale agirà successivamente snappyHexMesh.

Per la generazione della mesh attraverso blockMesh è innanzitutto necessario compilare il file blockMeshDict in diverse parti:

- Vertici, in cui vengono definite le coordinate degli stessi in ( x y z );
- Blocchi, in cui vengono specificati i vertici, il numero di celle e il grading;
- Patches, in cui vengono specificate le facce.

Il file blockMeshDict, all'interno della cartella System è il seguente:

```
FoamFile
{
    version 2.0;

    format ascii;

    class dictionary;
```

---

```

location system;

object blockMeshDict;

}

convertToMeters 1;

vertices
(
    ( -1.215 -0.792 -0.0335) // 0
    ( 4.05 -0.792 -0.0335) // 1
    ( 4.05 0.0 -0.0335) // 2
    ( -1.215 0.0 -0.0335) // 3
    ( -1.215 -0.792 1.62) // 4
    ( 4.05 -0.792 1.62) // 5
    ( 4.05 0.0 1.62) // 6
    ( -1.215 0.0 1.62) // 7
);

blocks
(
hex (0 1 2 3 4 5 6 7) (130 20 40 ) simpleGrading (1 1 1)
);

edges ( );

patches
(
wall inlet ( (0 4 7 3) )

```

---

```

wall outlet ( (1 2 6 5) )
wall side ( (0 1 5 4) )
wall symmetry ( (3 7 6 2) )
wall ground ( (0 3 2 1) )
wall top ( (4 5 6 7) )

);

mergePatchPairs ( );

```

Notiamo dapprima che nel file è presente la funzione ”*convert to meters*” che è un fattore di scala che permette di definire le coordinate dei vertici in metri.

Tali vertici vengono poi specificati: essi sono identificati dal rispettivo *label*, ed in accordo con l’enumerazione adottata nel linguaggio C, al primo sarà assegnato il valore 0, e poi a seguire fino ad arrivare a 7.

I blocchi vengono descritti, come già anticipato, in funzione dei vertici, del numero di celle e del fattore di espansione delle stesse. Troviamo dapprima indicati i vertici, che vengono enumerati come descritto poco sopra, antepoendo però la scritta *hex*, ad indicare il fatto che tutti i blocchi creati saranno esaedrici. Viene inoltre esplicitato il numero di celle in ogni direzione (xyz) seguito dal ”*simple grading*”: questo setting fornisce una misura di espansione delle celle nelle tre direzioni e ciò consente di refinarle in maniera precisa.

Troviamo poi la dicitura ”*edges*”: qui vengono specificati i tipi di linee che uniscono i vertici. BlockMesh offre varie possibilità a questo scopo: *arc* (arco circolare), *simpleSpline* (una spline curva) o ancora *polyLine* che fornisce un insieme di linee. In questo caso non è stato specificato nulla, ad indicare che verrà utilizzata il settaggio di default, ovvero *line* (linea dritta).

Successivamente si presentano le ”*patches*”, ovvero le regioni. Queste vengono denominate arbitrariamente dall’utente: abbiamo *wall inlet*, *wall outlet*, *wall side*, *wall symmetry*, *wall ground* e *wall top*. All’interno vengono dichiarate le facce con la lista dei label corrispondenti ai vertici. L’ordine di tali vertici, nella definizione della regione, avviene in senso orario.

Troviamo infine il *facemerging*, in cui viene descritta l’unione di due facce appartenenti a regioni distinte; ciò avviene attraverso la definizione di una *master patch*, ovvero la faccia che mantiene la conformazione e

---

la posizione originariamente definite, e di una *slave patch* che non vedrà invece necessariamente preservata la propria geometria. In questo caso tale setting non è stato applicato.

---

L'utility `snappyHexMesh` articola la generazione della griglia di calcolo in tre fasi successive:

1. *Castellated Mesh*;
2. *Snapping*;
3. *Aggiunta dei layers*.

Si analizzano ora le loro funzioni.

## 2.2 Castellated Mesh

Definita la regione di background in `blockMesh`, avrà inizio il cosiddetto “*cellsplitting*” caratteristico della prima fase di `snappyHexMesh`: la *Castellated Mesh*. Lo *splitting* avviene in accordo ai settaggi scelti dall'utente nel sotto-dizionario *castellatedMeshControls* all'interno del file *SnappyHexMeshDict*. Il processo incomincia in virtù delle specifiche in merito alle features all'interno del file “*edgeMesh*”; in questa fase viene creata una mesh grossolana che andrà poi rifinita e resa “body-fitted” con le procedure successive. Viene qui presentato il codice all'interno dei *castellatedMeshControls*:

```
castellatedMeshControls
{
    features
    (
        {
            file "asmoCar.eMesh";
            levels ( ( 0.0 0 ) );
        }
    );
    refinementSurfaces
    {
        asmoCar
```

---

```
    {
      regions
      {
        body
        {
          level ( 4 4 );
        }
        frontWheel
        {
          level ( 4 4 );
        }
        back
        {
          level ( 4 4 );
        }
        backWheel
        {
          level (4 4 );
        }
      }
      level ( 4 4 );
    }
  }
```

---

```

refinementRegions
{
    cylinder
    {
        mode inside;

        levels (( 1E5 2 ));
    }
}

```

Il file *asmoCar.eMesh* è il file contenente le *features*. La dicitura *levels (a b)* sotto refinement regions indica il livello minimo e massimo di refinement sulla superficie in questione.

Il livello di refinement è definito come:

$$\Delta X_n = \frac{\Delta X_0}{2^n} \quad (2.1)$$

Dove  $\Delta X_0$  è la dimensione della cella corrispondente al livello 0, mentre  $\Delta X_n$  è quella corrispondente al livello n-esimo.

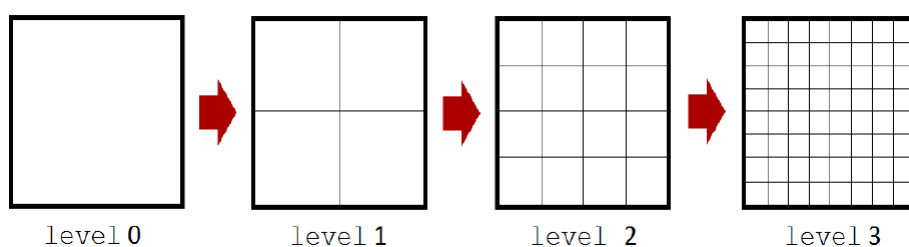


Figura 2.1: Grado di refinement

Il refinement può essere di tre tipi:

- **inside**: refines all'interno della regione volumetrica;
- **outside**: refines all'esterno della regione volumetrica;
- **distance**: refines in accordo con la distanza dalla superficie.

Il *refinement surface* è specificato dai livelli minimo e massimo di refinement, in corrispondenza della regione che verrà considerata. Il livello minimo è applicato in prossimità della superficie, il secondo sarà invece dipendente dal *resolvefeatureangle*. Quest'ultimo, e tutti gli altri parametri caratteristici della *castellatedMesh* e delle due successive fasi, verranno indagati nel prossimo capitolo. La voce *refinementRegions* contiene

---

le caratteristiche di refinement delle regioni volumetriche specificate nel sotto dizionario *geometry*. In questa circostanza il refinement è di tipo inside.

Una volta ultimato lo splitting avviene la fase del removing, ovvero di rimozione delle celle. Verranno qui rimosse tutte le celle che hanno almeno il 50 % del loro volume all'interno del dominio geometrico.

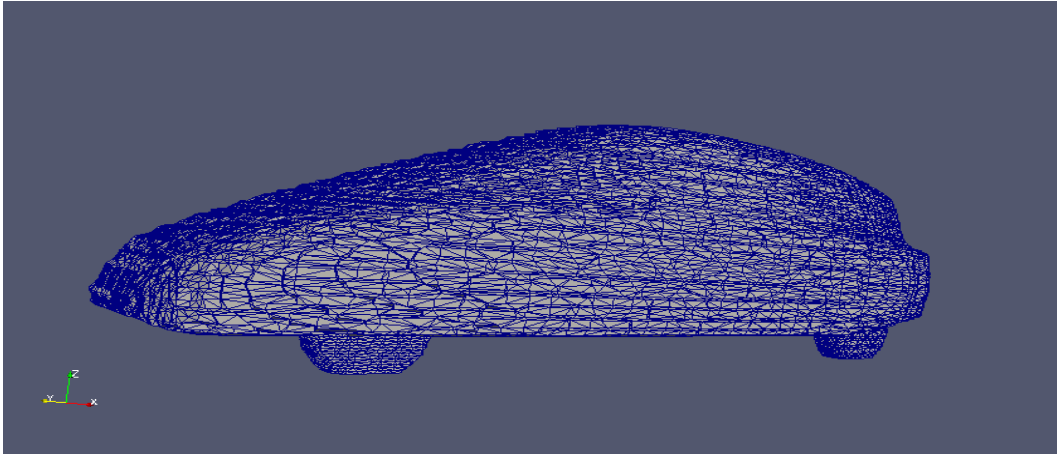


Figura 2.2: CastellatedMesh: Vista dell'intero body e delle ruote

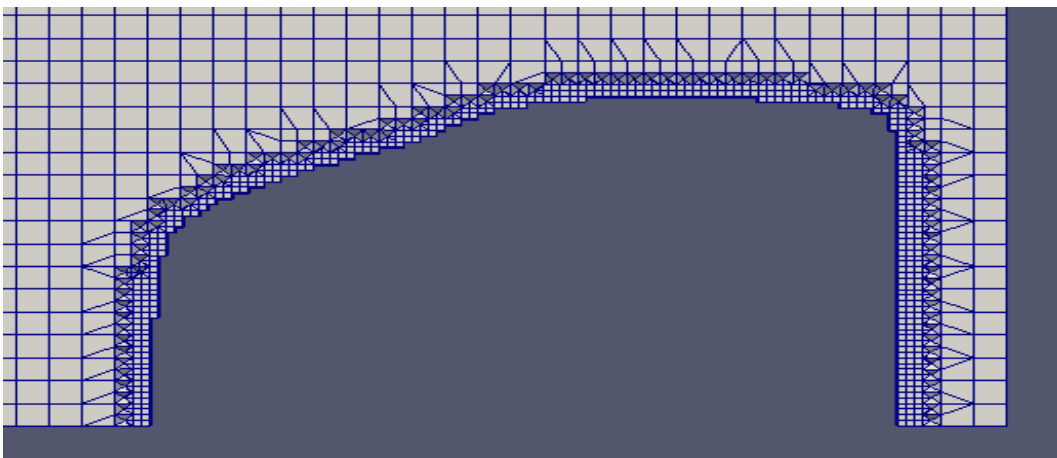


Figura 2.3: CastellatedMesh: Vista lungo il piano sagittale mediano

Nelle figure 2.2 e 2.3 sono riportati i risultati della mesh dopo la Castellated Mesh.



---

## 2.3 Snapping

Lo step successivo nel processo di creazione della mesh è quello dello *snapping*.

Si tratta di spostare i punti relativi ai vertici delle celle tagliate dalla superficie della geometria in maniera tale da rimuovere la superficie frastagliata ottenuta durante la *castellatedmesh* ed ottenere una griglia il più possibile simile al file di input (*body fitted mesh*).

In particolare, in questa fase verrà prestata attenzione al rispetto dei parametri di qualità della mesh, presenti tra i *meshQualityControls*: nel caso in cui tali requisiti non siano assolti si riproiettano i punti con uno spostamento inferiore, e si itera fino a che la mesh volumetrica non sia corretta.

I parametri caratteristici dello Snapping sono *nSmoothPatch*, *tolerance*, *nSolveIter* e *nRelaxIter* ed anche per questi verrà fatto un excursus nel capitolo successivo.

Nelle figure 2.4 e 2.5 sono riportati i risultati della mesh dopo lo snapping.

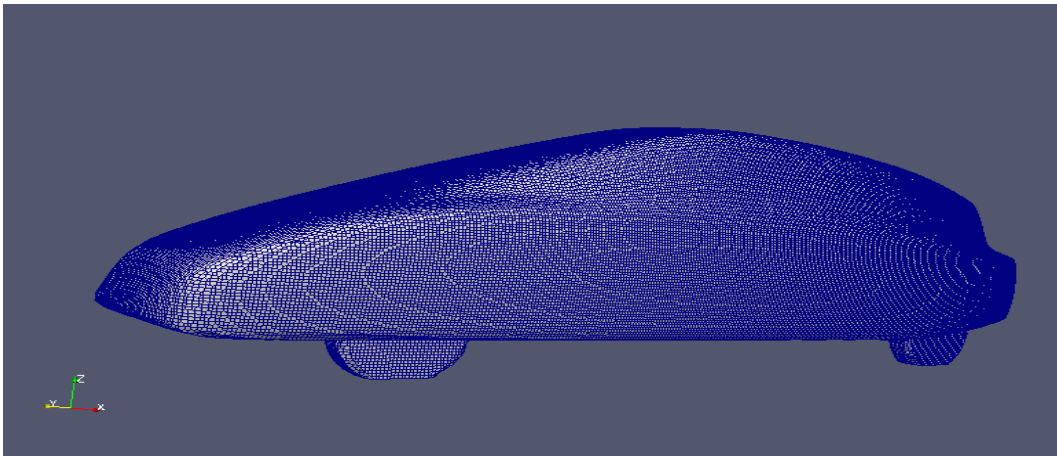


Figura 2.4: Post-snapping Mesh: Vista dell'intero body e delle ruote

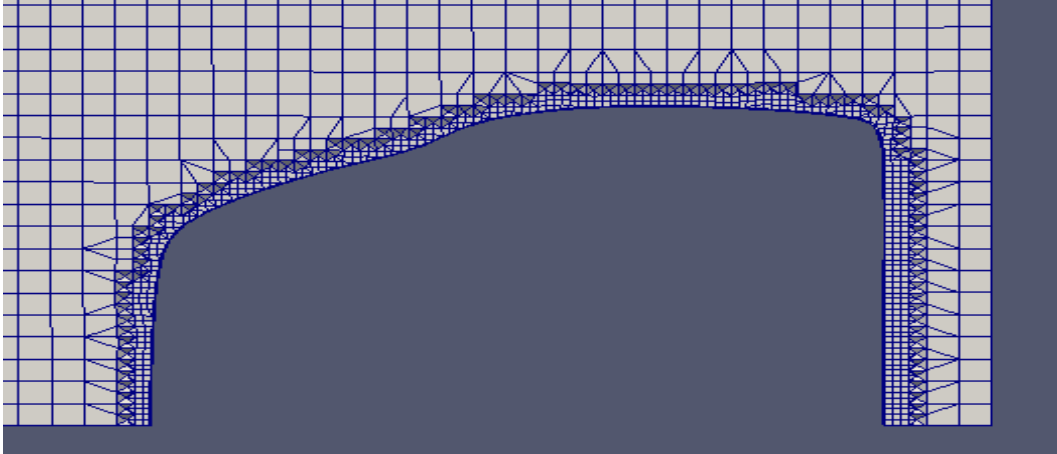


Figura 2.5: Post-snapping Mesh: Vista lungo il piano sagittale mediano

## 2.4 Aggiunta dei layers

L'ultimo passo nella creazione della mesh riguarda l'aggiunta dei layers: questo è un settaggio opzionale, ma l'esigenza di rimpiazzare le celle irregolari in corrispondenza dei confini del dominio geometrico, al fine di fornire una migliore risoluzione dello strato limite, lo rende necessario.

Il processo si articola in più fasi:

In primis la mesh è proiettata lontano dalla superficie di un certo spessore (*thickness*) nella direzione normale alla superficie stessa. In secundis avviene una fase iterativa, in cui si controlla il rispetto dei dettami di qualità: nel caso in cui questi non siano onorati, lo spessore di cui la mesh è allontanata viene ridotto. Nel caso limite in cui non si riesca ad accondiscendere ai criteri di validazione per qualsivoglia spessore dei layers, questi ultimi non verranno aggiunti. In caso positivo invece, ne avverrà l'introduzione e seguirà poi un'ultima fase di checking.

Per via delle difficoltà relative al sovramentionato rispetto dei dettami di qualità in corrispondenza di punti critici e degli spigoli, può essere necessario agire sui parametri contenuti nel sotto dizionario *addLayersControls*, nonchè utilizzare i *relaxed parameters* alla voce *meshQuality-controls*. Ciò avviene a discapito della bontà della mesh: infatti, come già discusso, SnappyHexMesh rimuove gli strati di prismi dalle zone che non rispettano le specifiche.

```
addLayersControls
```

```
{
    layers
    {
        asmoCar_body
```

---

```
{  
    nSurfaceLayers 5;  
    relativeSizes true;  
    expansionRatio 1.2;  
    finalLayerThickness 0.2;  
}  
asmoCar_frontWheel  
{  
    nSurfaceLayers 5;  
    relativeSizes true;  
    expansionRatio 1.2;  
    finalLayerThickness 0.2;  
}  
asmoCar_back  
{  
    nSurfaceLayers 5;  
    relativeSizes true;  
    expansionRatio 1.2;  
    finalLayerThickness 0.2;  
}  
asmoCar_backWheel  
{  
    nSurfaceLayers 5;
```

---

```
        relativeSizes true;  
        expansionRatio 1.2;  
        finalLayerThickness 0.2;  
    }  
    ground  
    {  
        nSurfaceLayers 5;  
        relativeSizes true;  
        expansionRatio 1.2;  
        finalLayerThickness 0.2;  
    }  
}
```

All'interno della voce "layers" è possibile visualizzare i 4 parametri caratteristici:

- *nSurfacelayers*: specifica il numero di layers che si desidera aggiungere;
- *relativeSizes*: la dicitura "true" indica che lo spessore del primo layer e dell'ultimo sono definiti relativamente al lato della superficie della mesh subito adiacente; la dicitura "false" indica invece che tali spessori sono definiti come lunghezze assolute;
- *expansionRatio*: indica il fattore di espansione tra un layer ed il successivo;
- *finalLayerThickness*: indica, infine, lo spessore dell'ultimo layer.

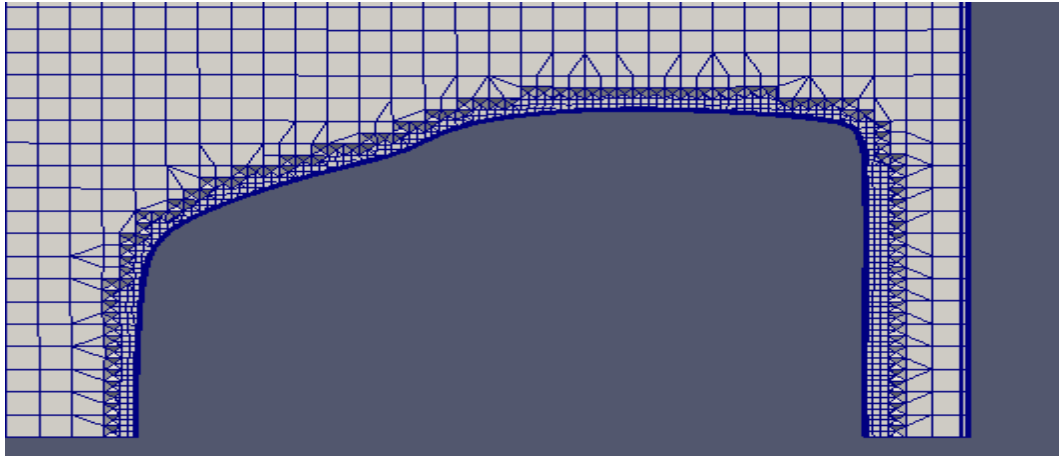


Figura 2.6: Mesh dopo l'aggiunta dei layers: vista lungo il piano sagittale mediano

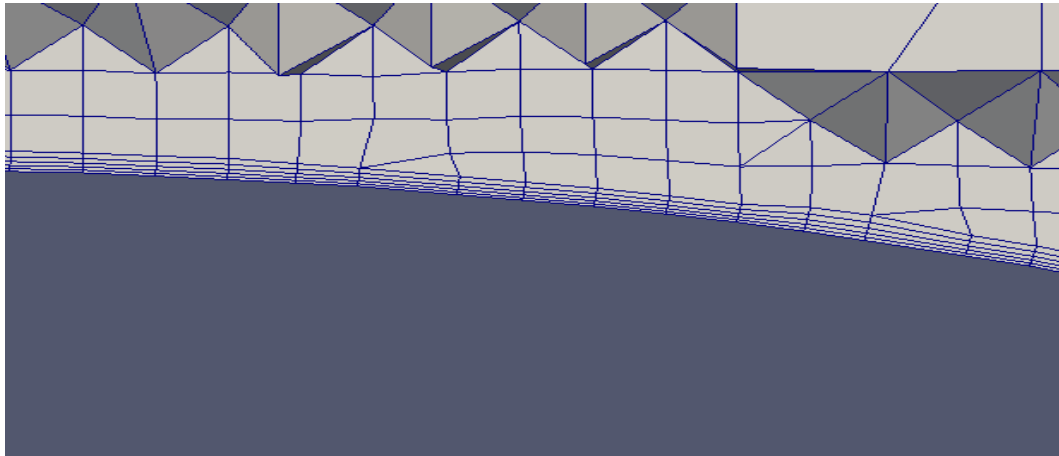


Figura 2.7: Mesh dopo l'aggiunta dei layers: zoom in corrispondenza del bordo dell'asmoCar

Nelle figure 2.6 e 2.7 sono presentati i risultati dopo l'aggiunta dei layers.



# Capitolo 3

## Analisi parametrica

In questa sezione si presenta l'analisi dei risultati, ottenuti a valle delle simulazioni e corredati dalle rispettive elucubrazioni in merito ai parametri.

Alcuni di questi sono stati mantenuti con il loro standard setting; altri invece sono stati studiati entro un adeguato range di valori al fine di rilevarne il comportamento in termini di massima non ortogonalità, massima skewness, numero di celle e tempo di creazione della mesh; i risultati sono poi stati plottati con GnuPlot.

Le analisi sono state effettuate in seriale, così da creare le medesime condizioni operative per ogni parametro. I tempi di generazione sono stati graficati dopo un'opportuna normalizzazione rispetto al valore di default di ogni parametro.

Nelle sezioni 4.1, 4.2 e 4.3 verranno indagati i parametri relativi alla **Castellatedmesh**; a seguire quelli sullo **Snapping** nella 4.4, ed infine quelli sulla **Layer addiction** (4.5, 4.6, 4.7 e 4.8).

Laddove i settaggi relativi alla qualità della mesh non potevano essere soddisfatti dalla combinazione di parametri selezionata, si sono adottati i cosiddetti relaxed parameters, al fine di imporre condizioni meno stringenti. Tali condizioni sono le seguenti:

- MaxNonOrthogonality 75;
- MaxBoundarySkewness 20;
- MaxInternalSkewness 6.

---

## 3.1 Refinement Parameters

### 3.1.1 MaxLocalCells

Rappresenta il numero di celle locali per processore. Tale parametro è stato settato a 100000.

### 3.1.2 MaxGlobalCells

Rappresenta il numero limite di celle nella fase di refinement immediatamente precedente a quella di removing. Nel caso si raggiunga un valore superiore a quello stabilito, il refinement termina immediatamente impedendo il raggiungimento del livello desiderato. Questo valore è stato fissato a 200000.

