

Politecnico di Torino

Corso di Laurea Magistrale in *Ingegneria Aerospaziale*



Tesi di laurea magistrale

Simulazione numerica diretta del flusso turbolento in un canale piano con riblet

Relatore:
Prof.re Michele Iovieno

Candidato:
Salvatore Sabbatini
mat: s230951

Sessione di Laurea: Aprile 2020

*A tutti quelli che hanno
sempre creduto in me
e mi hanno supportato
nel raggiungimento di
questo grande traguardo*

Indice

Sommario	3
1 Introduzione	4
2 Fenomenologia del Problema	8
2.1 Fluido Incomprimibile	8
2.1.1 Equazioni di Navier-Stokes per fluidi incomprimibili	9
2.1.2 Soluzioni delle equazioni per un flusso viscoso newtoniano	11
2.1.3 Soluzioni approssimate	12
2.2 Strato limite	14
2.2.1 Grandezze integrali dello strato limite	16
2.2.2 Cenni sullo strato limite su lamina piana e soluzione di Blasius	19
2.2.3 Caratterizzazione dello strato limite turbolento	20
2.2.4 Struttura dello strato limite	22
2.3 Turbolenza e flusso turbolento	27
2.3.1 Natura della turbolenza	27
2.3.2 Flusso turbolento	28
2.3.3 Instabilità e transizione	29
2.3.4 Moto turbolento	30
2.3.5 La cascata energetica e le ipotesi di Kolmogorov	32
2.3.6 Descrizione statistica della turbolenza	35
2.3.7 Decomposizione di Reynolds	37
2.3.8 Canale piano turbolento	39
2.3.9 Metodo di Clauser	47
3 Metodi riduzione resistenza	50
3.1 Storia riduzione resistenza	50
3.2 Tecniche riduzione resistenza	55
3.3 Tecniche attive	61
3.4 Tecniche passive	81
3.4.1 Riblet	86
4 Simulazione numerica (riblet)	109
4.1 Svolgimento dello studio	109
4.1.1 Inquadramento del problema	110
4.1.2 Resistenza di un Flusso Turbolento	112
4.1.3 Metodi di calcolo e discretizzazione della turbolenza utilizzati	113
4.2 DNS	118

4.2.1	Creazione della mesh	118
4.2.2	Svolgimento simulazioni (run simulazioni)	129
4.3	Codice di postprocessing	137
4.3.1	Riordino dei dati di postprocessing	138
4.3.2	Realizzazione Immagini	139
5	Dati e Risultati	141
5.1	Risultati ottenuti	141
5.2	Scelta intervallo temporale	143
5.3	Validazione dei risultati: canale liscio	151
5.4	Visualizzazioni del Flusso	153
5.5	Analisi statistica	161
5.5.1	Velocità media	161
5.5.2	Varianza della velocità	165
5.5.3	Sforzi	168
5.6	Resistenza	177
6	Conclusioni finali	180
	Appendici	183
A	Codici simulazioni numeriche	184
A.1	blockMeshDict	185
A.2	controlDictRun	219
A.3	Altri codici della cartella system	220
A.3.1	decomposeParDict	220
A.3.2	fvSchemes	221
A.3.3	fvSolution	222
A.3.4	mapFieldsDict	223
A.4	Codici della cartella constant	223
A.4.1	transportProperties	224
A.4.2	turbulenceProperties	224
A.4.3	fvOptions	224
A.5	Codici cond. cont. e iniz.	225
A.5.1	U e P	225
A.6	Altri codici	228
A.6.1	Newton	229
A.6.2	Submit Run	231
B	Codici post processing	232
B.1	controlDictReynolds	232
B.2	Riordina Reynolds	316
B.2.1	Input codice Riordina	319
B.2.2	Coordinate Y	319
B.2.3	Interpolazione Gnuplot	321
B.3	Submit Post processing	323
	Bibliografia	324

Sommar

Lo scopo di questo lavoro di tesi è stato quello di andare ad analizzare una tecnica passiva di riduzione della *resistenza d'attrito* di un *flusso in un canale turbolento*, mediante una *simulazione numerica diretta*. La tecnica che è stata analizzata è quella dell'utilizzo delle *riblet*, inserite in un *canale piano*.

La tecnica di utilizzo delle *riblet* consiste nel realizzare delle scanalature, molto piccole rispetto alla *dimensione caratteristica* del problema sul quale vengono applicate, che servono a modificare la struttura del flusso vicino alla parete dove sono inserite, al fine di ridurre la *resistenza d'attrito*. In particolare, in questa tesi, mediante l'utilizzo del programma *OpenFOAM*, sono state effettuate diverse *simulazioni fluidodinamiche*, atte a valutare lo *sforzo di attrito medio di un flusso turbolento che scorre in un canale piano* in presenza di *riblet triangolari e parallele alla direzione del flusso medio*. Tali simulazioni sono state ripetute per diverse altezze delle *riblet*, in modo da valutare l'effetto di questo parametro di forma sulla *resistenza d'attrito* del flusso, confrontandosi con lo stato dell'arte, dedotto dagli studi presenti in letteratura sui flussi con *riblet* di diversa forma e dimensione.

Sono stati confrontati gli *sforzi medi a parete* ottenuti dalle simulazioni a parità di *numero di Reynolds* globale, al fine d'individuare la configurazione di *riblet* che li minimizzavano. In questo modo, è stato possibile trovare quali valori dei parametri di forma delle *riblet*, tra i casi analizzati, riduce maggiormente la *resistenza d'attrito*.

Si è svolta inoltre anche un'analisi dei dati per individuare e caratterizzare meglio i *meccanismi fisici* che portano a tale effetto, modificando la struttura del flusso vicino alla parete.

Capitolo 1

Introduzione

Questo progetto di tesi consiste nell'andare ad ipotizzare, valutare, ricercare, studiare per poi realizzare ed analizzare, mediante una simulazione numerica diretta, una tecnica di riduzione della *resistenza d'attrito a parete* di un *flusso turbolento* in un *canale piano*, e, nello specifico, una tecnica di riduzione *passiva*, che in questo caso è la tecnica delle *riblet*¹. In particolare si è andati ad effettuare delle simulazioni numeriche dirette, mediante l'utilizzo del programma *OpenFOAM*, utilizzando linguaggio di programmazione *C++*, per calcolare lo *sforzo d'attrito a parete* nell'*inner sublayer* di un *flusso turbolento in un canale piano*².

In un primo momento, in particolare, si è andati ad analizzare e valutare un canale piano semplice, senza nessun fattore esterno, per conoscere la resistenza iniziale di quello specifico caso, che poi si è voluto andare a ridurre, mediante l'utilizzo di una tecnica passiva, nel nostro caso. Dopo di che è stata introdotta nel canale, dal punto di vista computazionale, la soluzione delle *riblet*, per ridurre la resistenza, rispetto al caso base. Nello specifico le *riblet* sono delle scanalature, molto più piccole rispetto alla *dimensione caratteristica* del problema sul quale vengono applicate, che servono a modificare la struttura del flusso vicino alla parete dove sono realizzate, al fine di ridurre lo *sforzo d'attrito a parete*. Generalmente nell'utilizzo delle *riblet*, queste vengono utilizzate, generalmente, in una configurazione in cui sono *longitudinali rispetto alla direzione del flusso*, ovvero che corrono nella stessa direzione della componente media di velocità del flusso. Si ha quindi, che il loro utilizzo, influenzando lo strato limite del flusso che scorre, va a ridurre la *resistenza d'attrito viscosa*.

Nel nostro caso, si è scelto di utilizzare delle *riblet* di forma triangolare, per le quali si è tenuta fissa la dimensione della base, mentre si è fatta variare, come parametro di valutazione della tecnica, l'altezza delle scanalature realizzate. Inoltre, per gli studi che sono stati effettuati, il punto di partenza è stato dato sia da alcune delle conoscenze attuali in merito alla riduzione dello sforzo d'attrito a parete con tecniche passive, e in particolare con *riblet*, e sia da uno studio progressivo dell'impatto di questa soluzione nel canale, dato dai limiti di calcolo che la simulazione numerica diretta ci pone.

Nella figura 1.1 è possibile osservare un esempio di canale, in cui è stata introdotta

¹In questo lavoro di tesi, quando ci si riferisce alla tecnica utilizzata, si usa sempre la parola "riblet" poiché, essendo una parola di origine inglese, in italiano, anche quando è usata con significato plurale, non richiede la "s" finale; tuttavia, in questo lavoro, è presente anche il termine "riblets", in quanto nel testo sono stati citati alcuni articoli in merito, di lingua inglese, nei quali, la parola presenta differenze tra singolare e plurale, mediante l'aggiunta della "s" finale.

²Questa tesi è stata scritta utilizzando \LaTeX , ed in particolare grazie alla guida *L'arte di scrivere con \text{\LaTeX}*, di L. Pantieri & T. Gordini

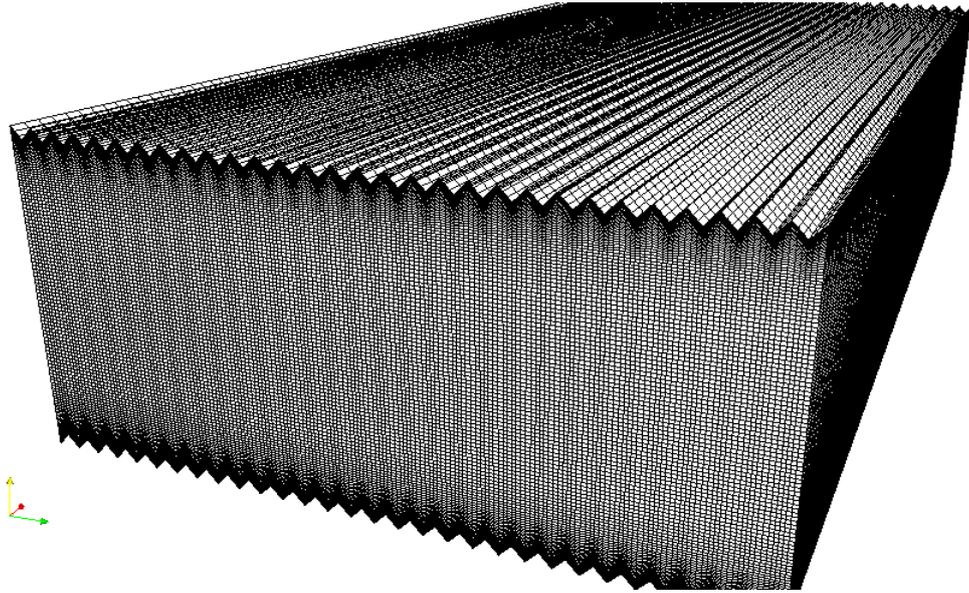


Figura 1.1: Esempio di rappresentazione su *ParaFOAM*, di un canale contenente *riblet*, in particolare nel caso di una mesh realizzata per le simulazioni di questa tesi, pronta per essere utilizzata, quindi già discretizzata nel numero di celle richiesto

la soluzione delle *riblet*, discretizzato come mesh pronta per un'analisi, realizzato su *OpenFOAM* e visualizzato con *ParaFOAM*. In particolare questa è la mesh di una delle simulazioni che sono state effettuate per ottenere i dati per questo lavoro di tesi.

Le simulazioni numeriche dirette, come detto, sono state effettuate mediante l'utilizzo di *OpenFOAM*, nel quale si va prima a realizzare la mesh che si vuole analizzare, poi si vanno a specificare le condizioni al contorno dell'analisi, ed in fine a specificare i dati relativi allo svolgimento della simulazione stessa, quali per esempio, passo temporale o durata della simulazione.

Sono state analizzate, nello svolgimento del lavoro, diverse configurazioni di *riblet*, per le quali a variare sono stati solamente pochi parametri del problema; nello specifico si è andati a confrontare il comportamento a pari *numero di Reynolds* globale, al variare dell'altezza delle *riblet*.

Per ogni blocco di dati caratterizzante i diversi casi ottenuti, si è valutato lo *sforzo medio a parete*, per capire quindi come le diverse configurazioni, ottenute per differenti parametri di forma delle *riblet*, andavano ad influenzare lo strato limite, e quindi a ridurre la *resistenza d'attrito viscoso* e lo *sforzo medio a parete*. Sono stati poi confrontati i diversi *coefficienti di sforzo d'attrito* ottenuti dalle simulazioni, al variare dei parametri utilizzati nell'analisi, con lo scopo di trovare infine il valore più basso d'attrito, e quindi la configurazione con la minore resistenza, ovvero quella che minimizza gli *sforzi medi a parete*.

Si è andati inoltre anche a svolgere un'analisi dei dati ottenuti, calcolando altre grandezze caratteristiche del *flusso in un canale*, per individuare e meglio caratterizzare, i *meccanismi fisici* caratterizzanti i flussi simulati, che portano alla modifica della struttura del flusso vicino alla parete, causando così la riduzione di resistenza cercata. In questo modo, inoltre, si è potuto comprendere anche in che condizioni di flusso si era, studiare l'accuratezza delle simulazioni, e trarre il maggior numero possibile di dati dalle simulazioni effettuate. Si ci è andati, inoltre a confrontare, con lo stato dell'ar-

te in merito a questa tecnica, dedotto dagli studi presenti in letteratura su flussi con *riblet* di diversa forma e dimensione, per avere, oltre che come già detto, un punto di partenza per lo studio che si è andato ad effettuare, anche un parametro di confronto per i risultati ottenuti.

Alla fine di questo lavoro di tesi, si sarà quindi trovata la configurazione di *riblet*, fra quelle studiate, che riduce maggiormente la resistenza d'attrito del flusso nel canale, rispetto al caso iniziale di canale piano liscio, senza *riblet*.

Nella figura 1.2 è riportata una simulazione *CFD*, tratta da Fu *et al* [1], in cui è pos-

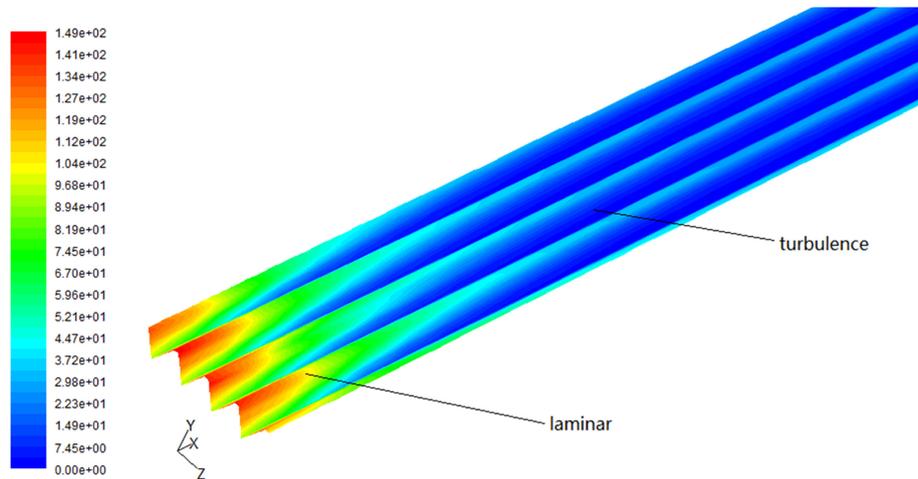


Figura 1.2: Esempio di utilizzo di *riblet*, in particolare "Counters of wall shear stress of *riblet* surfaces and smooth surfaces in different flow regime", tratta da simulazioni di Fu *et al* [1].

sibile osservare un flusso che scorre su di una parete su cui sono state realizzate delle *riblet*.

A questo punto della descrizione del lavoro svolto, prima di andare direttamente a vedere i casi analizzati, come questi sono stati realizzati, ed i dati da questi ottenuti, si andrà prima a richiamare la fenomenologia fisica che è stata alla base dei problemi trattati in questa tesi. In particolare, si va prima a richiamare i concetti fondamentali di un *fluido incompressibile*, poi dello *strato limite*, ed infine della *turbolenza di parete e dei flussi turbolenti*. Dopo di che, andandoci a concentrare maggiormente sulla *resistenza d'attrito a parete* di un *flusso turbolento che scorre in un canale piano*, e sulle tecniche utilizzate per ridurla, ed in particolare su quelle passive. Nello specifico si farà un rapido excursus su quello che è lo stato dell'arte in merito alle principali tecniche di riduzione della resistenza di parete di un flusso turbolento in un canale piano, andandoci a concentrare maggiormente su quelle passive ed ancora di più sulle conoscenze attuali in merito all'uso delle *riblet*, essendo stato questo il punto di partenza per il lavoro di tesi svolto.

Dopo aver effettuato questi richiami teorici, ed aver visto le soluzioni già conosciute in merito, si andrà quindi a vedere tutto il procedimento riportato, che è stato necessario per realizzare lo studio effettuato, sia per quanto riguarda i parametri e le soluzioni da analizzare scelte, sia per quanto riguarda la vera e propria realizzazione dei singoli casi su *OpenFOAM*, con dei brevi cenni al programma stesso, con il codice di calcolo riportato poi in appendice a questo lavoro, e sia il lavoro di post processing effettuato per ottenere i valori numerici cercati, da confrontare. Infine si riporteranno dunque tutti i

dati ottenuti, sia dalle simulazioni numeriche dirette, e sia dai programmi successivi di post processing utilizzati, visualizzando i risultati ottenuti da questo lavoro, confrontando i dati con gli obiettivi del lavoro, e potendo infine apprezzare, commentando i dati ottenuti, quelle che sono le conclusioni del lavoro, in merito all'obiettivo realizzato.

Capitolo 2

Fenomenologia del Problema

2.1 Fluido Incomprimibile

Un problema fluidodinamico viene detto di flusso *incomprimibile*, o anche di flusso *incompressibile*, quando le variazioni di densità del fluido non hanno effetti apprezzabili sulla soluzione (e quindi la densità può essere considerata con buona approssimazione costante). In particolare, dal punto di vista fisico avremo che *una corrente è detta incomprimibile se il volume Ω di ogni elemento fluido è costante nel tempo* [2]. Ricordando inoltre che la *divergenza* del campo di velocità $\nabla \cdot \mathbf{V}$ rappresenta la velocità di variazione del volume, cioè la dilatazione volumetrica di un elemento di fluido, allora la condizione di corrente incomprimibile è equivalente alla *condizione di incomprimibilità*

$$\nabla \cdot \mathbf{V} = 0 \quad (2.1)$$

Considerando inoltre l'equazione della conservazione della massa

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

si ottiene

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{V} \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{V} \quad (2.2)$$

e poiché la densità non può tendere a zero nel caso del modello di fluido continuo, ne consegue che la condizione di incompressibilità vista in precedenza, (2.1), è equivalente alla condizione

$$\frac{D\rho}{Dt} = 0, \quad (2.3)$$

e quindi si ha che *un elemento fluido durante la sua evoluzione lungo una traiettoria rimane a densità costante*.

Si può anche avere che una corrente risulta incomprimibile anche se la densità non è costante, come per esempio l'acqua marina, che pur essendo incomprimibile, ha una densità variabile che dipende dal grado di salinità dell'acqua; in questo caso, invece, noi considereremo solamente *fluidi omogenei*, caratterizzati da densità costante nello spazio, $\nabla \rho = 0$, e quindi avremo, che la relazione precedente (2.2), diventerà:

$$\frac{\partial \rho}{\partial t} = 0 \quad (2.4)$$

Da ciò ne consegue che un *fluido incomprimibile omogeneo* ad un certo istante, rimane *omogeneo*, cioè ρ rimane costante nello spazio e nel tempo.

2.1.1 Equazioni di Navier-Stokes per fluidi incomprimibili

Consideriamo ora le equazioni di Navier-Stokes per un fluido comprimibile newtoniano [2]:

$$\frac{D\rho}{Dt} = -\rho \cdot \nabla \mathbf{V} \quad (2.5a)$$

$$\rho \frac{D\mathbf{V}}{Dt} = -\nabla p + \nabla \cdot \left[\mu(\nabla \mathbf{V} + \nabla \mathbf{V}^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{V})\mathbf{I} \right] + \rho \mathbf{f} \quad (2.5b)$$

$$\rho \frac{D\varepsilon}{Dt} = -p\nabla \cdot \mathbf{V} + \rho\Phi + \nabla \cdot (\kappa \nabla T) + Q_v \quad (2.5c)$$

Nel caso di un fluido incomprimibile newtoniano, invece, la dipendenza degli sforzi in funzione del gradiente della velocità si semplifica nella forma:

$$\tau = \mu(\nabla \mathbf{V} + \nabla \mathbf{V}^T) \quad (2.6)$$

a causa della (2.1).

Inoltre l'energia interna ε dipende in generale dall'entropia e dalla temperatura, $\varepsilon = \varepsilon(s, T)$ [3], ed è possibile definire il *calore specifico* c come

$$c = \frac{\partial \varepsilon}{\partial T} \quad (2.7)$$

In questo caso la definizione di calore specifico è univoca, poiché per un fluido incomprimibile non si ha differenza fra *calore specifico a pressione costante* c_p e *calore specifico a volume costante* c_v . Nel caso di una corrente incomprimibile, avremo quindi, per la condizione di incomprimibilità, (2.1), che dalle equazioni di Navier-Stokes (2.5) si ottengono le *equazioni di Navier-Stokes per una corrente incomprimibile*

$$\frac{D\rho}{Dt} = 0 \quad (2.8a)$$

$$\rho \frac{D\mathbf{V}}{Dt} = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{V} + \nabla \mathbf{V}^T)] + \rho \mathbf{f} \quad (2.8b)$$

$$\rho \frac{D\varepsilon}{Dt} = \rho\Phi + \nabla \cdot (\kappa \nabla T) + Q_v \quad (2.8c)$$

$$\nabla \cdot \mathbf{V} = 0 \quad (2.8d)$$

alle quali si è aggiunta la condizione di incomprimibilità (2.1).

La funzione $\Phi = \frac{1}{\rho}(\tau \cdot \nabla) \cdot \mathbf{V}$ è la *funzione di dissipazione*, e rappresenta la dissipazione di energia meccanica per unità di massa dovuta alla *viscosità*.

Le equazioni (2.8) contengono comunque alcuni termini non lineari. In un caso di fluido omogeneo (ρ costante), in condizioni di pressione e temperatura standard, ed in assenza di forti variazioni di temperatura, il coefficiente di viscosità μ ed il coefficiente di conducibilità termica κ possono essere considerati costanti. In questo caso, quindi, si parla di *fluido incomprimibile a proprietà costanti*, e le (2.8) si semplificano, diventando:

$$\nabla \cdot \mathbf{V} = 0 \quad (2.9a)$$

$$\rho \frac{D\mathbf{V}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{V} + \rho \mathbf{f} \quad (2.9b)$$

$$\rho c \frac{DT}{Dt} = \rho\Phi + \kappa \nabla^2 T + Q_v \quad (2.9c)$$

dove l'equazione della continuità è sostituita dalla condizione (equivalente) di incomprimibilità.

La soluzione delle equazioni di Navier-Stokes (2.9), in particolare quella della continuità e quella della quantità di moto, dipendono da due parametri costanti: la densità ρ e la viscosità ν del fluido, oltre che dalla scala del dominio e da una scala temporale nel caso di soluzioni non stazionarie. Si possono dunque ottenere delle *leggi di scala* riscrivendo le equazioni del moto in forma *adimensionale*; per fare ciò è necessario introdurre delle grandezze di riferimento rispetto alle quali scalare le grandezze fisiche: in particolare le lunghezze possono essere scalate rispetto ad una lunghezza di riferimento L caratteristica del problema, ad esempio un diametro, una corda o una semi altezza; le velocità rispetto ad una velocità V , ad esempio la velocità della corrente indisturbata; il tempo è scalato rispetto ad un tempo caratteristico, che viene definito, avendo già introdotto una velocità ed una lunghezza caratteristiche, come $T = L/V$.

Si otterranno quindi in questo modo le forme adimensionali delle equazioni di Navier-Stokes della continuità e della quantità di moto; in particolare, in quest'ultima comparirà un termine, Re , definito come $Re = \frac{\rho V L}{\mu} = \frac{V L}{\nu}$ che è il *numero di Reynolds*. Questo è l'unico parametro adimensionale che compare nelle equazioni di Navier-Stokes in forma adimensionali; ciò significa che la soluzione di queste equazioni dipenderà anche dal valore di questo parametro. In particolare attraverso il *numero di Reynolds* di un flusso è possibile definire il *principio di similitudine dinamica*: *tutti i flussi che soddisfano le stesse condizioni al contorno ed iniziali, quando espresse in termini adimensionali, pur avendo valori differenti di ρ , μ , V ed L , ma tali che la loro combinazione dia lo stesso valore di Re , sono descritti dalla stessa soluzione adimensionale. In questo caso di ha che queste soluzioni, sono in similitudine dinamica [2].*

Similitudine dinamica

La *similitudine dinamica* è una condizione che permette di confrontare le soluzioni ottenute per flussi con parametri fisici differenti, a patto che questi, però, siano caratterizzati dall'aver determinati parametri adimensionali uguali. In questo caso si dice appunto che i flussi sono in *similitudine dinamica*, e si ha che le soluzioni trovate per uno, generalmente di più facile ottenimento, valgono anche per l'altro, per il quale può essere più difficile ottenerle. Nell'ambito di un problema fluidodinamico, generalmente, il più importante parametro adimensionale che si osserva per determinare la *similitudine dinamica* è il *numero di Reynolds*, in quanto si ha che *la similitudine dinamica di due sistemi geometricamente simili di tipo fluidodinamico esiste se loro sono caratterizzati dall'aver lo stesso numero di Reynolds* [4]. Oltre al numero di Reynolds esistono molti altri parametri adimensionali che servono a determinare la similitudine fra diversi casi analizzati, che variano in base al contesto che si sta analizzando.

Equazione della vorticità

Accanto al campo di velocità è possibile introdurre il campo vettoriale vorticità $\boldsymbol{\omega} \equiv \nabla \times \mathbf{V}$. Nel caso di fluido incompressibile a proprietà costanti si semplifica nella forma

$$\frac{D\boldsymbol{\omega}}{Dt} = \frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{V} \cdot \nabla) \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{V} + \nu \nabla^2 \boldsymbol{\omega} + \nabla \times \mathbf{f} \quad (2.10)$$

In assenza di forze esterne \mathbf{f} o esprimibili come il gradiente di un potenziale, il termine $\nabla \times \mathbf{f}$ è nullo. Nell'equazione (2.10) non compare alcun termine contenente la pressione.

L'equazione indica che la vorticità generata, ad esempio, lungo le pareti, in conseguenza della condizione di aderenza, è diffusa per effetto della viscosità e trasportata dalla corrente. Per bassi numeri di Reynolds la vorticità è diffusa in gran parte del campo di moto, mentre nel caso di alti numeri di Reynolds essa rimane confinata in regioni limitate, quali lo strato limite e la scia, in quanto la diffusione è sovrastata dal trasporto convettivo.

Nel caso di corrente bidimensionale, il vettore $\boldsymbol{\omega}$ è perpendicolare al piano del moto, $\boldsymbol{\omega} = \omega \mathbf{k}$, quindi nell'equazione (2.10) il termine $\boldsymbol{\omega} \cdot \nabla \mathbf{V} = 0$, e nel caso bidimensionale l'equazione della vorticità si riduce alla forma

$$\frac{D\boldsymbol{\omega}}{Dt} = \nu \nabla^2 \boldsymbol{\omega} \quad (2.11)$$

che rappresenta il semplice trasporto e diffusione della vorticità associata all'elemento fluido. La stessa situazione si può avere in casi particolari di flusso tridimensionale. Ma nel caso di moto genuinamente tridimensionale il termine $\boldsymbol{\omega} \cdot \nabla \mathbf{V}$ può essere positivo o negativo, ma non rappresenta una sorgente (positiva o negativa) di vorticità. Se la vorticità è nulla non si ha creazione di vorticità. Il meccanismo di variazione della vorticità può essere interpretato come effetto della contrazione o estensione della linea vorticosa, assunta coincidente istantaneamente con la traiettoria locale, che provoca una redistribuzione della vorticità per il principio di conservazione del momento angolare. In base a questa interpretazione tale termine è denominato *vortex stretching*.

2.1.2 Soluzioni delle equazioni per un flusso viscoso newtoniano

Le equazioni di continuità, momento ed energia, precedentemente ottenute, (2.5), costituiscono un sistema formidabile di equazioni non lineari alle derivate parziali. Non esistono né metodi analitici, né esistono teoremi di unicità o di esistenza generali. Inoltre mentre per un flusso laminare, che non è caratterizzato da fluttuazioni turbolente di piccola scala, una simulazione fluidodinamica risulta sufficientemente accurata, per i flussi turbolenti, fatta eccezione per la simulazione numerica diretta, la quale però può essere utilizzata solamente per bassi Re , si devono utilizzare correlazioni empiriche o modelli CFD, che però coprono solamente un limitato set di modelli fluidi. Sotto determinate ipotesi, invece, risulta possibile, a patto di soddisfare le condizioni richieste, semplificare notevolmente le equazioni di *Navier-Stokes*, ed ottenere sistemi più semplici, che ammettono soluzioni esatte. Fondamentalmente, ci sono due tipi di soluzioni esatte [5], che sono:

1. Soluzioni lineari, dove le accelerazioni convettive scompaiono;
2. Soluzioni non lineari, dove i termini convettivi non scompaiono.

È inoltre possibile classificare le soluzioni per il tipo di geometria o di flusso coinvolto; in questo modo abbiamo le seguenti soluzioni esatte [5]:

1. Couette, flussi stazionari, guidati da parete;
2. Poiseuille, flussi stazionari in un condotto, guidati dalla pressione;
3. Flussi non stazionari in un condotto;
4. Flussi non stazionari con contorni mobili;

5. Flussi in condotti, con aspirazioni e soffiaggi;
6. Flussi “wind-driven” (Ekman);
7. Soluzioni di similarità (dischi rotanti, flussi stagnanti, ecc...).

Quindi è possibile trovare delle soluzioni esatte alle equazioni di *Navier-Stokes*, (2.5), a patto però di essere in specifiche condizioni di esercizio, e quindi di avere delle ipotesi che vanno a semplificare le equazioni. Tuttavia essendo queste soluzioni possibili solamente per casi particolari, si va a vedere come è possibile semplificare le equazioni in altro modo, ottenendo delle soluzioni semplificate, o per meglio dire approssimate, che però valgono per un range di casi molto più ampio.

2.1.3 Soluzioni approssimate

Come si è appena visto, la soluzione analitica delle equazioni di Navier-Stokes è possibile solo in alcuni casi limitati, specifici e con geometrie semplici e ben definite. In altri casi, comunque, è comunque possibile poter semplificare le equazioni del moto; infatti analizzando la forma adimensionale delle equazioni, nelle quali il *numero di Reynolds* compare come parametro, si osserva come in funzione di questo valore, sia possibile distinguere due casi limite, che possono essere semplificati, rispetto alle equazioni di partenza [2]. In particolare si distingueranno due casi, un per *numero di Reynolds* molto basso, ed uno opposto, cioè per *numero di Reynolds* molto alto. Nell'andare a fare la distinzione fra questi due casi specifici è necessario ricordare come il *numero di Reynolds* vada, anche, ad indicare il rapporto fra le *forze d'inerzia* e le *forze viscosse* [4].

Flussi di Stokes

Nel caso di correnti estremamente lente e/o di viscosità molto elevate (comunque *numero di Reynolds* molto basso) le forze viscosse sono predominanti rispetto a quelle d'inerzia. Se inoltre il flusso è stazionario tutto il termine inerziale è piccolo rispetto alle forze viscosse e di pressione. I flussi caratterizzati da *numero di Reynolds* $Re \rightarrow 0$ sono detti *flussi di Stokes*. In questo caso è possibile trascurare i termini inerziali e le equazioni di Navier-Stokes si riducono alla forma approssimata:

$$\nabla \cdot \mathbf{V} = 0 \quad (2.12)$$

$$\nabla p = \frac{1}{Re} \nabla^2 \mathbf{V} \quad (2.13)$$

che in forma dimensionale si possono esprimere come

$$\nabla \cdot \mathbf{V} = 0 \quad (2.14a)$$

$$\nabla p = \mu \nabla^2 \mathbf{V} \quad (2.14b)$$

Il sistema di equazioni ottenuto (2.14) sono dette *equazioni di Stokes*. Queste sono lineari e di più facile soluzione rispetto al sistema completo di Navier-Stokes; inoltre il loro ordine è lo stesso del sistema completo e quindi è possibile imporre lo stesso numero di condizioni al contorno, ed in particolare è possibile imporre la condizione di aderenza, $\mathbf{V} = 0$, lungo una parete. Da ciò, attraverso una serie di passaggi matematici, è possibile giungere alla soluzione delle *equazioni di Stokes*.

Flussi ad alto numero di Reynolds

I flussi ad alto *numero di Reynolds*, rappresentano il caso opposto rispetto ai *flussi di Stokes*, e costituiscono una categoria di grande importanza pratica. Infatti questi rappresentano la maggior parte delle applicazioni più comuni della fluidodinamica, ed in particolare dell'aerodinamica, in quanto sia l'aria, ma anche l'acqua, hanno una viscosità molto piccola, e quindi, anche per velocità moderate, si raggiungono valori di Re elevati in molte applicazioni.

In particolare per *numero di Reynolds* elevato, si ha il caso limite in cui $Re \rightarrow \infty$ in cui la semplificazione delle equazioni consiste nel trascurare il termine viscoso $\mu \nabla^2 \mathbf{V}$, ottenendo il *sistema ridotto*, che viene rappresentato dalle *equazioni di Eulero*. La soluzione di questo sistema ridotto non soddisfa più tutte le condizioni al contorno, ed in particolare non è più possibile imporre la condizione di aderenza a parete.

A questo punto, si può definire, per i problemi ad alto *numero di Reynolds*, una regione di *strato limite*, vicino la parete, dove valgono ancora le condizioni di aderenza, mentre nel resto del flusso i termini viscosi sono trascurabili. La soluzione del problema completo consiste quindi in due zone: una zona *esterna* ottenuta risolvendo le equazioni di Eulero (relative ad un fluido perfetto e non viscoso), che abbraccia buona parte del flusso, ed una *interna*, quella dello *strato limite*, che occupa un sottile strato lungo la parete. Le due soluzioni, inoltre, devono raccordarsi in modo che esista una zona di transizione, dove entrambe le fenomenologie sono valide.

2.2 Strato limite

Nel 1904 Ludwig Prandtl introdusse uno dei concetti più importanti per lo studio delle correnti fluide caratterizzate da un alto *numero di Reynolds*, la *teoria dello strato limite*. Questa teoria dice che, per alti *Re*, il flusso lungo un corpo è caratterizzato da due zone principali:

zona esterna in cui le linee di corrente e velocità si discostano poco da quelle ottenute con un calcolo del flusso potenziale di un fluido ideale, inviscido, e dove quindi i termini viscosi possono essere trascurati, non si hanno gradienti di velocità elevati ed è possibile dunque applicare l'approssimazione di fluido inviscido;

zona interna dove il flusso, a differenza del flusso potenziale, il fluido non *scivola* lungo le pareti, ma *aderisce* su di esse, e dove quindi i termini viscosi non sono trascurabili, ma hanno, anzi, notevole importanza; inoltre si ha che il gradiente $\partial u_t / \partial n$ è molto grande.

Si ha dunque che la transizione da velocità nulla a parete ad un valore vicino a quello predetto dalla teoria potenziale avviene in un sottile strato, detto *strato limite*. Lo spessore di questo strato è δ , il quale è $\delta \ll L$, dove L indica una grandezza caratteristica del corpo nel fluido; si ha inoltre che δ diminuisce al crescere del *Re*.

La teoria dello *strato limite* è stata poi formalizzata in seguito mediante l'applicazione del metodo delle espansioni asintotiche congruenti, ed ulteriormente sviluppata definendo la teoria dello *strato limite* del secondo ordine per introdurre vari effetti, quali la curvatura della parete. Nel tentativo, infatti, di risolvere numericamente le equazioni dello *strato limite*, si realizzò che la soluzione è singolare se il profilo di velocità ha la derivata nulla a parete, cioè non è possibile calcolare lo sviluppo dello strato limite per stazioni a valle del punto di separazione, ovvero dove si ha appunto la derivata nulla.

Come si vedrà poi nei paragrafi successivi, la derivata del profilo di velocità ed il punto di separazione, sono collegati anche con l'andamento del gradiente di pressione.

Comunque per risolvere il problema della singolarità nella risoluzione numerica delle equazioni dello *strato limite*, è stato necessario adottare un metodo di calcolo inverso, anzichè imporre l'andamento della velocità esterna, occorre imporre l'andamento di una quantità associata allo *strato limite* stesso [2].

Equazioni dello strato limite

Consideriamo le equazioni di Navier-Stokes, in coordinate cartesiane per un dominio bidimensionale, per poter fare alcune considerazioni sul comportamento dello *strato limite*.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.15a)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = - \frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (2.15b)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = - \frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (2.15c)$$

Nel caso di corrente bidimensionale lungo una parete piana con l'asse x parallelo alla parete e l'asse y perpendicolare ad essa, le quantità sono adimensionalizzate rispetto ad una lunghezza di riferimento L , ed ad una velocità di riferimento V_∞ ; così facendo,

si ha che il *numero di Reynolds* è definito come : $Re = \frac{V_\infty L}{\nu}$.

Da questo è possibile definire un parametro ε , definito come $\varepsilon = \frac{1}{\sqrt{Re}}$. Questo parametro rappresenta il comportamento viscoso delle equazioni; per alti valori di Re , avremo bassi valori di ε , ed un contributo poco importante dei termini viscosi.

In particolare con $\varepsilon = 0$ si ottengono le *equazioni di Eulero*, che in questo specifico caso andiamo a considerare per una corrente esterna inviscida. Le condizioni al contorno sono la condizione di corrente uniforme all'infinito e la condizione di tangenza lungo la parete. Avremo che la soluzione esterna soddisfa solo parzialmente la condizione lungo la parete, e quindi nella regione in prossimità della parete si avranno *due scale di lunghezze*: lungo la direzione della parete le variazioni avvengono su una scala L , mentre invece, lungo la direzione perpendicolare alla parete le variazioni avvengono su una scala $\delta \ll L$. La relazione fra queste due scale è proporzionale alla relazione che c'è fra il *tempo caratteristico diffusivo* ed il *tempo caratteristico convettivo*, legati fra loro, a loro volta, dal *numero di Reynolds*.

Riscrivendo il sistema in funzione delle variabili interne, ponendo $\varepsilon = 0$, con le derivate che hanno tutte lo stesso ordine di grandezza, dal sistema ridotto, si ottengono le *equazioni dello strato limite*.

$$\frac{\partial U}{\partial X} + \frac{\partial V}{\partial Y} = 0 \quad (2.16a)$$

$$\frac{\partial U}{\partial T} + U \frac{\partial U}{\partial X} + V \frac{\partial U}{\partial Y} = -\frac{\partial P}{\partial X} + \frac{\partial^2 U}{\partial Y^2} \quad (2.16b)$$

$$0 = -\frac{\partial P}{\partial Y} \quad (2.16c)$$

Dalle equazioni riportate, in particolare dalla terza, risulta che *la pressione lungo la direzione perpendicolare alla parete è costante. La pressione attraverso lo strato limite è uguale al valore della pressione della corrente esterna p_0* . Si ha quindi che la pressione è impressa dal flusso esterno, inviscido.

Inoltre, limitando il ragionamento al solo caso di corrente stazionaria, la procedura di calcolo della corrente si divide in due parti:

- I) le equazioni di Eulero (quelle ottenute dalle (2.15), ponendo $\varepsilon = 0$) sono risolte indipendentemente dalle equazioni dello strato limite, con le condizioni all'infinito di corrente indisturbata e la condizione $v_0(x, 0) = 0$ a parete (ovvero la condizione di *tangenza*).
- II) la distribuzione della velocità $U_e(x) = u_0(x, 0)$ rappresenta la condizione esterna per le equazioni dello strato limite. Inoltre, poiché nello strato limite $p(x, y) = p_e(x)$, il sistema di equazioni diventerà:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.17a)$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{dp_e}{dx} + \nu \frac{\partial^2 u}{\partial y^2} \quad (2.17b)$$

La parete è una linea di corrente per il flusso esterno, ed il gradiente di pressione dp_e/dx è legato alla distribuzione delle velocità $U_e(x)$ dalla relazione

$$-\frac{1}{\rho} \frac{dp_e}{dx} = U_e \frac{dU_e}{dx}$$

Il calcolo della pressione avviene quindi mediante l'utilizzo di linee di corrente per il flusso, per le quali vale il teorema di Bernoulli, scritto come $p_e + \frac{1}{2}\rho U_e^2 = \text{cost}$; questo metodo, utilizzando una distribuzione di pressione p_e imposta è detto *metodo diretto*; tuttavia presenta problemi nel caso di separazione del flusso, in quanto le equazioni presentano una singolarità.

Influenza del gradiente di pressione sullo strato limite

Lo strato limite, essendo la parte di flusso a contatto con un corpo che si muove e/o che è investito da un fluido, risente dell'andamento del flusso lungo la sua traiettoria, in particolare se questo, subisce accelerazioni e/o decelerazioni dovute alla forma del corpo. Queste si ripercuotono quindi sul flusso come gradienti di pressione, poiché, considerando infatti che per ogni singola linea di flusso si ci trovi in una condizione ideale e valga la *Legge di Bernoulli*, il gradiente di pressione è opposto a quello di velocità. Le accelerazioni e le decelerazioni subite dal flusso, quindi, faranno sì che il flusso sarà caratterizzato da gradienti di pressione positivi e gradienti di pressione negativi; a seconda infatti se il flusso accelera o rallenta, avremo che la pressione diminuirà o aumenterà. Questi differenti gradienti di pressione, si ripercuoteranno quindi, anche sullo strato limite, ed a seconda dell'effetto che avranno su di esso, sono detti *gradiente di pressione favorevole* e *gradiente di pressione avverso*.

Nello strato limite, si ci può trovare quindi, a seconda del tipo di caso fluidodinamico che si sta considerando, in due condizioni di gradiente di pressione, *gradiente di pressione favorevole* e *gradiente di pressione avverso*. Il caso di *gradiente di pressione favorevole*, che si ha quando la pressione nello strato limite diminuisce, $dp_e/dx < 0$, non provoca il distacco dello strato limite dal corpo. Al contrario, il caso di *gradiente di pressione avverso*, definito come la situazione in cui si ha $dp_e/dx > 0$, è quello in cui si verifica la separazione dello strato limite dal corpo.

Nella figura 2.1 viene riportato l'intero andamento di uno strato limite sul dorso di un profilo alare, dove il flusso è costantemente accelerato; in questo caso è possibile osservare i diversi stati che lo strato limite può avere su di un corpo piano; avremo infatti inizialmente uno strato limite laminare, poi questo, dopo un punto di transizione diventerà turbolento, ed infine si arriverà ad un punto di separazione, dove si avrà il distacco dello strato limite dal corpo e si osserverà una zona di flusso inverso, con rimescolamento.

2.2.1 Grandezze integrali dello strato limite

Dall'andamento del profilo di velocità nello strato limite, è possibile ricavare diverse grandezze utili che caratterizzano un flusso che si sta considerando. Nello strato limite, infatti, abbiamo che la velocità non è uniforme, ma ha un andamento crescente lungo la direzione perpendicolare alla parete, diverso a seconda della situazione in cui ci si trova. Una prima grandezza che può essere calcolata, attraverso la velocità nello strato limite è la *resistenza d'attrito*, per profondità unitaria, dovuta allo strato limite, che si ottiene integrando lungo la parete il contributo dello sforzo di attrito a parete τ_w

$$D_f = \int_0^l \tau_w \cos(\phi) ds = \mu \int_0^l \left(\frac{\partial u}{\partial y} \right)_{y=0} dx, \quad (2.18)$$

dove s indica la coordinata curvilinea lungo la parete, localmente inclinata di un angolo ϕ rispetto all'asse x (per pareti con piccole curvature, come nel caso di profili alari

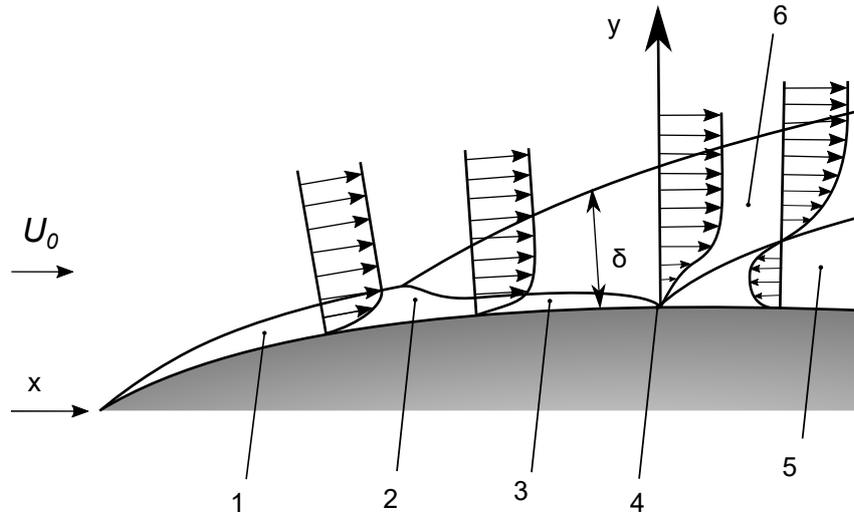


Figura 2.1: Andamento dello *strato limite* sul dorso di un profilo alare, con diversi gradienti di pressione, separazione, e strato limite laminare e turbolento. In particolare abbiamo diverse condizioni di strato limite, che sono:

- | | |
|---------------------------|-----------------------------|
| 1. Strato limite laminare | 4. Punto di separazione |
| 2. Transizione | 5. Strato separato |
| 3. Sottostrato viscoso | 6. Strato limite turbolento |

sottili, $ds \cos(\phi) \approx dx$).

È possibile inoltre ricavare, dallo strato limite, la *velocità di spostamento*, lo *spessore di spostamento* ed il *fattore di forma* dello strato limite. Abbiamo che, lungo il lato esterno dello *strato limite*, la componente della velocità parallela alla parete $u \rightarrow U_e$ e la componente perpendicolare v non è nulla; sia la corrente esterna che lo *strato limite*, infatti, soddisfano l'equazione di continuità.

Sottraendo all'equazione della continuità per il flusso esterno il bilancio di massa valido nello strato limite, si ha la relazione

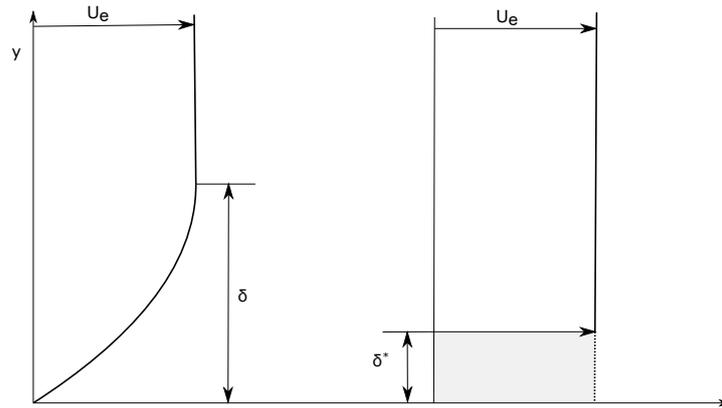
$$\frac{\partial v}{\partial y} - \frac{\partial v_e}{\partial y} = \frac{\partial u_e}{\partial x} - \frac{\partial u}{\partial x},$$

che, integrata attraverso lo *strato limite*, diventa

$$\lim_{n \rightarrow \infty} (v - v_e) = \frac{d}{dx} \int_0^\infty [U_e(x) - u] dy, \quad (2.19)$$

nella cui relazione si è posto $U_e(x)$ il valore della corrente esterna calcolata lungo la parete, in assenza di strato limite. Espandendo in serie di Taylor il valore di $u_e(x, y)$, $\left[u_e(x, y) = U_e(x) + \left(\frac{\partial u_e}{\partial y} \right)_{y=0} y + O(y^2) \right]$, e nel caso di flusso potenziale, per la condizione di irrotazionalità, $\left(\frac{\partial u_e}{\partial y} \right)_{y=0} = \left(\frac{\partial v_e}{\partial x} \right)_{y=0} = 0$, si ha che $u_e(x, y) = U_e(x) + O(y^2)$. Essendo $U_e(x) - u$ positivo ed essendo il termine destro della (2.19) una funzione crescente di x , allora $v - v_e > 0$, e ciò indica che lo *strato limite* ha un effetto di spostamento sul flusso esterno.

Definendo il termine sinistro dell'equazione (2.19), *velocità di spostamento* v_E , l'equa-

Figura 2.2: Spessore di spostamento δ^*

zione la si può riscrivere come:

$$v_E = \frac{d}{dx} \left[U_e \int_0^\infty \left(1 - \frac{u}{U_e} \right) dy \right] = \frac{d}{dx} (U_e \delta^*) \quad (2.20)$$

La lunghezza

$$\delta^* = \int_0^\infty \left(1 - \frac{u}{U_e} \right) dy \quad (2.21)$$

è lo *spessore di spostamento* (displacement thickness), e rappresenta la distanza dalla parete di cui si è spostata la corrente esterna in seguito alla diminuzione della velocità nello strato limite, come si può anche vedere nella figura 2.2.

Oltre alle grandezze appena riportate, per lo *strato limite*, in particolare per quello su di una lamina piana, che verrà brevemente descritto in seguito, esistono altre grandezze integrali che possono essere calcolate, e che servono a caratterizzare lo *strato limite* di un flusso su di una lamina piana.

Definiamo come prima cosa lo *spessore dello strato limite*. Questo è la distanza alla quale la velocità nello *strato limite* $u = 0.99 U_\infty$. Tuttavia a volte può risultare ambigua, in quanto si ha che la componente u della velocità tende *asintoticamente* al valore U_∞ . Oltre allo *spessore di spostamento*, visto in precedenza (equazione (2.21) e figura 2.2), si può analogamente definire lo *spessore della quantità di moto* θ , che tiene conto della diminuzione di flusso di quantità di moto nello strato limite rispetto al flusso potenziale; partendo da

$$\rho U_\infty^2 \theta = \rho \int_0^\infty u (U_\infty - u) dy \quad (2.22)$$

Otteniamo poi

$$\theta = \int_0^\infty \frac{u}{U_\infty} \left(1 - \frac{u}{U_\infty} \right) dy \quad (2.23)$$

Definiamo inoltre il *fattore di forma* H , come

$$H = \frac{\delta^*}{\theta} \quad (2.24)$$

Abbiamo infine lo *spessore di energia*, definito partendo da

$$\rho U_\infty^3 \delta_E = \rho \int_0^\infty u (U_\infty^2 - u^2) dy \quad (2.25)$$

ed ottenendo poi

$$\delta_E = \int_0^\infty \frac{u}{U_\infty} \left(1 - \frac{u^2}{U_\infty^2}\right) dy \quad (2.26)$$

I termini che sono stati ora riportati, compaiono inoltre nella prima versione dell'*equazione integrale della quantità di moto per il flusso turbolento* ricavata da *Karman*, nel 1921 [5], che aveva forma identica alla relazione per quella del flusso laminare, che è:

$$\frac{d\theta}{dx} + (2 + H) \frac{\theta}{U_e} \frac{dU_e}{dx} = \frac{\tau_w}{\rho U_e^2} = \frac{C_f}{2} \quad (2.27)$$

Per tutte le grandezze fin qui riportate, inoltre, esistono delle relazioni teoriche esatte per alcuni problemi specifici. In particolare, nel caso di un flusso che investe una lamina piana (vedi paragrafo successivo) esistono relazioni teoriche esatte, sia nel caso di flusso laminare, che nel caso di flusso turbolento (che verrà poi approfondito nella relativa sezione, vedi paragrafo 2.3).

Le relazioni per il caso di flusso laminare sono:

$$\delta = 5.0 \frac{x}{\sqrt{Re_x}} \quad (2.28a)$$

$$\delta^* = 1.7208 \sqrt{\frac{\nu x}{U_\infty}} \quad (2.28b)$$

$$\theta = 0.664 \sqrt{\frac{\nu x}{U_\infty}} \quad (2.28c)$$

$$H = 2.605 \quad (2.28d)$$

$$\delta_E = 1.044 \sqrt{\frac{\nu x}{U_\infty}} \quad (2.28e)$$

Per il caso turbolento, invece, le relazioni sono:

$$\delta^* = 0.04625 \frac{x}{Re_x^{1/5}} \quad (2.29a)$$

$$\theta = 0.035972 \frac{x}{Re_x^{1/5}} \quad (2.29b)$$

$$H = 1.286 \quad (2.29c)$$

2.2.2 Cenni sullo strato limite su lamina piana e soluzione di Blasius

Un caso particolare di strato limite è quello che si ha quando un flusso investe una lamina piana, ad incidenza nulla. In questo caso si avrà una prima zona (a seconda del valore del *numero di Reynolds*, se questo non è eccessivamente alto) di strato limite laminare, e poi una seconda zona (se viceversa il *numero di Reynolds* è sufficientemente alto) di strato limite turbolento. Qui parleremo brevemente dello strato limite laminare, mentre quello turbolento verrà trattato nella relativa sezione (vedi sottoparagrafo successivo e paragrafo 2.3).

Per il caso di strato limite, bidimensionale, stazionario, lungo una lamina piana, la prima soluzione esatta, per le equazioni (2.17), è stata ottenuta da Blasius (1908).

Si ha che, in assenza di gradiente di pressione, le equazioni dello strato limite diventano:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.30a)$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \nu \frac{\partial^2 u}{\partial y^2} \quad (2.30b)$$

con condizioni al contorno

$$y = 0 : u = v = 0 \quad \text{e} \quad y = \delta : u = U_\infty$$

Introducendo poi la funzione di corrente ψ , è possibile riscrivere le equazioni precedenti, e le relative condizioni al contorno, in funzione di questa.

Non essendoci una lunghezza di riferimento definita dalla geometria, o dal flusso potenziale, è possibile ipotizzare che *i profili di velocità al variare di x , siano in similitudine tra di loro*. I fattori di scala più opportuni per la velocità e la coordinata y sono rispettivamente U_∞ e δ , e quindi la proprietà di similitudine implica che

$$\frac{u}{U_\infty} = g\left(\frac{y}{\delta}\right)$$

dove la funzione g è la stessa per tutte le distanze x .

Dalle considerazioni svolte precedentemente si ha che $\delta \sim (\nu x / U_\infty)^{1/2}$, quindi introducendo la variabile $\eta = \frac{y}{\delta} = y \sqrt{\frac{U_\infty}{2\nu x}}$ e la funzione di corrente in forma adimensionale, $f(\eta) = \frac{\psi}{\sqrt{2\nu x U_\infty}}$, la componente della velocità u si ottiene dalla definizione di funzione di corrente:

$$u = \frac{\partial \psi}{\partial y} = \sqrt{2\nu x U_\infty} \frac{\partial f}{\partial \eta} \frac{\partial \eta}{\partial y} = U_\infty f'(\eta) \quad (2.31)$$

dove f' indica la derivazione rispetto alla variabile η . L'espressione appena ottenuta è consistente con l'ipotesi di similarità. Analogamente per la componente v della velocità si ha:

$$v = -\frac{\partial \psi}{\partial x} = -\frac{\partial}{\partial x} \left(\sqrt{2\nu x U_\infty} f(\eta) \right) = \frac{1}{2} \sqrt{\frac{2\nu U_\infty}{x}} (\eta f' - f) \quad (2.32)$$

Si ottiene così che l'equazione della funzione di corrente diventa un'equazione differenziale ordinaria per la funzione $f(\eta)$, ovvero *l'equazione di Blasius*

$$f''' + f f'' = 0 \quad (2.33)$$

con condizioni al contorno

$$\eta = 0 : f = f' = 0 \quad \text{e} \quad \eta = \infty : f' = 1$$

L'equazione (2.33) non è lineare. Blasius ottenne la soluzione in forma approssimata con l'espansione in serie, che può essere risolta numericamente. I valori delle funzioni $f(\eta)$, $f'(\eta)$ e $f''(\eta)$ sono stati tabellati, per poter essere confrontati con i dati sperimentali. La soluzione di Blasius si dimostrò poi essere in ottimo accordo con i dati sperimentali ottenuti da Nikuradse (1942).