MaxLocalCells e MaxGlobalCells sono parametri che non hanno valenza effettiva e sono solo una valvola di sicurezza numerica che inibisce problemi del tipo "out of memory" relativi alla memoria fisica a disposizione.



### 3.1.3 MinRefinementCells

Il loop di refinement potrebbe impiegare un elevato numero di iterazioni per raffinare soltanto un esiguo numero di celle. Questo setting causerà lo stop del refinement nel caso in cui il numero di celle da affinare sia inferiore a quello impostato. Si nota un netto peggioramento in termini di massima non ortogonalità per il valore 1000; parimenti si riscontra un aumento del tempo di creazione della mesh pari al 4%. Il valore preferibile risulta quindi "100" in quanto ad esso corrisponde un basso tempo di creazione, pur consentendo un elevato numero di iterazioni.

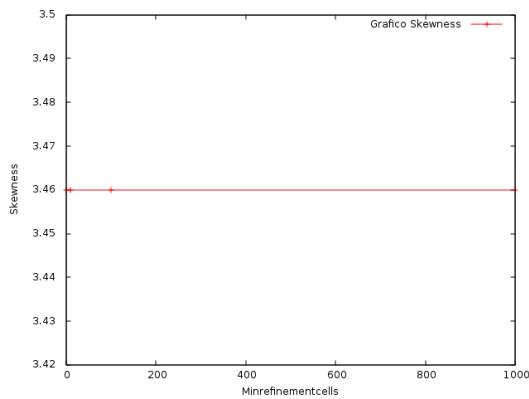


Figura 3.1: Skewness

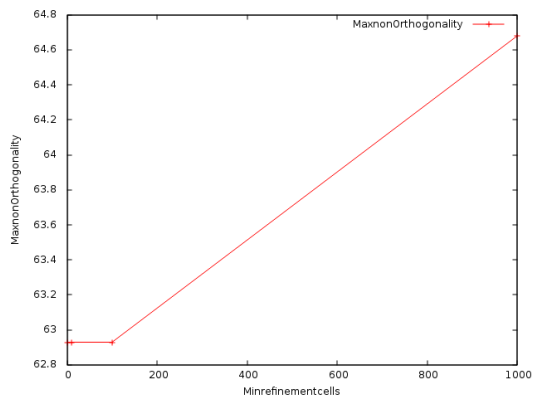


Figura 3.2: MaxnonOrthogonality

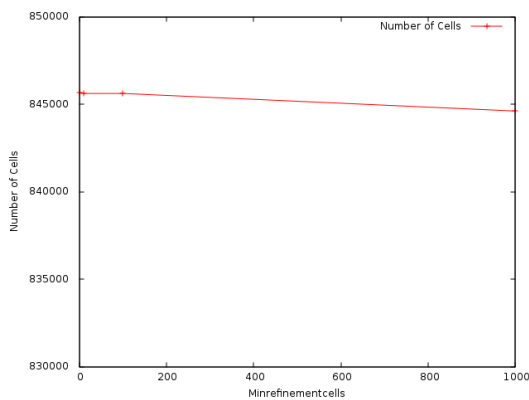


Figura 3.3: Numero di celle

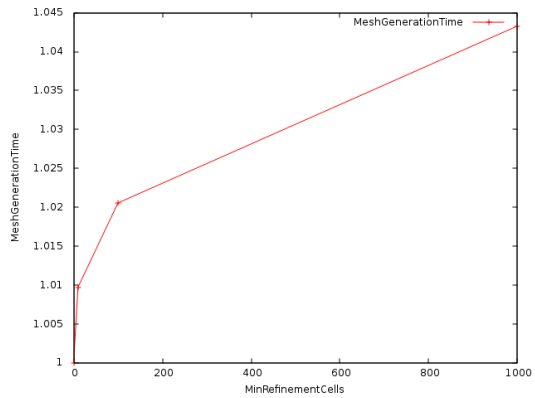


Figura 3.4: Tempo di creazione mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0	62.93	3.46	845690	1.00
1	62.93	3.46	845620	1.01
100	62.93	3.46	845620	1.02
1000	64.68	3.46	844613	1.04

Tabella 3.1: Minrefinementcells

### 3.1.4 nCellsBetweenLevels

Rappresenta il numero di buffer layers tra i diversi livelli di refinement. Tali livelli (levels) sono menzionati nei sub-dizionari refinementRegions e refinementSurfaces rispettivamente per le regioni e le superfici.

Nelle figure 3.5, 3.6, 3.7 e 3.8 sono riportati rispettivamente i valori di MaxNonOrthogonality, Max Skewness, numero di celle e tempo di creazione della mesh in corrispondenza dei tre valori studiati: 1, 2 e 3. "1" risulta il valore preferibile, in quanto presenta una bassa non ortogonalità. Anche il valore "3" potrebbe essere eleggibile, visto l'ottimo comportamento in termini di skewness. Ma l'elevato numero di celle che viene a prodursi comporta, come diretto effetto, un costo computazionale di calcolo ben più oneroso, il che non lo rende il valore prescelto.

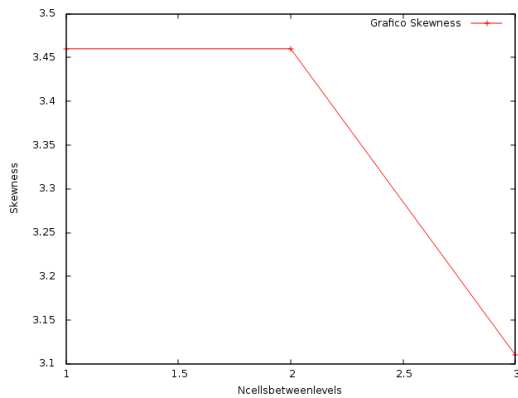


Figura 3.5: Skewness

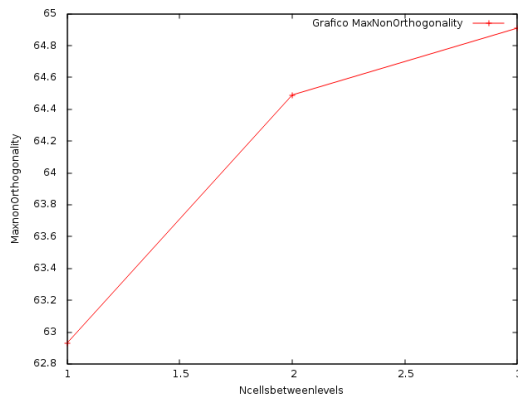


Figura 3.6: MaxnonOrthogonality

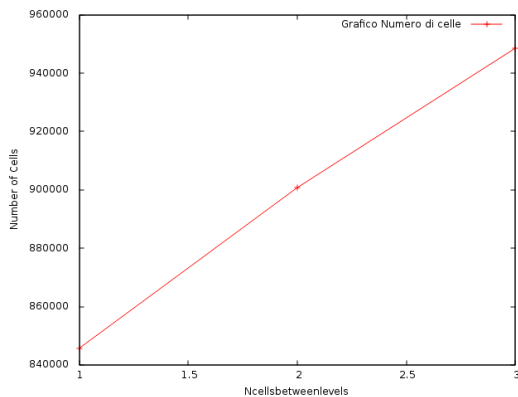


Figura 3.7: Numero di celle

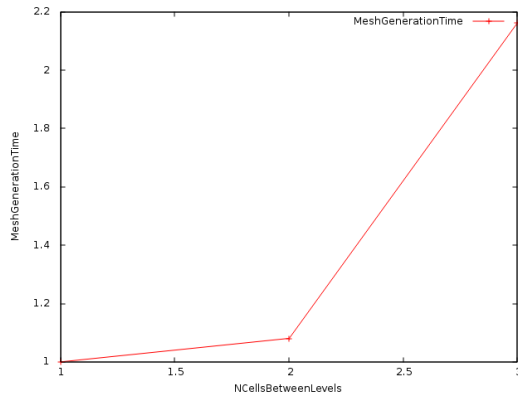
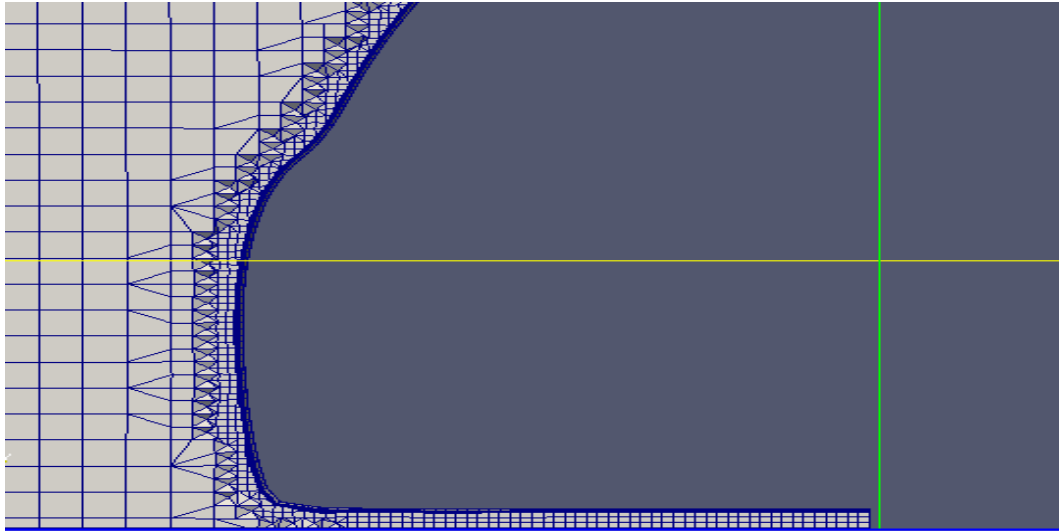
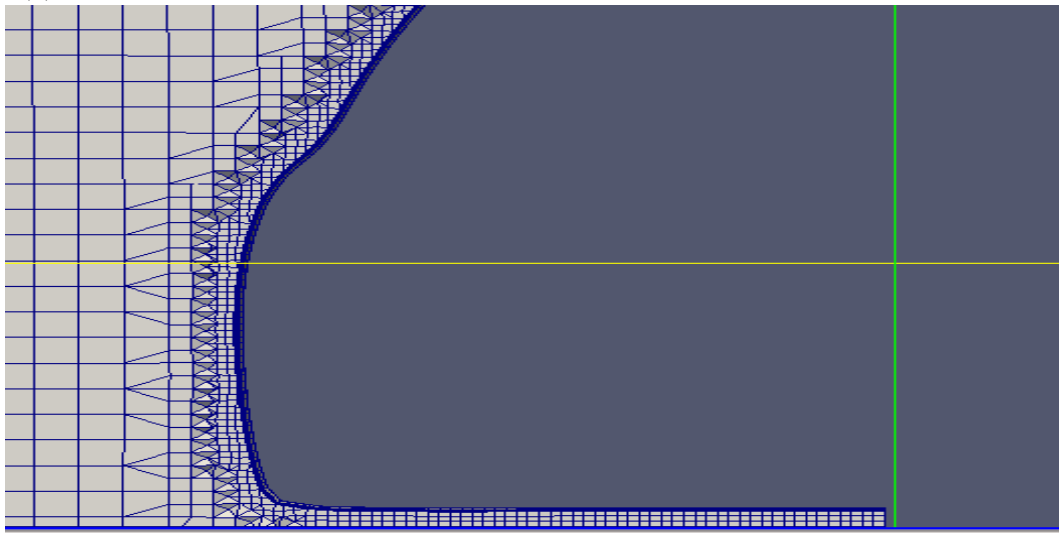


Figura 3.8: Tempo di creazione mesh normalizzato

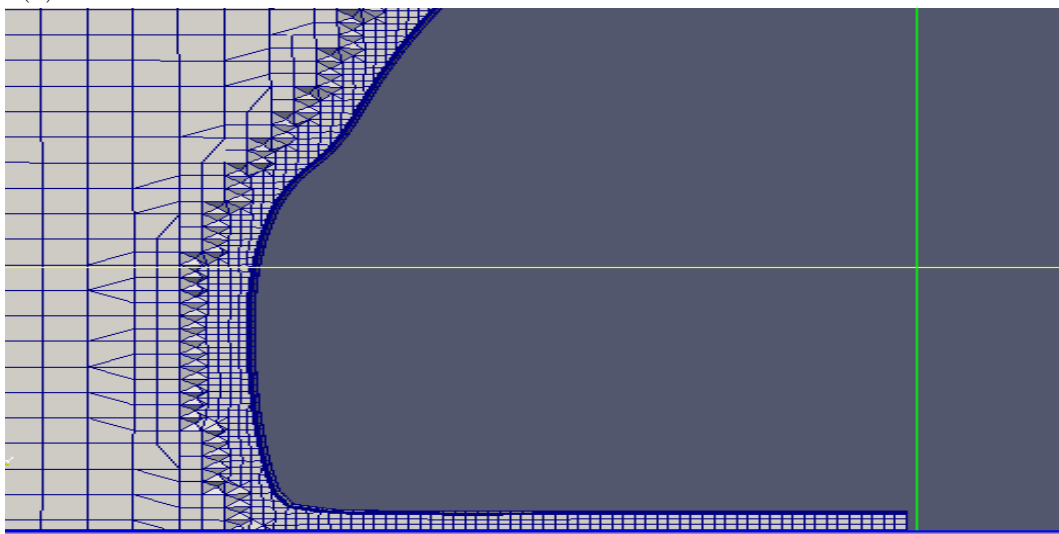
Si riportano inoltre, nel gruppo di figure 3.9a , 3.9b e 3.9c, i risultati del post processing con paraview, ottenuti a valle di una slice sulla macchina.



(a) *ncellsbetweenlevels1*



(b) *ncellsbetweenlevels2*



(c) *ncellsbetweenlevels3*

Figura 3.9: Tre screen con paraview

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
1	62.93	3.46	845690	1.00
2	64.49	3.46	900794	1.08
3	64.91	3.11	948569	2.16

Tabella 3.2: Ncellsbetweenlevels

### 3.1.5 MaxLoadUnbalance

Questo parametro consente un certo livello di squilibrio durante l'esecuzione del refinement in parallelo; è espresso come frazione del perfetto equilibrio, che a sua volta è ottenuto rapportando il numero totale di celle a quello dei processori (4). L'unico parametro soggetto a questo setting è il tempo: i valori "0.0" e "0.5" ne consentono una lieve riduzione pari al 3%.

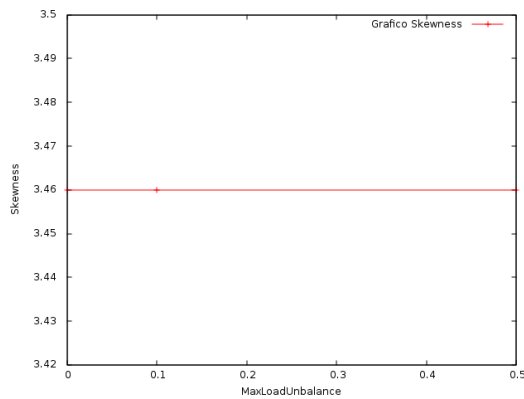


Figura 3.10: Skewness

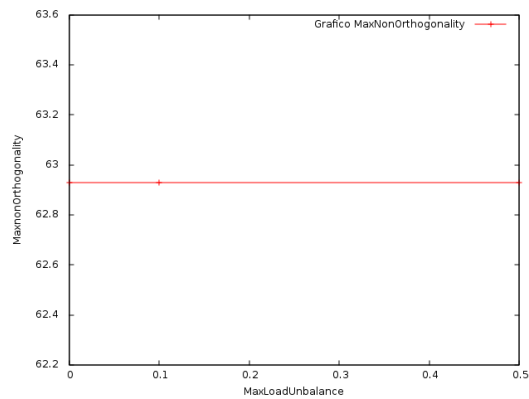


Figura 3.11: MaxnonOrthogonality

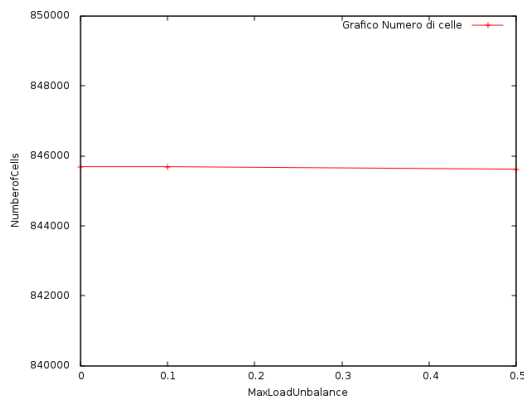


Figura 3.12: Numero di celle

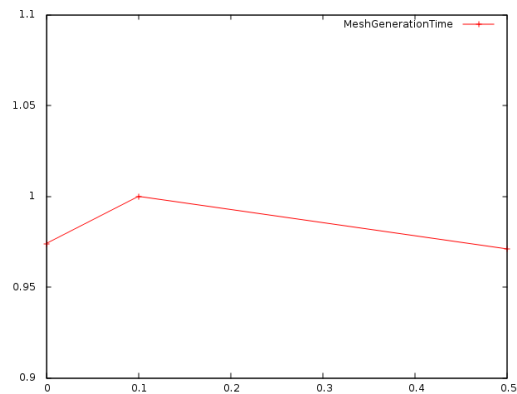


Figura 3.13: Tempo di creazione mesh normalizzato

---

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0.0	62.93	3.46	845690	0.97
0.1	62.93	3.46	845690	1.00
0.5	62.93	3.46	845620	0.97

Tabella 3.3: MaxLoadUnbalance

## 3.2 Mesh selection

### 3.2.1 LocationInMesh

Rappresenta le coordinate del punto materiale (-1.0 -0.5 0.5 ) che identifica l'interno del dominio da discretizzare.

### 3.2.2 AllowFreeStandingZoneFaces

Per ZoneFaces si intende un sottogruppo di facce della griglia organizzate come regione. Per sapere quali facce appartengono alla regione, è necessario accedere al file faceZoneMesh, dove, tramite l'usuale funzione di "lookup" esse possono essere individuate tramite il corrispettivo indice che ne fa da contrassegno. Il settaggio false indica il fatto che vengono prese in considerazione le sole facce lungo il bordo: questa risulta la scelta migliore, in quanto, con il settaggio true possono manifestarsi peggioramenti, durante la fase di snapping, in termini di qualità, come evidenziato in tabella 3.4.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
true	64.49	3.46	845620	1.00
false	62.93	3.46	845690	1.02

Tabella 3.4: AllowFreeStandingZoneFaces

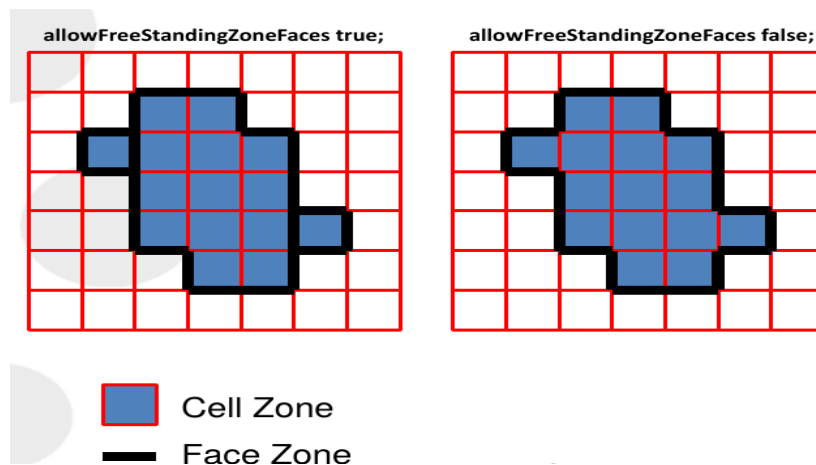


Figura 3.14: AllowFreeStandingZoneFaces [5]

## 3.3 SnapControls

### 3.3.1 ResolveFeatureAngle

Come già anticipato in capitoli precedenti, questo parametro fa sì che venga applicato il massimo livello di refinement alle celle le cui intersezioni individuano angoli che eccedono il valore settato. Il minimo livello di refinement viene invece applicato lungo la superficie del corpo geometrico. Affinchè sia in funzione, è necessario che il minimo ed il massimo livello di refinement differiscano tra loro, pertanto si è settato un livello minimo pari a 4, ed un livello massimo pari a 6. Entrambi, ovviamente, sono intesi sia rispetto al "body" e al "back" (ovvero l'automobile nella sua globalità meno che le ruote) sia rispetto al **frontwheel** e al **backwheel**. L'influenza di tale parametro è rilevabile sugli spigoli e su punti critici, come accade in figura 3.19 in cui si denota un, seppur lieve, decremento del numero di celle. In medio stat virtus: il valore 90 presenta le caratteristiche complessive migliori, pertanto risulta il valore selezionato.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
30	62.93	3.46	845690	1.00
90	63.39	3.23	844106	0.85
300	63.39	3.23	844029	0.84

Tabella 3.5: ResolveFeatureAngle

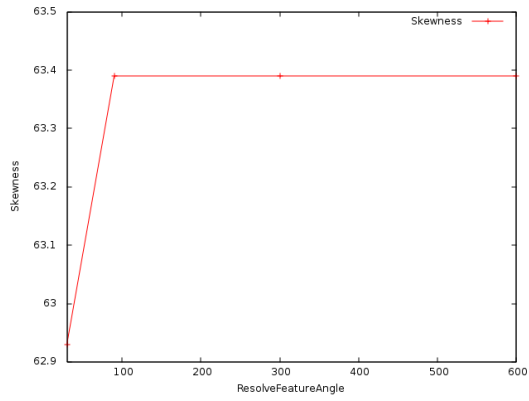


Figura 3.15: Skewness

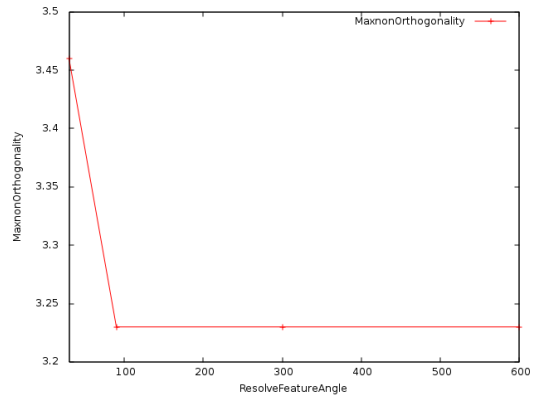


Figura 3.16: MaxnonOrthogonality

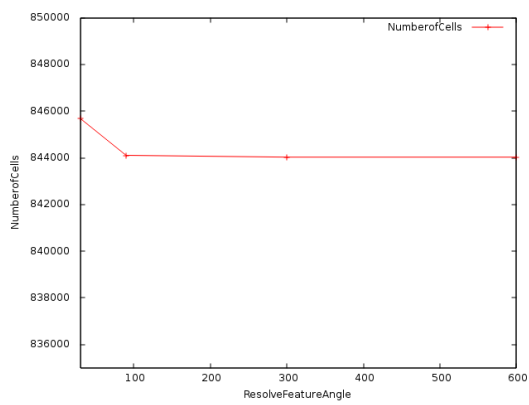


Figura 3.17: Numero di celle

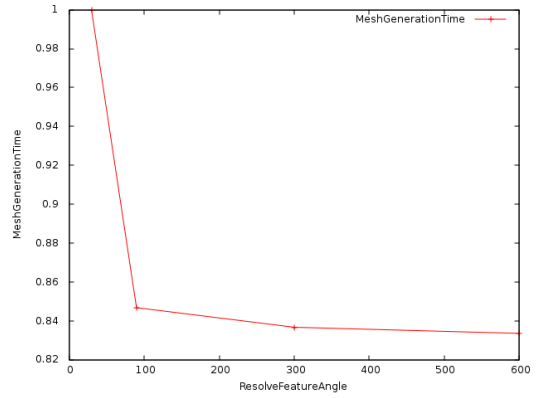
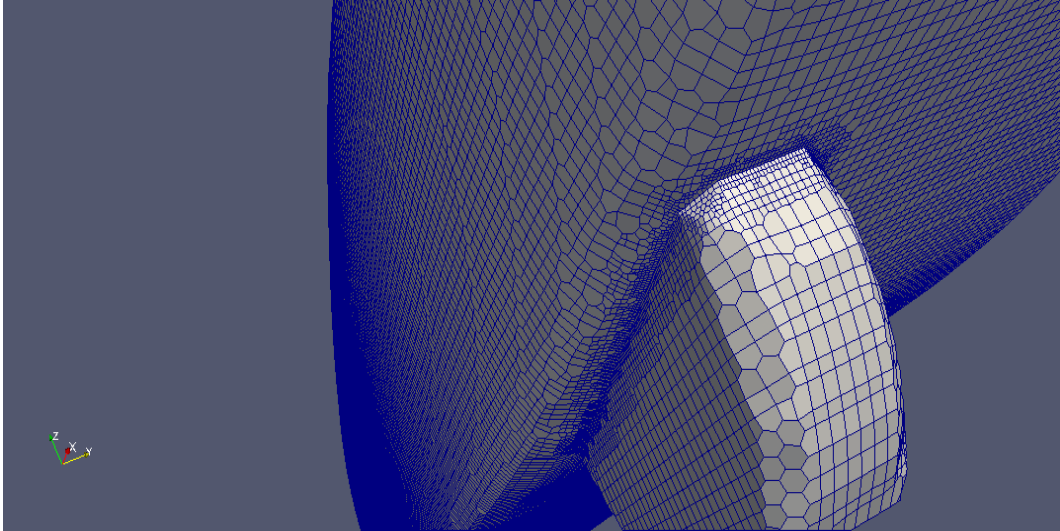
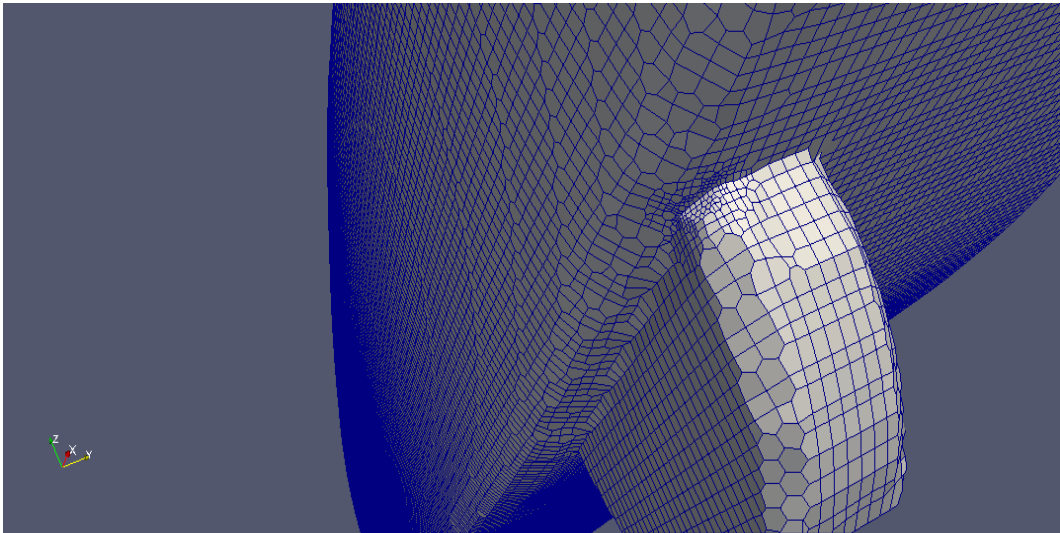


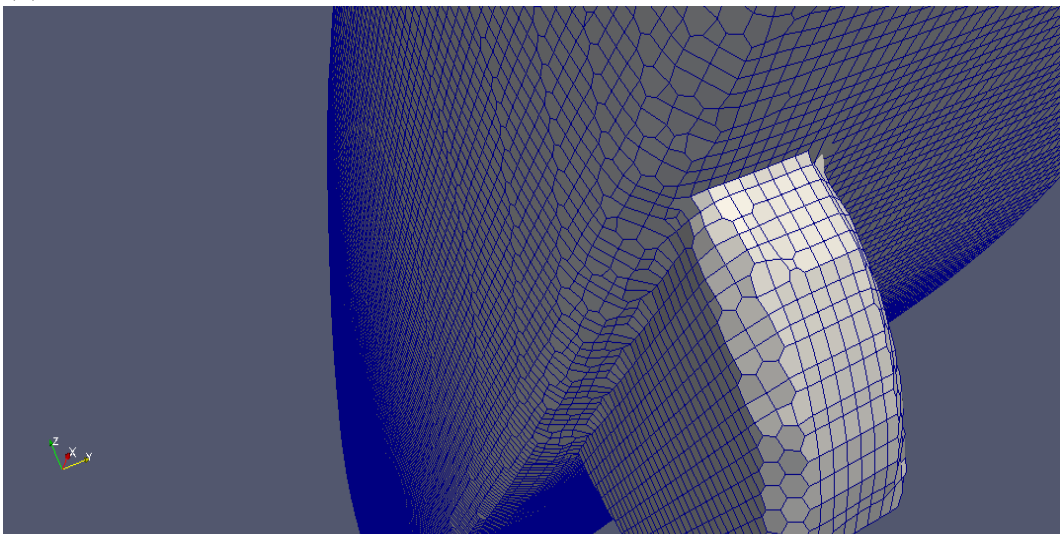
Figura 3.18: Tempo di creazione mesh normalizzato



(a) *resolve.featureangle30*



(b) *resolve.featureangle90*



(c)  
*resolve.featureangle300*

Figura 3.19: Tre screenshots con paraview



### 3.3.2 NSmoothPatch

Questo parametro regola la lisciatura della superficie: al suo accrescimento infatti, corrisponde un aumento del numero di iterazioni della lisciatura stessa al fine di ottenere una superficie più levigata e nitida. Il valore migliore risulta "3" per il quale si ha una non-ortogonalità massima leggermente inferiore.

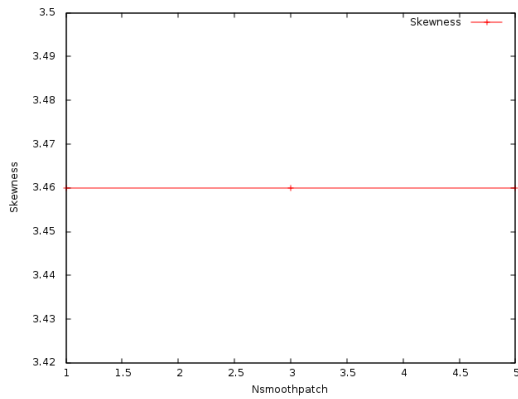


Figura 3.20: Skewness

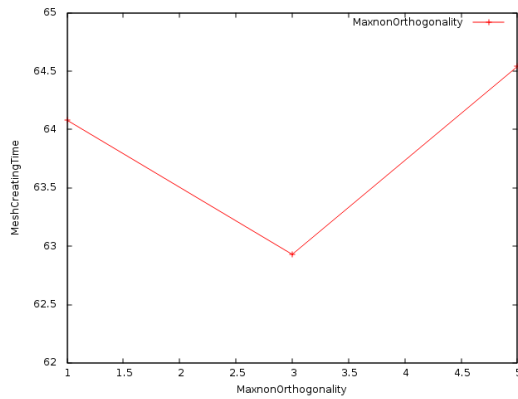


Figura 3.21: MaxnonOrthogonality

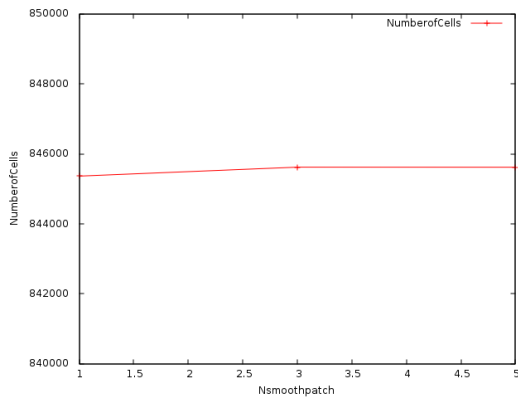


Figura 3.22: Numero di celle

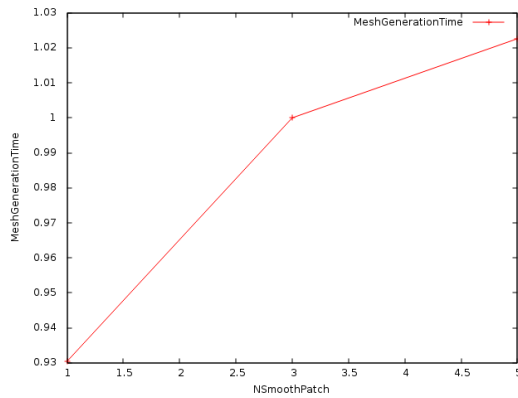


Figura 3.23: Tempo di creazione mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
1	64.08	3.46	845363	0.93
3	62.93	3.46	845620	1.00
5	64.54	3.46	845618	1.02

Tabella 3.6: NSmoothPatch

---

### 3.3.3 PlanarAngle

Parametro utilizzato per stabilire se le normali alla superficie coincidano o meno. E' stato settato ad un valore pari a 30.

### 3.3.4 NSolveIter

Questo settaggio agisce sul numero di iterazioni della fase di snapping. Tutto ciò ha una diretta conseguenza sulla qualità della mesh, in quanto più equidistante sarà la mesh in corrispondenza dei contorni tanto più tempo occorrerà per effettuare lo snapping. Questo parametro avrà particolare evidenza grafica qualora la superficie, per natura geometrica intrinseca, presenti considerevoli differenze di quota. Se la superficie è piatta l'effetto di questo parametro tende ad essere fortemente mitigato.

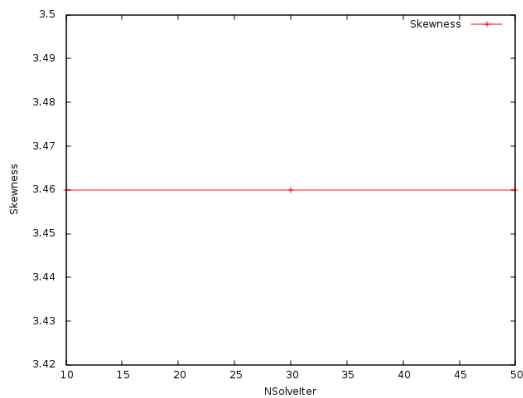


Figura 3.24: Skewness

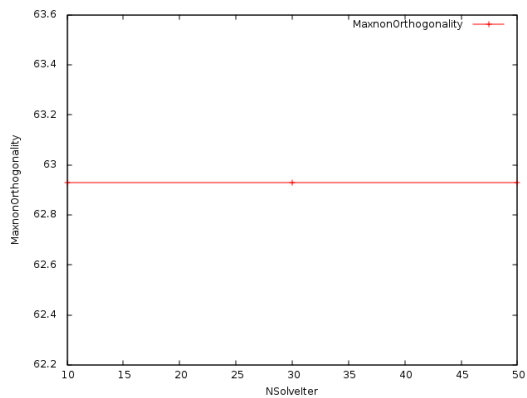


Figura 3.25: MaxnonOrthogonality

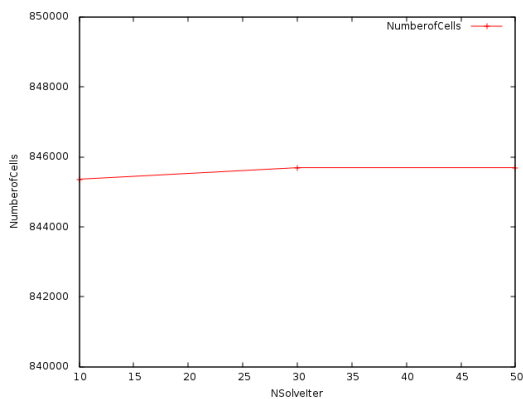


Figura 3.26: Numero di celle

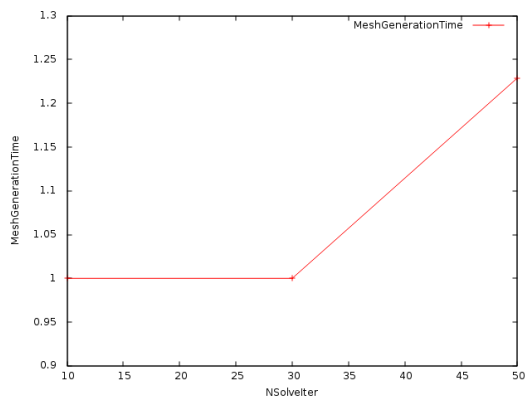


Figura 3.27: Tempo di creazione della Mesh

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
10	62.93	3.46	845360	1.00
30	62.93	3.46	845690	1.00
50	62.93	3.46	845690	1.23

Tabella 3.7: NSolveIter

### 3.3.5 NRelaxIter

Rappresenta il numero massimo di iterazioni in cui snappyHexMesh farà girare uno script relaxed, ovvero con i parametri di qualità peggiorati, al fine di rimuovere punti inadeguati nella mesh. Anche qui, come in nsolveiter, la scelta è quasi indifferente, vista la poca incidenza sulla qualità della mesh e la rara necessità di utilizzo dei relaxed parameters. L'unico fattore degno di nota è quello temporale: adottando il valore 2 si ottiene una riduzione del tempo di generazione della mesh pari al 9%.

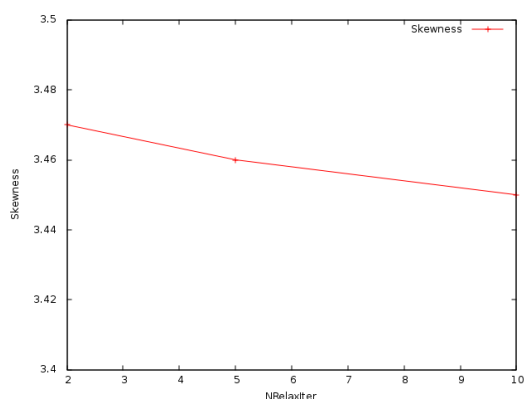


Figura 3.28: Skewness

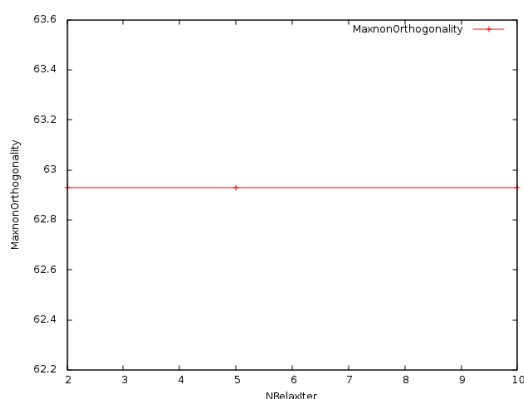


Figura 3.29: MaxnonOrthogonality

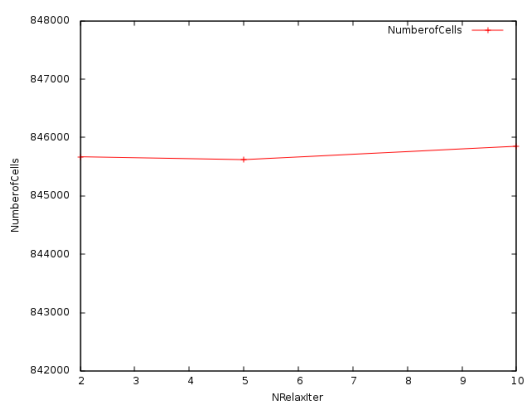


Figura 3.30: Numero di Celle

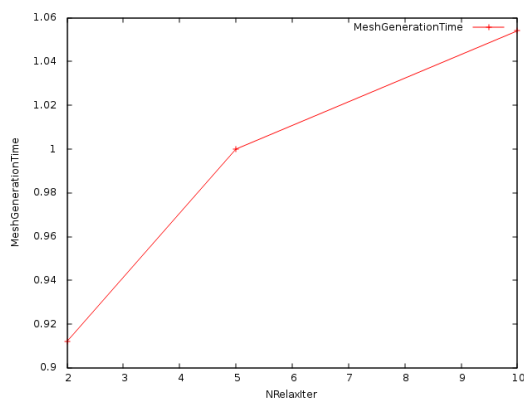


Figura 3.31: Tempo di Creazione Mesh normalizzato

---

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
2	62.93	3.47	845670	0.91
5	62.93	3.46	845620	1.00
10	62.93	3.45	845850	1.05

Tabella 3.8: NRelaxIter

### 3.3.6 Tolerance

Questa opzione governa la distanza in corrispondenza della quale SnappyHexMesh inizia a cercare un punto in cui effettuare lo snap. Il rapporto tra tale distanza, in cui i punti vengono annessi alla superficie, e la lunghezza massima del bordo locale fornisce questo settaggio di tolleranza, qui fissato a 2.0.

## 3.4 Feature snapping

### 3.4.1 NFeatureSnapIter

Rappresenta il numero di iterazioni da effettuare, durante lo snapping, sulle **features**, ovvero su tutte quelle zone, all'interno di una geometria più o meno complessa, che presentano criticità ovvero in cui si hanno improvvisi cambiamenti della direzione della normale alla superficie. Tali features possono essere migliorate eseguendo un opportuno refinement sulle celle locali. In questo caso si hanno due possibile scelte: la prima ricade sul valore "20" per il quale si registrano migliori prestazioni relative alle grandezze caratteristiche della qualità. La seconda, relativa al valore "5" fa registrare sì più alte skewness e non-ortogonalità, ma il risparmio in termini temporali rispetto al caso precedente è assai pregevole.

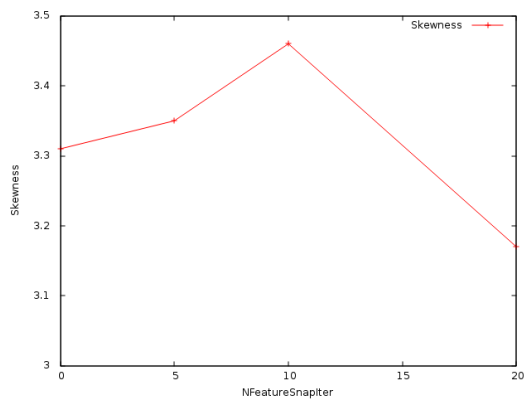


Figura 3.32: Skewness

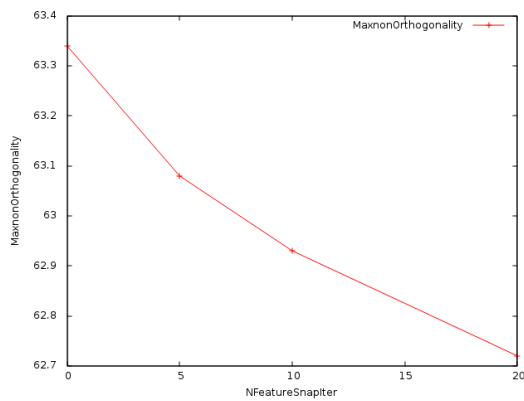


Figura 3.33: MaxnonOrthogonality

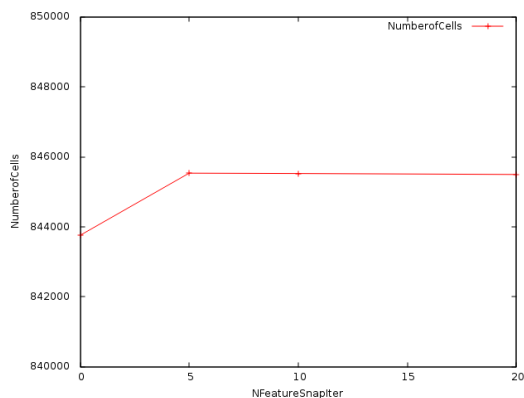


Figura 3.34: Numero di Celle

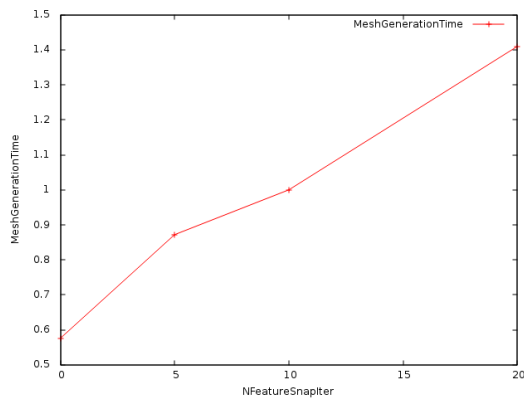


Figura 3.35: Tempo di creazione mesh normalizzato

---

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0	63.34	3.31	843757	0.58
5	63.08	3.35	845535	0.87
10	62.93	3.46	845525	1.00
20	62.72	3.17	845487	1.41

Tabella 3.9: NFeatureSnapIter

### 3.4.2 ImplicitFeatureSnap

Il metodo implicito non richiede all'utente di estrarre le features, ma effettua un campionamento della superficie per poi avvalersi del ResolveFeatureAngle per identificarle; ha il grosso vantaggio di essere un processo automatizzato.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
true	62.93	3.46	845690	1.00
false	64.49	3.46	845690	1.00

Tabella 3.10: ImplicitFeatureSnap

### 3.4.3 ExplicitFeatureSnap

Con l'ExplicitFeatureSnap, invece, è l'utente che estrae le features dalla superficie geometrica utilizzando la SurfaceFeatureExtractUtility e poi le specifica esplicitamente nei controlli della Castellated Mesh alla voce CastellatedMeshControls all'interno di snappyHexMeshDict, il dizionario di snappyHexMesh. Questo metodo esplicito offre un migliore controllo sulle features, in quanto manuale: pertanto verrà settato su true. Il metodo implicito verrà invece impostato su false.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
true	62.93	3.46	845690	1.00
false	63.39	3.23	844029	0.83

Tabella 3.11: ExplicitFeatureSnap

---

## 3.5 AddLayersControls

### 3.5.1 RelativeSizes

Specifica se lo spessore dei layers sia definito in funzione della dimensione della cella immediatamente antistante gli stessi (`true`) o se se sia invece definito come lunghezza assoluta (`false`).

### 3.5.2 ExpansionRatio

Fattore di espansione dei layers della mesh fissato a 1.2.

### 3.5.3 FinalLayerThickness

Spessore dell'ultimo layer, settato a 0.1.