Dalla soluzione di Blasius, è poi possibile ricavare altre grandezze dello strato limite laminare, come il *coefficiente di attrito (skin friction) τ_w* , il *coefficiente di resistenza c_D* , o le grandezze integrali dello strato limite, viste in precedenza.

2.2.3 Caratterizzazione dello strato limite turbolento

Per quanto riguarda il flusso turbolento, questo verrà trattato, essendo uno degli argomenti fondamentali su cui si basa la tesi, maggiormente nel dettaglio nella sezione successiva (vedi sezione 2.3). In questo paragrafo, quindi, ci limiteremo a completare il discorso relativo allo strato limite, includendo quindi, anche gli aspetti legati a quello

turbolento.

Nel caso dello strato limite turbolento lungo una parete, si ha uno sviluppo della corrente turbolenta nella direzione parallela alla parete. Lo spessore $\delta(x)$ aumenta, ed anche le altre grandezze integrali, quali lo spessore di spostamento δ^* e della quantità di moto θ , variano in x . Nel caso dello strato limite, inoltre, a differenza del canale piano e del tubo rettilineo, lo sforzo d'attrito non è noto a priori. Infine la parte esterna dello strato limite consiste in un'interfaccia variabile nel tempo e caratterizzata da un alto gradiente di intermittenza. Nel caso di strato limite con campo medio bidimensionale e staticamente stazionario, la statistica dipende da y , come succede nel caso del canale, ma anche da x , ovvero lungo la direzione della corrente. Nonostante ci siano delle differenze per un flusso turbolento di strato limite libero, rispetto ad un flusso confinato, per esempio in un canale, il comportamento del moto turbolento nella regione interna, ovvero quella per $y/\delta < 0.1$, è lo stesso che si ha anche per il flusso confinato; differenze evidenti, le si hanno, invece, nella regione esterna.

Analogamente al procedimento utilizzato nel caso dello strato limite laminare, anche per quello turbolento, è possibile eseguire un'analisi degli ordini di grandezza dei termini delle equazioni mediate di Reynolds. Infatti come si vedrà nel paragrafo 2.3.7, nel caso di un flusso turbolento, le equazioni di *Navier-Stokes*, le (2.42), laddove se ne va a fare la media nel tempo, diventano le *equazioni mediate alla Reynolds*, nello specifico le equazioni (2.44) e (2.45).

Nello specifico, nel caso di uno *strato limite turbolento*, considerando la componente lungo y dell'equazione della quantità di moto, la (2.45), si ottiene

$$0 = -\frac{\partial P}{\partial y} - \frac{\partial}{\partial y} (\overline{\rho v'^2}) \quad (2.34)$$

A differenza del caso laminare, attraverso lo strato limite, è costante la somma della pressione e dello sforzo turbolento normale $-\overline{\rho v'^2}$. Tuttavia, per le condizioni di aderenza a parete, $P + \overline{\rho v'^2} = P_w = p_e$. Le equazioni dello strato limite sono:

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \quad (2.35a)$$

$$U \frac{\partial U}{\partial x} + V \frac{\partial V}{\partial y} = -\frac{1}{\rho} \frac{dp_e}{dx} + \frac{\partial}{\partial y} \left(\nu \frac{\partial U}{\partial y} - \overline{u'v'} \right) \quad (2.35b)$$

Le equazioni differiscono dal caso laminare per il termine $-\overline{\rho u'v'}$ che rappresenta lo sforzo di Reynolds, dovuto alle fluttuazioni turbolente.

Se si vuole applicare l'equazione di *Von Karman* per lo strato limite, che è applicabile anche nel caso di flusso turbolento, si ci deve ricordare che in questo caso, il termine τ corrisponde allo sforzo totale, somma di contributo viscoso e dello sforzo di Reynolds. L'espressione per il sottostrato viscoso e la legge logaritmica, le equazioni (2.54) e (2.55), valgono anche per lo strato limite, con delle piccole correzioni per tenere conto della piccola influenza del gradiente di pressione. L'andamento di queste leggi può essere osservato nella figura 2.12.

Al contrario delle espressioni precedenti, quelle per il sottostrato viscoso e la legge logaritmica, nella regione esterna, la legge del *difetto di velocità*, l'equazione (2.56), è differente e dipende dal gradiente di pressione.

Nella zona esterna il profilo di velocità segue la *legge della scia (law of the wake)*. Questa è una legge empirica proposta da Cole basata su dati sperimentali e sulla considerazione che la parte esterna dello strato limite turbolento ha un comportamento simile a quello

di una scia (turbolenta), ovvero come in assenza di parete. L'andamento della velocità nello strato limite, in forma di *difetto di velocità*, è rappresentato dalla funzione:

$$\frac{U_e - U}{U_\tau} = \frac{1}{\kappa} \left\{ -\log\left(\frac{y}{\delta}\right) + \Pi \left[2 - w\left(\frac{y}{\delta}\right) \right] \right\} \quad (2.36)$$

La funzione $w\left(\frac{y}{\delta}\right)$ è la funzione di scia, proposta da Coles,

$$w\left(\frac{y}{\delta}\right) = 2 \sin^2\left(\frac{\pi}{2} \frac{y}{\delta}\right)$$

mentre il parametro Π dipende dalle condizioni della corrente esterna,

$$\Pi = 0.8 (\beta + 0.5)^{3/4} \quad \text{con} \quad \beta = \frac{\delta^*}{\tau_w} \frac{dp_e}{dx}$$

Attraverso poi alcune considerazioni fatte nel caso della legge di Blasius, si può supporre l'andamento della velocità e ottenere un'espressione per il coefficiente di attrito di facile utilizzo.

2.2.4 Struttura dello strato limite

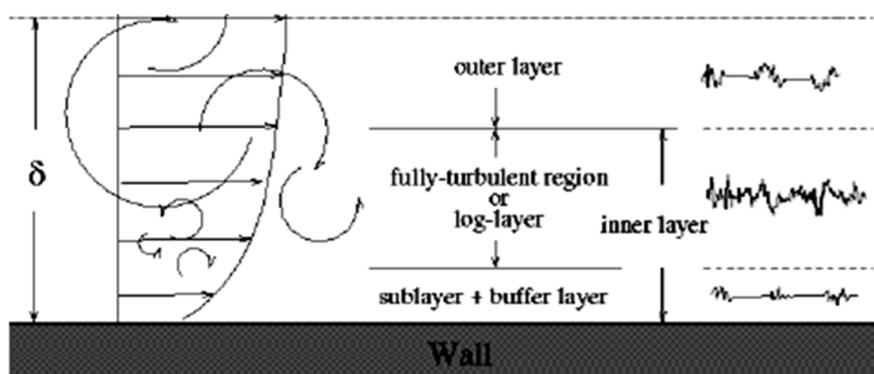


Figura 2.3: Schema riportante la divisione dello strato limite nelle diverse zone che lo compongono, con le indicazioni delle dimensioni dei vortici e degli spettri di turbolenza caratterizzanti ognuna delle diverse zone, indicando per ogni zona approssimativamente a quale parte del profilo di velocità nello strato limite corrisponde; schema tratto da [6].

Lo strato limite, sia per un flusso laminare, che per un flusso turbolento, è costituito al suo interno da diverse zone. In particolare questa divisione risulta essere più importante ed evidente per quanto riguarda lo strato limite turbolento. Queste sono caratterizzate in base a quali forze predominano, e al relativo andamento della velocità in funzione della distanza da parete. Inoltre si avrà, che lo spessore e l'importanza di queste zone cambierà fra strato limite laminare e strato limite turbolento, ma in entrambi i casi, saranno comunque tutte presenti. Nella figura 2.3 è riportata una prima e sommaria divisione dello strato limite nelle zone che lo compongono. In questa inoltre viene indicato anche a quale parte del profilo di velocità, dello strato limite, queste zone corrispondono. Vengono infine riportate delle indicazioni caratterizzanti la turbolenza e la vorticità nelle varie zone, caratteristica esclusiva dello strato limite turbolento, riportate come spettro di turbolenza e dimensione dei vortici.

Andando poi a vedere in modo più approfondito la divisione dello strato limite, si evidenzia in primo luogo, che le zone caratterizzanti lo strato limite sono:

- Inner layer
- Outer layer

A sua volta, poi, l'inner layer, che rappresenta la parte più importante dello strato limite, pur essendo la parte più piccola, ma essendo quella caratterizzata dai maggiori gradienti di velocità, è divisa in:

- Viscous layer
- Buffer layer
- Log. law region

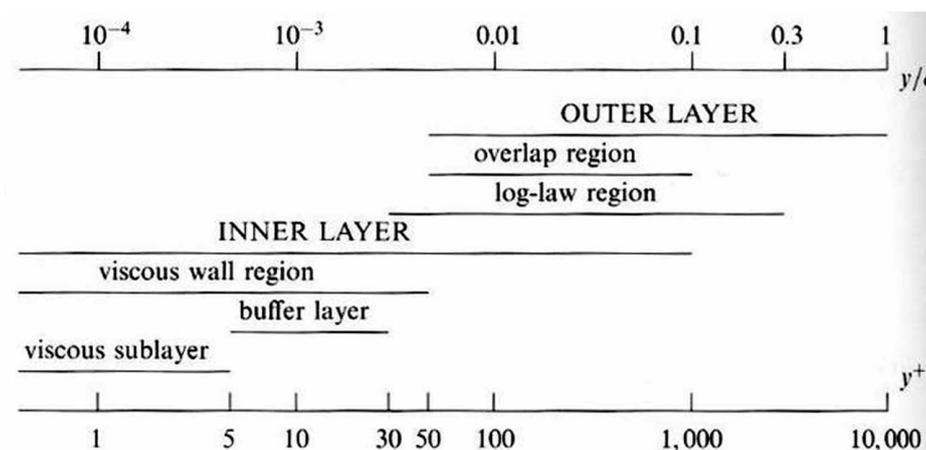


Figura 2.4: Divisione schematica di tutte le zone caratterizzanti lo strato limite, dove per ogni zona è indicata la porzione di strato limite che esse caratterizzano. Le dimensioni dello strato limite sono riportate sia in funzione della distanza adimensionalizzata con lo spessore di strato limite, y/δ e sia in funzione della distanza in termini di unità di parete y^+ ; immagine tratta da [6].

La divisione dello strato limite, sopra descritta, è riportata sia nella figura 2.3, vista in precedenza, che nella figura 2.4, in forma schematica. Nello schema, inoltre, sono presenti anche altri nomi che caratterizzano alcune parti dello strato limite, che non sono stati sopra riportati. Queste verranno poi prese in esame successivamente, mentre ora ci concentreremo sulle zone sopra riportate.

Andiamo poi a vedere un po' più nel dettaglio come sono caratterizzate queste sottozone dello strato limite, e quali sono gli aspetti che le caratterizzano.

L'*outer layer* è la zona più esterna dello strato limite, quella che congiunge la parte più interna dello strato limite (l'*inner layer* appunto) con la corrente esterna indisturbata. Poiché i maggiori gradienti di velocità si trovano, come detto nell'*inner layer*, nell'*outer layer*, al contrario, il gradiente di velocità non sarà molto accentuato. L'*outer layer* costituisce comunque la parte più grande dello strato limite. In particolare avremo che questo, nella maggior parte dei casi (o comunque in condizioni di strato limite standard), in termini di unità di misura di parete, va da $y^+ > 50$ a $y^+ \approx 10000$, che corrisponde circa alla fine dello strato limite. In questa zona, pur non avendo grandi gradienti di velocità, si avranno tuttavia i vortici di rimescolamento di dimensioni maggiori, per quanto riguarda il flusso turbolento; ciò è anche dovuto alle maggiori dimensioni di

questa zona, che permette di “ospitare” vortici di dimensioni maggiori. L’*outer layer*, inoltre, come è possibile vedere nella figura 2.4, condivide una parte della *log. law region* con l’*inner layer*. Questa, come si vedrà di seguito, è la zona detta di sovrapposizione, o *overlap region*, e costituisce una parte di strato limite in cui si è sia nell’*inner layer* che nell’*outer layer*.

L’*inner layer*, invece, pur essendo la parte più piccola dello strato limite, è quella che, sia a causa dei forti gradienti di velocità, e sia per le diverse forze che predominano al suo interno, rappresenta la parte più importante dello strato limite. In particolare avremo che l’*inner layer* occupa la parte di strato limite che va dalla parete, fino ad una distanza di circa $y/\delta \simeq 0.1$, oppure in termini di unità di parete $y^+ \simeq 1000$. Si può quindi riassumere dicendo che l’*inner layer* è la parte di strato limite per $y/\delta < 0.1$ o $y^+ < 1000$. In questa parte di strato limite avremo inoltre, a differenza dell’*outer layer*, che i vortici di rimescolamento hanno dimensioni ridotte, sia a causa delle dimensioni ridotte di questa parte di strato limite, e sia, soprattutto, a causa della forte influenza della viscosità (ciò è valido soprattutto nella *viscous wall region*, che verrà dopo introdotta e approfondita, mentre nella *log. law region* ciò non è del tutto vero). Inoltre, poiché, come detto, al suo interno prevalgono forze diverse, viene a sua volta diviso in alcune sottozone. Queste sono, come è possibile vedere nella figura 2.4, *viscous layer*, *buffer layer* e *log. law region*.

La *viscous layer* è la parte di strato limite a contatto con la parete solida, ed è anche la più piccola delle zone. Questa occupa la zona di strato limite: $y^+ < 5$ o $y/\delta < 5 \cdot 10^{-4}$; anche a causa di ciò, in molte misure sperimentali, risulta difficile misurare la velocità del flusso in questa parte dello strato limite, se non utilizzando sonde apposite (come per esempio l’*anemometro a filo caldo*); anche dal punto di vista computazionale risulta difficile, ed abbastanza complesso e costoso in termini di risorse di calcolo, simulare in modo preciso la velocità in questa zona, in quando è necessario avere delle celle di calcolo molto molto piccole, al fine di catturare bene i piccoli vortici che si hanno in questa zona. Infatti in questa parte dello strato limite, come già detto per l’*inner layer*, di cui la *viscous layer* fa parte, i vortici di rimescolamento hanno dimensioni molto piccole. Ciò è dovuto principalmente al forte effetto delle forze viscosi sul flusso (e da ciò appunto il nome *viscous*), che prevalgono su quelle inerziali. In questa zona l’andamento della velocità, in funzione della distanza da parete, espresso in termini di unità di parete, è dato dalla formula $u^+ = y^+$ [per capire come si ha questa relazione, vedere l’equazione (2.54)].

La *log. law region*, invece, al contrario della *viscous layer*, è la parte di strato limite in cui le forze di inerzia sono predominanti su quelle viscosi. Il nome di questa zona, o regione, è dato dal fatto che qui, la velocità, in funzione della distanza da parete, espresso in termini di unità di parete, è data dalla formula $u^+ = B + \frac{1}{\kappa} \log y^+$ con $\kappa = 0.41$ e $B = 5.2$ [anche qui, per capire come si ha questa relazione, vedere l’equazione (2.55)]. Questa zona è la più grande fra quelle presenti nell’*inner layer*, presenta grandi vortici di rimescolamento (soprattutto nella parte che condivide con l’*outer layer*) e fluttuazioni, per lo strato limite turbolento, molto forti. In questa zona infatti si ha la maggiore vorticità ed il maggior rimescolamento di velocità. Le dimensioni, e la collocazione, di questa zona nello strato limite, data in termini di unità di parete, sono $y^+ > 30$ e $y/\delta < 0.3$. Questa zona, inoltre, pur facente parte dell’*inner layer*, data la sua dimensione, ed anche i confini non netti che si hanno nello strato limite, presenta una zona (quella superiore) in comune con l’*outer layer*; avremo infatti che anche in quest’ultimo, in particolare nella parte più “bassa”, l’andamento della velocità sarà dato

dalla legge logartimica; questa zona in comune è chiamata, come citato in precedenza, zona di sovrapposizione, o *overlap region*, e verrà descritta alla fine del paragrafo. La *log. law region*, risulta, a differenza della *viscous layer*, molto più facile sia da simulare computazionalmente, e soprattutto da misurare sperimentalmente, sia grazie alle sue maggiori dimensioni ed anche al fatto di non essere attaccata alla parete (il che crea invece problemi ad utilizzare sonde non eccessivamente piccole ed adatte). Per questo motivo, i dati che vengono ottenuti in questa zona, sono molto sfruttati ed utilizzati (in elaborazioni di post processing), con numerose leggi empiriche, le quali permettono, grazie ai dati qui misurati, di ricavare anche grandezze o valori, relativi alla *viscous layer*, o che richiedono dati da quest'ultima parte; un esempio di ciò è il *metodo di Clauser*, vedi paragrafo 2.3.9, che permette di conoscere la resistenza d'attrito dello strato limite turbolento, pur non conoscendo la velocità nella *viscous layer*, attraverso l'utilizzo dell'equazione (2.61), che è quella che si ottiene e che caratterizza il *metodo di Clauser*, appunto.

Per quanto riguarda le equazioni che caratterizzano l'andamento della velocità nella *viscous layer* e nella *log. law region*, è possibile osservare l'andamento di queste leggi nella figura 2.12.

Mentre come detto il confine fra la *log. law region* e l'*outer layer* non c'è, in quanto le due zone condividono una parte in comune, la separazione fra la *log. law region* e la *viscous layer* è data dalla zona detta *buffer layer*, o zona di transizione.

La *buffer layer* è appunto la zona che separa la *log. law region* e la *viscous layer*, ed è la zona dove si ha la transizione dalla zona in cui domina la viscosità e quella in cui domina la turbolenza (o le forze di inerzia). La dimensione, e la collocazione, di questa zona nello strato limite, data in termini di unità di parete, è $5 < y^+ < 30$. Pur essendo la zona di separazione fra quella in cui prevalgono le forze di inerzia (*log. law region*) e quella in cui prevalgono le forze viscosi (*viscous layer*), in questa zona risultano essere ancora importanti le forze di inerzia. Infatti questa zona, insieme alla *viscous layer*, fanno parte della *viscous wall region*, che verrà a breve descritta.

Si ha infatti che oltre alle zone sopra descritte, che comunque coprono tutto lo strato limite, ci sono altre zone che caratterizzano lo strato limite, come più volte citato, e come è possibile osservare nella figura 2.4, che corrispondono a pezzi di zone già viste. Avremo in particolare:

- Viscous wall region
- Overlap layer

La *viscous wall region* è una parte dello strato limite in cui le forze viscosi hanno un ruolo prevalente. Questa corrisponde, come detto in precedenza, a *viscous layer* ed a *buffer layer*, ma non solo; infatti è composta anche ad una piccola parte iniziale della *log. law region*. Conseguentemente avremo che la sua dimensione/posizione nello strato limite è, in termini di unità di parete, $y^+ < 50$. La differenza principale fra questa zona e quella *viscous layer*, visto che si è detto che per entrambe sono importanti le forze viscosi, è il ruolo che proprio quest'ultime ricoprono. Infatti per la *viscous layer* si è detto che qui le forze viscosi sono predominanti rispetto a tutte le altre forze; la *viscous wall region*, è invece, la regione dove le forze viscosi hanno un contributo significativo per lo sforzo d'attrito, e quindi sono presenti in modo significativo. Quindi le forze viscosi sono significative in entrambe le zone, ma mentre nel *viscous layer* sono le uniche rilevanti, e quindi predominanti, nella *viscous wall region*, coesistono anche con altre forze anch'esse rilevanti (principalmente nella parte che non corrisponde al

viscous layer.

L'*overlap layer* è la zona di sovrapposizione fra l'*inner layer* e l'*outer layer*, come detto in precedenza. Corrisponde inoltre anche ad una parte della *log. law region*, e nello specifico la sua dimensione/posizione nello strato limite è, in termini di unità di parete, $y^+ > 50$ ed anche $y/\delta < 0.1$. Questa regione come si può evincere dai numeri appena riportati, inizia laddove termina la *viscous wall region*, ovvero quando le forze viscosi diventano trascurabili; per questo motivo, come in tutta la *log. law region*, sono predominanti le forze di inerzia, e quindi a causa dei forti rimescolamenti dovuti ai vortici presenti in questa zona, i confini fra inner ed outer layer diventano labili, e quindi nasce questa zona di sovrapposizione. Anche l'*overlap layer* ha una sua legge che descrive l'andamento della velocità; inoltre essendo molto influenzata dalla *log. law region*, questa legge deriva proprio da quella, imponendole però le condizioni di raccordo; da ciò si ottiene l'equazione caratteristica di questa zona, che è l'equazione (2.57).

In merito all'*overlap layer*, che nasce, come detto, dai forti vortici di rimescolamento presenti in quella parte dello strato limite, è possibile dire, che la parte di strato limite corrispondente a quella zona, ed a quella a y^+ maggiori della *log. law region*, è detta anche *fully-turbulent region*, ovvero regione a piena turbolenza. Infatti, come è anche possibile osservare nella figura 2.3, questa è la zona dove la turbolenza ha la sua massima intensità, come si vede dallo spettro di turbolenza riportato in figura.

Nella tabella riportata in figura 2.5, sono riportate tutte le zone costituenti lo strato

Region	Location	Defining property
Inner layer	$y/\delta < 0.1$	$\langle U \rangle$ determined by u_τ and y^+ , independent of U_0 and δ
Viscous wall region	$y^+ < 50$	The viscous contribution to the shear stress is significant
Viscous sublayer	$y^+ < 5$	The Reynolds shear stress is negligible compared with the viscous stress
Outer layer	$y^+ > 50$	Direct effects of viscosity on $\langle U \rangle$ are negligible
Overlap region	$y^+ > 50, y/\delta < 0.1$	Region of overlap between inner and outer layers (at large Reynolds numbers)
Log-law region	$y^+ > 30, y/\delta < 0.3$	The log-law holds
Buffer layer	$5 < y^+ < 30$	The region between the viscous sublayer and the log-law region

Figura 2.5: Tabella che riporta tutte le zone costituenti lo strato limite, le relative dimensioni/posizioni nello strato limite, e le loro proprietà; tabella tratta da [6].

limite descritte, con indicato per ognuna la relativa dimensione/posizione nello strato limite, ed anche una caratteristica peculiare.

2.3 Turbolenza e flusso turbolento

2.3.1 Natura della turbolenza

Il fenomeno della *turbolenza* è qualcosa che può essere osservato tutti i giorni, in qualsiasi momento, per esempio osservando le nuvole in cielo, ma che tuttavia risulta difficoltoso da descrivere con una precisa definizione di turbolenza. Tuttavia è possibile fare un elenco delle caratteristiche della turbolenza, descrivendo per ognuna in cosa consiste.

Irregolarità L'irregolarità, o casualità, di un *flusso turbolento* fa sì che un approccio deterministico ai problemi turbolenti sia impossibile, rendendo necessario un approccio statistico [7].

Diffusività La diffusività della *turbolenza* causa rapidi mescolamenti ed incrementi dei rapporti di trasferimento delle grandezze che caratterizzano il flusso turbolento. Questa distingue inoltre un flusso turbolento da uno casuale, ma che non è turbolento. Inoltre la diffusività della turbolenza è il *fattore fondamentale*, che dal punto di vista applicativo, la rende importante, poiché è quella che *previene la separazione dello strato limite* ad angoli di incidenza per cui, nel caso di flusso laminare, si avrebbe la separazione. Ciò tuttavia non impedisce la separazione in modo assoluto, laddove gli angoli diventino troppo elevati. Inoltre è la *causa principale della resistenza dei flussi nei condotti*, ed incrementa il trasferimento di quantità di moto nelle correnti [7].

Numeri di Reynolds elevati Per avere un *flusso turbolento* è necessario che questo sia caratterizzato da un alto *numero di Reynolds*. La turbolenza spesso è generata dall'instabilità di un flusso laminare che si ha quando il *numero di Reynolds* diventa troppo alto. Le instabilità sono legate all'interazione dei termini viscosi e quelli non lineari di inerzia nelle equazioni del moto, ma queste interazioni sono molto complesse, in quanto, matematicamente, sono descritte da equazioni differenziali alle derivate parziali, per le quali non esiste una soluzione generale. La *casualità* e la *non linearità*, combinate, rendono le equazioni della turbolenza quasi intrattabili, e ciò fa sì che la *teoria della turbolenza soffra dell'assenza di sufficienti metodi matematici potenti*. Questa è una dei principali problemi non risolti della fisica moderna [7].

Fluttuazioni vorticosi tridimensionali La turbolenza è *rotazionale e tridimensionale*. La turbolenza è inoltre caratterizzata da un alto livello di *fluttuazioni vorticosi*. Per questo motivo la *dinamica della vorticità* ha un ruolo essenziale nella descrizione dei flussi turbolenti. Le *fluttuazioni vorticosi casuali* che caratterizzano la turbolenza non potrebbero mantenersi se le fluttuazioni fossero bidimensionali, dato che un importante meccanismo di *mantenimento della vorticità*, ovvero il *vortex stretching*, è assente nel flusso bidimensionale [7].

Dissipazione I flussi turbolenti sono sempre dissipativi. Lo *sforzo di taglio viscoso* genera un lavoro di deformazione che *incrementa l'energia interna* del flusso, a *discapito dell'energia cinetica* della turbolenza. La turbolenza richiede quindi continuo supporto energetico per sostenere queste perdite viscosi. Se non c'è

energia, la turbolenza decade velocemente. La dissipazione distingue i flussi turbolenti da quelli di onde casuali, i quali sono essenzialmente non dissipativi, anche se spesso sono *dispersivi* [7].

Continuità La turbolenza è un *fenomeno continuo*, governato dalle equazioni della *meccanica dei fluidi*. Anche le *più piccole scale* che si hanno in un flusso turbolento, sono di *diversi ordini più grandi della scala di grandezza molecolare* [7].

I flussi turbolenti sono flussi La turbolenza non è una caratteristica dei fluidi, ma dei *flussi fluidi*. La maggior parte della dinamica della turbolenza è la stessa in tutti i fluidi, sia liquidi che gassosi, se il *numero di Reynolds* della turbolenza è grande abbastanza; le principali caratteristiche dei flussi turbolenti non sono controllate dalle proprietà molecolari dei fluidi, dove si ha la turbolenza. Poiché le equazioni del moto sono non lineari, ogni modello di flusso ha delle caratteristiche uniche, che sono associate alle sue condizioni iniziali ed a quelle al contorno. Non sono note soluzioni generali delle *equazioni di Navier-Stokes*; di conseguenza non sono disponibili soluzioni generali per problemi con flussi turbolenti. Poiché ogni flusso è differente, anche ogni flusso turbolento sarà differente, anche se tutti i flussi turbolenti hanno le principali caratteristiche in comune [7].

Dopo aver visto quindi quali sono le principali caratteristiche che caratterizzano il fenomeno della turbolenza in generale, e quindi anche i flussi turbolenti, si vanno ad analizzare più nello specifico proprio questi ultimi, andando a studiare prima l'origine della turbolenza, che come si è visto deriva dall'instabilità, e poi i flussi turbolenti veri e propri.

2.3.2 Flusso turbolento

Le soluzioni delle equazioni di *Navier-Stokes*, o di modelli ridotti, rappresentano principalmente delle correnti *laminari*. Tuttavia è noto sperimentalmente che tali correnti esistono solo in particolari situazioni, cioè quando le inevitabili perturbazioni sono smorzate. Ma quando il numero di *Reynolds* aumenta, l'effetto smorzante della viscosità del fluido diminuisce e la corrente diviene *turbolenta*. La presenza o meno di una corrente *laminare* viene verificata attraverso un'analisi di *stabilità*, e nel caso di *instabilità*, si parla di *transizione* della corrente, da laminare a turbolenta. Nello studio della corrente turbolenta si utilizza il *metodo statistico*, che permette di studiare le grandezze medie del moto turbolento; in questo modo è possibile ottenere delle relazioni valide per il calcolo delle caratteristiche dello strato limite turbolento e della resistenza d'attrito [2].