### 3.5.4 MinThickness

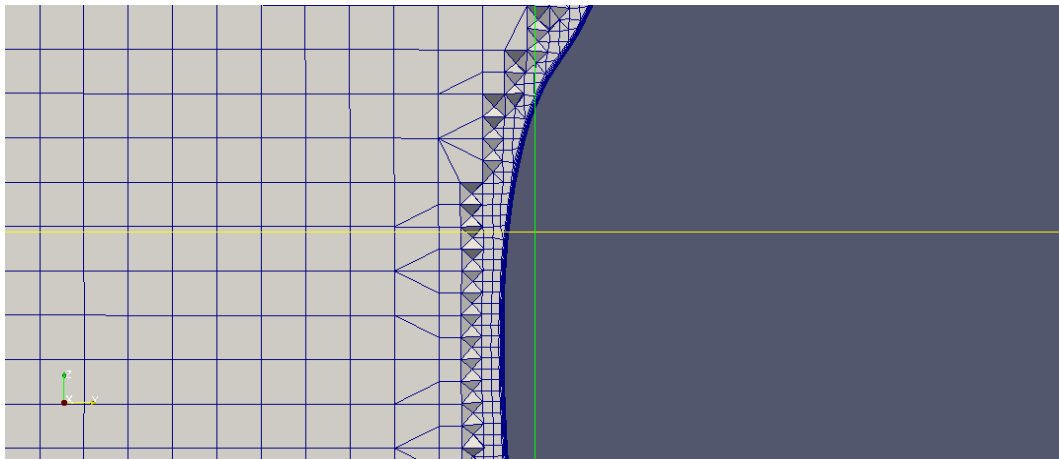
Minimo spessore totale dei layers settato a 0.0001. Se, per varie ragioni, durante la creazione della mesh, si dovesse raggiungere un valore inferiore a questo, allora il processo di introduzione dei layers verrebbe brutalmente interrotto da `SnappyHexMesh`, impedendone l'aggiunta.

### 3.5.5 Ngrow

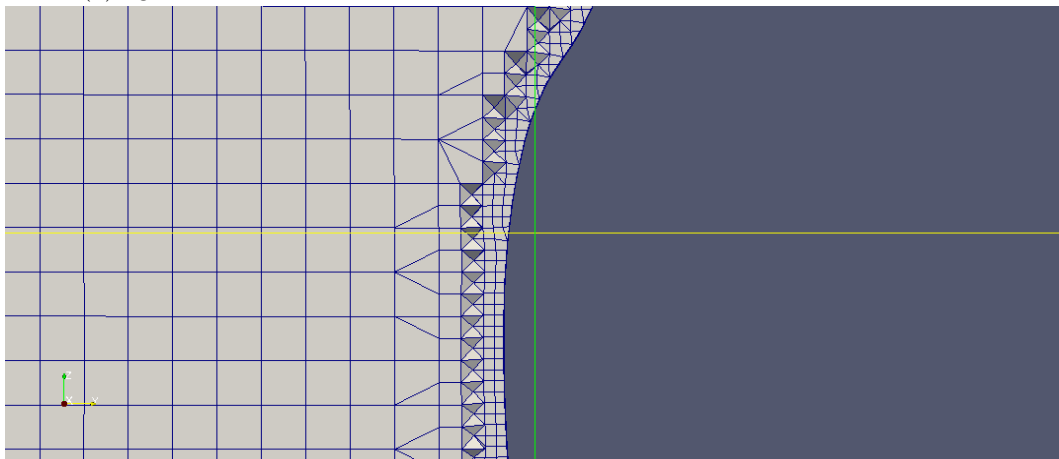
Ngrow è un parametro che regola l'aggiunta dei layers, come si evince dal post-processing in paraview. Al valore 0 corrisponde uno spessore uniforme dei layers sulla superficie. Il valore 1 fornisce invece una condizione meno stringente, garantendo eventuali disuniformità. Tuttavia, per questo valore, snappyHexMesh fallisce ad introdurre i layers e pertanto il valore "0" risulta una scelta obbligata.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0	62.93	3.46	845690	1
1	56.82	2.85	564911	1.7805

Tabella 3.12: NGrow



(a)  $ngrow = 0$



(b)  $ngrow = 1$

Figura 3.36: Screen con paraview dopo una slice



---

## 3.6 Static analysis of starting mesh

### 3.6.1 FeatureAngle

Tramite questo parametro si specificano le zone dei contorni in cui verranno aggiunte le celle. Rappresenta l'angolo oltre il quale non vengono estruse le superfici e per cui si ha il collasso automatico dei layers. Come si evince dalla tabella 3.13, il valore preferibile in termini di ortogonalità è "300"; d'altro canto "130" fornisce un significativo calo della skewness: essendo questo il valore più basso riscontrato nell'intera analisi parametrica è possibile e leggerlo come possibile valida alternativa.

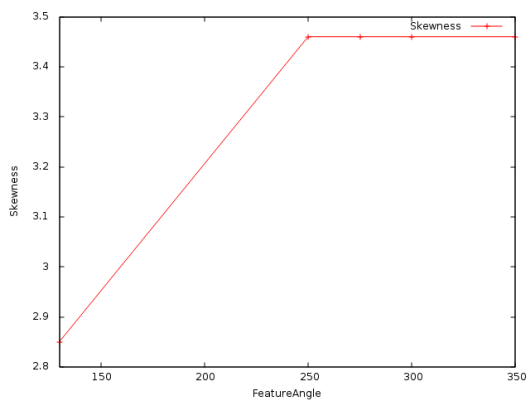


Figura 3.37: Skewness

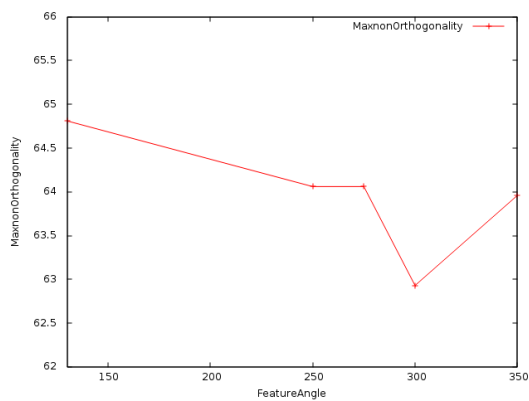


Figura 3.38: MaxnonOrthogonality

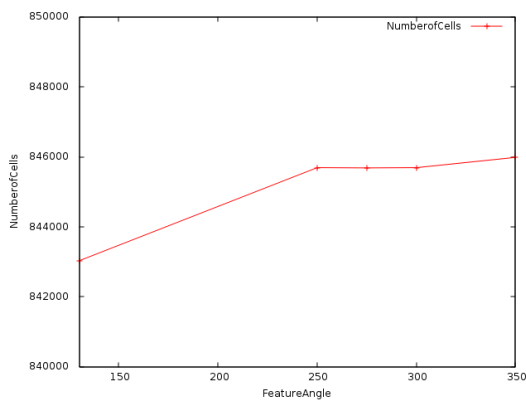


Figura 3.39: Numero di Celle

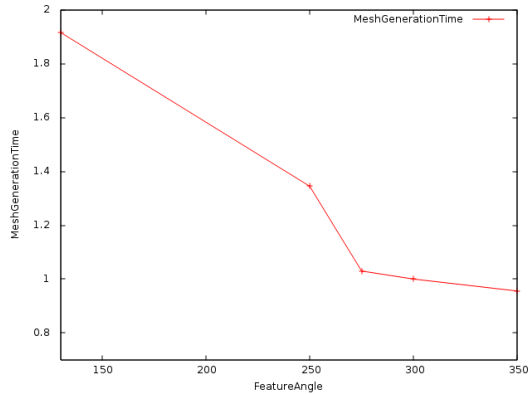
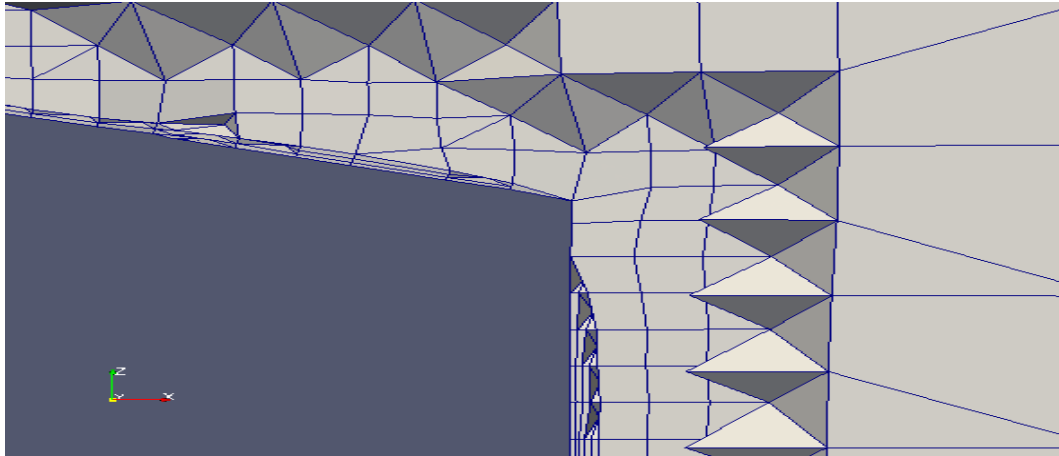


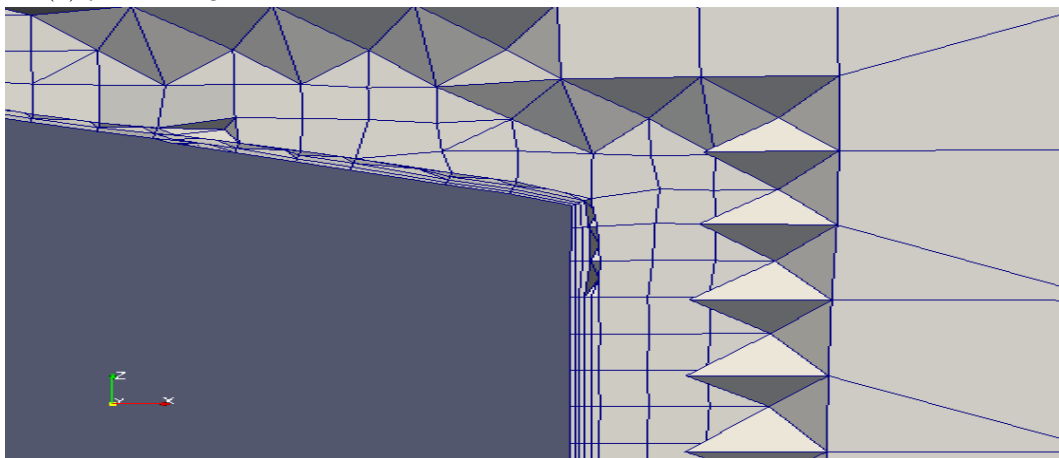
Figura 3.40: Tempo di creazione mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
130	64.81	2.85	843031	1.92
250	64.06	3.46	845690	1.34
275	64.06	3.46	845684	1.03
300	62.93	3.46	845690	1.00
350	63.96	3.46	845983	0.96

Tabella 3.13: FeatureAngle



(a) *featureangle130*



(b) *featureangle300*

Figura 3.41: Screen con paraview dopo una slice

### 3.6.2 MaxFaceThicknessRatio

Questo parametro è utilizzato per individuare le facce deformate così da impedire l'aggiunta dei layers sulle stesse. Per il valore 1 si rileva un miglioramento della skewness, a discapito però della non-ortogonalità.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0	62.93	3.46	845690	1.09
0.5	62.93	3.46	845470	1.00
1	64.91	3.11	845525	1.00

Tabella 3.14: MaxFaceThicknessRatio

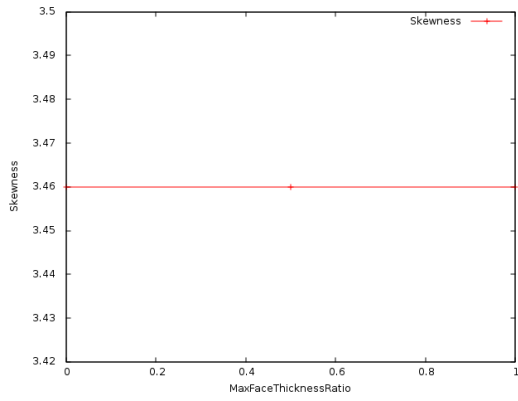


Figura 3.42: Skewness

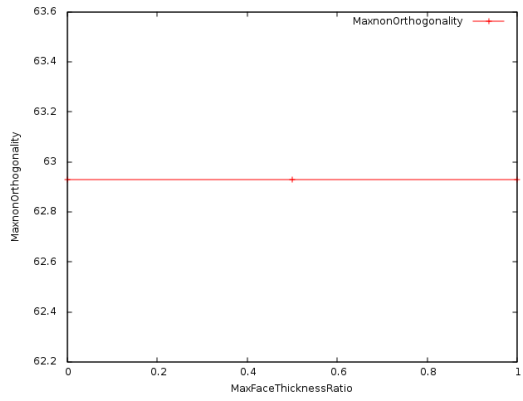


Figura 3.43: MaxnonOrthogonality

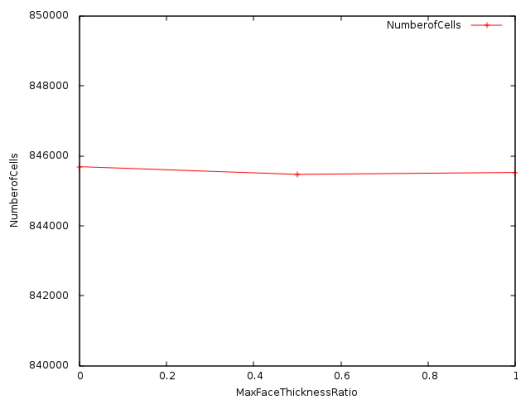


Figura 3.44: Numero di Celle

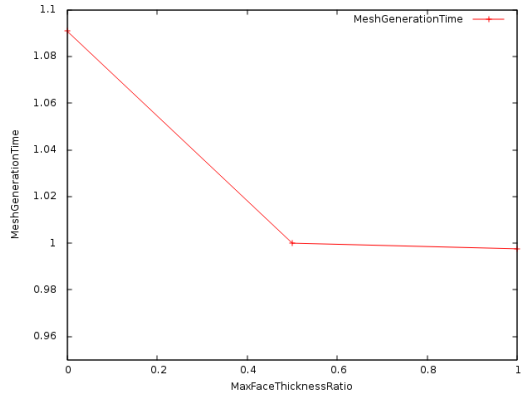


Figura 3.45: Tempo di creazione mesh normalizzato

## 3.7 Patch displacement

### 3.7.1 NSmoothSurfaceNormals

Numero di iterazioni di smoothing delle normali alle superfici. All'aumentare di questo valore si osserva, come evidenziato nel gruppo di figure in 3.50, come la normale alla superficie tenda progressivamente ad inclinarsi e ad assumere la stessa direzione di quella in corrispondenza dello spigolo. Vista il basso valore di skewness, il valore preferibile in questa sezione risulta "5".

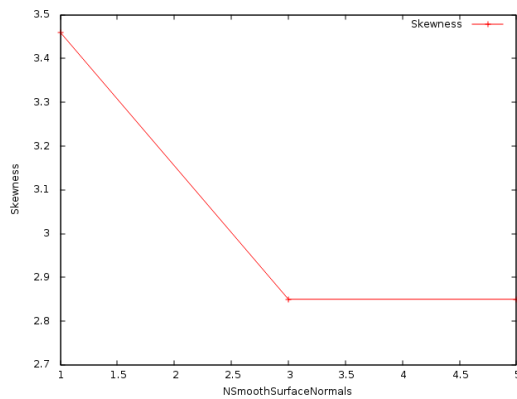


Figura 3.46: Skewness

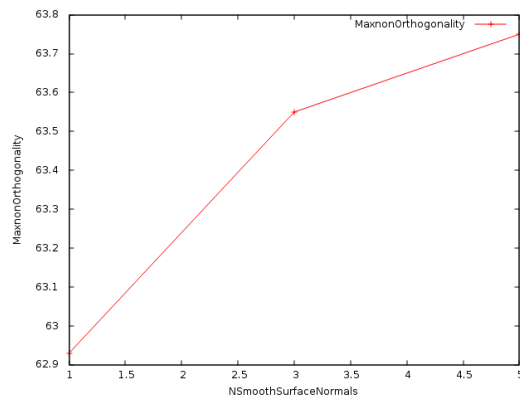


Figura 3.47: MaxnonOrthogonality

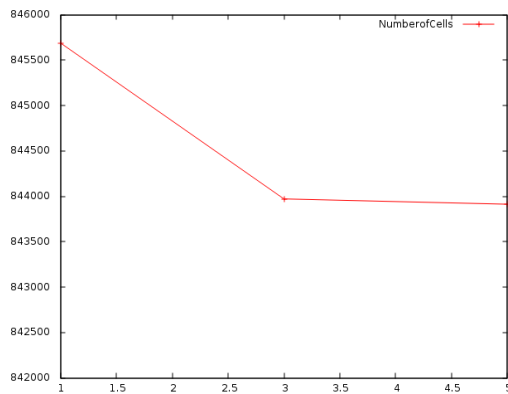


Figura 3.48: Numero di Celle

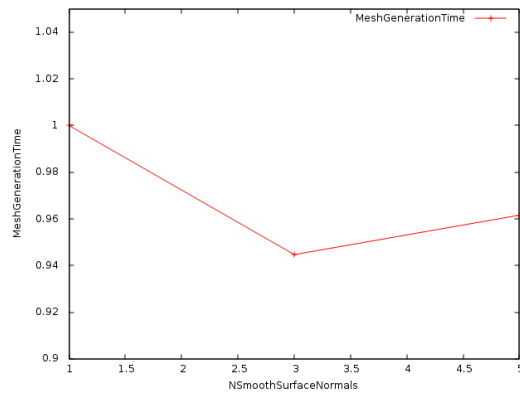
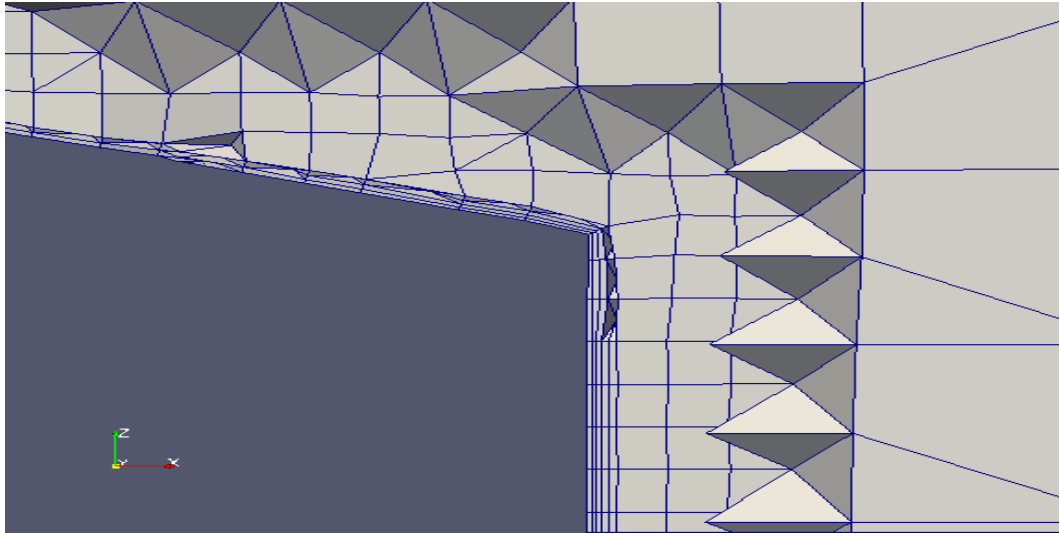


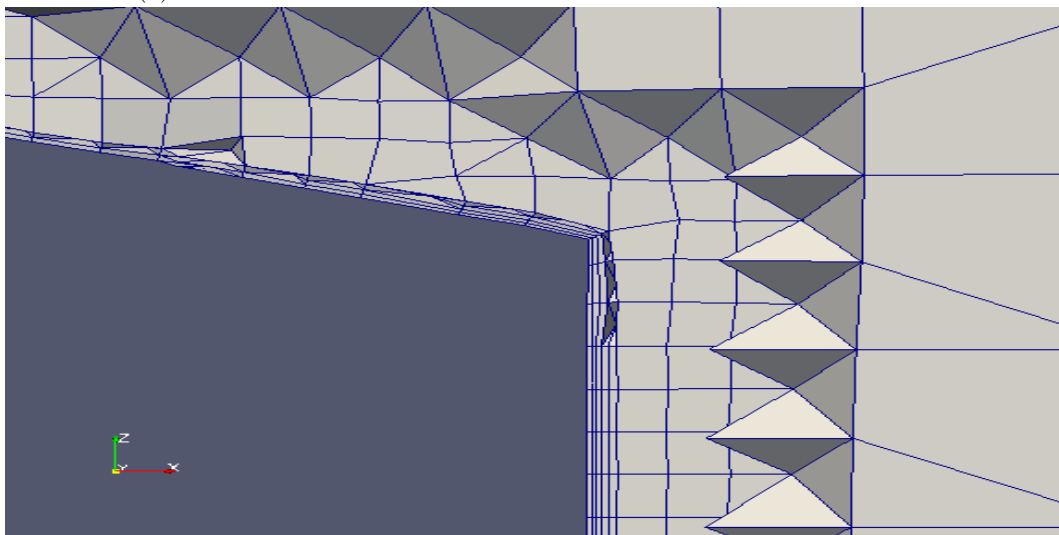
Figura 3.49: Tempo di creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
1	62.93	3.46	845690	1.00
5	63.55	2.85	843971	0.95
10	63.75	2.85	843913	0.96

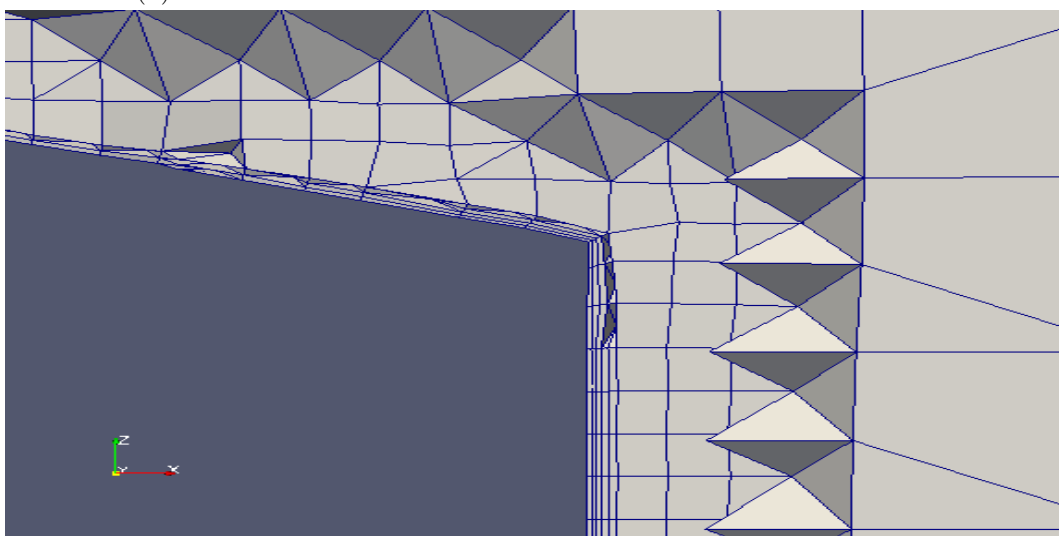
Tabella 3.15: NSmoothSurfaceNormals



(a) 1



(b) 5



(c) 10

Figura 3.50: Screen con paraview nel retro dell'automobile

### 3.7.2 NSmoothThickness

Numero di iterazioni di smoothing relativamente ai punti considerati sul viscous layer. Come evidenziato nel gruppo di figure 3.55 e 3.56 ,si rileva come lo strato di layer aggiunti tenda progressivamente a raffinarsi, seguendo fedelmente il profilo geometrico della macchina. Visto l'ottimo comportamento rilevato attraverso il post processing e il modesto valore di skewness si sceglie il valore "20".