Storicamente, le principali caratteristiche del moto del flusso turbolento sono state studiate inizialmente nel caso di correnti in condotti e tubi, in quanto la loro analisi e misura sperimentale risultava più semplice rispetto ai casi di getti, scie e strati limiti liberi turbolenti.

Più recentemente, invece, lo sviluppo dei metodi numerici applicati alla fluidodinamica, e le maggiori potenze di calcolo degli elaboratori elettronici, ha permesso di ottenere nuove informazioni sulla dinamica del moto turbolento; ad oggi, infatti, la simulazione numerica dei flussi turbolenti, rappresenta un potente mezzo di analisi e di predizione. Poiché come si è detto, per verificare se ci si trova in condizione di *flusso laminare*,

o *flusso turbolento*, è necessario effettuare un'analisi di stabilità, per definire eventualmente poi una condizione ed un punto di *transizione*, la descrizione dei flussi turbolenti inizierà proprio da quest'ultimi due aspetti, l'*instabilità* e la *transizione*.

2.3.3 Instabilità e transizione

Sperimentalmente si osserva che, al variare di alcuni parametri, una corrente fluida, sia in un condotto, che lungo una parete, o una scia, o un getto, subisce un drastico cambiamento passando da un moto laminare ad uno turbolento.

Nel caso di un condotto, o tubo, il fenomeno dipende dalla dimensione caratteristica della geometria (es. diametro del condotto), dalla viscosità del fluido e dalla velocità della corrente. Reynolds fu il primo ad osservare che la transizione avviene quando il rapporto fra questi parametri, il *numero di Reynolds*, supera un certo valore critico. Nel caso di un tubo si ha

$$Re_{crit} = \frac{U_{mean} d}{\nu} \approx 2300,$$

dove d è il diametro del condotto, U_{mean} è la velocità media del flusso che scorre nel condotto, e ν è la viscosità cinematica del fluido.

Nel caso di una lamina piana, invece, in assenza di un gradiente di pressione, si ha che

$$Re_{x,crit} = \frac{U_{\infty} x}{\nu} \approx 3.5 \cdot 10^5 \div 10^6,$$

dove x è la distanza del punto che si sta considerando dal bordo d'attacco della lamina piana, U_{∞} è la velocità del flusso indisturbato che investe la lamina, e ν è la viscosità cinematica del fluido.

Nel caso di uno strato limite, inoltre, la posizione del punto di transizione è influenzata anche dal gradiente di pressione. La transizione è accompagnata da un forte incremento dello spessore dello strato limite e dalla variazione della forma del profilo di velocità (media), il fattore di forma su una lamina piana decresce dal valore del profilo di Blasius, da $H = 2.59$ a $H = 1.4$, con un aumento della resistenza d'attrito. In questo caso si ha anche che la variabilità del valore critico dipende dalle influenze esterne allo strato limite.

Da queste osservazioni si deduce che la transizione da laminare a turbolento risulta essere un problema di stabilità del flusso, in quanto a bassi valori di Reynolds l'azione delle forze viscosive è sufficientemente alta per smorzare tutte le perturbazioni, mentre al contrario, al crescere del numero di Reynolds, l'azione smorzante della viscosità non è più sufficiente per impedire la crescita in ampiezza delle perturbazioni, le quali si amplificano ed intensificano, iniziando la transizione alla turbolenza.

Sperimentalmente si è osservato che i primi stadi della transizione sono caratterizzati da piccole oscillazioni bidimensionali; quindi avremo che l'analisi di stabilità più semplice si basa sullo studio bidimensionale di piccole perturbazioni di un profilo di velocità laminare stazionario. Ipotizzando perturbazioni ondose, si individua, per ogni numero d'onda, un corrispondente valore critico del *numero di Reynolds*, risolvendo un problema linearizzato agli autovalori. Il minore dei numeri trovati è il *Reynolds critico* della corrente.

Ci sono però delle differenze fra i casi precedentemente visti. Nel caso di un condotto, il superamento del valore critico implica la presenza di un moto turbolento in tutta la regione dove il flusso è completamente sviluppato. Nel caso di uno strato limite, o di una scia o di un getto, invece, il numero di Reynolds ha carattere locale, in quanto la

lunghezza e la velocità sono i valori presenti soltanto in una particolare posizione, per esempio lo spessore dello strato limite e la velocità esterna. In conseguenza uno strato limite, inizialmente laminare, subisce la transizione a turbolento quando raggiunge localmente le condizioni di instabilità. Per esempio, per una lamina piana, dall'analisi linearizzata della stabilità, si trova la condizione

$$Re_{\delta^*,crit} = \frac{U_\infty \delta^*}{\nu} = 520. \quad (2.37)$$

La scelta di utilizzare come lunghezza di riferimento una quantità integrale (δ^* o θ) permette di evitare le ambiguità connesse con la definizione di spessore dello strato limite δ . Utilizzando poi le relazioni di Blasius per δ e δ^* , dall'equazione (2.37) si ottiene $Re_{x,crit} = 3 \cdot 10^5$. Si avrà quindi, in conclusione, che lungo una parete una parte dello strato limite può essere laminare, seguita poi eventualmente da una parte turbolenta. L'analisi teorica della transizione nello strato limite, ha inoltre evidenziato un aspetto importante, già visto in precedenza: nel limite di fluido inviscido, si rileva che tutti i profili di velocità con un flesso sono sempre instabili (come si è visto in una sezione del paragrafo 2.2).

Poiché i profili di velocità presentano un flesso in presenza di un gradiente avverso di pressione, $dp_e/dx > 0$, si conclude che, ad esempio sul dorso di un profilo alare, nella zona di recupero della pressione, i profili sono sempre instabili, e quindi la transizione avviene prima che lo strato limite laminare raggiunga la condizione di separazione. In prima approssimazione si può dire che sul dorso di un profilo alare la transizione avviene alla stazione di minima pressione. Nel caso di correnti con un *numero di Reynolds* basso, la transizione può avvenire ad una stazione leggermente più a monte, ma sempre prima che avvenga la separazione laminare. Come eccezione a ciò, però, si ha il caso di un profilo alare ad alta incidenza e basso *numero di Reynolds*: il gradiente di pressione subito dopo il punto di minimo della pressione, localizzato presso il bordo di attacco, può essere molto alto. Se lo strato limite laminare supera il punto di minima pressione senza subire la transizione, può separare immediatamente. In questo caso si parla di separazione laminare, generalmente abbinata a condizioni di stallo. Se lo strato di taglio, ovvero lo strato limite che ha separato, separato dalla superficie, diventa turbolento immediatamente a monte del punto di separazione e, crescendo improvvisamente ed abbastanza in spessore, può riattaccarsi alla parete. In questo caso si ha la presenza di bolle di separazione sul bordo d'attacco.

In quest'ultima parte, si è fatto riferimento alla separazione che avviene solo se il flusso rimane laminare, in quanto, come si è visto in precedenza, per il flusso turbolento, cambiando l'andamento del profilo di velocità, essendo lo strato limite molto più spesso ed essendo il flusso molto più energetico a causa delle forti rimescolamenti ed oscillazioni della corrente, il flusso turbolento separa più difficilmente, e sicuramente in punti molto più a valle, rispetto al flusso laminare.

2.3.4 Moto turbolento

La turbolenza è uno stato di moto dei fluidi descritto dalle equazioni di *Navier-Stokes*, nel caso di corrente incompressibile, dalle equazioni (2.38). Quindi è soluzione di queste equazioni, tuttavia la difficoltà risiede nell'estrema complessità di questa ipotetica soluzione (seppur si potrebbe obiettare sulla validità dell'ipotesi di linearità tra sforzi e velocità di deformazione, per la presenza di grandi accelerazioni dovute alla

turbolenza).

$$\nabla \cdot \mathbf{V} = 0 \quad (2.38a)$$

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{V} \quad (2.38b)$$

$$\frac{\partial T}{\partial t} + (\mathbf{V} \cdot \nabla) T = \frac{Ec}{Re} [(\nabla \mathbf{V})^2 + \nabla \mathbf{V} \cdot \nabla^T \mathbf{V}] + \frac{1}{Re Pr} \nabla^2 T \quad (2.38c)$$

con $Pr = c\mu/\kappa$ numero di Prandtl e $Ec = \mathbf{V}^2/c\Delta T$ numero di Eckert

Le principali caratteristiche del moto turbolento sono la presenza di fluttuazioni sia spaziali che temporali, di vorticità, di alti livelli di dissipazione e diffusività, e l'estrema non linearità del moto.

Le *fluttuazioni turbolente*, sia spaziali che temporali, delle quantità del campo fluido (V, p, T, \dots), rendono il moto turbolento *non stazionario e tridimensionale* anche nel caso in cui le condizioni al contorno sono stazionarie e/o bidimensionali. Inoltre le fluttuazioni appaiono irregolari, caotiche e non prevedibili.

Il moto turbolento è altamente *non lineare*, infatti avviene quando il valore del numero di Reynolds, parametro che misura l'importanza dei termini convettivi, non lineari, supera un certo valore critico oltre il quale si ha il processo di transizione: perturbazioni, inizialmente piccole, che evolvono linearmente nelle prime fasi della loro crescita, poi si sviluppano non linearmente. La turbolenza è la fase finale di questo processo non lineare di transizione [2].

La turbolenza è caratterizzata da fluttuazioni tridimensionali di *vorticità*. Se si visualizza una sezione del campo di moto turbolento, esso appare come una collezione di striature (*streaks*), regioni con concentrazioni di sforzi (*strain regions*), vortici (*swirls*) che si deformano, si uniscono e dividono, e ruotano per mutua induzione. Si possono identificare regioni caratterizzate da particelle fluide dotate di vorticità, e quindi velocità angolare, denominate *eddy*¹, le cui dimensioni variano su varie scale (in questo caso il termine *eddy* non indica un vortice, ma piuttosto una regione di fluido caratterizzata da elevata vorticità; infatti la traduzione di *eddy* più vicina all'effettivo significato inglese è *massa vorticoso*). Gli *eddy* più grandi hanno dimensioni dell'ordine di grandezza dell'intera regione turbolenta, lo spessore δ nel caso di uno strato limite. Le dimensioni dei più piccoli, invece, dipendono dal numero di Reynolds, in particolare, maggiore è il Re e minore è la sua dimensione. Il rapporto tra le due scale è dell'ordine del numero di Reynolds, quindi diversi ordini di grandezza. Gli *eddy* di grande scala contengono la maggior parte dell'energia cinetica associata alle fluttuazioni, mentre quelli più piccoli rappresentano la scala alla quale avviene il processo di dissipazione in calore dell'energia meccanica.

Il moto turbolento è contraddistinto da un alto livello di *dissipazione*. Per comprendere questo meccanismo è utile richiamare l'equazione della vorticità ω (2.10)

$$\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + (\mathbf{V} \cdot \nabla) \omega = \omega \cdot \nabla \mathbf{V} + \nu \nabla^2 \omega + \nabla \times \mathbf{f} \quad (2.39)$$

Il termine $\omega \cdot \nabla \mathbf{V}$, che si ottiene applicando l'operatore $\nabla \times$ al termine (non lineare) convettivo delle equazioni di *Navier-Stokes*, è responsabile del meccanismo detto *vortex stretching*. Questo meccanismo, di natura tridimensionale, trasferisce l'energia associata alle fluttuazioni, dalle grandi scale verso le piccole scale attraverso interazioni non

¹Anche per il termine "eddy" non si fanno distinzioni fra singolare e plurale, come per le "riblet".

lineari, fino a quando i gradienti di velocità sono sufficientemente elevati da provocare la dissipazione in calore dell'energia cinetica, per effetto delle azioni viscosse. Questo trasporto di energia dalle grandi scale verso le piccole scale è denominato *cascata inerziale*, o *cascata energetica*, ed è dovuto esclusivamente all'azione delle forze di inerzia. Il moto turbolento è altamente *diffusivo* in quanto la presenza di grandi fluttuazioni implica un tasso di mescolamento (diffusione) di quantità di moto, massa ed energia termica molto più alto rispetto al caso laminare, dove la diffusione è solo dovuta all'agitazione molecolare.

Riassumendo, fluttuazioni stocastiche in una corrente fluida sono turbolente se esse sono contemporaneamente dissipative, vorticosse e non lineari. È possibile introdurre la seguente definizione di *turbolenza*: *un moto fluido caratterizzato da vorticità tridimensionale, non stazionaria che interagisce non linearmente.*

2.3.5 La cascata energetica e le ipotesi di Kolmogorov

Esaminando il flusso turbolento, si è osservato come il suo moto è caratterizzato da ordini di grandezza, delle grandezze caratteristiche, che vanno dalle dimensioni dello strato limite δ a scale molto più piccole, che diventano tanto più piccole, quanto il numero di Reynolds cresce. Si vedrà, in seguito, anche l'importanza dell'*energia cinetica turbolenta*, e dell'*anisotropia* negli sforzi di Reynolds, mentre si va a vedere subito come queste si distribuiscono sulle varie scale. In particolare, i concetti chiave alla base di questa distribuzione, sono la *cascata energetica*, il *vortex stretching*, la *microscala di Taylor* e l'*ipotesi di Kolmogorov*.

Si considera un flusso completamente turbolento, ad alto numero di Reynolds, con una velocità caratteristica U ed una lunghezza caratteristica L . La prima cosa che si deve tenere in considerazione in merito alla *cascata energetica*, è che la turbolenza la si può considerare come composta da *eddy* di diverse dimensioni. Un *eddy* è un *vortice di rimiscelamento di flusso* che si crea all'interno di un flusso turbolento, ed è la struttura che contiene l'*energia cinetica turbolenta* di un flusso, concentrata maggiormente negli *eddy* più grandi, che si trovano nella parte inerziale dello strato limite. Si ha inoltre che gli *eddy* più grandi sono le caratteristiche di più lunga durata di un flusso turbolento [4], in quanto sono le prime che si generano quando un flusso diventa turbolento, ed anche le prime che sono state osservate e studiate, in merito al comportamento di un flusso turbolento. Queste strutture, inoltre, essendo caratterizzate da proprietà ben definite, sono dette *strutture coerenti* del flusso turbolento.

Ogni *eddy*, inoltre, di differente dimensione l , è caratterizzato dall'aver una sua velocità ed un suo tempo caratteristico, funzioni della sua dimensione. Inoltre un *eddy*, pur non avendo una precisa definizione, è caratterizzato da moto turbolento, localizzato in una determinata regione; una regione occupata da un grande *eddy*, può comunque contenere anche *eddy* più piccoli.

Gli *eddy* di scala più grande, sono caratterizzati da una lunghezza di scala, che è confrontabile con quella del problema che si sta considerando, e la loro velocità caratteristica è dell'ordine della *r.m.s.* (*root mean square*) dell'intensità di turbolenza

$$u' \equiv \left(\frac{2}{3} k \right)^{\frac{1}{2}}$$

che è confrontabile con la velocità caratteristica del problema che si considera.

Per quanto riguarda gli *eddy* più piccoli invece, attraverso lo studio della turbolenza e della dissipazione, la quale è causata da questi *eddy*, si ottiene che le strutture di

piccola scala tendono ad essere *isotrope*. Dall'analisi della *turbolenza isotropa*, si arriva ad ottenere che la scala degli *eddy* più piccoli è λ , la quale è chiamata *microscala di Taylor*, in onore di *G.I. Taylor*, che per primo la definì. La *microscala di Taylor* è anche associata con la curvatura delle autocorrelazioni spaziali della velocità [7]. Inoltre la *microscala di Taylor*, essendo quella a cui si hanno gli *eddy* più piccoli, è anche quella a cui avviene la maggior dissipazione.

Tornando invece ad analizzare gli *eddy* più grandi, si nota però che questi sono molto instabili, e si rompono, trasferendo la loro energia a qualche *eddy* più piccolo. Questi *eddy* più piccoli, a loro volta, subiscono un processo simile, trasferendo quindi la loro energia ad *eddy* ancora più piccoli. Questa *cascata energetica*, nella quale l'energia viene quindi trasferita in successione ad *eddy* sempre più piccoli, continua fino a che il numero di Reynolds caratterizzante gli *eddy*, non è tale da rendere il loro moto stabile, laddove la *viscosità molecolare* poi, ha l'effetto di *dissipare l'energia cinetica*. Si ha quindi che *l'energia cinetica media del flusso*, trasferita dagli *sforzi di Reynolds* nello strato superficiale, è convertita prima in *energia cinetica turbolenta (produzione di turbolenza)* e poi in calore (*dissipazione viscosa*) [7].

Si nota quindi l'importanza che la dissipazione occupa alla fine di questa sequenza di processi. Il rapporto di dissipazione ε , è determinato comunque dal primo processo della sequenza, che è il trasferimento di energia dalle scale più grandi, o dagli *eddy*; questi avendo energia dell'ordine di u_0^2 e scala temporale $\tau_0 = l_0/u_0$, danno un rateo di trasferimento dell'ordine di

$$\frac{u_0^2}{\tau_0} = \frac{u_0^3}{l_0}$$

Una rappresentazione schematica del trasferimento di energia, che rappresenta appunto



Figura 2.6: Rappresentazione schematica della cascata energetica a numeri di Reynolds molto alti; nel diagramma si osserva come l'energia viene prodotta nelle strutture più grandi, trasferita progressivamente a strutture sempre più piccole, e da queste infine dissipata; immagine tratta da [6].

la *cascata energetica*, è riportato nella figura 2.6; in questo diagramma viene riportato il processo di trasferimento dell'energia, partendo dalla *produzione*, che avviene per gli *eddy* più grandi, il trasferimento di energia, ad *eddy* sempre più piccoli, per arrivare poi alla dissipazione che si ha per i più piccoli della scala. Nel diagramma, inoltre, sono indicati anche dei range costituenti lo strato limite che verranno di seguito meglio descritti e caratterizzati.

Nel *trasferimento di energia che caratterizza la cascata energetica*, appena vista, ha un ruolo importante anche il fenomeno del *vortex stretching*. Richiamando infatti l'equazione della vorticità, (2.10), si è visto che in questa il termine $\boldsymbol{\omega} \cdot \nabla \mathbf{V}$ rappresenta la

variazione di vorticità, ma non inteso nel termine di una sorgente (positiva o negativa), ma dal punto di vista che la variazione della vorticità è un effetto della contrazione e/o estensione della linea vorticoso. Questo fenomeno è detto appunto *vortex stretching*. Dal punto di vista energetico, invece, il *vortex stretching* è coinvolto nello *scambio di energia*, perché la velocità di deformazione esegue lavoro di deformazione sui vortici che sono *stretchati*, o *stirati*. Si ha infatti che l'energia totale ottenuta da una perturbazione, in una deformazione, è uguale al prodotto delle sue componenti di velocità, per il rapporto di deformazione. Da ciò si ottiene che, quando un vortice si deforma, acquisisce energia nella direzione in cui la sua dimensione cresce, mentre la perde in quella che si riduce [7].

In merito al trasferimento di energia appena descritto, molti aspetti fondamentali rimanevano inizialmente senza risposta, quali quale fosse la scala di *eddy* più piccola, che determinasse poi la dissipazione, oppure se al diminuire della scala degli *eddy*, se le loro velocità e scale temporali rimanessero le stesse, o aumentassero o diminuissero.

Questi ed altri aspetti poco chiari vengono risolti dalla teoria proposta da *Kolmogorov*, che venne proposta in forma di ipotesi. Una conseguenza della teoria, che *Kolmogorov* motivava con le ipotesi, è che sia la velocità, che la scala temporale, diminuiscono al diminuire della scala degli *eddy*.

Le ipotesi di *Kolmogorov*, [6], sono:

Ipotesi di Kolmogorov di isotropia locale Ad un numero di Reynolds sufficientemente alto, le piccole scale di moto turbolento sono statisticamente isotrope.

Prima ipotesi di Kolmogorov di similarità In ogni flusso turbolento ad un numero di Reynolds sufficientemente alto, le medie statistiche dei moti di piccola scala ($l < l_{EI}$), hanno una forma universale che è solamente determinata da ν e ε .

Seconda ipotesi di Kolmogorov di similarità In ogni flusso turbolento ad un numero di Reynolds sufficientemente alto, le medie statistiche dei moti di scala l nel range $l_0 \gg l \gg \eta$, hanno una forma universale, che è solamente determinata da ε , ed è indipendente da ν .

La prima ipotesi riguarda l'isotropia dei moti di piccola scala; in generale gli *eddy* più grandi sono anisotropi e sono influenzati dalle condizioni al contorno del flusso. Kolmogorov ci dice che le distorsioni direzionali delle grandi scale, vanno perduti nel processo caotico di riduzione di scala, nel quale l'energia è trasferita in successione ad *eddy* sempre più piccoli.

La seconda ipotesi (la prima di similarità), invece riguarda quali parametri influenzano lo stato statisticamente universale delle piccole strutture. Nel processo della *cascata energetica*, i due processi dominanti sono il trasferimento progressivo di energia alle piccole scale, e la dissipazione viscosa. Si ha quindi che i parametri importanti sono il rateo di trasferimento per cui le piccole scale ricevono energia da quelle più grandi, e la viscosità cinematica ν . Inoltre il rapporto di dissipazione ε è determinato proprio dal rateo di trasferimento di energia. Di conseguenza si ha l'ipotesi che lo stato statisticamente universale delle piccole scale è determinato da ν e da ε che a sua volta dipende dal rateo di trasferimento di energia. Inoltre la dimensione del range indicato, $l < l_{EI}$, si riferisce ad un *range di equilibrio universale*. In questo range la scala temporale sono confrontabili con quelle del problema globale, in modo tale che i piccoli *eddy*, si possono velocemente adattare a mantenere un equilibrio dinamico con il trasferimento di energia, imposto da quelli grandi. Andando poi ad analizzare le piccole scale nel dettaglio si

arriva alla conclusione che *per le piccole scale, tutti i campi di velocità ad un numero di Reynolds sufficientemente alto, sono statisticamente simili; si ha quindi che loro sono statisticamente identici quando rientrando anche nelle scale di Kolmogorov.*

Il rapporto fra le scale più grandi e quelle più piccole è determinato dalla definizione delle scale di Kolmogorov, e dal ridimensionamento di $\varepsilon \sim \frac{u_0^3}{l_0}$. Si ottiene che, ad alti numeri di Reynolds, le scale di velocità e temporali degli *eddy* più piccoli, sono, come si supponeva, piccole se confrontate con quelle di quelli più grandi.

Come conseguenza del diminuire del rapporto η/l al crescere del Re , a Reynolds sufficientemente alti, c'è un range di scale l che sono molto piccole confrontate con quella del problema, l_0 , ma comunque molto grandi se confrontate, invece, con η ; si ha quindi $l_0 \gg l \gg \eta$. Poiché gli *eddy* in questo range sono più grandi di quelli dissipativi, si può supporre che il loro Reynolds sia più grande, e conseguentemente che il loro moto sia meno affetto dalla viscosità. Da ciò si ottiene la terza ipotesi di Kolmogorov (la seconda di similarità), in merito al range in cui si hanno questi *eddy*.

Da questa, inoltre, introducendo delle opportune scale di grandezza, si può dividere il

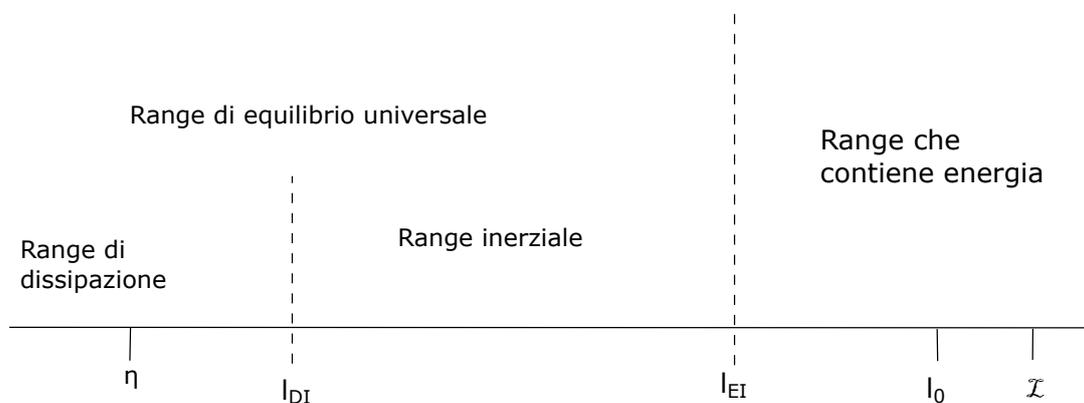


Figura 2.7: Suddivisione delle dimensioni degli *eddy*, in scala logaritmica, per numeri di Reynolds molto alti; vengono mostrate inoltre le varie scale di grandezza ed i range caratteristici; immagine tratta da [6].

range di equilibrio universale in due parti, il *sottorange inerziale* ed il *sottorange dissipativo*. Come il nome richiama, ed in accordo con la seconda ipotesi di similarità, il moto nel sottorange inerziale è determinato dagli effetti inerziali, mentre quelli viscosi sono trascurabili, mentre nel sottorange dissipativo gli effetti viscosi sono significativi, e sono responsabili per tutta la dissipazione. Una rappresentazione schematica di questa divisione è riportata nella figura 2.7, nella quale viene riportata la suddivisione nei vari range appena descritti, caratterizzanti il fenomeno della *cascata energetica*.

2.3.6 Descrizione statistica della turbolenza

Il problema della descrizione del moto turbolento consiste quindi nello sviluppare una rappresentazione del moto complesso, tridimensionale, non stazionario, e caratterizzato da scale di vari ordini di grandezza, in modo da poter essere utilizzate sia per lo studio teorico della turbolenza che per applicazioni tecnologiche.

Gli *eddy* consistono in regioni di fluido irregolari che si comportano in maniera stocastica, sia spazialmente che temporalmente. Per ottenere una descrizione *riproducibile* occorre introdurre una descrizione statistica: la velocità media, e la sua varianza, hanno

un comportamento temporale e spaziale ordinato. La *teoria statistica della turbolenza* descrive le proprietà delle variabili del moto turbolento in base alle loro statistiche, o momenti, e costituisce la base per la descrizione teorica della turbolenza e per lo sviluppo di modelli predittivi.

Sia $u(x, t)$ una variabile stocastica, si definisce insieme una collezione di N realizzazioni indipendenti della variabile u . La statistica più semplice di una variabile stocastica è la *media di insieme*, intesa come meda aritmetica dell'insieme di N realizzazioni,

$$\bar{u}(x, t) \equiv \frac{1}{N} \sum_{n=1}^N u(x, t; n)$$

per $N \rightarrow \infty$ la *media di insieme* è il valore atteso (*expected value*). Il primo effetto dell'operazione di media è quella di rimuovere le fluttuazioni aleatorie. La *media di insieme* rappresenta il momento primo; generalizzando si hanno i momenti di ordine m , definiti come

$$\bar{u}^m(x, t) \equiv \frac{1}{N} \sum_{n=1}^N (u(x, t; n))^m$$

Nel caso in cui la statistica è indipendente dal tempo, cioè $\bar{u}(x)$, si dice che la variabile stocastica $u(x, t)$ è statisticamente stazionario. In questo caso la *media di insieme* può essere ottenuta utilizzando misure della variabile u ottenute per tempi diversi, ogni misura rappresenta una realizzazione indipendente. Lo stesso vale per i momenti di ordine superiore. Dalla definizione di *media di insieme*, si ha

$$\bar{u}(x) = \frac{1}{N} \sum_{n=1}^N u(x, t; n) = \frac{1}{N} \sum_{n=1}^N \int_{t_0+(n-1)\Delta t}^{t_0+n\Delta t} u(x, t) \frac{dt}{\Delta t}$$

dove l'integrale rappresenta una singola realizzazione. Per $T = N\Delta T$, si ottiene la *media temporale*

$$\bar{u}(x) = \frac{1}{T} \int_{t_0}^{t_0+T} u(x, t) dt \quad (2.40)$$

T è l'intervallo di tempo di campionamento e deve essere $T \rightarrow \infty t$, cioè sufficientemente più grande del tempo caratteristico delle oscillazioni.