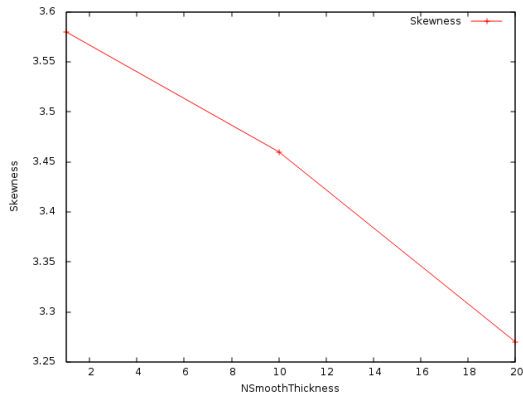


Figura 3.51: Skewness

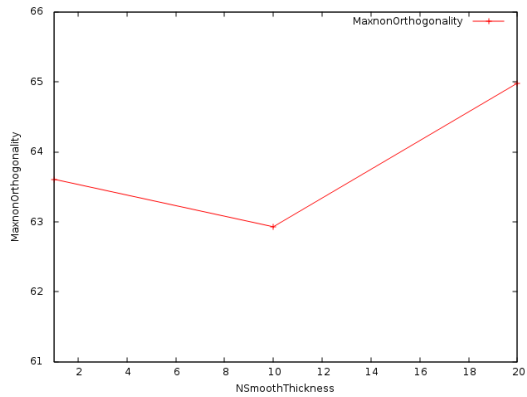


Figura 3.52: MaxnonOrthogonality

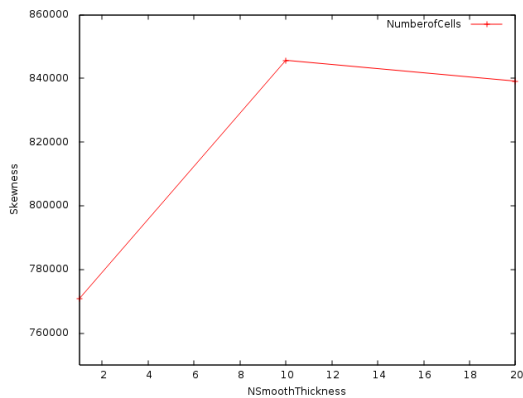


Figura 3.53: Numero di Celle

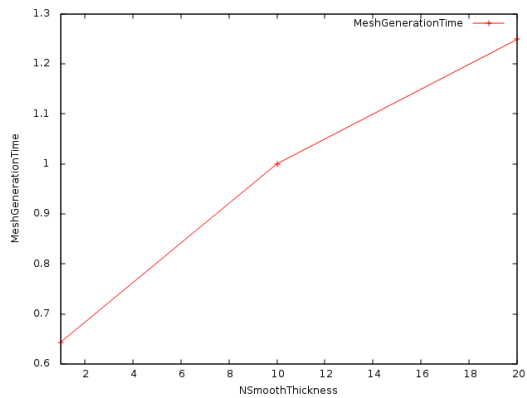
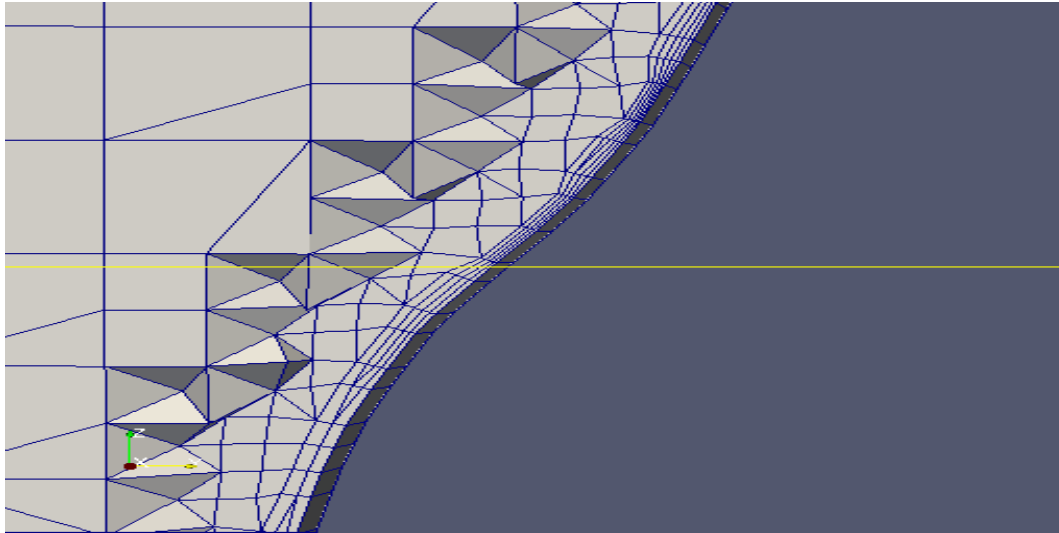


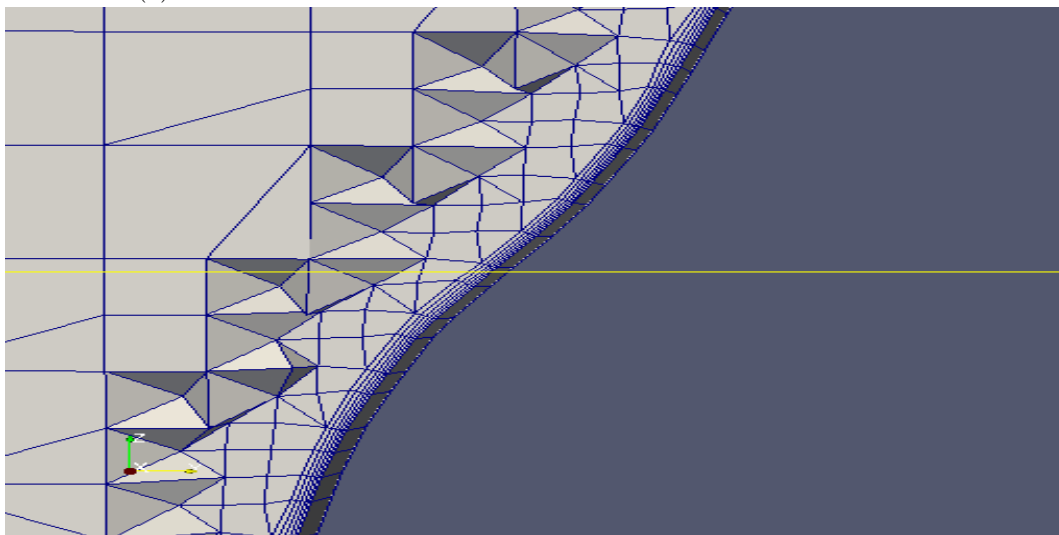
Figura 3.54: Tempo di Creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
1	63.61	3.58	770742	0.64
10	62.93	3.46	845690	1.00
20	64.98	3.27	839149	1.25

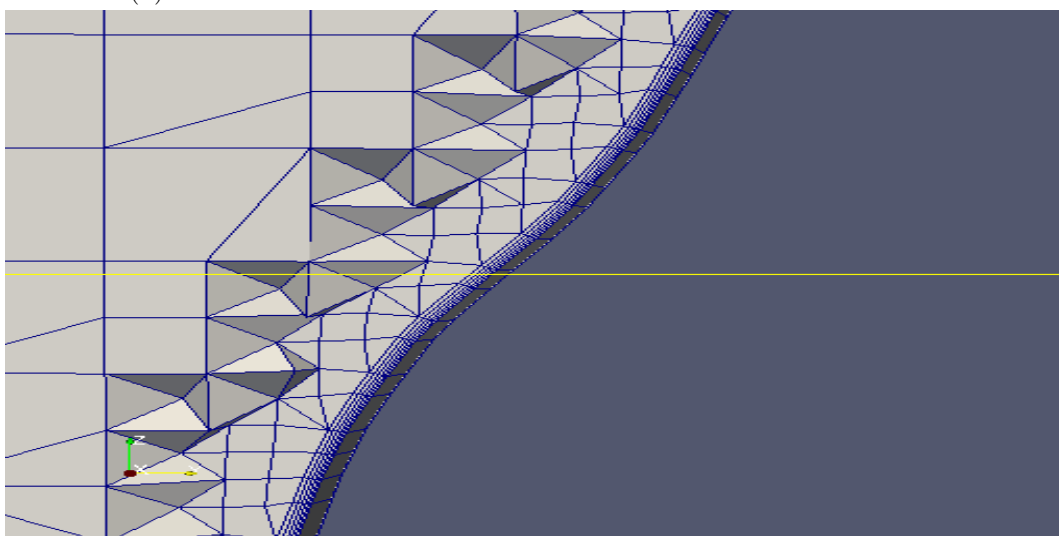
Tabella 3.16: NSmoothThickness



(a) 1



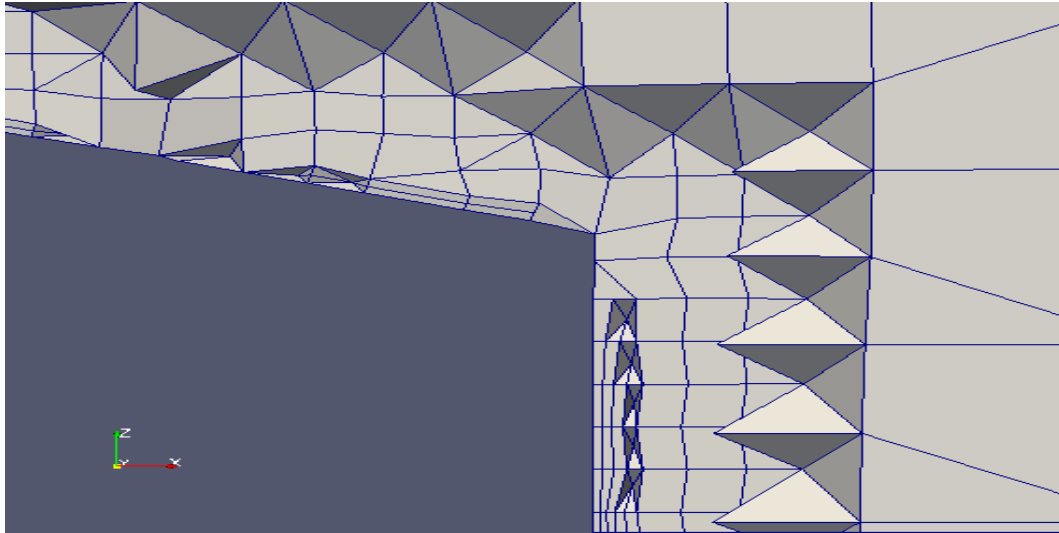
(b) 10



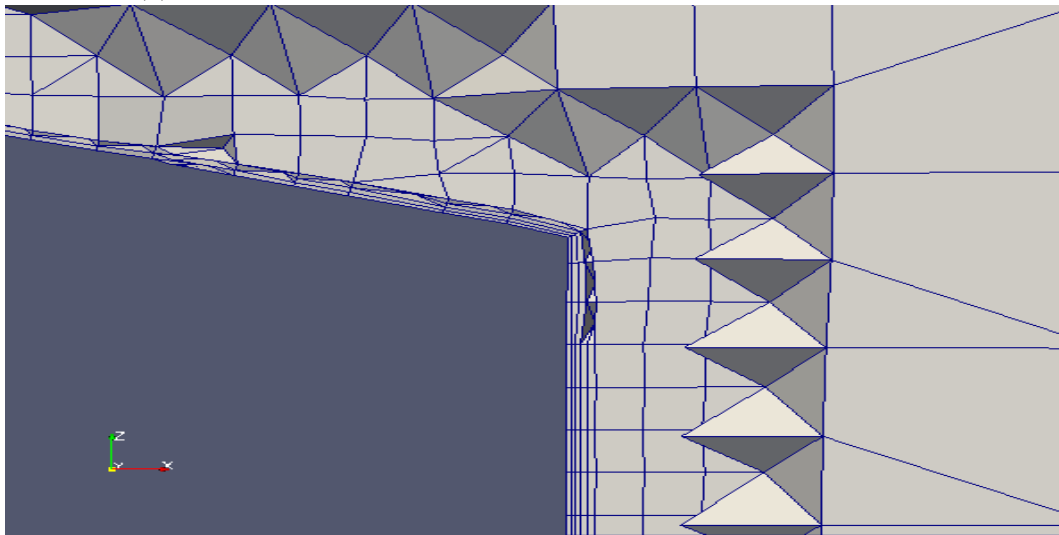
(c) 20

Figura 3.55: Screen con paraview

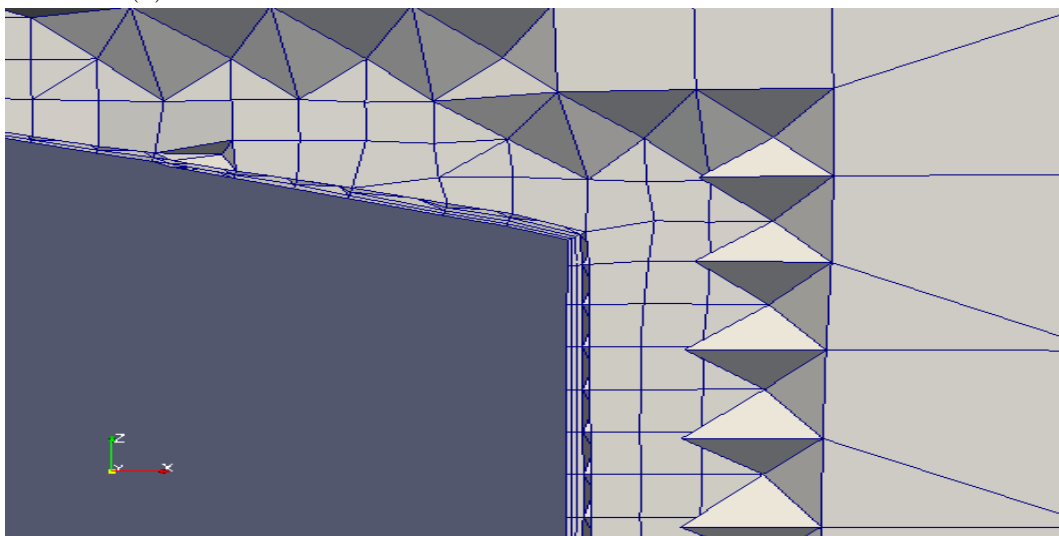




(a) 1



(b) 10



(c) 20

Figura 3.56: Screen con paraview

## 3.8 Medial axis analysis

### 3.8.1 MinMedianAxisAngle

Quest'angolo è utilizzato per definire un asse medio nel momento in cui, dovendo aggiungere i layers, si allontana la mesh dalla superficie.

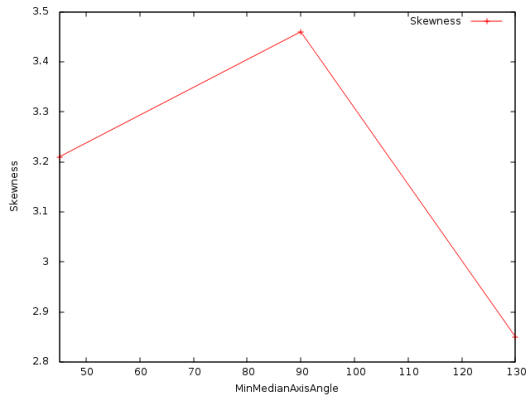


Figura 3.57: Skewness

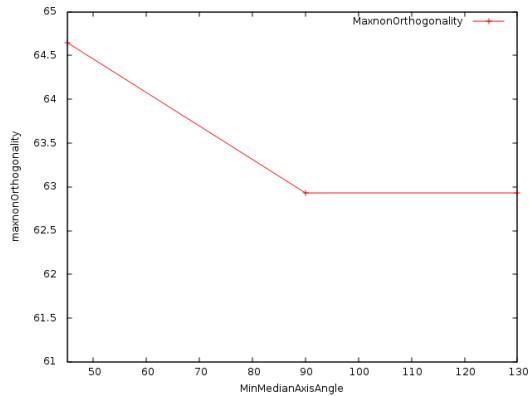


Figura 3.58: MaxnonOrthogonality

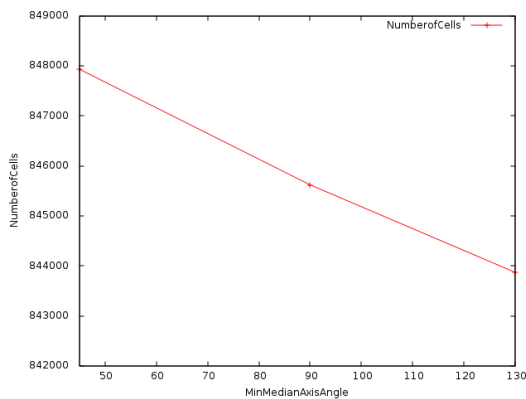


Figura 3.59: Numero di Celle

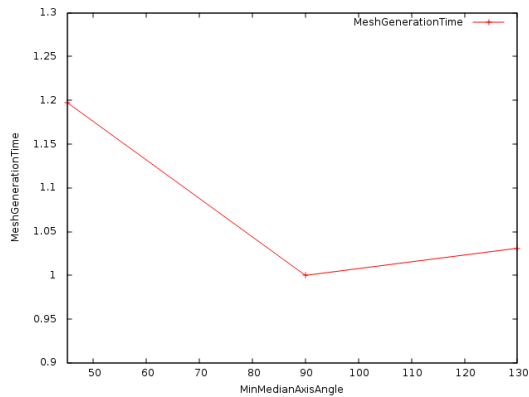


Figura 3.60: Tempo di creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
45	64.65	3.21	847934	1.20
90	62.93	3.46	845620	1.00
130	62.93	2.85	843868	1.03

Tabella 3.17: MinMedianAxisAngle

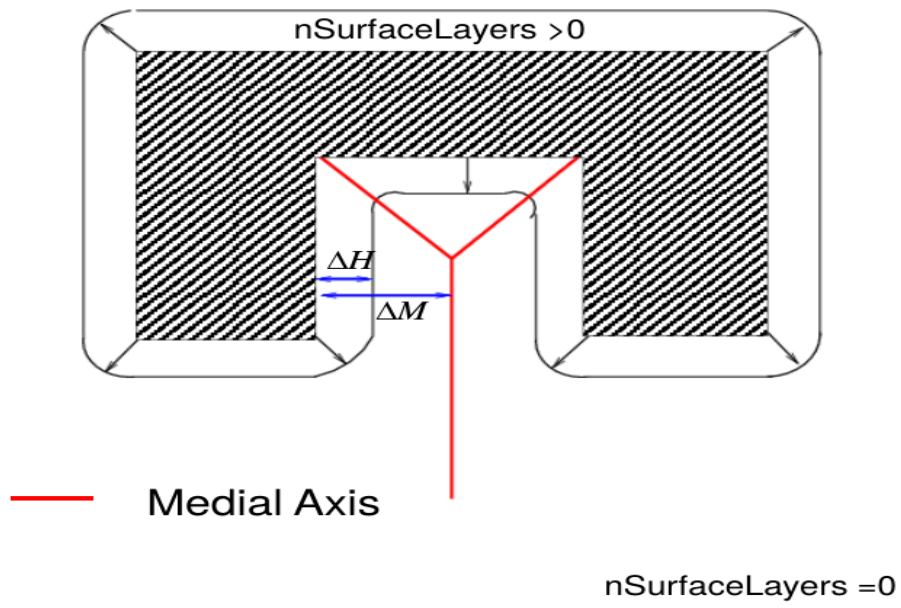
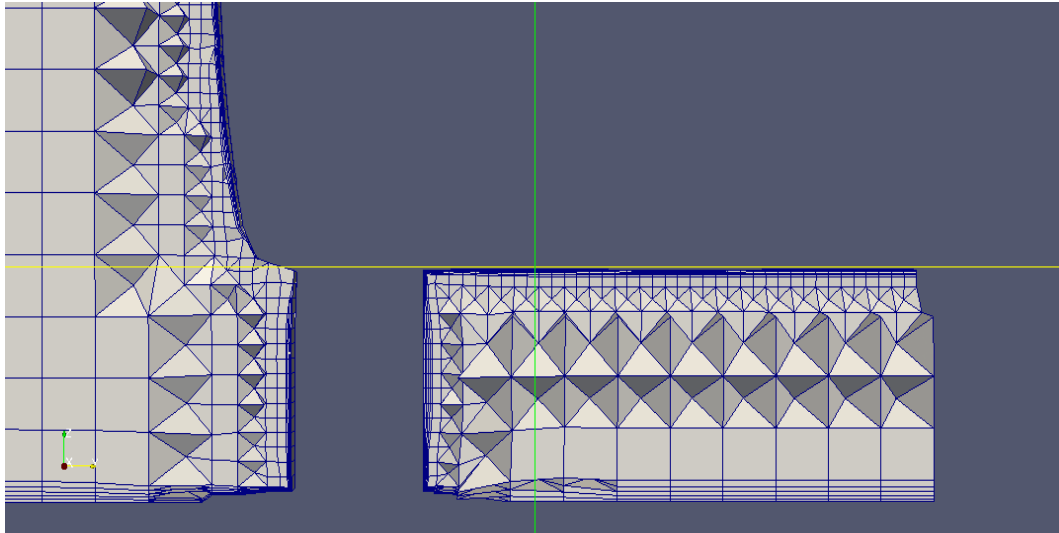
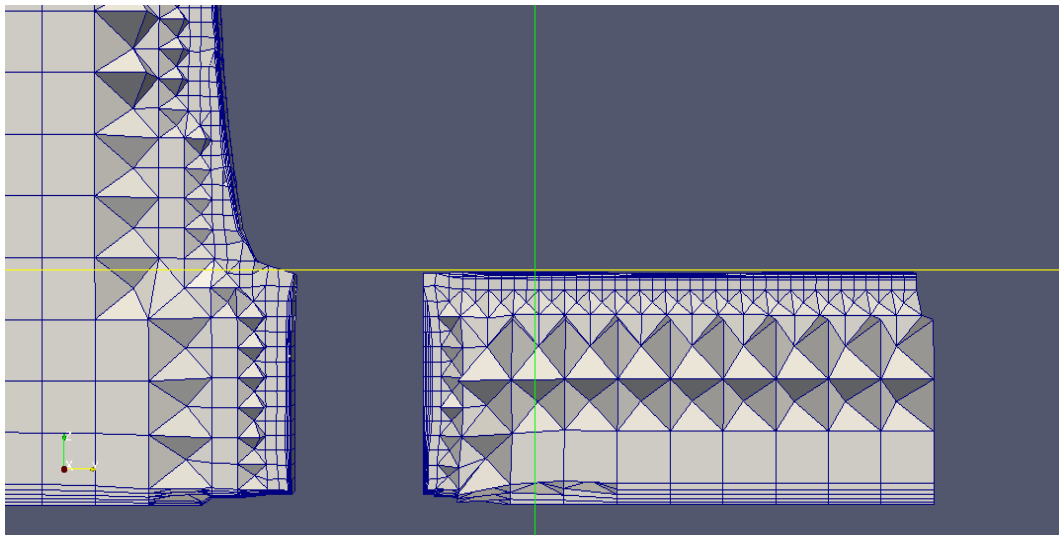


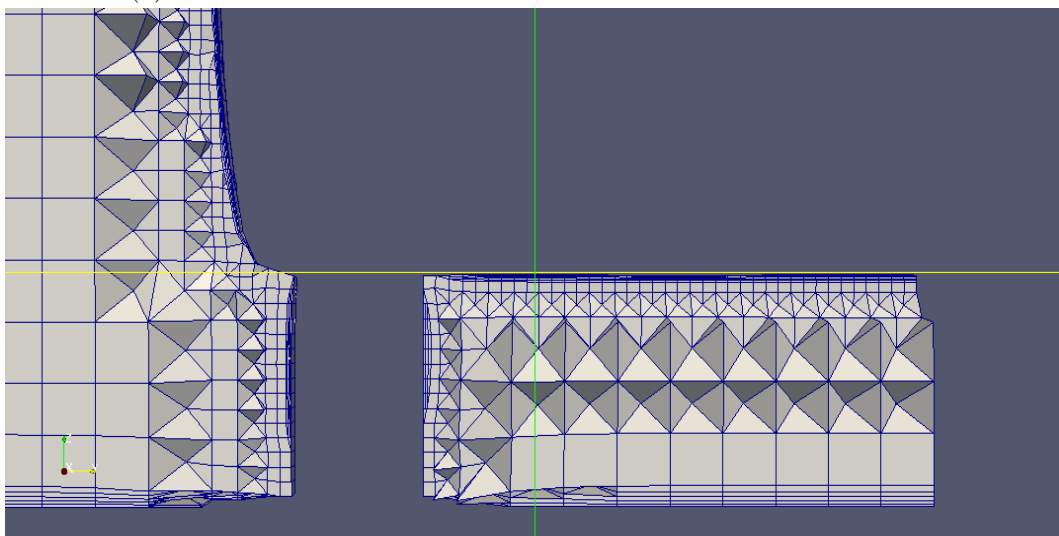
Figura 3.61: MinMedianAxisAngle [5]



(a) 45



(b) 90



(c) 130

Figura 3.62: Screen con paraview

### 3.8.2 MaxThicknessToMedialRatio

Questo setting è utilizzato per ridurre lo spessore dei layer aggiunti qualora, in accordo alla figura 3.62 a due pagine precedenti, il rapporto tra lo spessore del layer,  $\Delta H$ , e la distanza dall'asse medio,  $\Delta M$ , diventi troppo elevato. In questa sede si sceglie il valore "0.3" in quanto presenta le migliori performance complessive.

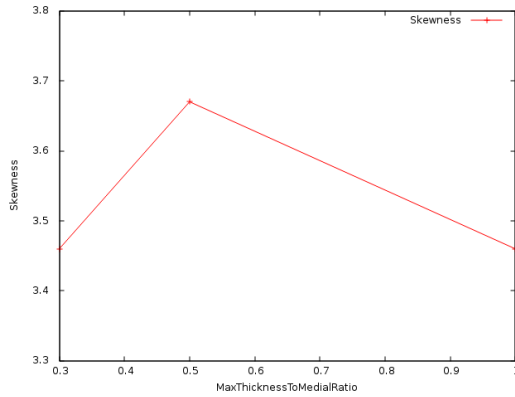


Figura 3.63: Skewness

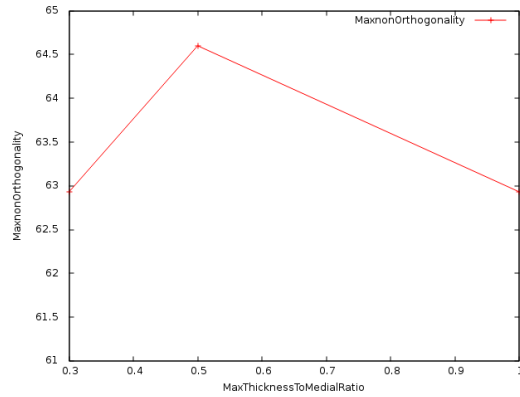


Figura 3.64: MaxnonOrthogonality

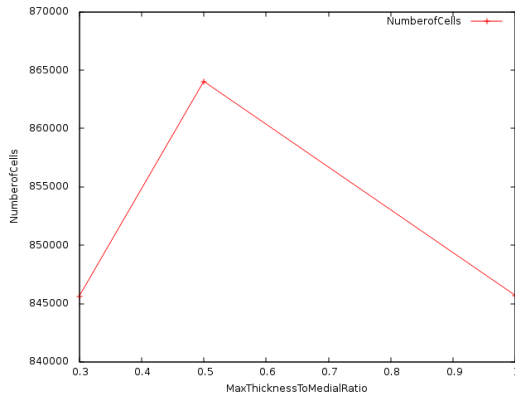


Figura 3.65: Numero di Celle

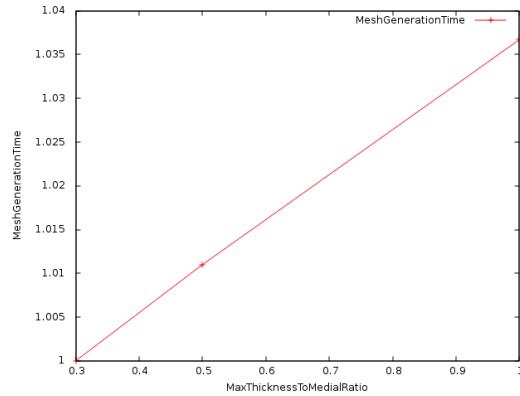
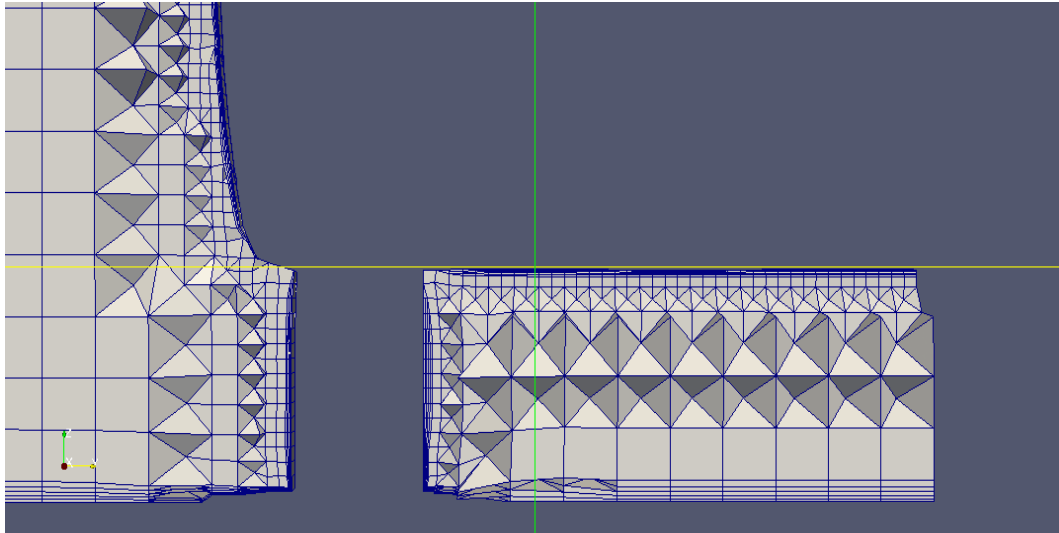


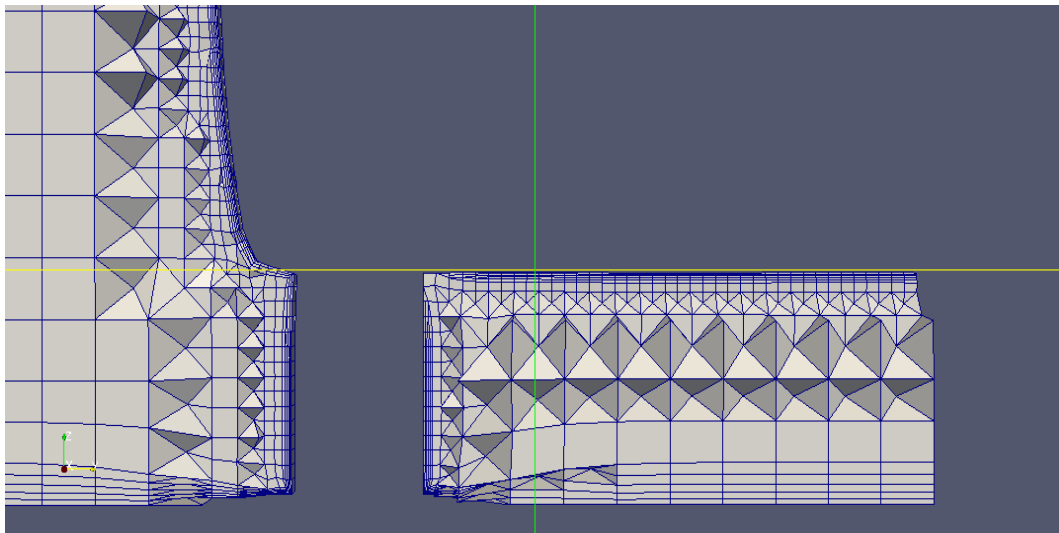
Figura 3.66: Tempo di creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0.3	62.93	3.46	845620	1.00
0.5	64.6	3.67	864039	1.01
1	62.93	3.46	845690	1.04

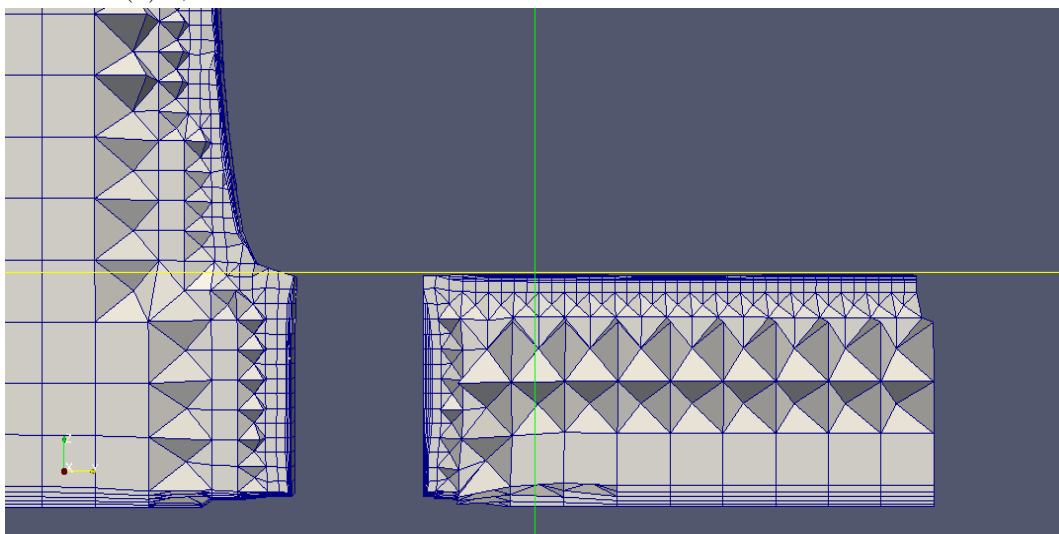
Tabella 3.18: MaxThicknessToMedialRatio



(a) 0,3



(b) 0,5



(c) 1

Figura 3.67: Screen con paraview

### 3.8.3 NSmoothNormals

Rappresenta il numero di iterazioni di smoothing sul campo di spostamento interno, ovvero prima dell'aggiunta dei layers. Vista la scarsa influenza sulle grandezze oggetto di studio, si sceglie il valore massimo di iterazioni: "5".

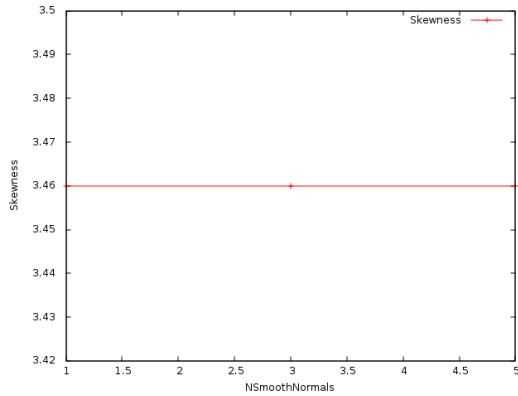


Figura 3.68: Skewness

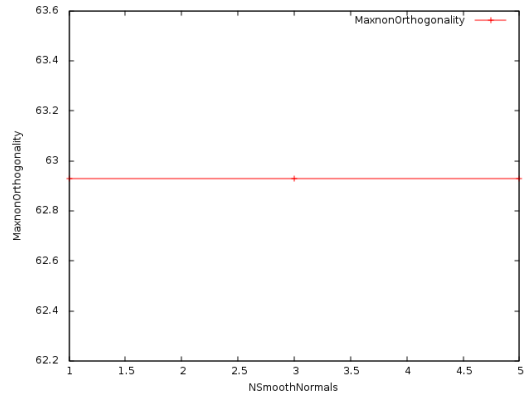


Figura 3.69: MaxnonOrthogonality

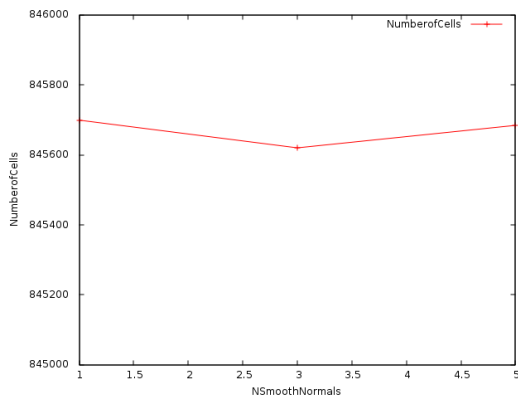


Figura 3.70: NumberofCells

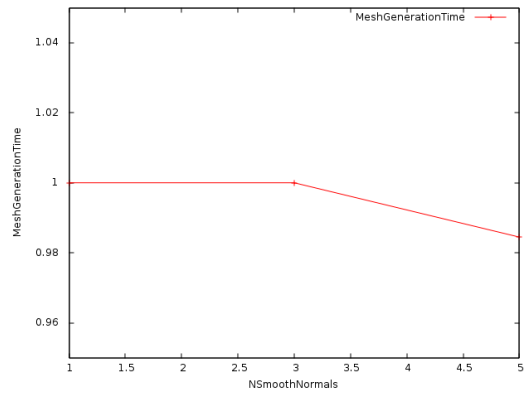
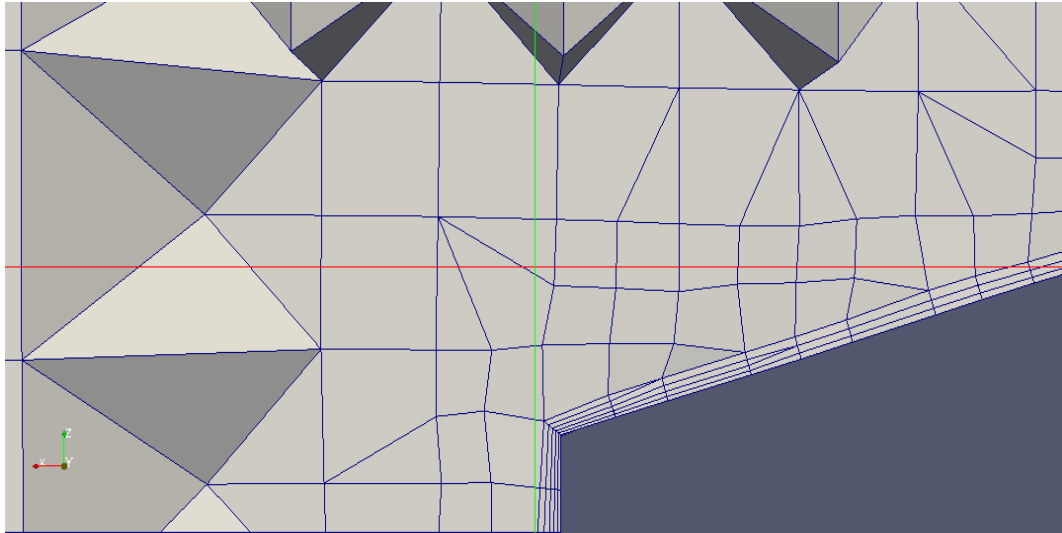


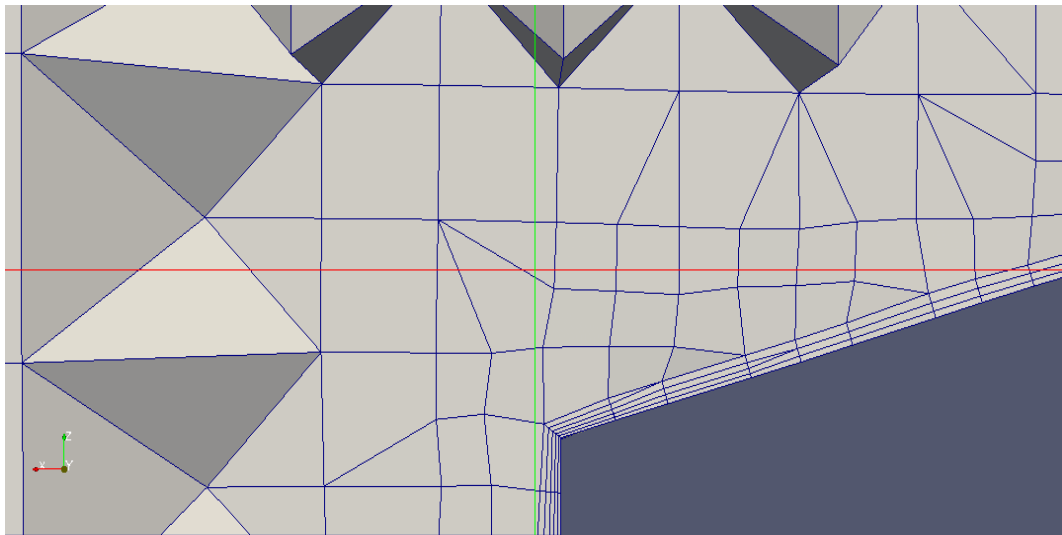
Figura 3.71: Tempo di creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
1	62.93	3.46	845699	1.00
3	62.93	3.46	845620	1.00
5	62.93	3.46	845684	0.99

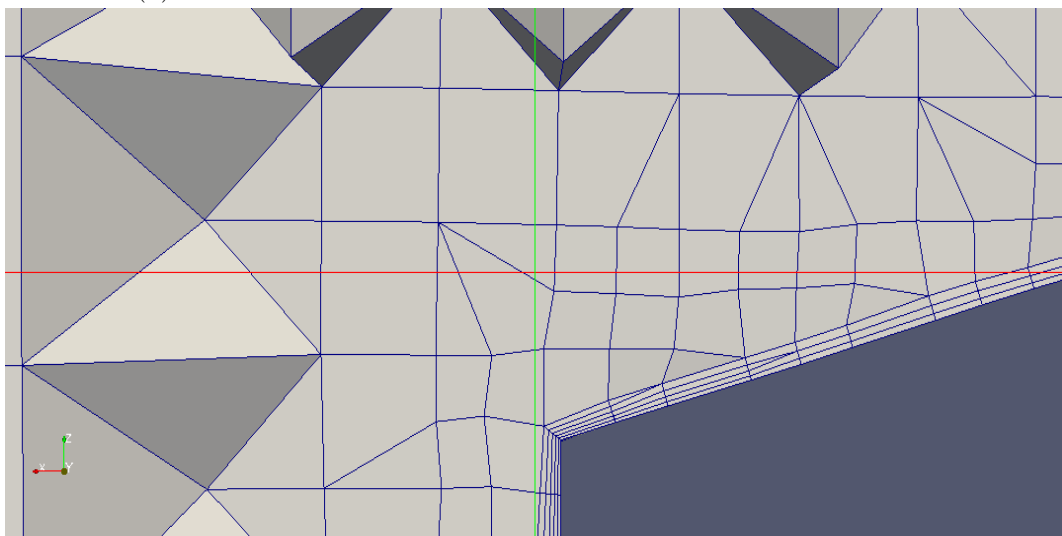
Tabella 3.19: NSmoothNormals



(a) 1



(b) 3



(c) 5

Figura 3.72: Screen con paraview



### 3.8.4 NMedialAxisIter

Durante il procedimento di inserzione dei layers i vertici delle celle vengono spostati lontano dalla superficie così da creare spazio per i layers stessi. Con questo setting è possibile regolare lo spostamento dei vertici e confinarli entro uno specificato numero di layers dalla superficie. L'immagine *c* nel gruppo di figure 3.77 mostra il caso in cui questo setting è disattivato. In *a* e *b* si può notare invece come la distorsione delle celle sia leggermente aumentata in corrispondenza della superficie; questa peculiarità assume particolare rilevanza qualora nella mesh si manifestino dislocazioni: il valore "10" ne consentirebbe la rimozione.