L'operatore di media, (2.40), è lineare e valgono le seguenti proprietà, con u e v due variabili stocastiche statisticamente stazionarie,

$$\begin{aligned} \overline{u+v} &= \bar{u} + \bar{v} \\ \overline{ku} &= k\bar{u} \quad (k \text{ costante}) \\ \overline{\bar{u}} &= \bar{u} \\ \overline{u - \bar{u}} &= 0 \end{aligned}$$

ma

$$\overline{uv} \neq \bar{u} \bar{v}$$

Generalizzando

$$\begin{aligned} \overline{u^m + v^m} &= \bar{u}^m + \bar{v}^m \\ \overline{ku^m} &= k\bar{u}^m \end{aligned}$$

Inoltre

$$\frac{\overline{\partial u^m}}{\partial x} = \frac{\partial \overline{u^m}}{\partial x}$$

$$\overline{\int u^m dx} = \int \overline{u^m} dx$$

2.3.7 Decomposizione di Reynolds

Considerato un moto turbolento statisticamente stazionario, è possibile scomporlo in un moto medio e nelle fluttuazioni, intese come deviazione del segnale istantaneo dal valor medio, mediante la *decomposizione di Reynolds*; considerate le componenti cartesiane della velocità, possiamo scrivere questa come

$$u_i(\mathbf{x}, t) = U_i(\mathbf{x}) + u'_i(\mathbf{x}, t) \quad (2.41)$$

dove abbiamo che

$$U_i(\mathbf{x}) \equiv \overline{u_i}(\mathbf{x}) = \frac{1}{T} \int_{t_0}^{t_0+T} u_i(\mathbf{x}, t) dt$$

è il campo medio stazionario della velocità, mentre le fluttuazioni $u'_i(\mathbf{x}, t)$ sono definite come

$$u'_i(\mathbf{x}, t) \equiv u_i(\mathbf{x}, t) - U_i(\mathbf{x})$$

Dalla definizione di valor medio, e per le proprietà dell'operatore di media, si ha

$$\overline{U_i}(\mathbf{x}) = U_i(\mathbf{x})$$

$$\overline{u'_i} = \frac{1}{T} \int_{t_0}^{t_0+T} [u_i(\mathbf{x}, t) - U_i(\mathbf{x})] dt = U_i(\mathbf{x}) - \overline{U_i}(\mathbf{x}) = 0$$

Analogamente anche pressione e temperatura possono essere scomposte come

$$p(\mathbf{x}, t) = P(\mathbf{x}) + p'(\mathbf{x}, t)$$

$$\mathbf{T}(\mathbf{x}, t) = \Theta(\mathbf{x}) + \mathbf{T}'(\mathbf{x}, t)$$

dove abbiamo rispettivamente che

$$P(\mathbf{x}) \equiv \overline{p}(\mathbf{x}) = \frac{1}{T} \int_{t_0}^{t_0+T} p(\mathbf{x}, t) dt$$

e

$$\Theta(\mathbf{x}) \equiv \overline{\mathbf{T}}(\mathbf{x}) = \frac{1}{T} \int_{t_0}^{t_0+T} \mathbf{T}(\mathbf{x}, t) dt$$

sono i campi medi stazionari di pressione e temperatura, mentre le fluttuazioni $p'(\mathbf{x}, t)$ e $\mathbf{T}'(\mathbf{x}, t)$ sono definite come

$$p'(\mathbf{x}, t) \equiv p(\mathbf{x}, t) - P(\mathbf{x})$$

e

$$\mathbf{T}'(\mathbf{x}, t) \equiv \mathbf{T}(\mathbf{x}, t) - \Theta(\mathbf{x})$$

Mediando nel tempo le equazioni di *Navier-Stokes* si ottengono le equazioni che governano il moto medio. Riscrivendo le equazioni in coordinate cartesiane nella forma

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (2.42a)$$

$$\rho \frac{\partial u_i}{\partial t} + \rho \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j^2} \quad (2.42b)$$

e mediando la prima equazione delle (2.42), si ottiene

$$\frac{\partial \bar{u}_i}{\partial x_i} = \frac{\partial \bar{u}_i}{\partial x_i} = \frac{\partial}{\partial x_i} (U_i + \bar{u}'_i) = 0 \quad (2.43)$$

da cui

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (2.44)$$

Analogamente, mediando ambo i membri della seconda equazione delle (2.42), considerando i singoli termini, si ha

$$\begin{aligned} \frac{\partial \bar{u}_i}{\partial t} &= \frac{\partial U_i}{\partial t} + \frac{\partial \bar{u}'_i}{\partial t} = 0 \\ \frac{\partial}{\partial x_j} (\bar{u}_i u_j) &= \frac{\partial}{\partial x_j} [(U_i + u'_i) + (U_j + u'_j)] = \frac{\partial}{\partial x_j} (U_i U_j + \bar{u}'_i u'_j) \\ \frac{\partial \bar{p}}{\partial x_i} &= \frac{\partial P}{\partial x_i} \\ \frac{\partial^2 \bar{u}_i}{\partial x_j^2} &= \frac{\partial^2 U_i}{\partial x_j^2} \end{aligned}$$

L'equazione per la quantità di moto, per il moto medio, è

$$\rho \frac{\partial}{\partial x_j} (U_i U_j) = -\frac{\partial P}{\partial x_i} + \mu \frac{\partial^2 U_i}{\partial x_j^2} - \rho \frac{\partial}{\partial x_j} (\bar{u}'_i u'_j)$$

la quale, per l'equazione (2.44), può essere riscritta nella forma

$$\rho U_j \frac{\partial U_i}{\partial x_i} = -\frac{\partial P}{\partial x_j} + \frac{\partial}{\partial x_j} (\tau_{ij} - \rho \bar{u}'_i u'_j) \quad (2.45)$$

dove τ_{ij} è lo sforzo viscoso

$$\tau_{ij} = \mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$$

Le equazioni (2.44) e (2.45) sono le *equazioni di Navier-Stokes mediate alla Reynolds* (*Reynolds averaged Navier-Stokes equations* o *RANS*), che governano il moto medio. Esse sono formalmente uguali alle equazioni di Navier-Stokes (2.42) ad eccezione della derivata del termine $-\rho \bar{u}'_i u'_j$. Esso è un tensore i cui termini, le correlazioni $R_{ij}(\mathbf{X}, t) = -\rho \bar{u}'_i u'_j$, rendono conto degli effetti delle fluttuazioni sul moto medio. In analogia agli sforzi viscosi τ_{ij} che tengono conto degli effetti dell'agitazione molecolare sul moto delle particelle fluide, i termini R_{ij} rappresentano gli *sforzi turbolenti* medi, esercitati dalle fluttuazioni turbolente sul campo di moto medio. Il tensore R_{ij} è detto *tensore degli*

sforzi di Reynolds.

La correlazione $-\rho \overline{u'_i u'_j}$ rappresenta il trasferimento della componente della quantità di moto totale lungo la direzione i (o j) per effetto delle fluttuazioni lungo la direzione j (o i). Ad esempio per $i = 1$ e $j = 2$

$$\rho \overline{(U + u')v'} = \rho \overline{Uv'} + \rho \overline{u'v'} = \rho \overline{u'v'}$$

Nella regione interessata dal moto turbolento gli sforzi di Reynolds $R_{ij} \gg \tau_{ij}$, ad eccezione delle zone adiacenti ad una parete, dove le fluttuazioni turbolente sono nulle ed i gradienti della velocità media alti (come visto nel paragrafo 2.2.4). Il tensore di Reynolds è simmetrico, i termini fuori diagonale ($i \neq j$) rappresentano degli sforzi di taglio, mentre i termini diagonali ($i = j$) sono degli sforzi normali che contribuiscono ad aumentare la pressione media.

In maniera analoga per l'equazione dell'energia (2.9c), scritta in coordinate cartesiane, e trascurando il termine Φ

$$\rho c \left\{ \frac{\partial}{\partial t} (\Theta + T') + \frac{\partial}{\partial x_i} [(U + u'_i)(\Theta + T')] \right\} = \kappa \frac{\partial^2}{\partial x_i^2} (\Theta + T') \quad (2.46)$$

mediando ambo i membri, si ottiene poi

$$\rho c U_i \frac{\partial \Theta}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\kappa \frac{\partial \Theta}{\partial x_i} - \rho c \overline{u'_i T'} \right) \quad (2.47)$$

dove il vettore $-\rho c \overline{u'_i T'}$ rappresenta il contributo delle fluttuazioni turbolente al flusso di calore.

Per un moto turbolento, statisticamente stazionario, si hanno quattro equazioni indipendenti, l'equazione scalare (2.44) e l'equazione vettoriale (2.45). Tuttavia oltre alle quattro incognite \mathbf{U} e P , si hanno anche sei termini del tensore di Reynolds R_{ij} . Il sistema non è quindi chiuso. Per poter risolvere le equazioni di Reynolds occorre introdurre delle relazioni aggiuntive per la chiusura del sistema. Il tentativo di derivare delle equazioni di conservazione per le componenti R_{ij} comporta la comparsa di correlazioni triple nelle equazioni degli sforzi di Reynolds. Analogamente delle equazioni di conservazione per le correlazioni triple porta ad avere delle correlazioni di ordine superiore di incognite. Il problema dunque persiste a tutti i livelli di correlazione e rappresenta il *problema della chiusura* per il calcolo dei flussi turbolenti [2].

2.3.8 Canale piano turbolento

Considerata la corrente in un canale piano di altezza $2H$ (vedi figura 2.8), per

$$Re_H = \frac{U_m H}{\nu} > 1800$$

dove U_m è la velocità media, si ha una corrente turbolenta completamente sviluppata. Abbiamo infatti, che un flusso che scorre in un canale piano, assume un andamento ben definito, quando ci si trova nelle condizioni che definiscono il range di turbolenza. Prendendo in considerazione un caso reale, ovvero una misurazione sperimentale, si ha che, inizialmente, all'ingresso del canale, c'è una prima zona di convergenza, dove il flusso entra nel canale, e poi, risentendo dell'effetto dello strato limite del canale, viene indirizzato, leggermente accelerato e guidato all'interno del canale. Dopo di che, dopo un primo tratto di adeguamento e di transizione, ovvero dove il flusso mantiene una zona

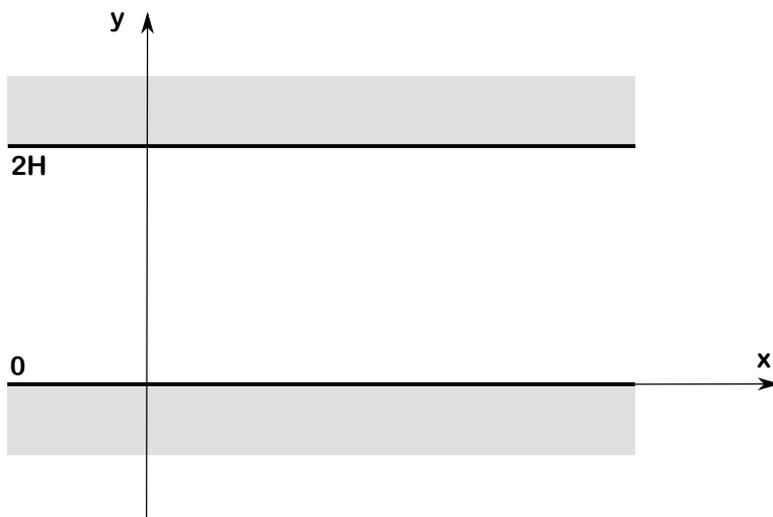


Figura 2.8: Esempio di configurazione di canale piano, in cui si ha una corrente turbolenta che scorre al suo interno.

di cuore potenziale a velocità circa costante lungo la direzione longitudinale, si ha che questa si riduce sempre di più, per effetto dei gradienti di velocità e dei rimescolamenti, fino a che si arriva ad una zona di flusso completamente sviluppato, e completamente turbolento. Nel canale, infatti, gli sforzi d'attrito dello strato limite, vanno velocemente ad erodere la zona di cuore potenziale del flusso, fino a che tutto il flusso diventa di rimescolamento, e turbolento; in particolare tutto il flusso diventerà flusso di *strato limite*. A questo punto, si parla di flusso congelato, o completamente sviluppato. Dopo la prima zona, infatti, si ha che il flusso nel canale assume una velocità, massima al centro del canale, costante all'asse, ovvero che non diminuisce lungo tutto il canale e caratterizzata da un profilo di velocità, che in forma adimensionale, è costante per tutto il canale, ovvero assume sempre la stessa forma. Per il profilo di velocità adimensionale si intende che la velocità è adimensionalizzata rispetto a quella al centro del canale, e la coordinata radiale è adimensionalizzata rispetto alla dimensione trasversale del canale; per questo motivo si parla di flusso congelato, in quanto il suo comportamento in direzione longitudinale non varia più dopo il transitorio iniziale.

Dopo la prima zona di transizione, si ha quindi la zona di flusso congelato, che è quella principale, che caratterizza il flusso in un canale, ed è quella che si deve considerare quando si parla appunto di flusso in un canale piano. Inoltre il profilo di velocità che si ha dopo la transizione, rimane lo stesso per tutto il canale; è possibile osservare un possibile profilo di velocità che si può avere in un canale piano, per un flusso turbolento, in situazione di flusso congelato, in figura 2.9.

Assumendo che la turbolenza sia non omogenea solo lungo la direzione y , si ha un moto statisticamente stazionario ed unidimensionale, in quanto le derivate lungo le direzioni x e z e la derivata nel tempo sono nulle. Il moto medio è un flusso parallelo $U(y)$ e le condizioni al contorno sono:

$$\begin{aligned} y = 0 : \quad U = V = 0 \quad \text{e} \quad u' = v' = w' = 0 \\ y = 2H : \quad U = V = 0 \quad \text{e} \quad u' = v' = w' = 0 \end{aligned}$$

Inoltre per il flusso nel canale piano, andiamo a considerare le ipotesi di *incompressibilità*, *stazionarietà* e *unidimensionalità*. Avremo infatti che i termini di U lungo y e z

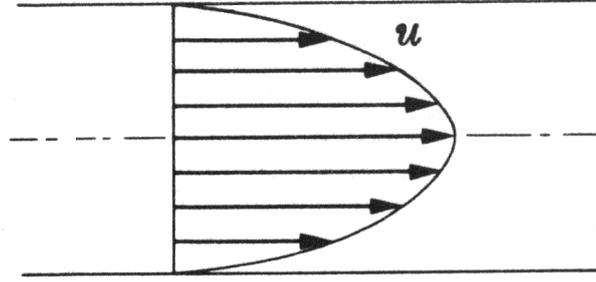


Figura 2.9: Profilo di velocità di un flusso che scorre in un canale piano, in particolare il profilo di velocità di un flusso turbolento, completamente sviluppato, in condizione di flusso congelato.

saranno nulli nel canale a livello di grandezze medie; saranno presenti solamente come fluttuazioni dovute alla turbolenza o ai flussi di spigolo.

Volendo poi analizzare il flusso nel canale nel dettaglio, andiamo a considerare le equazioni di conservazione. Per prima consideriamo l'equazione di continuità,(2.44), e considerando le velocità longitudinali nulle, otteniamo:

$$\frac{\partial U}{\partial x} = 0 \quad (2.49)$$

Dopo di che, considerando l'equazione del bilancio della quantità di moto, l'equazione (2.45), poiché $U(y)$ e per le condizioni di aderenza a parete, si ha

$$\frac{\partial V}{\partial y} = 0 \Rightarrow V = 0$$

Le equazioni di Reynolds, in particolare la (2.45), lungo x ed y si riducono ad un sistema di due equazioni per le variabili del campo medio U e P

$$-\frac{1}{\rho} \frac{\partial P}{\partial x} = \frac{d}{dy} \left(\overline{u'v'} - \nu \frac{dU}{dy} \right) \quad (2.50a)$$

$$-\frac{1}{\rho} \frac{\partial P}{\partial y} = \frac{d}{dy} (\overline{v'^2}) \quad (2.50b)$$

Integrando la (2.50b) lungo y si trova che attraverso la sezione del canale

$$P + \rho \overline{v'^2} = \text{cost} = P_w(x)$$

dove $P_w(x)$ è la pressione lungo le pareti, dove le fluttuazioni turbolente sono nulle. Inoltre se ne consegue che

$$\frac{\partial P}{\partial x} = \frac{dP_w(x)}{dx}$$

Nell'equazione (2.50a) il termine a sinistra è funzione di x , mentre il termine a destra è funzione della sola variabile y , quindi ambo i membri sono costanti, ed in particolare

$$\frac{dP_w}{dx} = \text{cost}$$

Indicando con $\tau = \mu dU/dy$ lo sforzo viscoso, l'equazione (2.50a) può essere riscritta nella forma

$$\frac{d}{dy} (\tau - \overline{\rho u'v'}) = \frac{dP_w}{dx}$$

ed integrata attraverso il canale si ottiene, per le condizioni al contorno,

$$\tau(2H) - \tau(0) = 2H \frac{dP_w}{dx}$$

Per la simmetria del campo di moto $\tau(0) = -\tau(2H) = \tau_w$, e si ricava che la caduta di pressione lungo il canale è bilanciata dal gradiente degli sforzi viscosi a parete

$$-\frac{dP_w}{dx} = \frac{\tau_w}{H}$$

Integrando poi l'equazione (2.50a), da parete fino ad una generica posizione y , si ottiene

$$\mu \frac{dU}{dy} - \overline{\rho u'v'} = \tau_w \left(1 - \frac{y}{H}\right)$$

Introducendo infine le *variabili di parete*, adimensionali, abbiamo

$$y^+ = \frac{yu_\tau}{\nu} \quad U^+ = \frac{U}{u_\tau} \quad \overline{u'v'}^+ = \frac{\overline{u'v'}}{u_\tau^2}$$

dove $u_\tau = \sqrt{|\tau_w/\rho|}$ è la velocità di attrito; abbiamo quindi che la relazione precedente, in termini di *variabili di parete*, può essere espressa in forma adimensionale

$$\frac{dU^+}{dy^+} - \overline{u'v'}^+ = 1 - \frac{y^+}{Re_\tau} \quad \left(Re_\tau = \frac{u_\tau H}{\nu}\right) \quad (2.51)$$

La somma degli sforzi viscosi e turbolenti varia linearmente con $0 \leq y^+ \leq yH/\nu = Re_\tau$

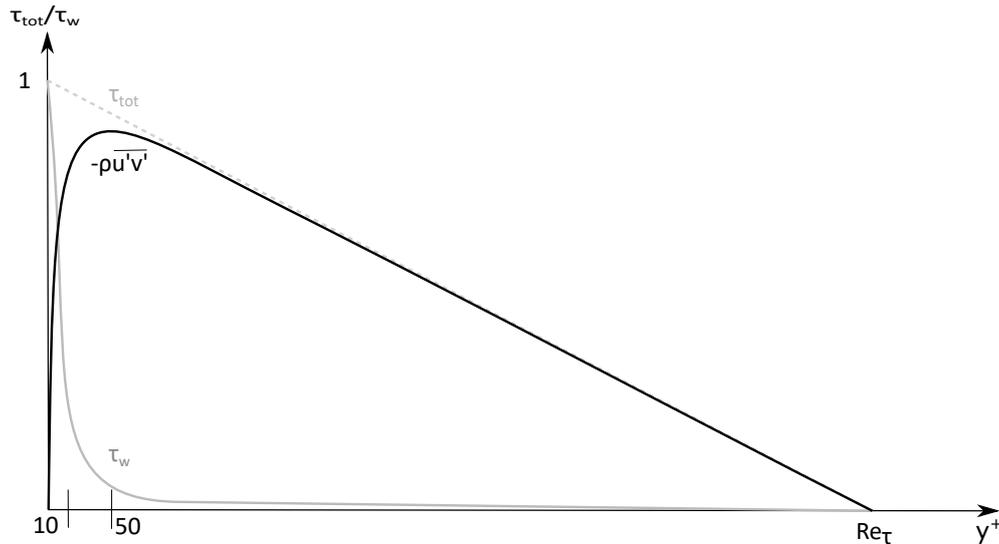


Figura 2.10: Andamento qualitativo degli sforzi turbolenti e viscosi in un canale piano, riportati in funzione della distanza espressa in unità di parete

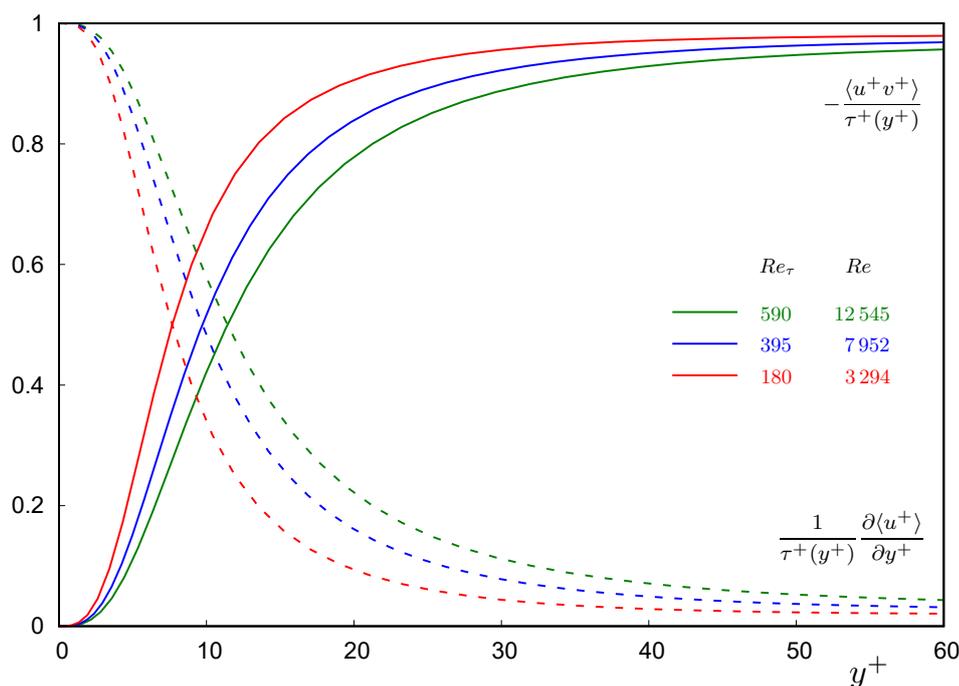


Figura 2.11: Andamento dei contributi relativi degli sforzi viscosi e di Reynolds allo sforzo totale, riportati in funzione della distanza espressa in unità di parete. Sono riportati tre diversi andamenti, per Re_τ 180, 395 e 590. I dati riportati sono tratti dalle DNS di Kim, Moser, Mansour, [8].

e si annulla nella mezzera. Dalle simulazioni numeriche e dalle analisi sperimentali risulta che il termine viscoso è importante solo in prossimità della parete $y^+ \approx 1$, mentre per $y^+ > 10$ esso è trascurabile rispetto allo sforzo turbolento, che è dominante nella maggior parte del canale. Nelle figure 2.10 e 2.11 sono riportati gli andamenti degli sforzi sopra descritti. In particolare nella 2.10 viene riportato il confronto fra gli andamenti, dal punto di vista qualitativo, riportando il contributo di questi allo sforzo totale, mostrando anche come questo diminuisca all'allontanarsi dalla parete; nella 2.11, invece, vengono riportati gli andamenti dei contributi di ognuno dei due sforzi, rapportati rispetto a quello totale, per alcuni Re_τ , dove si può osservare come, al variare di questo, cambi il comportamento dei contributi relativi, i quali hanno zone di predominanza leggermente diverse al variare del Re_τ , così come cambia anche il punto di intersezione delle curve, ovvero il punto, che in questo caso indica la distanza da parete, dove i due contributi hanno lo stesso peso per lo sforzo totale. Questi dati sono presi da simulazioni DNS, effettuate da Kim, Moser, Mansour, [8].

Volendo riassumere il comportamento di un flusso turbolento in un canale piano, abbiamo che l'equazione di continuità dice che la velocità si mantiene costante lungo l'asse del canale; questo conferma, come detto in precedenza, che il flusso è congelato e non cambia per tutto il canale; dice inoltre che non ci sarà accelerazione lungo l'asse x. Il primo termine dell'equazione della quantità di moto dice che la pressione si mantiene costante lungo le direzioni (entrambe in quanto il comportamento è lo stesso sia lungo y che lungo z) perpendicolari l'asse del condotto; ciò risulta essere particolarmente importante quando si vuole effettuare delle misure sperimentali in un canale piano, in quanto è impossibile misurare la pressione lungo l'asse del condotto, in quanto la

presenza di una sonda in quel punto disturberebbe notevolmente il flusso alterandolo; grazie alla costanza della pressione, invece, si ha che la pressione in un punto all'asse del canale si mantiene costante in direzione longitudinale fino alle pareti e quindi può essere lì misurata. Il secondo termine dell'equazione di quantità di moto dice invece che le forze di pressione uguagliano quelle di attrito; non essendoci accelerazioni del flusso, non ci sono forze di inerzia. Inoltre gli sforzi d'attrito, sono proporzionali ai gradienti di velocità, e saranno quindi molto più intensi in prossimità delle pareti. Si ha inoltre che le perdite di pressione sono proporzionali agli sforzi d'attrito. Nel canale la velocità si mantiene costante; tuttavia le perdite causano la diminuzione della pressione del flusso. Ci saranno perdite di carico lungo tutto il canale. È possibile infine esprimere le perdite di pressione nel canale in forma adimensionale. Queste vengono indicate con il termine λ , che è dato da

$$\lambda = \frac{\frac{\Delta p}{\Delta x}}{\frac{1/2\rho U^2}{D}} \quad (2.52)$$

Dove D indica il diametro idraulico del canale, e nel caso di un canale reale, può essere calcolato come $D = 4 * A/P$ con $A = B * H$ e $P = 2 * (B + H)$, dove B e H sono rispettivamente la larghezza e l'altezza di un canale piano.

A questo punto, dopo aver visto a livello generale il flusso nel canale piano, ed avendo parlato, in merito degli sforzi, del fatto che alcuni sono predominanti vicino alle pareti, mentre altri lo sono muovendoci verso il centro del canale, si va a vedere maggiormente nel dettaglio lo strato limite di un flusso che scorre in un canale piano. Nello specifico si vanno a vedere, maggiormente nel dettaglio, caratterizzandole anche con le loro equazioni caratteristiche, le zone che costituiscono lo strato limite turbolento, viste in precedenza nel paragrafo 2.2.4. Qui sono descritte per un flusso che scorre in un canale, ma, in particolare le equazioni caratteristiche, valgono anche per uno strato limite turbolento libero, soprattutto vicino alla parete.

La regione interna e la legge di parete per la velocità

Nella regione adiacente la parete, dove $y^+ = O(1)$, il termine $y^+/Re_\tau \ll 1$, quindi l'equazione 2.51 si semplifica

$$\frac{dU^+}{dy^+} - \overline{u'v'^+} = 1 \quad (2.53)$$

Lo sforzo totale è costante ed uguale al valore τ_w . In questa regione non si ha una dipendenza esplicita dalla geometria (Re_τ), e per questo motivo è denominata *regione di equilibrio universale*. E le considerazioni ottenute nel caso del canale, sono quindi, come già accennato in precedenza, estendibili a tutti i casi in cui si ha una parete.

Nella zona dove $\frac{dU^+}{dy^+} \gg -\overline{u'v'^+}$, per $y^+ \ll 10$ si ha la regione detta *sottostrato viscoso*, o *viscous sublayer*, dove il profilo di velocità è dato da

$$\frac{dU^+}{dy^+} \approx 1$$

Integrando si ottiene la *legge della velocità nel sottostrato viscoso*

$$U^+ = y^+ \quad (2.54)$$

Nella zona, invece, dove $1 \ll y^+ \ll Re_\tau$, che si estende dal sottostrato viscoso fino ad $y/H \approx 0.1$, lo sforzo turbolento è dominante, e dalla relazione (2.53) si ha che

$-\overline{u'v'}^+ \approx 1$, cioè che lo sforzo turbolento è costante. Per ottenere l'andamento della velocità in questa regione è utile osservare che la corrente è completamente definita dai parametri ρ , ν , H e dP_w/dx (o u_τ), con i quali è possibile formare due gruppi adimensionali: y/H e Re_τ ; da ciò è lecito supporre

$$U = u_\tau f\left(\frac{y}{H}, Re_\tau\right)$$

dove f è una funzione adimensionale, oppure, definendo $\delta_v = \nu/u_\tau$ in modo che $Re_\tau = Hu_\tau/\nu = (y/\delta_v)/(y/H)$, si ha

$$U = u_\tau f\left(\frac{y}{\delta_v}, \frac{y}{H}\right)$$

da cui

$$\frac{dU}{dy} = \frac{u_\tau}{y} F\left(\frac{y}{\delta_v}, \frac{y}{H}\right)$$

Prandtl ipotizzò che per valori piccoli di y/H , e quindi nella regione di parete, la velocità non dipenda da H , avendo quindi

$$\frac{dU}{dy} = \frac{u_\tau}{y} F_I\left(\frac{y}{\delta_v}\right)$$

Introducendo le variabili di parete, si ha

$$\frac{dU^+}{dy^+} = \frac{1}{y^+} F_I(y^+)$$

Inoltre, essendo gli effetti viscosi ininfluenti, $F_I \approx \text{cost} = 1/\kappa$ e

$$\frac{dU^+}{dy^+} = \frac{1}{\kappa y^+}$$

Si ottiene la *legge logaritmica*

$$U^+ = \frac{1}{\kappa} \log(y^+) + B \quad (2.55)$$

dove κ è la *costante di Von Karman*. A conferma dell'universalità della legge si è trovato sperimentalmente che entrambe le costanti κ e B sono sensibilmente universali, e valgono

$$\kappa = 0.41 \quad B = 5.2$$

Nella figura 2.12, le curve delle leggi 2.54 e 2.55 sono confrontate con dei dati ottenuti numericamente risolvendo le equazioni di *Navier-Stokes* [2]. In particolare nella figura 2.12, vengono riportati i dati di tre simulazioni DNS, ottenuti per diversi valori di Re_τ , ed è possibile osservare come i valori delle simulazioni vadano a seguire molto bene l'andamento delle leggi, sia la prima che la seconda, mentre se ne discostano leggermente nella zona in cui le due curve vanno a raccordarsi. Ciò è tuttavia in linea con le aspettative, in quanto si ha che la regione di raccordo tra le due leggi, quella situata per $y^+ \approx 10 \div 15$ è detta *buffer layer*, come si era anche già visto in precedenza, nel paragrafo 2.2.4, ed è la regione in cui le due leggi valgono entrambe, ma in modo meno netto; ciò spiega il leggero discostarsi dei punti dalle curve in quella zona. Le DNS, da cui sono presi i dati riportati, sono effettuate da Kim, Moser, Mansour, [8].

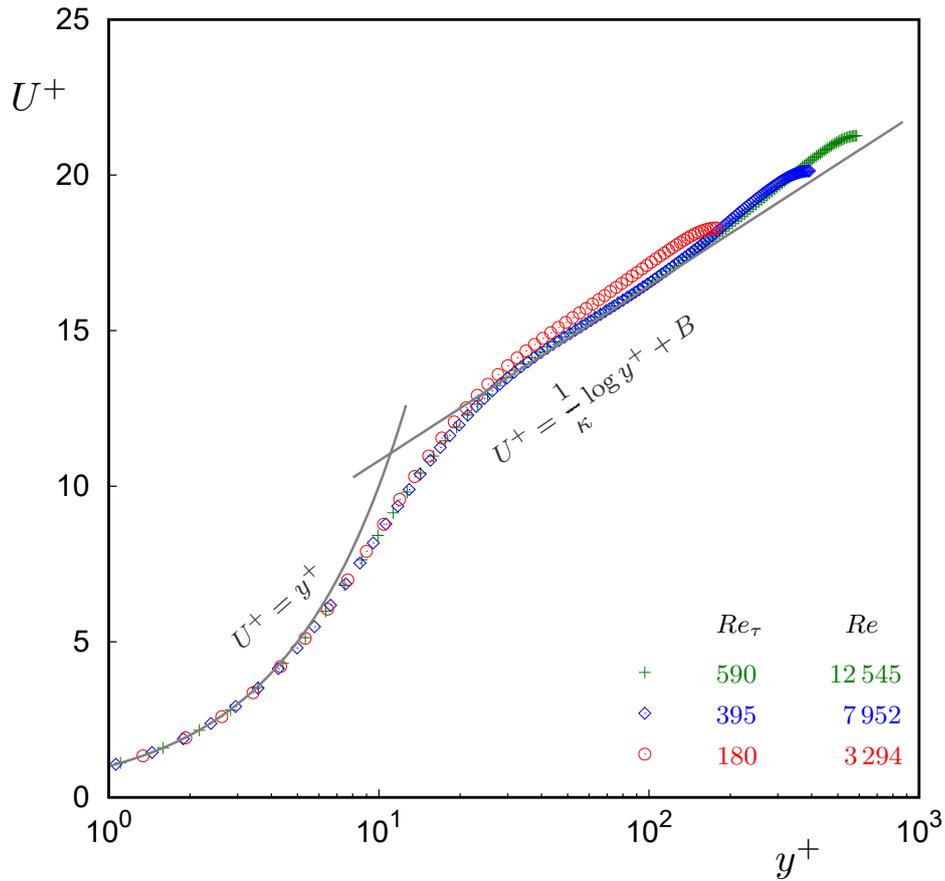


Figura 2.12: Confronto legge universale di parete, lineare e logaritmica, con dati numerici, tratti dalle DNS di Kim, Moser, Mansour, [8], ottenuti per Re_τ 180, 395 e 590.

Regione esterna

Per valori di $y^+ \gtrsim 1000$ la legge *legge logaritmica* perde di validità, in quanto l'ipotesi di y/H piccolo non è più valida. Nella *regione esterna* il parametro y/H diventa importante, mentre si può supporre che il profilo di velocità non dipenda dal parametro y/δ_v . In questo caso si ha

$$\frac{dU}{dy} = \frac{u_\tau}{y} F_O\left(\frac{y}{H}\right)$$

Integrando da y ad H , ponendo $U_0 = U(H)$, si ricava la *legge del difetto di velocità*

$$\frac{U_0 - U}{u_\tau} = F_D\left(\frac{y}{H}\right) \quad (2.56)$$

dove abbiamo la funzione F_D , definita come

$$F_D\left(\frac{y}{H}\right) = \int_y^H \frac{1}{y} F_O\left(\frac{y}{H}\right) dy$$

Questa funzione non è universale, e dipende infatti dalla geometria del problema. La *legge logaritmica*, (2.55), e la *legge del difetto di velocità*, (2.56), sono entrambe valide

in una regione di sovrapposizione tra la regione interna e quella esterna. Imponendo la condizione di raccordo è possibile determinare la forma di F_D

$$\frac{y}{u_\tau} \frac{dU}{dy} = F_I \left(\frac{y}{\delta_v} \right) = F_O \left(\frac{y}{H} \right) = \frac{1}{\kappa}$$

poiché F_I e F_O devono essere costanti. Integrando poi si arriva alla legge

$$\frac{U_0 - U}{u_\tau} = -\frac{1}{\kappa} \log \left(\frac{y}{H} \right) + B_1 \quad (2.57)$$

dove la costante B_1 dipende dal tipo di corrente. Nel caso del canale piano si ha $B_1 = 0.2 \div 0.7$.

Coefficiente di attrito

Nota la legge del profilo di velocità, (2.57), è possibile ottenere l'espressione del coefficiente di attrito in funzione del numero di Reynolds $Re_0 = HU_0/\nu$. Utilizzando le espressioni (2.57) e (2.55), abbiamo

$$\begin{aligned} \frac{U_0 - U}{u_\tau} + \frac{U}{u_\tau} &= -\frac{1}{\kappa} \log \left(\frac{y}{H} \right) + B_1 + \frac{1}{\kappa} \log y^+ + B \\ \frac{U_0}{u_\tau} &= \frac{1}{\kappa} \log \left[Re_0 \left(\frac{u_\tau}{U_0} \right) \right] + B + B_1 \end{aligned}$$

e dalla definizione del coefficiente di attrito $c_f = \tau_w/(1/2\rho U_0^2) = 2(u_\tau/U_0)^2$, si ottiene la relazione (implicita) del $c_f(Re_0)$

$$\sqrt{\frac{2}{c_f}} = \frac{1}{\kappa} \log \left[\frac{Re_0}{2} \sqrt{\frac{c_f}{2}} \right] + (B + B_1) \quad (2.58)$$

2.3.9 Metodo di Clauser

Nel caso di un flusso turbolento, come detto nel paragrafo 2.2.4 parlando dei diversi sottostrati che compongono lo strato limite, si ha che l'*inner layer*, e più nello specifico la *viscous sublayer* assume dimensioni minori rispetto al caso di flusso laminare. Ciò porta che, soprattutto nel caso di misure sperimentali, ma anche nel caso di simulazioni numeriche, i dati di questo sottostrato, diventano difficili da ottenere. A causa di ciò, quindi, non è possibile conoscere la velocità, ed il suo andamento, vicino alla parete. Tuttavia questo è un dato fondamentale, se si vuole conoscere lo *sforzo d'attrito* che quel flusso ha, in quanto si ha, per la relazione di Newton, che lo *sforzo d'attrito a parete* è dato da

$$\tau_w = \mu \left(\frac{\partial u}{\partial y} \right)_w$$

dove il pedice w indica appunto le misure effettuate vicino alla parete.

Laddove ciò non è possibile, come detto a causa della grande difficoltà di ottenere dati vicino alla parete in modo accurato, è possibile comunque conoscere il valore dello *sforzo d'attrito a parete*, utilizzando il *metodo* (o *formula*) di Clauser.

Questa formula si basa sull'andamento della velocità nel *log. law layer*, e sfrutta proprio

i dati ottenuti in questa zona, per calcolare lo *sforzo d'attrito a parete*. Come prima cosa, si introduce il *coefficiente d'attrito adimensionale*, C_f , che è definito come

$$C_f = \frac{\tau_w}{1/2\rho U^2}$$

Noto questo, si ha, invertendo la formula, che lo *sforzo d'attrito a parete* è dato da

$$\tau_w = \frac{1}{2}\rho U^2 C_f$$

Dopo di che, si va a richiamare la formula dell'andamento della velocità nel *log. law layer*, l'equazione (2.55)

$$U^+ = \frac{1}{\kappa} \log(y^+) + B$$

Ora, nell'equazione appena riportata, si va ad esplicitare i termini funzioni di unità di parete, e poi si vanno a moltiplicare entrambi i termini esplicitati per il rapporto $\frac{U_{\text{inf}}}{U_{\text{inf}}}$

$$\frac{u}{u_\tau} \frac{U_{\text{inf}}}{U_{\text{inf}}} = \frac{1}{\kappa} \left(\log \frac{y u_\tau U_{\text{inf}}}{\nu U_{\text{inf}}} \right) + B \quad (2.59)$$

A questo punto si vanno a separare i termini al secondo membro, utilizzando le proprietà del logaritmico, e dopo di che si vanno a riordinare i termini dell'equazione

$$\frac{u}{U_{\text{inf}}} = \frac{u_\tau}{U_{\text{inf}}} \left[\frac{1}{\kappa} \left(\log \frac{y U_{\text{inf}}}{\nu} + \log \frac{u_\tau}{U_{\text{inf}}} \right) + B \right] \quad (2.60)$$

Ora, si può riscrivere il termine $\frac{u_\tau}{U_{\text{inf}}}$, dalla definizione di u_τ come

$$\frac{u_\tau}{U_{\text{inf}}} = \sqrt{\frac{\tau_w}{\rho U_{\text{inf}}^2}}$$

Poi, dalla definizione del coefficiente d'attrito C_f sopra riportata, si ha che

$$\frac{u_\tau}{U_{\text{inf}}} = \sqrt{\frac{C_f}{2}}$$

Andando quindi a sostituire i termini appena trovati nell'equazione (2.60), si ottiene

$$\frac{u}{U_{\text{inf}}} = \sqrt{\frac{C_f}{2}} \left[\frac{1}{\kappa} \left(\log \frac{y U_{\text{inf}}}{\nu} + \log \sqrt{\frac{C_f}{2}} \right) + B \right] \quad (2.61)$$

Nell'equazione appena ottenuta si ha che C_f rappresenta un valore parametrico libero. L'equazione ottenuta, la (2.61), è la *relazione di Clauser*, ed attraverso questa è possibile ottenere il C_f e da questo il τ_w di un flusso per il quale non si conosce la velocità vicino alla parete, ma ad una distanza maggiore, ovvero nella *log. law layer*.

Nota l'equazione appena ricavata, il procedimento è il seguente:

1. Si calcolano, a partire dai dati misurati (u e la y a cui sono stati ottenuti), i rapporti presenti nell'equazione, $\frac{u}{U_{\text{inf}}}$ e $\log \frac{y U_{\text{inf}}}{\nu}$ e li si vanno a riportare su di un grafico;
2. Ora, facendo variare il parametro libero dell'equazione (2.61), il C_f , si ottengono e si diagrammano diverse curve a pendenza variabile, funzione di C_f appunto;

3. Fra tutte le curve riportate si andrà ad individuare quella che meglio interpola la rappresentazione dei dati misurati; il C_f utilizzato per ottenere quella curva, sarà quello del problema che si sta analizzando;
4. Dal C_f ottenuto, si può poi ricavare il τ_w , e da esso tutte le grandezze espresse in termini di unità di parete.

Capitolo 3

Metodi di riduzione della resistenza d'attrito dello strato limite

3.1 Storia della riduzione della resistenza fluidodinamica

Il problema della *riduzione della resistenza d'attrito* in un *flusso turbolento*, o generalmente per un qualsiasi flusso, risulta essere ad oggi ancora uno dei più importanti fra quelli della fluidodinamica, sia dal punto di vista teorico, che da quello pratico. In particolare dal punto di vista teorico, per poter affrontare al meglio questo problema, è necessario conoscere e caratterizzare al meglio il fenomeno della turbolenza, anche se ad oggi questo rimane uno dei problemi fondamentali irrisolti della fluidodinamica; dal punto di vista pratico, invece, l'importanza della *riduzione della resistenza d'attrito* è data dal fatto che meno resistenza significa minori consumi o minori perdite, in funzione dell'ambito di interesse, e quindi prestazioni migliori e più risparmio.

Lo scopo di *ridurre la resistenza d'attrito* nello *strato limite turbolento* nasce alla fine degli anni 70, diventando importante ai fini pratici dello studio della fluidodinamica, e lo è ancora oggi.

L'approccio generale che si è avuto nell'affrontare questo problema, è stato quello di andare ad intervenire in diversi modi sul flusso, al fine di modificare il comportamento del *flusso a parete*, ed ottenere da ciò una diminuzione della resistenza. Nell'articolo *Active and passive turbulent boundary-layer drag reduction* di T.C. Corke et al [9], si osserva come, prendendo in esame i dati delle simulazioni *Kline et al.*, come il flusso turbolento, pur essendo caratterizzato da un moto disordinato, tende ad assumere, sorprendentemente, delle strutture di moto ben organizzate nelle regioni vicino alla parete. Queste strutture portano a loro volta alla formazione di *strisce* a bassa velocità vicino alla parete, che interagiscono con la parte esterna dello strato limite, attraverso un processo di graduale *separazioni, oscillazioni improvvise, rimescolamenti e dissipazioni*. Questa sequenza viene ora semplicemente conosciuta come *rimescolamenti* e si sa che svolge un ruolo importante nella *produzione* di turbolenza vicino alle pareti.

Negli esperimenti presi in considerazione nell'articolo, si è osservato inoltre come, per il *numero di Reynolds* utilizzato, la regione di *strisce di parete* scoperta da *Kline et al.*, coincide con la regione di massima produzione di turbolenza, e ciò richiede ulteriori indagini in merito ai *fenomeni vicino alla parete*.

Data l'importanza della produzione di turbolenza, un primo argomento è stato come la

frequenza di questa produzione dipende dalle scale caratteristiche e dalle velocità dello strato limite. In particolare, le *strisce di parete* passano attraverso le diverse scale del fenomeno, seguendo le variabili u_τ e ν .

Anche poi per le strutture di grande scala nello strato di parete dei flussi turbolenti, si è ritenuto che avessero un ruolo chiave nel generare aumento di resistenza e rateo di trasferimento di calore. Anche questo aspetto è stato poi studiato ed analizzato maggiormente nel dettaglio, in diversi studi, al fine di caratterizzare queste strutture di grande scala. Si è mostrato come le strutture di grande scala hanno la forma di vortici a ferro di cavallo. Queste sono dinamicamente importanti, ed in diversi studi (*Robinson et Adrian* si descrive come questi vortici a ferro di cavallo sono responsabili, in larga parte, sia dei rimescolamenti a bassa velocità, e sia degli scambi ad alta velocità, due fenomeni che contribuiscono all'aumento della produzione di sforzi di *Reynolds* e di conseguenza al trasposto normale a parete di quantità di moto. Da simulazioni numeriche dirette, emerge inoltre come la nascita dei vortici, sia quelli di *upstream* che di *downstream*, sia un processo di autogenerazione, che appare molto robusto con l'emergere di nuovi vortici all'aumento dell'alto livello di simulazione artificiale dato dalle piccole scale di turbolenza imposte alle simulazioni. L'autogenerazione dei vortici è quindi un aspetto chiave del processo di generazione della turbolenza di parete.

A questo punto dello studio, si evidenzia come, il primo modo con cui si va a cercare di ridurre la resistenza d'attrito, sia quello di andare ad intervenire direttamente sul flusso, attraverso il suo controllo. In particolare l'approccio generalmente utilizzato per la riduzione della resistenza è stato cercare di intervenire sulle varie strutture coerenti di moto, come si è già detto. In particolare alcuni di questi approcci sono passivi, mentre altri sono attivi. Un esempio di questi è quello di andare ad intervenire sulle strutture chiamate *large eddy*, ovvero sulle grandi strutture vorticosi presenti in un flusso turbolento. Questi metodi poi, oltre alla parte puramente teorica di riduzione della resistenza, hanno dovuto analizzare, in particolare per quelli attivi che hanno anche un costo in termini di utilizzo, il loro impatto sul caso a cui venivano applicati, e si è cercato infatti di andare a minimizzare l'aggiunta di elementi necessari per modificare il flusso. Un esempio di ciò lo si può trovare nello studio della soluzione *LEBU* [9], sigla che indica *Large Eddy Break Up*, ovvero un sistema per *rompere* le grandi strutture vorticosi, attraverso l'inserimento sulle pareti su cui scorre il flusso di superfici mobili. In questo studio si evidenzia che anche se 4 superfici mobili influenzano di più il flusso, queste producono anche, per la loro presenza, più resistenza della presenza di solo 2 superfici.

L'importanza di conoscere, capire e controllare la turbolenza vicino alla parete in un flusso, al fine di ridurre la resistenza, viene anche evidenziata nell'articolo *Physics and control of wall turbulence for drag reduction* di *John Kim* [10]. In questo articolo si evidenzia come la fisica della turbolenza è responsabile dell'alta *resistenza d'attrito* nello strato limite turbolento. Si discute poi del processo auto sostenente delle strutture turbolente vicino alla parete, dalla prospettiva di controllare questo processo con l'obiettivo di ridurre la *resistenza d'attrito*. Emerge come le parti fondamentali del processo auto sostenente sono lineari, e quindi si parla di sistemi lineari come approccio al controllo dello strato limite. Si mostra come un'analisi di decomposizione ad un singolo valore del sistema lineare ci permette di esaminare approcci differenti per il controllo dello strato limite senza dover sostenere il costo di simulazioni non lineari. I risultati che si ottengono da analisi lineari sono infatti consistenti e validi con quelli che si osservano per simulazioni non lineari, e ciò dimostra la validità delle analisi lineari.

Si parla anche del fondamentale limite di performance che ci si aspetta dall'input di controllo ottimale.

Andando quindi ad approfondire questo articolo [10] maggiormente nel dettaglio, al fine di coglierne gli aspetti fondamentali dello studio che si è svolto, partiamo dall'obiettivo che lo ha reso necessario. Il controllo dei flussi, con conseguente riduzione della resistenza, significa sia importanti benefici finanziari dal punto di vista economico, ma rappresenta anche un motivo di grande interesse per la ricerca fluidodinamica. In particolare, grazie all'uso sempre maggiore della fluidodinamica computazionale, si è potuti fare grandi progressi in questi studi, in quanto la possibilità di avere la *timeline* completa e tridimensionale dei campi di velocità e pressione, rende possibile capire molti più fenomeni legati alla turbolenza. Le simulazioni numeriche sono state anche estremamente utili ed importanti nel testare diverse ipotesi, conducendo abilmente esperimenti numerici mirati, con cui modificare il modo in cui le equazioni di *Navier-Stokes* vengono risolte, in modo da esaminare il ruolo di alcuni meccanismi della turbolenza. Ciò ha permesso quindi lo sviluppo di nuovi metodi di controllo del flusso, facendo emergere nuovi metodi di approccio al problema. In particolare i nuovi approcci, a differenza di quelli precedenti che sono principalmente basati sul capire la fisica all'interno del flusso, incorporano le moderne teorie di controllo nel design di strumenti di controllo.

La scoperta di strutture turbolente ben organizzate ed il capire che queste strutture giocano ruoli importanti nelle dinamiche dello strato limite sono fra i maggiori progressi nella ricerca sullo strato limite turbolento, che si sono ottenuti nelle scorse decadi.

Si è scoperto che le onnipresenti caratteristiche strutturali nella regione vicino a parete nello strato limite turbolento sono quindi *strisce* a bassa ed alta velocità, che consistono principalmente in una modulazione di velocità lungo il flusso. Queste strisce sono create da vortici che si sviluppano lungo il flusso, i quali sono anche approssimativamente allineati nella direzione del flusso. È stato dunque riconosciuto, in gran parte grazie ad indagini numeriche, che sono anche i vortici i responsabili dell'elevata resistenza d'attrito. Questi vortici si trovano principalmente nel *buffer layer* ($y^+ = 10 - 50$), con il loro diametro tipico che è dell'ordine di $d^+ = 20 - 50$. Vi è una forte evidenza che la maggior parte delle regioni ad alto attrito presenti negli strati limite turbolenti sono indotte dai vicini vortici lungo il flusso. Da ciò si ha che una caratteristica comune di tutti i flussi a *resistenza ridotta*, indipendentemente da come questa sia stata ridotta (per esempio mediante l'utilizzo di riblet o pareti oscillanti, per fare alcuni esempi) è l'indebolimento vicino alla parete dei vortici lungo il flusso. Anche la forza delle *strisce* è significativamente ridotta, mentre l'allargamento trasversale medio è incrementato.

Studiando il fenomeno maggiormente nel dettaglio, si osserva come i vortici si formano e si sostengono autonomamente, ovvero indipendentemente dallo strato esterno del flusso, tramite un processo auto sostenente, che coinvolge le *strisce* dello strato di parete, e le instabilità associate a queste. Si ha poi un fenomeno di generazione ciclica, che riguarda gli elementi della turbolenza, in cui i vortici interagiscono con il flusso esterno, mentre le *strisce* si formano a partire dai vortici. Le strisce create in questo meccanismo lineare, sono instabili per piccoli disturbi, ovvero sono linearmente instabili. Inoltre *Schoppa & Hussain* hanno mostrato che le *strisce* sono soggetti a modi di instabilità non normali, riferendosi alla crescita progressiva delle *strisce*, dovuto alla natura non *auto-aggiustante* della linearizzazione delle equazioni di *Navier-Stokes*.

Il riconoscere che un meccanismo lineare gioca un ruolo significativo nei *TBLs* (*Turbulent Boundary Layers*) ha portato *Lim & Kim* ad esaminare i *TBLs* da una prospettiva lineare. Nello specifico loro hanno applicato le analisi di *decomposizione al singolo*

valore (*SVD*), (*Single Value Decomposition*) per avere maggiore comprensione del meccanismo attraverso il quale diversi strumenti di controllo permettono di realizzare una riduzione della resistenza d'attrito viscosa nei *TBLs*.

L'analisi *SVD* da loro effettuata [10], ha mostrato come diversi strumenti di controllo sono efficaci nel ridurre i valori singolari associati al sistema lineare che si ottiene dalla riduzione delle equazioni di *Navier-Stokes*. Si osserva che la riduzione del valore singolare è relativa alla riduzione della *non-normalità* del sistema flusso, che è parzialmente responsabile del sostentamento delle strutture turbolente vicino alla parete (le quali sono a loro volta responsabili dell'alta resistenza d'attrito nei *TBLs*).

Tuttavia comprendere le limitazioni delle performance è di grande importanza nel design di strumenti di controllo per i flussi turbolenti, in modo da fornire una guida che può accompagnare gli input dei controlli. In merito a ciò, vale la pena discutere la congettura di Bewley sul limite di prestazione fondamentale:

La resistenza più bassa sostenibile di un flusso incompressibile a flusso di massa costante, quando controllato tramite una distribuzione di risultante netta nulla di flusso di massa di aspirazione/soffiaggio sulle pareti senza scivolamento di un canale, è esattamente quella che si avrebbe per un flusso laminare.

La congettura di Bewley può essere chiarita mediante l'utilizzo dell'equazione della resistenza d'attrito in un flusso in un canale completamente sviluppato:

$$D = \frac{1}{2} \left(\left. \frac{dU}{dy} \right|_{-1} - \left. \frac{dU}{dy} \right|_1 \right) = 2 + \frac{3}{2} \text{Re} \int_{-1}^1 \overline{u'v'} dy$$

In questa si ha che tutte le quantità sono normalizzate dalla velocità a centro canale e dalla semialtezza del canale, e quindi U indica la velocità media, Re il numero di Reynolds basato sulla velocità laminare a centro canale e $\overline{u'v'}$ è lo sforzo di taglio di Reynolds. Dal punto di vista dell'equazione, la congettura di Bewley equivale a dire che l'integrale lungo y di $\overline{u'v'}$, con o senza un input di controllo, è sempre positivo nei flussi in un canale. Guardando poi i termini si vede come la resistenza d'attrito in un flusso transitorio e turbolento in un canale è più alta che nel corrispondente flusso laminare con lo stesso flusso di massa.

In merito alla congettura di Bewley sono stati fatti poi numerosi altri studi, dai quali è emerso sia che questa congettura risultava essere errata, in quanto ci potevano essere flussi in cui la resistenza si riduceva maggiormente, e sia altri in cui si vedeva invece, che questi casi erano molto limitati e quindi rimaneva, dal punto di vista generale, la validità di questa congettura.

Le conclusioni dello studio effettuato [10] dicono che le strutture turbolente vicino parete, responsabili dell'incremento della resistenza d'attrito nei *TBLs*, si rigenerano autonomamente nella regione di parete tramite un processo autosostenente. Gli elementi chiave nel processo autosostenente sono lineari, e sono soggetti ad analisi lineari. Queste mostrano che gli strati limite possono essere analizzati da una prospettiva di sistema lineare. L'analisi *SVD* da utili informazioni riguardo la capacità degli strumenti di controllo di attenuare la crescita transitoria dei disturbi nei *TBLs*. I trend che si osservano nelle analisi *SVD* sono simili a quelli osservati nelle simulazioni numeriche dirette (*DNS*) per riduzione di resistenza in flussi turbolenti, mostrando come il modello di sistema lineare può descrivere una parte importante della dinamica vicino a parete e può essere utilizzato come schema guida per il design di diversi strumenti

di controllo per la riduzione della resistenza; per esempio può risultare utile per ottimizzare i parametri di controllo di uno strumento nella prima fase di progetto, senza le attuali costose performance di calcoli non lineari. Buona parte delle discussioni fatte nell'articolo sono basate sulle equazioni di *Navier-Stokes* linearizzate, il cui uso è giustificato sia dal fatto che un meccanismo lineare gioca un ruolo fondamentale nel processo di autosostentamento della turbolenza vicino alla parete, e sia dal confronto fatto fra le simulazioni linearizzate svolte e i dati ottenuti, per lo stesso problema, da quelle numeriche dirette (*DNS*).

La necessità di conoscere le caratteristiche fisiche del flusso turbolento, al fine di poterlo controllare per ridurre la resistenza, ha in seguito, portato e dato sempre più importanza allo studio di soluzioni pratiche, per ridurre la resistenza, dal punto di vista di metodi applicativi.