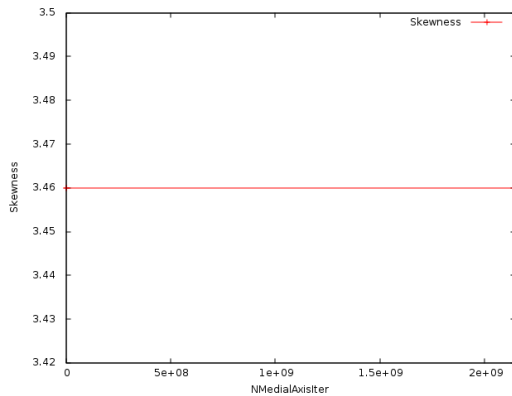


Figura 3.73: Skewness

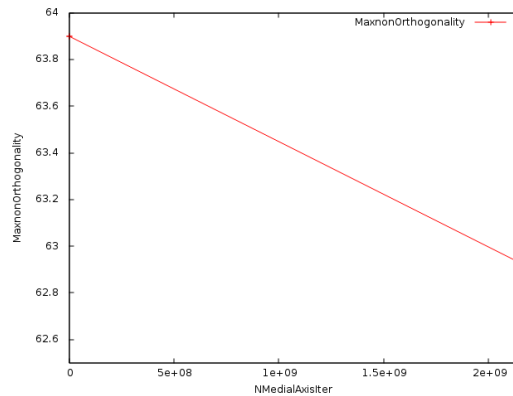


Figura 3.74: MaxnonOrthogonality

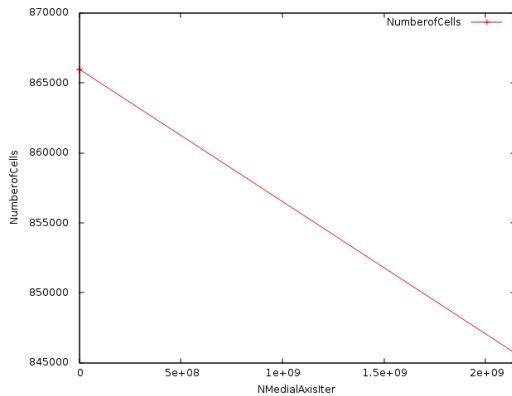


Figura 3.75: Numero di Celle

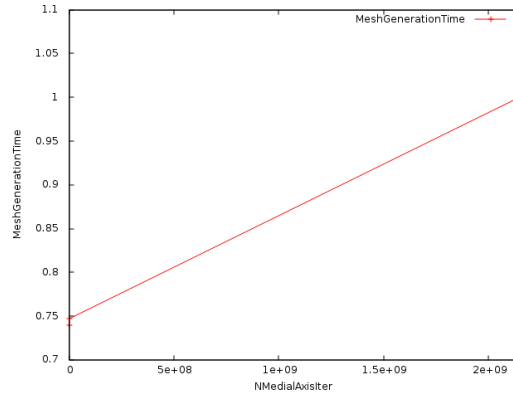
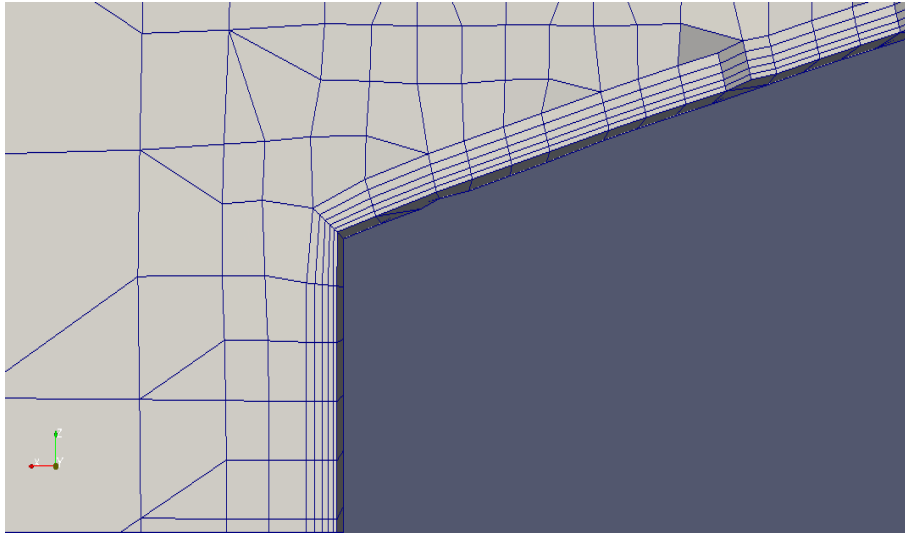


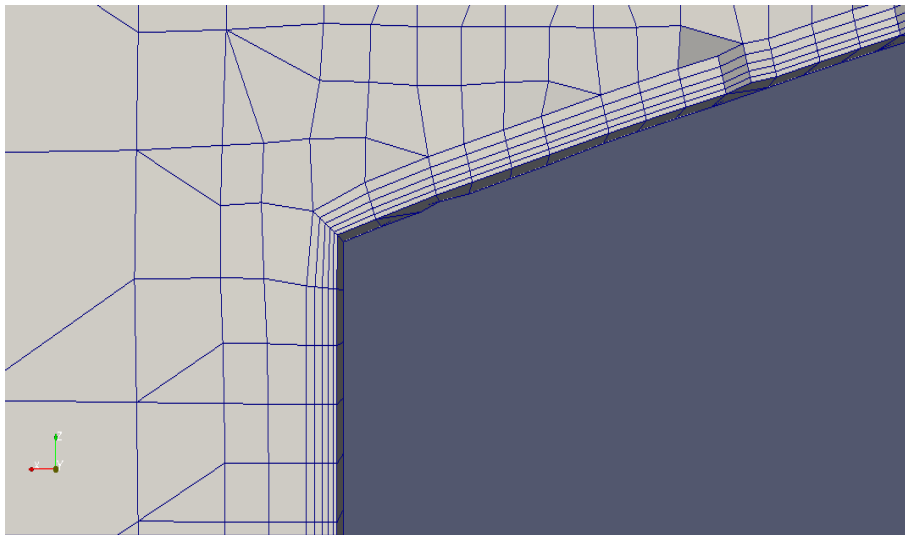
Figura 3.76: Tempo di Creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
10	63.9	3.46	865974	0.73
100	63.9	3.46	865974	0.75
2 <sup>31</sup>	62.93	3.46	845684	1.00

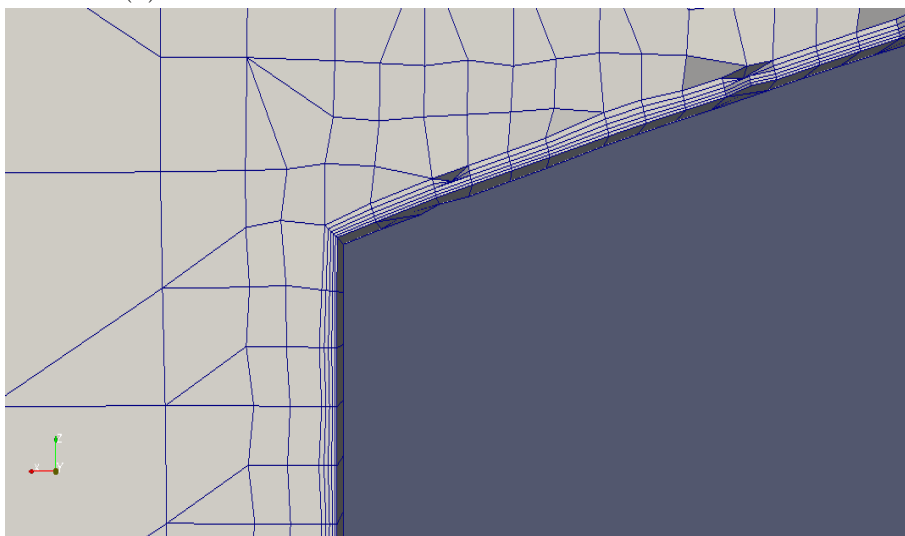
Tabella 3.20: NMedialAxisIter



(a) 10



(b) 100



(c)  $2^{31}$

Figura 3.77: Screen con paraview

### 3.8.5 NSmoothDisplacement

Parametro opzionale che gestisce lo spostamento delle regioni e che si attiva una volta avvenuta la determinazione dell'asse medio. Il valore di default, "0", risulta quello preferibile: i tempi di calcolo sono leggermente inferiori.

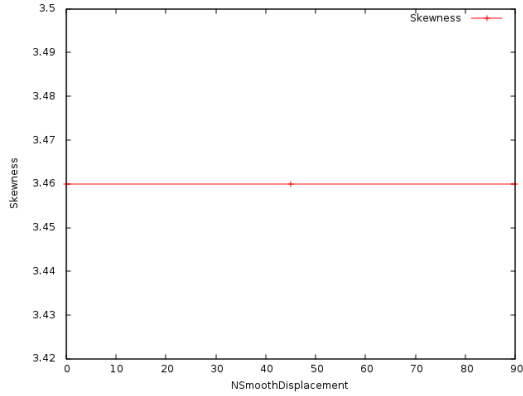


Figura 3.78: Skewness

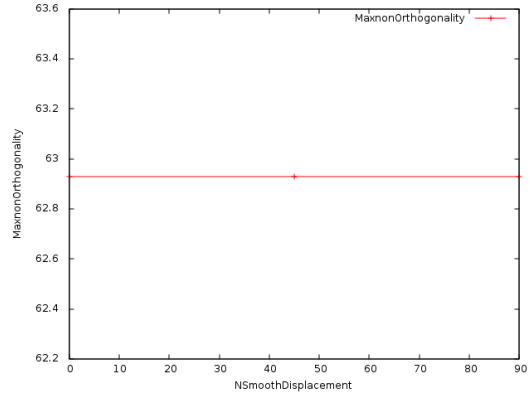


Figura 3.79: MaxnonOrthogonality

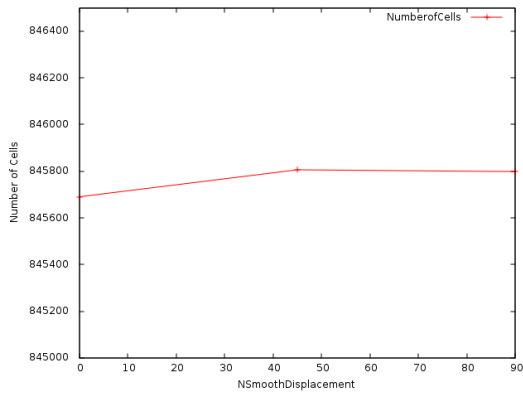


Figura 3.80: Numero di Celle

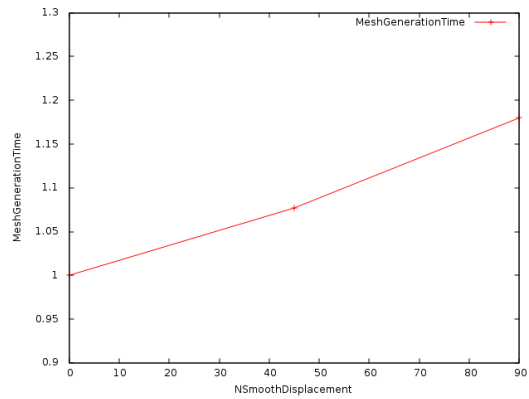


Figura 3.81: Tempo di creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0	62.93	3.46	845690	1.00
45	62.93	3.46	845806	1.08
90	62.93	3.46	845978	1.18

Tabella 3.21: NSmoothDisplacement

---

## 3.9 Mesh shrinking

### 3.9.1 SlipFeatureAngle

Questo setting è di recente introduzione nella suite di OpenFOAM®: nelle versioni precedenti, durante il processo di aggiunta dei layers, i vertici in corrispondenza dei bordi venivano sistematicamente fissati.

Infatti se l'utente avesse deciso di aggiungere un certo numero di layers ad una specifica regione, questi sarebbero stati soggetti al cosiddetto pinching, ovvero sarebbero stati indissolubilmente schiacciati al bordo esterno della regione stessa.

La nuova versione, adottata in questa tesi, consente invece ai vertici di scivolare lungo il bordo (sliding), permettendo la generazione dei layers senza che questi siano inchiodati. Ed è qui che entra in gioco lo SlipFeatureAngle: lo sliding avverrà quando l'angolo formato tra la regione in cui sono stati aggiunti i layers e quella immediatamente vicina è superiore a quello settato per questo parametro.

Di default tale parametro vale la metà del FeatureAngle, ma visto il notevole beneficio in questa fase di restringimento della mesh (shrinking) è bene adottare il valore più basso possibile: nell'analisi qui effettuata, questo risulta il valore "20" per il quale si mantengono comunque inalterate le varie grandezze, come specificato in tabella 3.22.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
20	62.93	3.46	845690	0.98
30	62.93	3.46	845620	1.00
45	62.93	3.46	845690	0.99

Tabella 3.22: SlipFeatureAngle

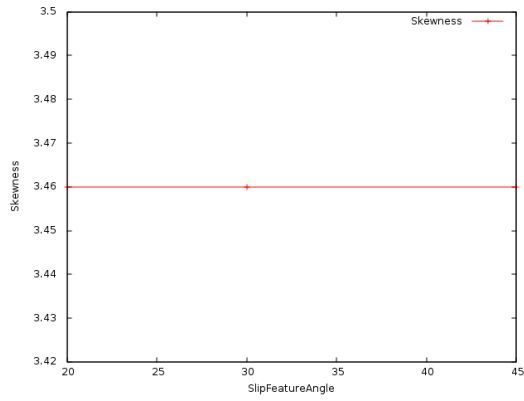


Figura 3.82: Skewness

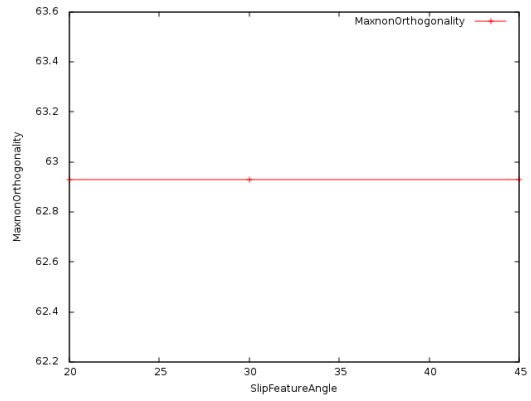


Figura 3.83: MaxnonOrthogonality

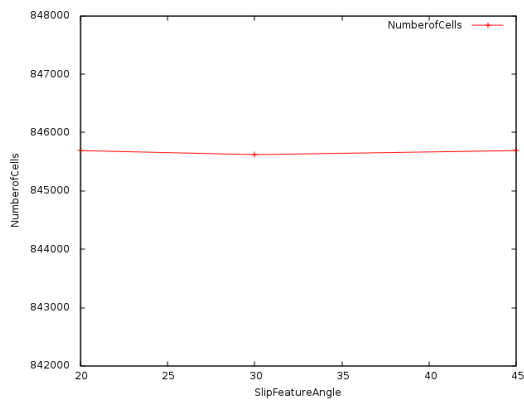


Figura 3.84: Numero di Celle

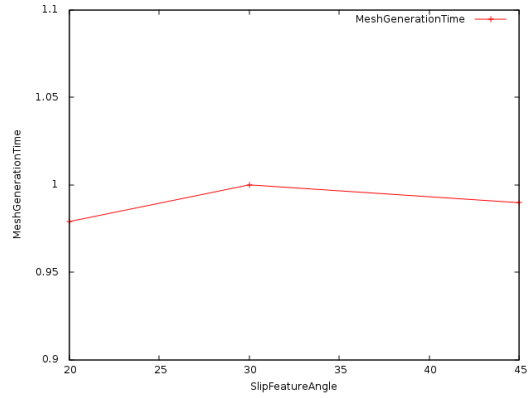


Figura 3.85: Tempo di Creazione Mesh normalizzato

### 3.9.2 NRelaxIter

NRelaxIter rappresenta il numero massimo di iterazioni di utilizzo dei relaxed parameters. Il processo di iterazione si arresta una volta ottenuta una mesh le cui qualità soddisfano le nuove specifiche apportate in tali parametri (come anticipato nella nota introduttiva, si tratta di MaxnonOrthogonality pari a 75 e MaxInternalSkewness pari a 4). Il valore migliore risulta "5".

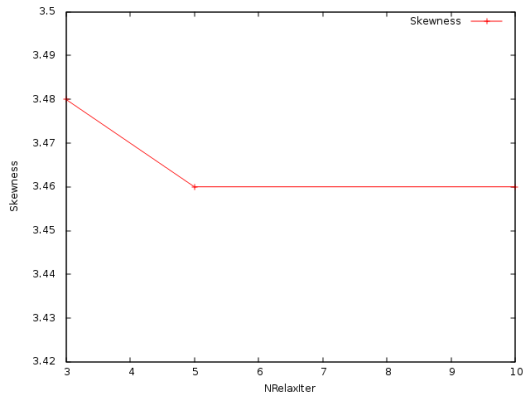


Figura 3.86: Skewness

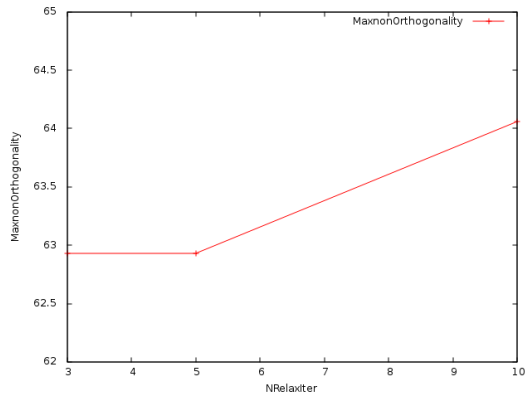


Figura 3.87: MaxnonOrthogonality

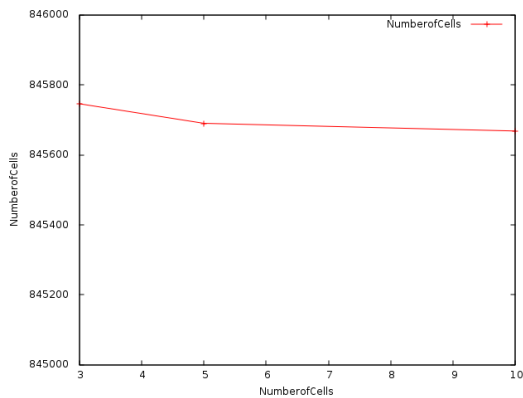


Figura 3.88: Numero di Celle

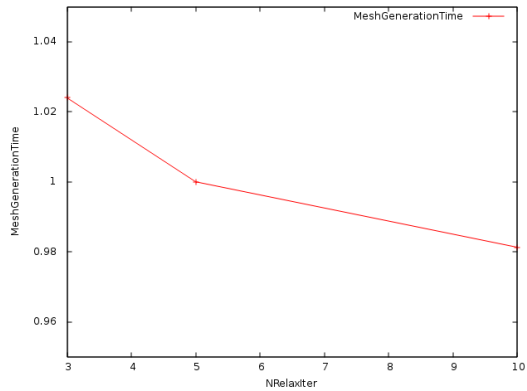


Figura 3.89: Tempo di Creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
3	62.93	3.48	845746	1.02
5	62.93	3.46	845690	1.00
10	64.06	3.46	845668	0.98

Tabella 3.23: NRelaxIter

### 3.9.3 NBufferCellsNoExtrude

Questo parametro crea buffer regions. Queste sono regioni cuscinetto che separano la regione interna (inner viscous layer) ed esterna (boundary layer) dello strato limite. Tali regioni si ottengono rimuovendo le celle in corrispondenza dei bordi e ciò si effettua al fine di inserire i buffer layers. Come si evince dal post processing, il parametro di default, "0", risulta la scelta migliore.

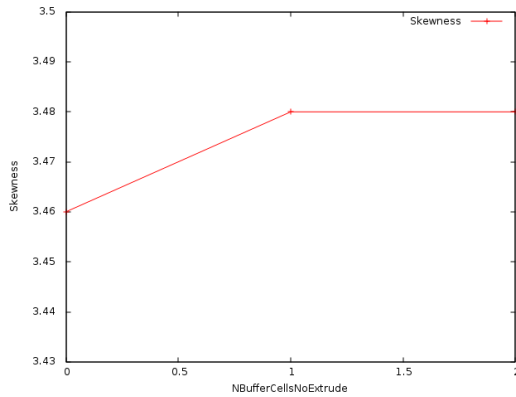


Figura 3.90: Skewness

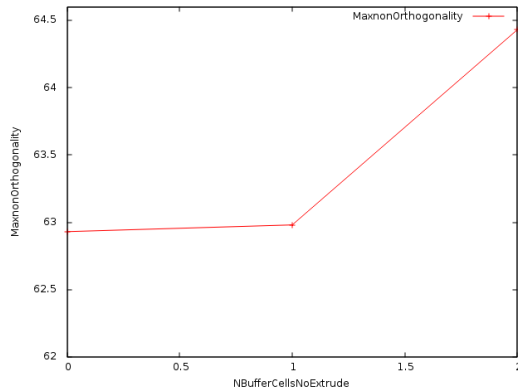


Figura 3.91: MaxnonOrthogonality

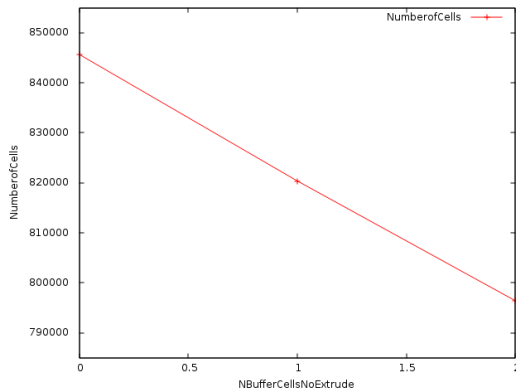


Figura 3.92: Numero di Celle

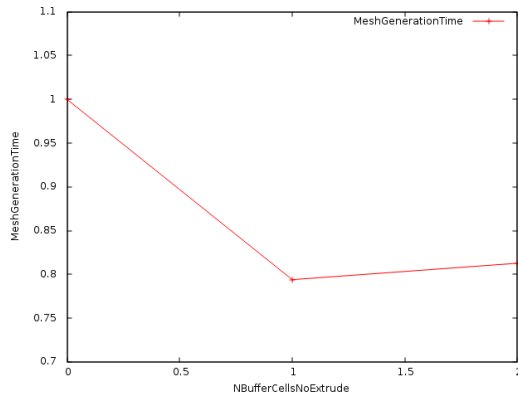
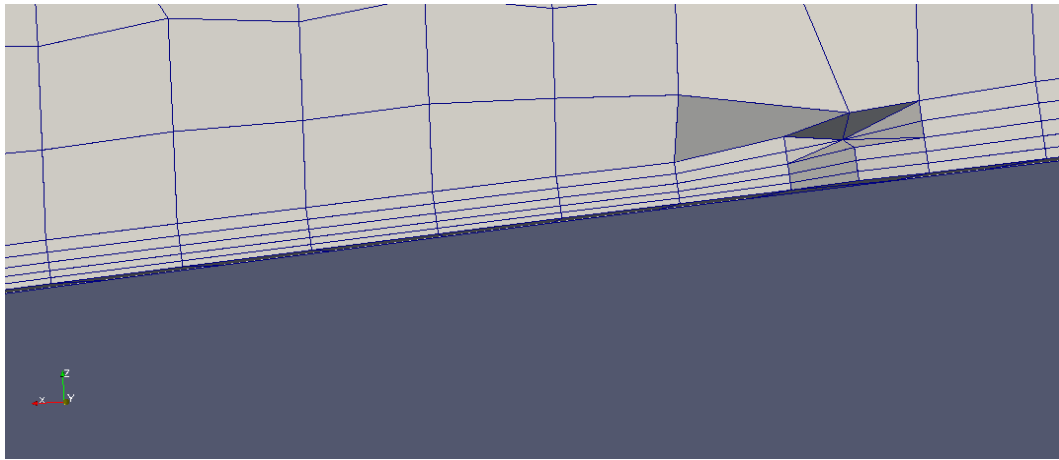


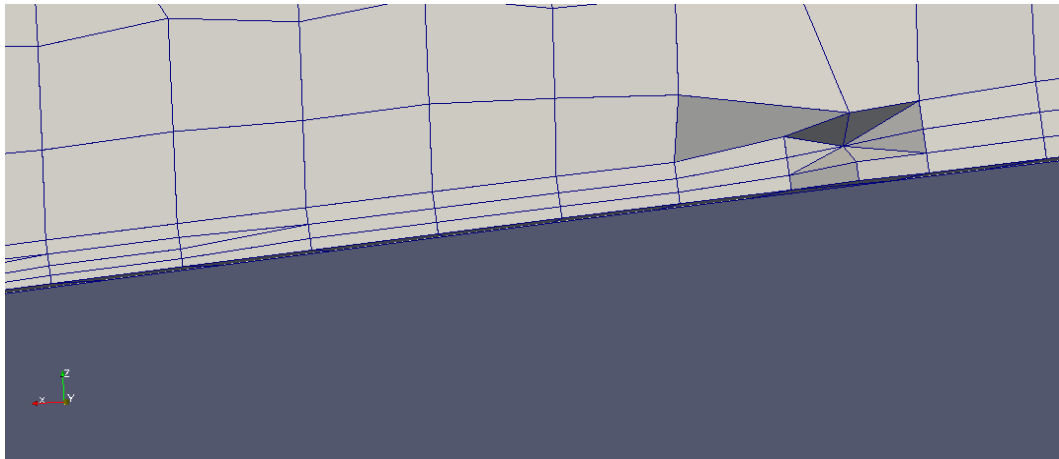
Figura 3.93: Tempo di Creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0	62.93	3.46	845690	1.00
1	62.98	3.48	820374	0.79
2	64.43	3.48	796426	0.81

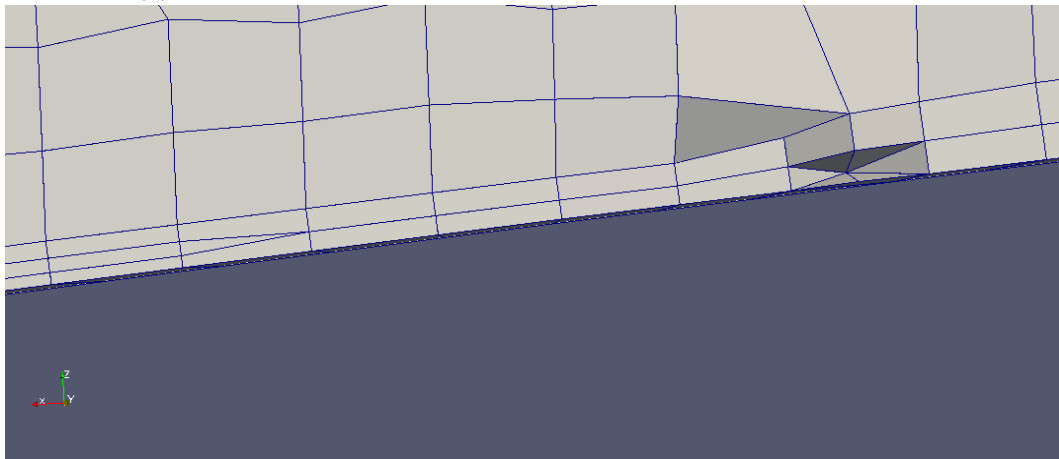
Tabella 3.24: NBufferCellsNoExtrude



(a) 0



(b) 1



(c) 2

Figura 3.94: Post processing sul bordo



### 3.9.4 NLayerIter

Numero totale di iterazioni che portano all'aggiunta dei layers. Nel caso in cui snappyHexMesh sia chiamato ad effettuare un numero di iterazioni superiore si avrà un crash repentino della simulazione. Per questo settaggio non si hanno particolari modifiche della qualità: il valore preferibile è quindi "50" per il quale si ha il costo minore in termini di tempo di calcolo.

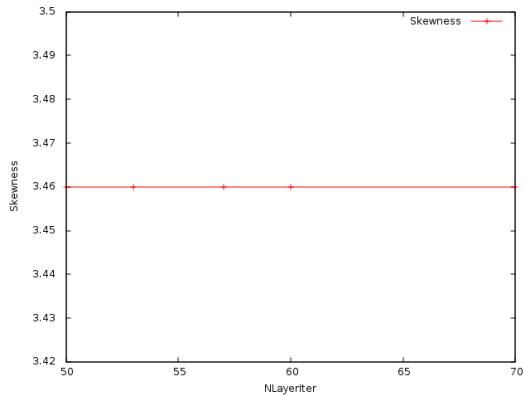


Figura 3.95: Skewness

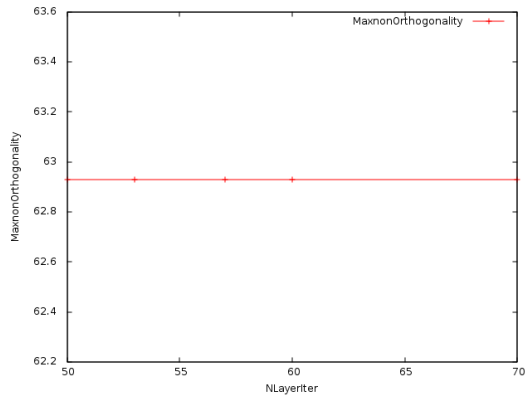


Figura 3.96: MaxnonOrthogonality

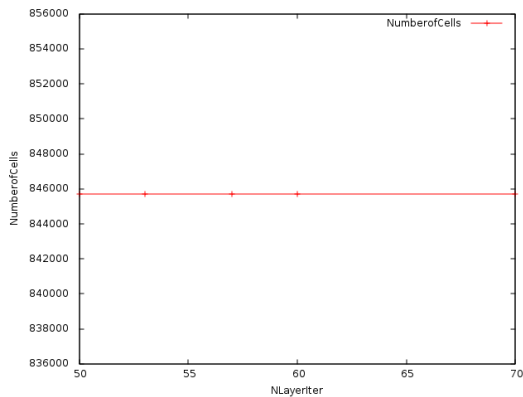


Figura 3.97: Numero di Celle

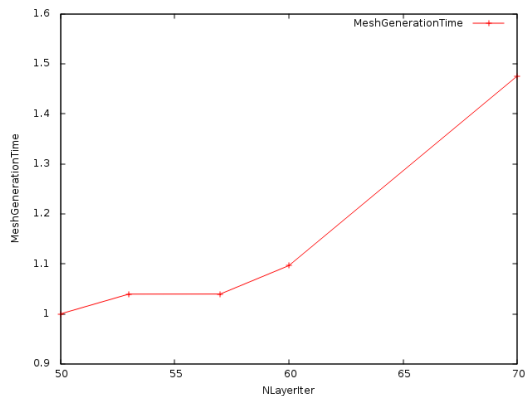


Figura 3.98: Tempo di Creazione Mesh normalizzato

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
50	62.93	3.46	845690	1.00
53	62.93	3.46	845690	1.03
57	62.93	3.46	845690	1.03
60	62.93	3.46	845690	1.10
70	62.93	3.46	845690	1.48

Tabella 3.25: NLayerIter

---

### 3.9.5 NRelaxedIter

Numero massimo di iterazioni con i parametri di qualità standard, dopo le quali vengono usate i relaxed parameters. Ovvero, per valori di iterazioni inferiori, verranno applicati i settaggi contenuti nel file mesh-QualityControls; per valori superiori subentreranno i relaxed parameters: in questo caso il numero di iterazioni sarà gestito, come esplicito poc'anzi, dal parametro NRelaxIter. Considerato che, in questa trattazione, le iterazioni con i parametri standard si sono sempre mantenute inferiori al valore massimo settato, e che non si riscontrano modifiche circa il tempo di generazione, si può adottare il valore "10": questo risulta già piuttosto conservativo, difatti se dopo 10 iterazioni snappyHexMesh non riesce ancora a produrre una mesh con le specifiche di default, è meglio affidarsi ai relaxed parameters.

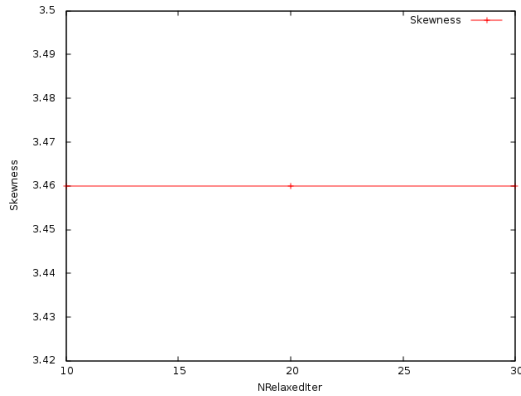


Figura 3.99: Skewness

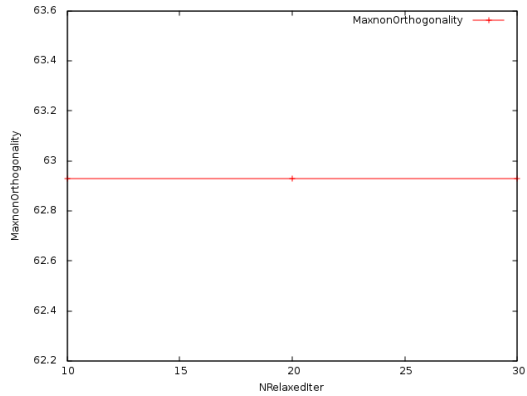


Figura 3.100: MaxnonOrthogonality

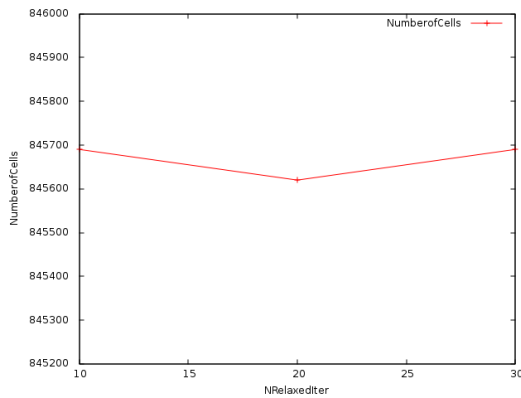


Figura 3.101: Numero di Celle

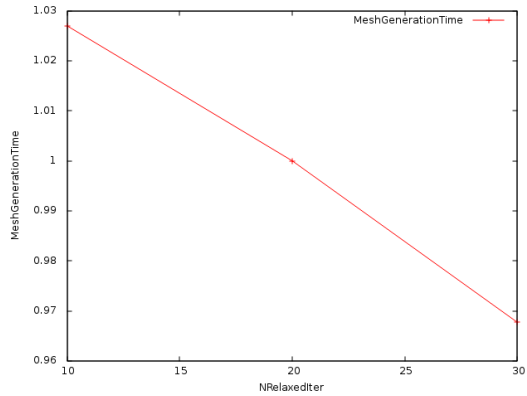


Figura 3.102: Tempo di Creazione Mesh normalizzato

---

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
10	62.93	3.46	845690	1.03
20	62.93	3.46	845620	1.00
30	62.93	3.46	845690	0.97

Tabella 3.26: NRelaxedIter

## 3.10 Mesh Quality Controls

### 3.10.1 MaxnonOrthogonality

In questa sezione si è cercato di fare leva sulla qualità della mesh imponendovi condizioni più o meno stringenti. Si noti come, snappyHexMesh, per generare una mesh di migliore caratura richieda ben il doppio del tempo (valore "50"); settando a "60" invece si riesce ad avere un'ottima non-ortogonalità, ottenendo un migliore responso anche in termini di tempistica.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
50	49.99	3.46	819785	2.12
60	59.91	3.46	844846	1.16
65	62.93	3.46	845690	1.00
70	62.93	3.46	845823	1.08

Tabella 3.27: MaxNonOrthogonality

### 3.10.2 MaxInternalSkewness

Vale anche qui il discorso espresso in precedenza: si agisce sulla qualità, imponendo a SnappyHexMesh di soddisfare i però requisiti sulla skewness. Interessante osservare come, per il valore limite fissato 3, venga a generarsi una mesh con un valore di 2.85 senza un consistente aggravio del tempo di creazione.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
3	62.93	2.85	844748	1.11
4	62.93	3.46	845690	1.00
5	62.93	3.46	845760	1.00

Tabella 3.28: MaxInternalSkewness

### 3.10.3 MaxBoundarySkewness

Questo setting è del tutto simile a quello precedente: è solo una nuova definizione di skewness per le celle localizzate ai confini del dominio.

---

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
10	62.93	3.46	845690	1.10
20	62.93	3.46	845690	1.00

Tabella 3.29: MaxBoundarySkewness

### 3.10.4 MinDeterminant

Parametro di controllo, ottenuto effettuando il determinante del tensore calcolato dai vettori normali alla superficie della faccia.

$$A = \sum_{faces} |\vec{A}_i|$$

$$t_{i,j} = \sum_{faces} \vec{A}_i * \frac{\vec{A}_i}{|\vec{A}_i|}$$

$$mindeterminant = \frac{|t_{i,j}|}{A} / 0.03703$$

dove  $\vec{A}_i$  è il vettore normale alla singola faccia. Il valore 1 corrisponde ad una cella perfettamente esaedrica.

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
0.001	62.93	3.46	845690	1.00
0.01	64.49	3.46	844965	0.63

Tabella 3.30: MinDeterminant

### 3.10.5 MinTriangleTwist

Questa grandezza è calcolata decomponendo la faccia in parti triangolari; dal centro di queste nuove facce triangolari si tracciano i vettori normali: il TriangleTwist è definito come prodotto scalare normalizzato tra i vettori di due facce consecutive. Il valore 0 corrisponde al caso in cui le facce prese in considerazione siano tra loro ortogonali; il valore -1 corrisponde al caso in cui tali normali abbiano verso opposto ovvero per cui si abbia il collasso della faccia. Essendo questo parametro una misura preventiva nei confronti dei controlli circa la plurimenzionata qualità, è bene settarlo a -1 in maniera tale da permettere ogni conformazione possibile.

$$triangletwist = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$$

---

	MaxnonOrthogonality	Max Skewness	Number of cells	Generation time
-1	62.93	3.46	845620	1.00
0	64.06	3.46	845684	0.67

Tabella 3.31: MinTriangleTwist

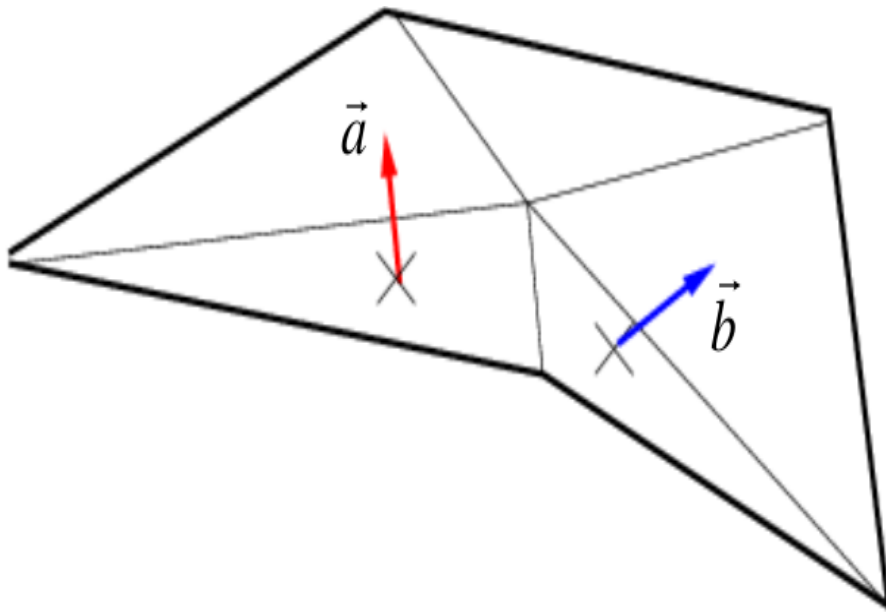


Figura 3.103: MinTriangleTwist [5]



# Capitolo 4

## Conclusioni

Si riportano, nelle due tabelle che seguono, i valori selezionati al termine dell'analisi.

Parametro	Valore selezionato
MinRefinementCells	100
NcellsBetweenLevels	1
MaxLoadUnbalance	0.5
AllowFreeStandingZoneFaces	false
ResolveFeatureAngle	90
NSmoothPatch	3
NSolveIter	30
NRelaxIter	2
NFeatureSnapIter	20
ImplicitFeatureSnap	false
ExplicitFeatureSnap	true
NGrow	0
FeatureAngle	300
MaxFaceThicknessRatio	1

Tabella 4.1: 1

---

Parametro	Valore selezionato
Nsmoothsurfacenormals	5
NSmoothThickness	20
MinMedianAxisAngle	130
MaxThicknessToMedialRatio	0.3
NSmoothNormals	5
NMedialAxisIter	10
NSmoothDisplacement	0
SlipFeatureAngle	20
NRelaxIter	5
NBufferCellsNoExtrude	0
NLayerIter	50
NRelaxedIter	10
MinDeterminant	0.001
MinTriangleTwist	-1.0

Tabella 4.2: 2

I parametri che hanno apportato le più significative migliorie in termini di skewness sono stati:

- *Feature Angle*, per un valore pari a 130 (valore alternativo);
- *NSmoothSurfaceNormals* per un valore pari a 5;
- *MinMedianAxisAngle* per un valore pari a 130.

In questi tre casi si è riusciti ad avere un valore di skewness pari a 2.85 quando invece con le impostazioni di default si otteneva 3.46.

Queste risultano quindi le soluzioni applicative ideali al fine di generare una mesh di migliore qualità.

Il numero di celle si è mantenuto quasi sempre costante (attorno alle 845000) salvo per alcune eccezioni:

- *NCellsBetweenLevels*: qui sono stati aggiunti i buffer layers per cui si è passato da circa 845000 (1) a 900000 (2) per poi arrivare a 950000 (3);
- *NGrow*: per il valore 1, non essendo stati aggiunti i layers si ha avuto una drastica riduzione fino a 565000 celle;
- *NSmoothThickness*: per il valore 1 si è riscontrato uno scarso livello di refinement in corrispondenza dei layers e si è quindi ottenuto un numero inferiore di celle, pari a 770000;
- *NMedialAxisIter*: per 10 e 100 si sono rilevati, in corrispondenza della superficie, una distorsione delle celle nonché un aumento del loro numero, per cui si sono ottenute circa 20000 celle in più rispetto al settaggio di default;



- 
- *NBufferCellsNoextrude*: Per via della rimozione delle celle in corrispondenza dei bordi di superficie al fine di introdurre i buffer layers, si sono rilevate 820000 celle per il valore 1 e 796000 celle per il valore 2.

Le diminuzioni nel numero di celle riscontrate, sono quindi imputabili all'impossibilità di inserire i layers per il valore settato: ciò non si manifesta sulla qualità intrinseca della mesh, che si mantiene inalterata, ma chiaramente questo fenomeno è sintomatico di una mesh non propriamente corretta.

I parametri più cruciali sono stati quindi quelli relativi al tempo di generazione. Questo è risultato essere di circa 590 secondi per una simulazione in seriale con le regolazioni standard. I parametri che hanno prodotto le più significative diminuzioni sono stati:

- *ResolveFeatureAngle*: per "90" e "300" si è avuta una riduzione, rispettivamente, del 15% e del 16% , in quanto aumentando questo valore sono risultati inferiori le celle la cui intersezione individuava un angolo superiore a quello impostato, da cui è disceso il minor tempo necessario per eseguire il loop di refinement in tali zone;
- *NSmoothPatch*: questo parametro è di natura sequenziale; al suo crescere è corrisposto un maggior numero di iterazioni per raffinare le regioni, pertanto il valore "1", il più basso tra quelli del range studiato, ha fornito una riduzione pari al 7%;
- *NRelaxIter*: per il valore "2" si è ottenuta una riduzione del 9% in quanto si è diminuito il numero di cicli effettuati con i relaxed parameters;
- *NFeatureSnapIter*: per valori "0" e "5" si sono ottenute riduzioni del 42% e del 13%; questo parametro è riferito però al numero di cicli di refinement in corrispondenza delle features ed è necessario che sia sufficientemente alto affinché la mesh abbia una buona qualità;
- *NSmoothThickness* per il valore "1" si è ottenuta una riduzione del 36% ; anche qui vale lo stesso discorso del parametro precedente: in questo caso il refinement è attuato in corrispondenza dei layers.
- *MinDeterminant* per "0.01" si è ottenuta una riduzione del 37%.

---

## 4.1 Sviluppi futuri

- L’analisi effettuata fornisce solo una delle molteplici possibilità offerte da questo tool. Infatti è stato variato un solo parametro alla volta al fine di individuarne il valore più performante. Per ottenere un effetto più accurato si potrebbero variare più parametri in contemporanea, così da ricercare una combinazione migliore.
- Un’ulteriore maniera per valutare le prestazioni di snappyHexMesh potrebbe essere quella di raffrontarlo con altri generatori di Mesh OpenSource forniti in OpenFOAM<sup>®</sup> quali:
  - \* cfMesh [15]
  - \* enGrid [16]
  - \* Netgen [17]
  - \* Salome [18]
  - \* GMSH [19]
  - \* MeshKIT [20]
- Infine, il Parallel Computing, che è un metodo di computazione che fa leva sul principio ”analitico” nel senso greco del termine, ovvero di decomposizione: i problemi vengono suddivisi in parti gestibili, per poi essere risolti simultaneamente da più processori.

# Bibliografia

- [1] A. Quarteroni, F.Saleri, P. Gervasio, *Calcolo Scientifico*, Springer, 2012.
- [2] S.Battaini, *Sensitività aerodinamica nelle autovetture da competizione:uno studio basato su codice open source ed operatore aggiunto*, 2013.
- [3] S.Corti, *Simulazione numerica nel flusso delle cavità nasali*, 2011.
- [4] J.Guerrero, *Introduction to computational fluid dynamics: governing equations, turbulence modeling introduction and finite volume discretization basics*, 2014.
- [5] *A comprehensive tour of SnappyHexMesh*, Engys ltd., 2012.
- [6] <https://www.openfoamwiki.net/index.php/SnappyHexMesh>.
- [7] *OpenFoam user guide*, OpenFoam foundation, 2014.
- [8] <http://www.openfoam.org>
- [9] J.Aguilar, *Fondamenti di programmazione in C++*, McGraw-Hill, 2008.
- [10] Karalit, *CFD validation: Asmo Model*, 2012.
- [11] <http://www.cfd-online.com>
- [12] *Mesh generation in OpenFoam with SnappyHexMesh:an user experience*, ATS4I, 2010.
- [13] <http://www.foamcfd.org>
- [14] Roberto Pieri, *A tool for pre-processing:SnappyHexMesh*, 2014
- [15] <http://www.c-fields.com/>
- [16] <http://www.engits.eu/en/engrid>
- [17] <http://www.hpfem.jku.at/netgen/>
- [18] <http://www.salome-platform.org/>
- [19] <http://www.geuz.org/gmsh/>
- [20] <http://www.press3.mcs.anl.gov/sigma/meshkit-library/>



# Ringraziamenti

Desidero ringraziare infinitamente il Prof. Bottaro per la possibilità di lavorare a questo tipo di progetto, ed inoltre per l' aiuto e la disponibilità che mi ha fornito nell'arco di questi mesi.

Un ringraziamento particolare anche per l'Ing.Matteo Bargiacchi, per la sua cortesia e la costante gentilezza nell'assistermi durante lo sviluppo della tesi.

Ringrazio infine i miei genitori, per l'incondizionato supporto che mi hanno sempre dato.