3.2 Tecniche di riduzione della resistenza d'attrito

A causa della necessità di trovare sempre nuovi metodi per ridurre la resistenza d'attrito, in un flusso turbolento, oltre allo studio del flusso vero e proprio, come si è visto, si è iniziato a studiare sempre più nuove tecniche e soluzioni pratiche per questo scopo. Diverse tecniche sono state sviluppate, e poi negli anni analizzate, per raggiungere questo obiettivo. Alcuni esempi di tecniche, analizzate al fine di valutarne le prestazioni e di stabilire quali siano effettivamente più efficaci, si possono trovare sia nell'articolo, già preso precedentemente in considerazione, *Active and passive turbulent boundary-layer drag reduction* di T.C. Corke et al [9] e sia nella dissertazione, *Experimental Investigation of Passive and Active Friction Drag Reduction*, di W. Li, basata sul suo lavoro di scienziato ricercatore presso *Institute of Aerodynamics (AIA)* del *RWTH Aachen University* [11]. In entrambi questi articoli vengono mostrate ed analizzate alcune tecniche di riduzione della resistenza d'attrito per un flusso turbolento. Nello specifico si vanno ora a riportare le tecniche descritte negli articoli, mentre in seguito, alcune di queste tecniche saranno maggiormente approfondite mediante l'utilizzo di altri articoli specialistici in merito.

La prima tecnica che viene esposta nell'articolo di T.C. Corke et al [9] è quella detta *LEBU* (*Large Eddy Breakup*), già brevemente citata in precedenza, la quale consiste in sottili piastre piane allineate con la direzione del flusso principale che vengono sospese a differenti altezze sopra la parete nello strato limite turbolento. Le piastre vengono utilizzate singolarmente, impilate verticalmente con spaziature diverse, o posizionate in tandem nella direzione principale del flusso. Nello studio effettuato e riportato nell'articolo, tutte le grandezze vengono normalizzate, rispetto ad alcuni parametri fondamentali del problema. Vengono inoltre analizzati diversi parametri di configurazione della soluzione, e particolare importanza viene data al confronto fra le configurazioni con 1, 2 e 4 piastre. Infatti uno dei risultati di maggiore importanza che si trovano è che la configurazione migliore risulta essere quella con 2 piastre, in quanto, nella configurazione con 4, pur andando maggiormente ad intervenire sul flusso, la presenza di questa soluzione genera un maggior incremento di resistenza. Questo studio evidenzia inoltre come si ha una significativa riduzione delle fluttuazioni di velocità vicino alla parete, a patto che si riesca però ad identificare la configurazione ottimale della posizione delle piastre, che risulta essere fra la parete e la vorticità esterna di grande scala dello strato limite, in funzione della configurazione del flusso. Si indica anche, inoltre, che la soppressione delle grandi scale di moto è uno dei meccanismi per la riduzione della resistenza, e l'interazione all'interno dei vortici dei *LEBU* con le strutture vicino a parete fornisce la diminuzione primaria. Questa tecnica utilizza il controllo della parte esterna dello strato limite, per la riduzione della resistenza.

Al contrario un'altra tecnica descritta nell'articolo [9], quella delle *Riblet*, va ad intervenire e controllare le scale interne dello strato limite. Data l'importanza di questa soluzione nel lavoro di tesi qui descritto, in quanto questo si basa su questa soluzione, questa verrà trattata maggiormente nel dettaglio più avanti in questo capitolo, (vedi 3.4.1), e qui vengono riportati solamente i dati e le osservazioni che sono state fatte nell'articolo, in quanto questo, essendo un articolo su più soluzioni diverse, fornisce meno dettagli in merito. Le *riblet* sono il metodo più comune ed utilizzato per andare ad alterare la zona più interna dello strato limite a parete, per ottenere una riduzione della resistenza d'attrito. Queste consistono in scanalatura con forma a V applicate sulla superficie delle pareti. Esistono diversi tipi di forma di *riblet*, che è possibile appli-

care. Nell'articolo [9] viene presa una forma triangolare, ma con i lati curvilinei, come

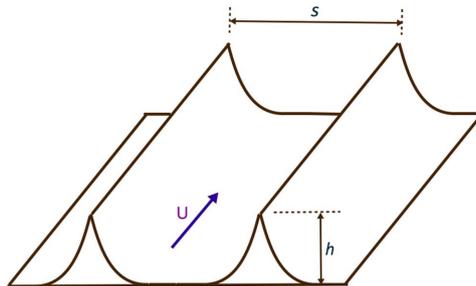


Figura 3.1: Design generico di riblet analizzato [9].

si può osservare nella figura 3.1. Qui si ha che le riblet non sono consecutive, ma si ha una distanza s fra le punte delle riblet, le quali a loro volta hanno un'altezza definita dal parametro h . Gli esperimenti svolti, nel caso qui considerato, ci dicono che per questa configurazione di riblet, si trova che le dimensioni ottimali per la riduzione della resistenza sono $h^+ = 8 - 12$ e $s^+ = 15 - 20$. Con questo design ottimale si traduce nella massima riduzione di resistenza, di circa un 8%, come risultato degli esperimenti. Tuttavia, già in questo articolo, viene detto che esistono altri studi in merito alle riblet, come quello di *Garcia-Mayoral* e *Jimenez*, che verrà analizzato in seguito, (vedi 3.4.1), in cui si studiano altre configurazioni di riblet e si ottengono altri risultati.

Dopo le riblet, nell'articolo vengono poi riportate altre due tecniche, *l'aspirazione e il soffiaggio di parete e le pareti oscillanti*. Queste due, a differenza delle riblet, che è una tecnica detta passiva, in quanto una volta realizzata non richiede energia, sono invece esempi di tecniche attive, ovvero tecniche, che pur dando effetti maggiori rispetto a quelle passive, richiedono costantemente energia nel loro funzionamento, ed è quindi necessario bilanciare i guadagni in termini di riduzione della resistenza, con i costi che queste comportano.

Andando a vedere quanto riportato nell'articolo, in merito all'*aspirazione/soffiaggio di parete*, viene detto che questa soluzione è generalmente utilizzata per intervenire nel ciclo di eventi di esplosione e rimescolamento che si ha nel flusso turbolento e che ha un ruolo chiave nella produzione degli *sforzi di Reynolds*. L'aspirazione può, per esempio, essere utilizzata per ridurre questi fenomeni in uno strato limite turbolento a gradiente di pressione nullo. L'aspirazione viene anche utilizzata per eliminare un susseguirsi di separazioni artificialmente create in uno strato limite laminare; l'approccio fondamentale è quello di utilizzare l'aspirazione per inibire la separazione di un flusso a bassa velocità dalla parete. Si possono anche creare *strisce artificiali* nello strato limite laminare mediante aspirazioni impulsive, ottenute da due distinti fori nel flusso, ad una ben definita distanza lungo il flusso. Al contrario il soffiaggio nel flusso può essere utilizzato per ridurre o sopprimere dinamicamente i fenomeni i moti ordinati nella regione vicino a parete; in particolare, l'obiettivo del sistema di controllo, realizzato mediante soffiaggio, è la soppressione dei fenomeni di separazione e rimescolamento. Per questo caso il soffiaggio viene applicato per opposti alla velocità normale alla parete, individuata nel cosiddetto *piano di rilevamento*, individuato ad una distanza fissa, normale alla parete del canale. In questo schema, quando il fluido nel piano di rilevamento si muove verso la parete, nel moto di rimescolamento, per contrastarlo viene utilizzato il soffiaggio da parete. Allo stesso modo, quando il fluido viene rilevato in allontanamento dalla parete, nel moto di separazione, viene contrastato da una uguale aspirazione a parete. La

relativa riduzione della resistenza all'attrito viene caratterizzata in termini di riduzione del gradiente di pressione medio lungo il flusso richiesto veicolare il flusso.

Le pareti oscillanti, (la cui tecnica sarà approfondita maggiormente di seguito, vedi 3.3), è un altro approccio attivo al controllo del flusso. Questa tecnica si concentra sull'effetto che questa ha sulle strutture ordinate del flusso nella regione di strato limite di parete, e come ciò influenza la resistenza viscosa. Da esperimenti si osserva come questa tecnica pur fornendo risultati eccezionali, come la riduzione di oltre il 40% della resistenza, arrivando fino ad una riduzione del 70% della fluttuazione di velocità u'/U_∞ , richiede però una potenza, necessaria per mantenere l'oscillazione, che quasi supera la potenza risparmiata con la riduzione della resistenza [9]. Qui viene comunque detto, che in ulteriori studi fatti in merito a questa tecnica, come quelli di *Quadrio*, che saranno poi presi in considerazione, si analizzeranno altre configurazioni di questa soluzione, tenendo anche in considerazione la soluzione laminare di Stokes, che porteranno a differenti risultati. Dopo di che nello studio che si sta considerando, sia in merito a soluzioni mediante l'uso delle pareti oscillanti, sia in merito ad altre soluzioni, vengono considerate altre soluzioni, le quali, però, invece che andare ad agire sul flusso in modo *fluidodinamico tradizionale*, utilizzano soluzioni legate ad effetti elettromagnetici e uso di attuatori al plasma. Poiché questi argomenti non riguardano lo scopo di questo lavoro di tesi, ora non vengono presi in considerazione. L'articolo si conclude dicendo come tutti gli approcci in esso descritti (sia quelli qui riportati che gli altri) danno una riduzione locale della resistenza. Un tema comune di questi approcci è quello di provare a stabilizzare i meccanismi di strisce nel flusso turbolento.

Nella dissertazione di W. Li, viene descritta l'importanza del problema della riduzione

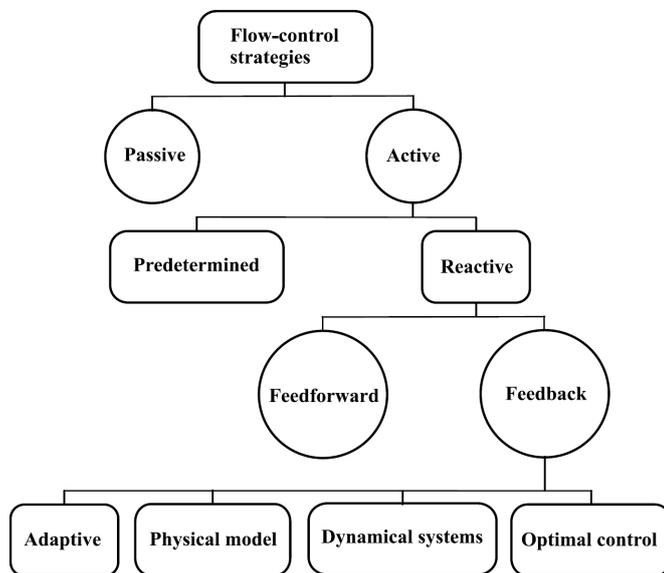


Figura 3.2: Schema riassuntivo della classificazione possibile dei metodi di controllo del flusso, proposto da *Gad-El-Hak* [9].

della resistenza d'attrito, sia come problema dal punto di vista economico, e sia dal punto di vista di obiettivo della fluidodinamica. Vengono inoltre classificati i metodi attualmente conosciuti, dividendoli in diverse categorie, in base ai principi che stanno alla base del loro funzionamento. Inizialmente i metodi di controllo dei flussi (o di ri-

duzione della resistenza) possono essere divisi in passivi, che come detto non richiedono potenza ausiliaria per il funzionamento, ed in attivi, i quali invece, richiedono input di energia e un sistema di controllo. I metodi di controllo, a loro volta sono divisi in reattivi e predeterminati, in funzione dei requisiti del sistema di controllo. Questi poi si possono dividere in ulteriori sottocategorie. Si può osservare un tentativo di schema riassuntivo, proposto da *Gad-El-Hak* [11], per la classificazione dei metodi di controllo del flusso, nella figura 3.2. Le riblet, come già visto, sono uno dei metodi passivi di riduzione della resistenza applicati con maggior successo, non solo in laboratorio, ma anche in soluzioni pratiche, ottenendo riduzioni della resistenza fino al 10%. Si ha che la riduzione di resistenza che si ottiene con le riblet non dipende solamente dall'altezza di queste ma anche dalla loro spaziature. Le soluzioni passive, come le riblet, non richiedono energia per funzionare. Tuttavia le soluzioni passive hanno generalmente una geometria fissa, la quale funziona solamente in un range di parametri limitato, fuori dal quale, può portare, oltre ad un non funzionamento, anche ad un incremento della resistenza. Al contrario dei metodi di controllo passivo, ci sono i metodi di controllo attivi, basati sul cambiare la turbolenza vicino alla parete, manipolando il flusso in quella zona. In particolare i metodi adattativi, diventati importanti negli ultimi anni, possono portare riduzioni della resistenza fino al 45%, grazie a superfici mobili e forzanti. Fra i metodi di controllo attivi ci sono pareti oscillanti, attuatori di forze di volume e superfici a movimento ondulatorio, tramite deformazioni delle pareti, i quali sono utilizzati per la manipolazione del flusso. L'oscillazione trasversale delle pareti risulta essere uno dei metodi più ampiamente analizzato fin dalla sua scoperta, ovvero dal comprendere che la resistenza d'attrito turbolenta può essere ridotta da oscillazioni di pareti trasversali ad alta frequenza. Queste possono portare riduzioni di resistenza in range fra il 10% ed il 40%. I diversi concetti di controllo del flusso, mostrati nella figura 3.2, sono stati analizzati intensivamente. Buona parte della ricerca, comunque, si è focalizzata su un solo concetto di controllo del flusso, o controlli attivi o passivi. Considerando la somiglianza nel meccanismo che porta alla riduzione della resistenza per riblet e per superfici mobili, ovvero l'influenza di mezzi passivi o attivi sulla distribuzione della velocità vicino alla parete e sui flussi di rimescolamento o separazione, è più o meno naturale collegare entrambi concetti, vale a dire, per studiare l'impatto delle superfici con riblet che subiscono anche un movimento trasversale al flusso sulla distribuzione degli sforzi di taglio a parete [11]. Questa combinazione di controlli attivi e passivi è possibile, come mostrato in esperimenti da *Roggenkamp et al.*, con una superficie ruvida di alluminio su cui può essere esercitato un moto trasversale ondulatorio. Ha senso eseguire tale analisi poiché l'effetto di riduzione della sollecitazione di taglio della parete osservata per la superficie mobile potrebbe essere migliorato dalle riblet e la suscettibilità alla geometria delle riblet potrebbe essere ridotta dal movimento della parete. Ciò viene effettuato eseguendo prima un'analisi distinta sui due metodi che si considerano, e poi andando ad unirli, per valutarne gli effetti globali.

Le riblet sono piccole sporgenze superficiali allineate nella direzione del flusso con forme diverse. Sono state riconosciute come una delle poche tecniche di controllo del flusso passivo che possono ridurre resistenza all'attrito in flussi turbolenti limitati da pareti. Numerosi studi hanno riportato che le riblet riducono la resistenza d'attrito fino al 10% nei flussi interni ed esterni. Esistono poi diverse tecniche per la realizzazione pratica delle riblet sulle superfici su cui devono essere applicate. Le riblet riducono la resistenza d'attrito impedendo il moto trasversale del flusso indotto dai vortici longitudinali nelle zone vicino alla parete. L'effetto delle riblet è quindi smorzante nei confronti dei flussi

trasversali. Tuttavia il meccanismo di interazione fra le riblet e la turbolenza vicino a parete è molto più complesso e non è ancora stato compreso completamente. Uno dei meccanismi generali è che le riblet riducono l'attrito prevenendo i vortici semilongitudinali da interazioni dirette con le pareti. Recentemente l'interazione di strutture fluide tra il flusso turbolento e le riblet e il suo impatto sulla riduzione della resistenza è stato rivisto da *Garcia-Mayoral e Jimenez* (ciò verrà riportato in seguito, vedi 3.4.1). Approcci indiretti sul flusso globale per ridurre la resistenza d'attrito possono essere realizzati su larga scala, senza sensori, con metodi di controllo attivi a circuito aperto, quali per esempio oscillazioni forzate, dischi rotanti vettori e moto di parete. I metodi di controllo del flusso vengono considerati per quanto riguarda le forze di eccitazione, come per esempio oscillazioni della parete trasversali al flusso, attuatori di forze di volume, moti ondulatori delle superfici attraverso la deformazione della parete. Inoltre, i metodi forzanti sono classificati per le direzioni cui sono applicate, cioè trasversale, longitudinale e normale alla parete. Le oscillazioni trasversali delle pareti sono state una delle tecniche più studiate fin dalla scoperta che queste potevano portare ad una riduzione della resistenza d'attrito, in particolare mediante oscillazioni ad alta frequenza. Utilizzando simulazioni numeriche dirette è stato possibile investigare le oscillazioni andando sotto la superficie dei flussi nei canali per periodi da 25 a 500 unità temporali adimensionali. Si è trovato che le oscillazioni trasversali ad alta frequenza possono portare riduzioni di resistenza dal 10% al 40%. Dopo di che l'interazione tra il flusso turbolento e le pareti oscillanti e il loro impatto sulla riduzione della resistenza è stato rivisto da *Quadrio et al.*, arrivando fino ad una massima riduzione di resistenza fino al 45% (ciò verrà riportato in seguito, vedi 3.3).

Alla fine del lavoro riportato [11] si vanno a vedere quali sono stati i risultati in merito alla riduzione della resistenza utilizzando insieme la tecnica passiva delle riblet e quella attiva delle pareti oscillanti. Gli studi sono stati condotti mediante analisi in galleria del vento a bassa velocità, utilizzando metodi *PIV*, analizzando prima la sola tecnica passiva e poi combinandola a quella attiva. La rateo di riduzione locale di resistenza a valle del flusso è stato determinato dal gradiente di velocità nel sottostrato viscoso. Le misure nei flussi *ZPG* mostrano una riduzione locale di resistenza del 4,7% a $Re_\theta = 1200$, ed una riduzione di resistenza del 0,7% a $Re_\theta = 2080$. In condizioni di flusso più realistico, cioè per flusso instazionario e *APG*, la riduzione è incrementata al 6 – 7%. Questi risultati sono quelli che sono stati trovati per l'uso delle sole riblet. Dopo aver analizzato le soluzioni attive, andando ad unire le tecniche, si osservano i risultati di entrambe. Si trova che le riblet e le pareti oscillanti, essendo complementari una all'altra per gli effetti di riduzione della resistenza, danno che la riduzione di resistenza locale del 4,7% per le sole riblet a $Re_\theta = 1200$, con l'utilizzo di anche le pareti oscillanti, sale al 9,4%. Al più alto valore di numero di Reynolds di $Re_\theta = 2080$ la combinazione di metodi di controllo attivi e passivi dimostra una riduzione di resistenza locale più alta del 2% in più rispetto alla superficie con riblet, ma non oscillante, indicando una minore sensibilità geometrica della soluzione, dovuta al moto ondulatorio trasversale della parete. Questo studio è un inizio di una fondamentale ricerca per aprire nuovi campi nel controllo dei flussi. Per futuro sviluppo, la comprensione dei fenomeni del flusso dietro la riduzione della resistenza d'attrito deve essere necessariamente estesa e ci si deve rivolgere a molti altri nuovi campi di interesse. In aggiunta, come risultato dello studio [11], si ha che ancora l'impatto delle grandi strutture turbolente per flussi ad alti numeri di Reynolds è ancora non chiaro. Ulteriori ricerche sulle riblet movimentate per numero di Reynolds elevato e sui flussi

ad alta velocità devono ancora essere condotte. Inoltre i flussi reali sono generalmente caratterizzati da raffiche instabili. Le riblet non movimentate in condizioni di flusso non stazionario sono state studiate e questo flusso instazionario deve essere applicato poi alla configurazione di riblet movimentate.

Dopo aver visto quindi, prima un insieme dei metodi più comuni utilizzati per la riduzione della resistenza d'attrito nello strato limite di un flusso turbolento, e poi uno studio che si pone l'obiettivo, per ora parzialmente raggiunto, di combinare insieme, una tecnica passiva (le riblet) ed una attiva (le pareti oscillanti), andiamo ora a vedere, tramite relativa documentazione in merito a studi specifici effettuati, alcune delle soluzioni più utilizzate, entrando un po' di più nel dettaglio del loro funzionamento ed osservando alcuni risultati raggiunti. Si vanno a vedere, riportandone le metodologie con cui queste sono state realizzate, delle tecniche utilizzate in merito alla *riduzione della resistenza d'attrito viscosa di un flusso turbolento*, dividendole, in prima approssimazione, in *tecniche attive* e *tecniche passive*. Tra le tecniche passive, troveremo poi, anche quella delle *riblet*, di particolare interesse per questo lavoro di tesi, essendo l'argomento su cui questo lavoro si basa.

3.3 Tecniche attive di riduzione della resistenza

Pareti Oscillanti

Tra le tecniche così dette attive, ovvero quelle che richiedono per il loro funzionamento un costante input di energia, e la presenza di un sistema di controllo, troviamo quella delle pareti oscillanti. Questa è sia una delle prime tecniche di questo tipo scoperte, ed anche una di quelle ancor oggi maggiormente utilizzate. Questa tecnica consiste nel rendere una o più superfici a contatto con la parete su cui scorre il flusso mobili, al fine di andare ad alterare le strutture turbolente e ridurre la resistenza d'attrito viscosa. In particolare si parla di tecnica delle pareti *oscillanti*, quando il moto delle pareti risulta essere trasversale alla direzione del flusso, periodico, e quindi oscillatorio, e generalmente ad alta frequenza. I parametri importanti di questo tipo di soluzione sono infatti sia la frequenza, o il periodo delle oscillazioni, e la loro ampiezza, oltre ovviamente alla dimensione delle superficie mobile. Per andare meglio ad analizzare questa tecnica, e vedere quali sono stati i progressi e i risultati raggiunti da questa, andiamo a prendere in considerazione due articoli in merito. Il primo è *Wall-oscillation conditions for drag reduction in turbulent channel flow* di *Pierre Ricco & Maurizio Quadrio* [12], mentre il secondo è *Streamwise oscillation of spanwise velocity at the wall of a channel for turbulent drag reduction* di *Claudio Viotti, Maurizio Quadrio & Paolo Luchini* [13]. Questi due articoli, che condividono un autore in comune, *M. Quadrio* trattano entrambi il tema dell'oscillazione delle pareti al fine di ridurre la resistenza d'attrito in un flusso turbolento. Nel primo articolo, quello di *Ricco & Quadrio*, vengono analizzate le proprietà di ridurre la resistenza di un flusso turbolento in un canale, modificato da oscillazioni trasversali sinusoidali delle pareti, tramite simulazioni numeriche dirette. Il lavoro li descritto si basa sulla relazione lineare tra la riduzione di resistenza ed un parametro S , funzione della massima velocità della parete mobile e del periodo delle oscillazioni. Questo parametro è stato trovato attraverso argomenti fisici relativi all'azione dello strato di Stokes oscillante sulla dinamica di turbolenza vicino a parete. Il potenziale predittivo del parametro di ridimensionamento viene sfruttato per ottenere informazioni sugli effetti di riduzione della resistenza della tecnica delle pareti oscillanti. Il periodo di oscillazione che garantisce la massima riduzione della resistenza per un dato spostamento massimo della parete viene studiato qui per la prima volta. Anche il problema dell'intensità minima della forzante di parete richiesta per produrre un effetto di riduzione della resistenza non nullo e la dipendenza della riduzione di resistenza al numero di Reynolds vengono indirizzati. I dati di riduzione della resistenza disponibili in letteratura sono quindi confrontata con la previsione fornita dal parametro di ridimensionamento, per poter così ottenere una visione completa dello stato dell'arte. Nel secondo articolo, invece, quello di *Viotti, Quadrio & Luchini*, viene studiato, attraverso sempre simulazioni numeriche dirette, una forzante stazionaria a parete di un flusso in un canale, per valutare la sua abilità intrinseca di riduzione della resistenza d'attrito turbolenta. La forzante di parete consiste in una distribuzione stazionaria di velocità trasversale che si alternano nella direzione longitudinale del flusso. L'idea alla base di una forzante si costruisce sulle tecniche esistenti si pareti trasversalmente oscillanti e sfrutta la natura convettiva del flusso per ottenere un'interazione non stazionaria con la turbolenza. Le analisi si avvantaggiano dall'equivalente flusso laminare, che viene analizzato analiticamente per mostrare che il costo energetico della forzante non è influenzato dalla turbolenza. In un flusso turbolento, si trova che il forzante alternante si comporta in modo analogo alla parete oscillante; in particolare si trova una lunghez-

za d'onda ottimale che produce la massima riduzione della resistenza turbolenta. Le prestazioni energetiche sono notevolmente migliorate, con più del 50% del massimo risparmio di attrito a grandi intensità di forzanti ed un risparmio energetico netto del 23% per intensità minori. Una forzante così stazionaria, basata su una parete, può aprire la strada ad interagire passivamente con il flusso turbolento per ottenere una riduzione della resistenza attraverso un'adeguata distribuzione di rugosità, progettato per eccitare una selezionata lunghezza d'onda lungo il flusso.

Entrando ora maggiormente nel dettaglio degli studi effettuati, in particolare quello di *Ricco & Quadrio* [12], si ha che questo presenta diversi studi numerici di flussi turbolenti in canali con oscillazioni trasversali sinusoidali delle pareti. Il flusso sulle oscillazioni è il risultato della combinazione di due semplici fattori, cioè il flusso turbolento canonico in un canale in direzione longitudinale e un flusso oscillante trasversalmente al livello dello strato limite. La caratteristica più rilevante di questo flusso turbolento modificato è la riduzione sostenuta nel tempo dello sforzo di taglio longitudinale; l'effetto di riduzione della resistenza è causato dall'indebolimento dei fenomeni rilevanti nella produzione di turbolenza, in prossimità delle pareti. Un passo importante verso l'applicazione pratica della tecnica delle pareti oscillanti è stata la recente scoperta di un parametro di scala S , che ci suggerisce essere collegato con il corrispettivo della riduzione di resistenza. Questo parametro, a sua volta, dipende dai parametri che definiscono l'oscillazione sinusoidale, ovvero il periodo delle oscillazioni T e la massima velocità di parete W_m . Il principale obiettivo dell'articolo è investigare attraverso la possibilità di stimare la riduzione di resistenza attraverso il parametro S . Qui l'analisi di correlazione fra S e la riduzione di resistenza è portata avanti sono attraverso i dati delle loro *DNS*.

Nell'articolo di *Viotti, Quadrio & Luchini* [13], invece, ci si concentra sempre su una

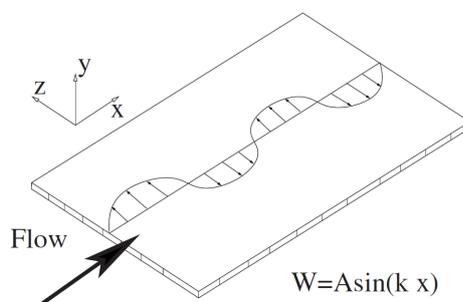


Figura 3.3: Schizzo della forzante trasversale di parete, descritta da *Viotti, Quadrio & Luchini* [13].

forzante a parete trasversale, ed in particolare su una classe di metodi forzanti disegnati per modificare in modo favorevole il flusso turbolento tramite l'introduzione di un'azione esterna diretta in direzione trasversale al flusso. In questo la componente trasversale della velocità viene indicata con W , e la legge forzante è definita come $W = A \sin\left(\frac{2\pi}{T}t\right)$, dove si ha che t è il tempo, A e T sono l'ampiezza e il periodo dell'oscillazione, rispettivamente. Si osserva come, una volta che A e T sono definiti in maniera ottimale, si osserva una forte soppressione di turbolenza. Vantaggi e svantaggi della tecnica delle pareti oscillanti sono ovvi. Presenta prestazioni energetiche che potrebbe valere la pena di implementare praticamente e grazie alla sua caratterizzazione ad anello aperto, non ha bisogno di sensori distribuiti o attuatori, che sarebbero ancora poco pratici con la

tecnologia disponibile oggi. D'altro canto, proprio per questa natura, questa tecnica richiede parti in movimento, e quindi non si presta ad essere implementato come dispositivo passivo, che è d'altra parte il possibile campo di applicazione più attraente. In questo articolo, [13], si vuole andare ad estendere questo concetto un passo avanti, trasformando la dipendenza temporale della forzante in una legge stazionaria; ciò può essere realizzato esplorando la natura convettiva del flusso di strato limite a parete. La legge della velocità trasversale W , viene ridefinita come $W = A \sin\left(\frac{2\pi}{\lambda_x} x\right)$, dove $\left(\frac{2\pi}{\lambda_x}\right)$ viene indicato con k , ed è il parametro che definisce la distribuzione di velocità. È possibile osservare uno schema della soluzione che si va a realizzare in questo studio, nella figura 3.3.

Nell'articolo di *Ricco & Quadrio* [12] la velocità è espressa in funzione del tempo, ovvero con la formula $W = W_m \sin\left(\frac{2\pi t}{T}\right)$, con la differenza che qui l'ampiezza delle oscillazioni viene indicata con W_m e non con A . Si fa riferimento, tuttavia, proprio al metodo utilizzato da *Viotti, Quadrio & Luchini*, indicando che in quel caso si va ad eseguire un'analisi *pseudo-spettrale*, che si basa sull'espansione di Fourier, in direzione omogenea ed al quarto ordine di accuratezza, utilizzando poi uno schema compatto alle differenze finite per la discretizzazione degli operatori differenziali, in direzione normale alla parete. Tornando al proprio lavoro, oltre a specificare come le simulazioni siano state svolte grazie ad un sistema di calcolo in modalità dedicata,¹ vengono indicati i tipi di simulazioni che vengono svolti. In particolare si ha il $Re_\tau = 200$, mentre i parametri geometrici sono $L_y = 2h$, $L_x = 21h$ e $L_z = 4.2h$. Viene utilizzata una griglia con rispettivamente 160, 321 e 129 punti di griglia e modi di Fourier in direzione normale alla parete, longitudinale e trasversale. Per il caso a $Re_\tau = 400$, invece, questi valori salgono a 256 punti di griglia in direzione normale alla parete, e 385 e 321 modi di Fourier in direzione longitudinale e trasversale.

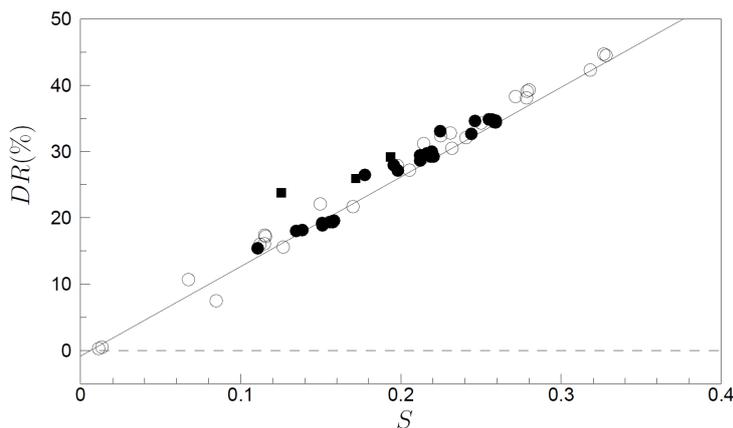
Nello studio di *Viotti, Quadrio & Luchini*, invece, viene analizzato principalmente un flusso laminare in un canale piano soggetto o a forzanti temporali a parete o forzanti spaziali a parete. Per i casi temporali è facile mostrare che la componente w dell'equazione della quantità di moto incompressibile si disaccoppia, in modo da poter descrivere il flusso con la soluzione parabolica classica di Poiseuille. Anche qui viene riportato, in merito a pareti oscillanti in funzione del tempo, come ci siano soluzioni già studiate, ed in particolare si fa riferimento all'articolo di *Ricco & Quadrio*, in cui si spiega come li altri termini possano essere trascurati, nelle equazioni da considerare. Anche in quest'altro, le analisi si basano sulla stretta concordanza fra la media spaziale trasversale al flusso e la soluzione laminare del secondo problema di Stokes. A frequenza relativamente alte, il flusso trasversale mediato nello spazio si disaccoppia dal flusso longitudinale mediato nello spazio, sebbene le fluttuazioni turbolente trasversali siano significativamente alterate dal moto della parete. Invece, quando il periodo T delle oscillazioni è grande, le strutture vicino a parete hanno sufficiente tempo per sviluppare la loro dinamica interna fra scambi successivi degli strati di Stokes. Entro questi limiti, il flusso si adatta ad una nuova condizione tridimensionale quasi stazionaria, dove il tempo può essere trattato come un parametro e l'effetto di riduzione della resistenza dello strato di Stokes si perde.

Guardando i parametri dello studio di *Viotti, Quadrio & Luchini*, abbiamo che qui, per i casi turbolenti analizzati, viene utilizzata la tecnica delle simulazioni numeriche dirette. Le simulazioni sono svolte a $Re_\tau = 200$, dove Re_τ è il numero di Reynolds

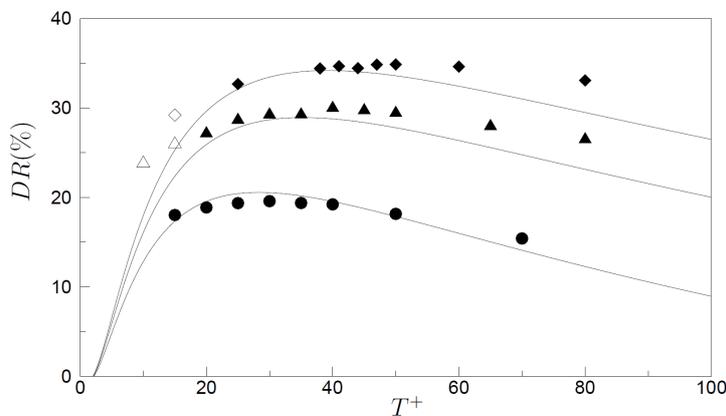
¹all'università di Salerno, grazie al lavoro di Luchini, presente nell'altro articolo analizzato.

d'attrito. Le dimensioni del dominio di calcolo sono $L_x = 6\pi h$, $L_y = 2h$ e $L_z = 3\pi h$. Il tempo di calcolo totale è molto grande, per garantire risultati mediati nel tempo ben convergenti. Il modello numerico utilizzato è formato da $N_x = 320$, $N_y = 160$ e $N_z = 320$. Vengono eseguite diverse simulazioni per diverse condizioni al contorno. Le simulazioni sono state eseguite sullo stesso sistema di calcolo utilizzato anche per gli altri studi.

Si vanno ora a vedere i risultati degli studi, partendo prima da quelli di *Ricco & Quadrio* [12]. Nel primo grafico dei due riportati in figura 3.4 (a), si osserva che i dati delle



(a) Riduzione della resistenza in funzione del parametro di scala S .



(b) Riduzione della resistenza in funzione del periodo delle oscillazioni, per fissati valori di D_m^+ , in particolare i tondi sono a $D_m^+ = 100$, i triangoli a $D_m^+ = 200$ e i rombi a $D_m^+ = 300$.

Figura 3.4: Alcuni dei risultati delle analisi di *Ricco & Quadrio* [12].

DNS mostrano una riduzione di resistenza in funzione del parametro S , a $Re_\tau = 200$, insieme ai dati con periodo $T^+ \leq 150$. Le condizioni di oscillazione sono largamente differenti, ma i dati collassano tutti sulla linea retta, che riporta l'andamento previsto, calcolato raggruppando i dati delle DNS con la maggior parte di quelli disponibili in letteratura. Nel secondo grafico della figura, il 3.4 (b), vengono confrontati i risultati delle simulazioni svolte, designate per identificare il $T_{\text{opt,D}}$, tramite le previsioni effettuate analiticamente. Si ha una buona corrispondenza dei dati tranne per 3 punti a

basso T (i simboli vuoti nella figura (b) e i quadrati neri in quella (a)). Per questi punti i valori previsti sono più bassi di quelli ottenuti dai dati delle DNS. Ciò perché non si è ancora bene in grado di spiegare il comportamento a periodi T molto bassi, i quali corrispondono a velocità W_m molto grandi. Queste discrepanza comunque non limitano le possibilità di sfruttare S per studiare con successo il flusso, poiché tutti i fenomeni rilevanti di riduzione della resistenza avvengono per W_m più piccole. Nella

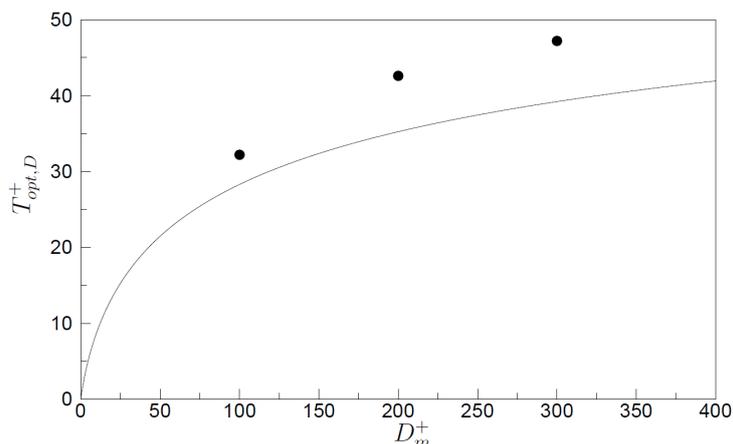


Figura 3.5: Periodo ottimale delle oscillazioni $T_{opt,D}^+$, come funzione di D_m^+ . La linea solida rappresenta l'equazione, mentre i punti indicano i valori di $T_{opt,D}^+$ trovati dalle analisi DNS di *Viotti, Quadrio & Luchini* [13].

figura 3.5 si mostra che il $T_{opt,D}^+$ previsto è buono su gran parte del range analizzato, confermato dai calcoli effettuati per confrontarlo con quello ottenuto dalle DNS. L'idea che un'intensità finita del forzante sia necessaria per influenzare l'attrito turbolento è contenuta nella definizione di S , dove viene introdotta una velocità di soglia W_{th} . Questi valori minimi sono mostrati nelle linee isolivello nelle figure 3.6, come le curve di riduzione di resistenza nulla. La riduzione di resistenza, mostrata nella prima delle due figure, aumenta senza limiti quando T diminuisce perché è necessaria una forzatura maggiore della parete per influire sul flusso turbolento man mano che la profondità di penetrazione $\delta_s \sim \sqrt{T}$ dello strato di Stokes svanisce. $W_{m,min}^+$ diventa approssimativamente costante a più alti periodi delle oscillazioni. Le figure 3.6 mostrano l'andamento della riduzione della resistenza, in funzione di diversi parametri. Valutando invece l'andamento del risparmio netto di energia, definito come la differenza tra la potenza risparmiata grazie alla riduzione della resistenza grazie alle pareti mobili, a velocità costante, e la potenza utilizzata per muovere le pareti a cui si oppongono gli sforzi viscosi. Gli andamenti di questa grandezza sono riportati, in funzione di diversi parametri, nelle figure 3.7. Dai calcoli eseguiti si osserva come quantità leggermente inferiori di riduzione della resistenza si ottengono raddoppiando il valore del numero di Reynolds, da $Re_\tau = 200$ a $Re_\tau = 400$ per $W_m^+ = 12$ e $T^+ = 30, 125, 200$ come mostrato nella figura 3.8. La riduzione di sforzo di taglio si riduce dal 21,7% al 20,3% a $T^+ = 30$, (variazione del 6,6%), dal 32,5% al 28,1% a $T^+ = 125$, (variazione del 13,4%), e dal 27,2% al 22% a $T^+ = 200$, (variazione del 19,2%). Queste variazioni aumentano così con T per W_m fissato. Studi sperimentali precedenti indicavano che la riduzione di resistenza non variava con il numero di Reynolds, per (piccole) variazioni di

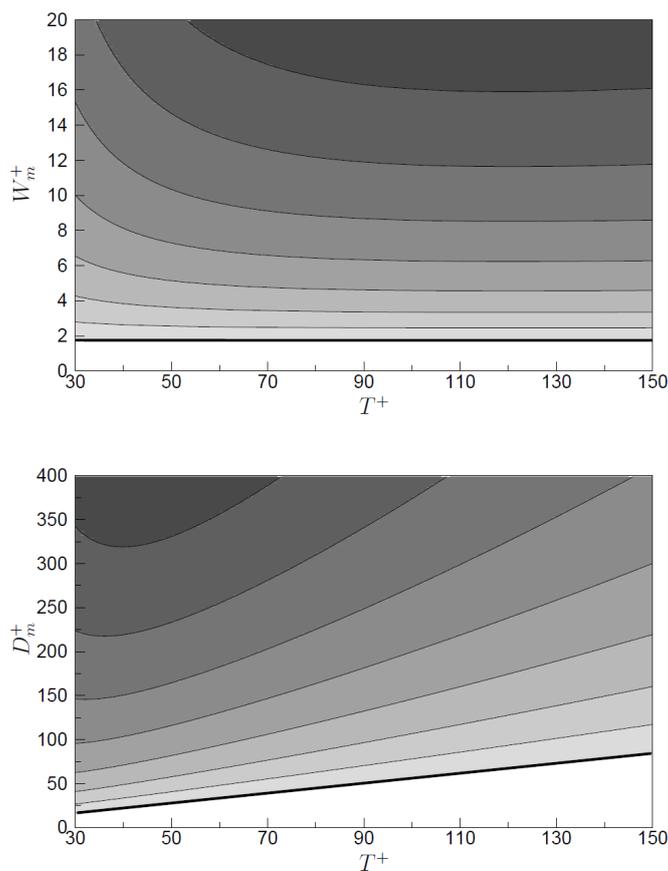


Figura 3.6: Linee isolivello della riduzione totale di resistenza in funzione di W_m^+, T^+ in quella superiore e di D_m^+, T^+ in quella inferiore. Le zone più scure indicano una più alta riduzione di resistenza e le linee isolivello si incrementano dallo 0 (linee più spesse) del 5% [13].

questi parametri. È possibile che le variazioni di riduzioni di resistenza non siano state individuate a causa dell'elevata incertezza sperimentale.

Concludendosi l'articolo [12], riassume quello che è il lavoro presentato, in merito agli effetti delle oscillazioni trasversali delle pareti in un flusso turbolento in un canale, basati su simulazioni numeriche dirette delle equazioni incompressibili di *Navier-Stokes*. Viene inoltre considerato un parametro di scala S , ed utilizzato per migliorare la comprensione delle principali proprietà di riduzione della resistenza di questa tecnica. Si è scoperto che S è un eccellente strumento predittivo per la riduzione della resistenza, per velocità non troppo elevate ($W_m^+ \leq 40$) e periodo $30 \leq T^+ \leq 150$. Questa regione di parametri include la maggior parte dei dati pubblicati sulla riduzione della resistenza ed è della massima importanza pratica. L'esistenza di due periodi ottimali per la riduzione della resistenza, $T_{\text{opt,D}}^+$ e $T_{\text{opt,W}}^+$, dedotti dall'espressione di S e verificati attraverso il database di dati di DNS, è chiaramente stabilita. $T_{\text{opt,D}}^+$, il periodo di oscillazione che garantisce la massima riduzione di resistenza a picchi fissati di spostamento delle pareti, riguarda principalmente gli sperimentatori, che sono costretti a variare T con D_m costante. $T_{\text{opt,W}}^+$, il periodo ottimale di oscillazione per la massima velocità fissa, è una

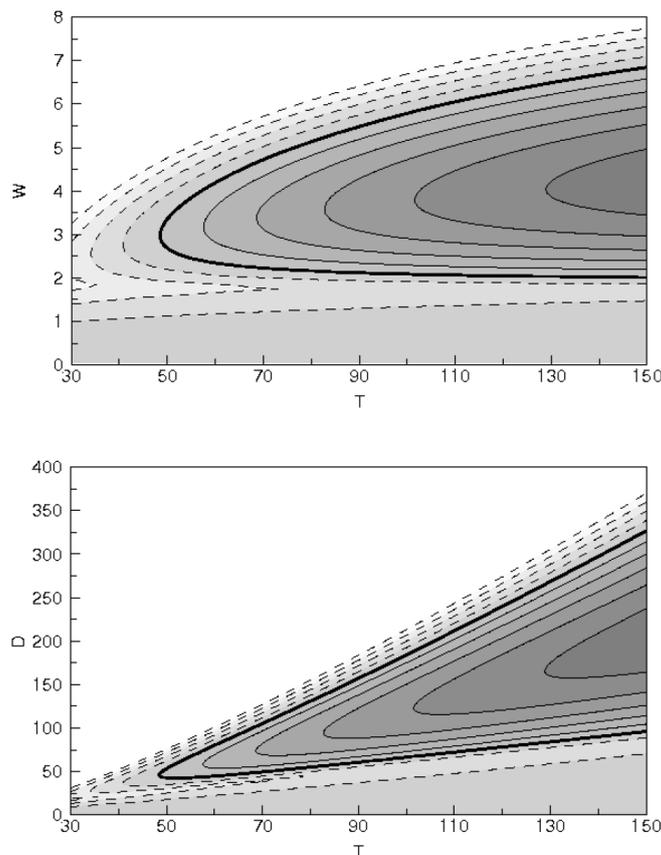


Figura 3.7: Linee isolivello della percentuale di energia risparmiata in funzione di W_m^+, T^+ in quella superiore e di D_m^+, T^+ in quella inferiore. Le zone più scure indicano una più alto risparmio di energia, le linee isolivello si incrementano dallo 0 del 1% e le linee tratteggiate riportano i valori negativi (solo i valori $\leq -4\%$ sono riportati). Le linee più spesse indicano un risparmio nullo di energia [13].

quantità tipicamente ricercata da numerosi studiosi numerici. La funzione di $T_{\text{opt},W}^+$ viene determinata dall'espressione di S , e confermata attraverso esperimenti numerici svolti per questo scopo. Si è anche stabilita la condizione minima forzante di parete portante riduzione di resistenza. Nonostante alcune discrepanze con dati esistenti, i grafici ottenuti possono essere usati per migliorare il design di strumenti esistenti per la riduzione della resistenza. È stato riscontrato che la quantità di riduzione della resistenza diminuisce leggermente al variare del numero di Reynolds, da $Re_\tau = 200$ a $Re_\tau = 400$ quando $W_m^+ = 12$. Questo l'effetto si amplifica all'aumentare del periodo di oscillazione. Ulteriori studi dovrebbero essere condotti per studiare come cambiano i periodi ottimali di oscillazione con il numero di Reynolds. Abbiamo anche stimato che il massimo risparmio di energia netto diminuisce leggermente con Re_τ . I dati esistenti, numerici e sperimentali, sono stati riesaminati attraverso un confronto tra le quantità misurate di riduzione della resistenza e loro stime basate su S : le differenze sono state discusse, caso per caso, in relazione all'accuratezza di ciascun set di dati. Si è infatti, infine svolto, un confronto fra i dati calcolati nell'articolo, e i set di dati già esistenti [12].

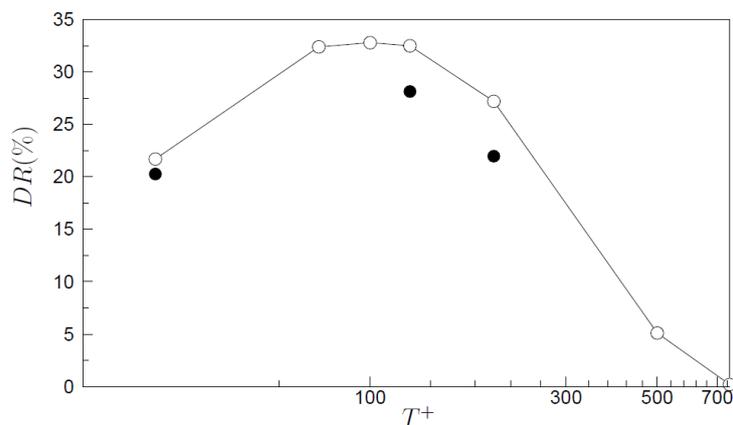


Figura 3.8: Riduzione della resistenza in funzione di T^+ a $W_m^+ = 12$ e $Re_\tau = 200$ (tondi vuoti) e $Re_\tau = 400$ (tondi pieni) [13].

Dopo aver visto i risultati del primo studio analizzato, si vanno a vedere ora, i procedimenti ed i risultati del secondo. L'efficacia della forzante spaziale stazionaria nel ridurre la resistenza di attrito viene valutata esaminando il valore del coefficiente di attrito superficiale ottenuto in diverse simulazioni differenti, in cui l'ampiezza A e la lunghezza d'onda λ_x della forzante di parete sono sistematicamente variate. I coefficienti misurati vengono quindi confrontati con il valore di riferimento del flusso non alterato (senza forzante). I risparmi nella potenza necessaria al moto sono quindi confrontati con il costo energetico della distribuzione della velocità della parete, al fine di valutare il risparmio energetico netto reso possibile dalla forzante spaziale. Le nostre simulazioni vengono eseguite per una portata fissa, quindi più che una diminuzione della resistenza di attrito, questa si traduce in una proporzionale diminuzione del gradiente di pressione medio lungo il flusso ed in una diminuzione della potenza richiesta per guidare il flusso. Si definisce la potenza necessaria per il moto del flusso a vincere la resistenza viscosa, e dopo di che si va anche a definire la percentuale di potenza risparmiata, espressa in percentuale della potenza necessaria richiesta per il flusso nel caso base. Oltre alla potenza necessaria per muovere il flusso nel canale, un'altra potenza aggiuntiva è richiesta per muovere le pareti, contro lo sforzo di taglio del flusso. Anche questa viene prima definita in modo assoluto, e dopo di che espressa in termini percentuali, rispetto alla potenza base necessaria al moto del flusso. Questa potenza assume valori negativi, in quanto effettua un lavoro contro la viscosità del flusso. Si possono così confrontare diversi metodi di forzanti di pareti, al fine di apprezzare i diversi costi energetici. Dalle due potenze così definite, è possibile definire una potenza netta risparmiata, come la differenza fra quella totale risparmiata e quella necessaria al moto delle pareti. Dalla figura 3.9 è possibile osservare che i guadagni positivi di potenza netta P_{net} si hanno per ampiezze maggiori: ad $A^+ = 12$ il guadagno netto è del 5 %, mentre per altre simulazioni riportate si ha, alla stessa ampiezza, una perdita netta di circa il 30 %, come si osserva anche nella figura 3.10. In questa infatti è riportato l'andamento della potenza netta risparmiata, P_{net} , dove si osserva anche che il massimo valore, di circa il 23 % lo si ha per $A^+ = 6$. Vale la pena notare che la ricerca del massimo di potenza netta P_{net} non può essere considerata esaustiva, e quindi il valore massimo attualmente osservato del 23 % con $A^+ = 6$ dovrebbe essere considerato solamente come un punto di partenza

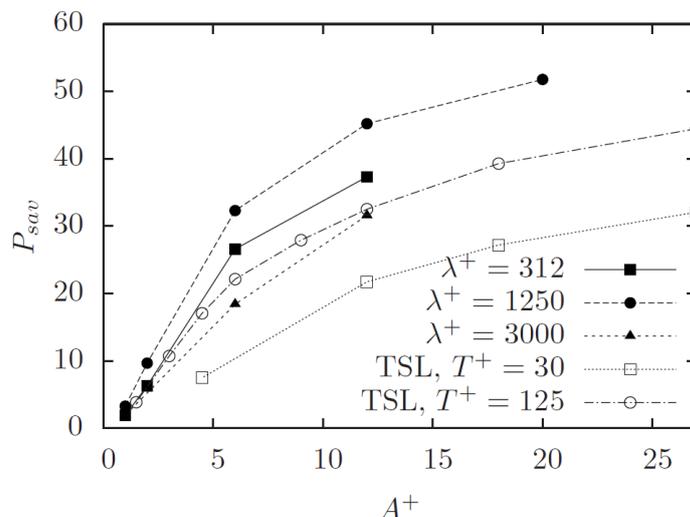


Figura 3.9: Percentuale di potenza netta risparmiata, P_{net} , in funzione dell'ampiezza della forzante [13].

per un'ulteriore ricerca più raffinata [13].

Riassumendo quanto visto in questo articolo, questo ha analizzato una nuova forma di forzanti di contorno per un flusso turbolento limitato da pareti, la quale consiste nell'imporre a parete una distribuzione stazionaria di velocità trasversali, modulate nella direzione longitudinale del flusso. Nello specifico, nell'articolo sono state considerate solamente modulazioni sinusoidali. Il motivo principale è stato quello di trovare una controparte stazionaria alla tecnica delle pareti oscillanti. Il collegamento fra i due tipi di forze è la velocità convettiva delle fluttuazioni turbolente, che assume un ben definito valore non nullo a parete, ed è capace di trasformare la scala temporale in quella spaziale e viceversa. Grazie a numerose DNS, il comportamento di questa nuova forzante nel parametro spaziale è stato determinato, ed una riduzione di resistenza fino al 52% è stata osservata per $A^+ = 20$ e $\lambda_x^+ = 1250$. Per tutte le ampiezze, la lunghezza d'onda della forzante che porta alla massima riduzione di resistenza è stata trovata in corrispondenza del periodo ottimale di oscillazione, convertito in lunghezza, confermando così la validità dell'analogia fra forzante spaziale e temporale. Questa analogia è stata ulteriormente estesa studiando il caso laminare; questo era noto per essere rilevante per la tecnica delle pareti oscillanti, poiché nel caso turbolento il profilo trasversale di velocità dopo la media spazio-temporale è identico alla soluzione laminare. La soluzione laminare è stata determinata anche per il caso spaziale, ed è stato inoltre verificato che il flusso turbolento trasversale quando mediato nella fase è identico alla soluzione laminare. Questa proprietà può essere sfruttata per prevedere la potenza richiesta per il controllo del flusso. Si spera quindi che una quantità predittiva, simile al parametro discusso da *Ricco & Quadrio* [12] ed in grado di descrivere gli effetti della forzante di riduzione della resistenza, possa essere trovata e definita sulla base di questa soluzione analitica. Insieme alle analogie qualitative, ci sono differenze quantitative tra forzanti temporali e spaziali. La forzante spaziale è più efficiente in termini di riduzione della resistenza, dal punto di vista sia di quella assoluta e sia dal punto di vista del risparmio energetico netto. In particolare, un risparmio netto fino al 23% è stato calcolato con

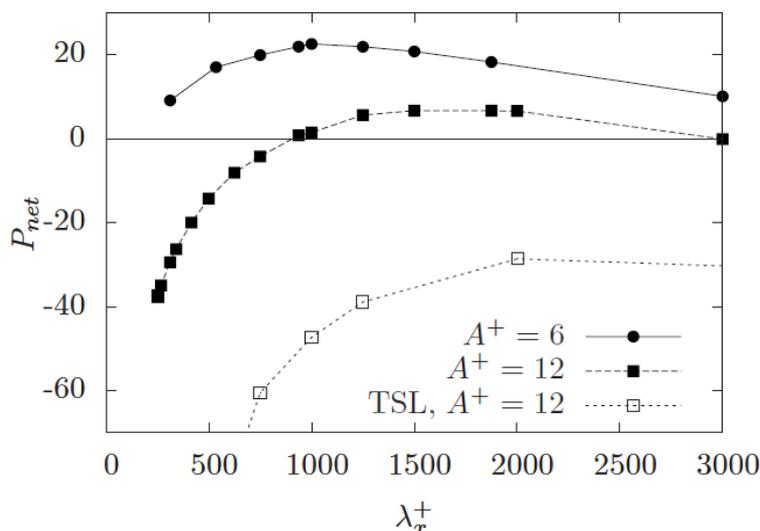


Figura 3.10: Percentuale di potenza netta risparmiata, P_{net} , in funzione della lunghezza d'onda della forzante [13].

$A^+ = 6$. Questo è più di 3 volte più grande del risparmio netto documentato per la parete oscillante, ed anche significativamente più grande del vantaggio ottenuto con dispositivi passivi come riblet, che secondo quanto riportato in letteratura sono in grado di risparmiare fino all'8% – 10% in condizioni di laboratorio. L'attuale forma di forzante spaziale è certamente realizzabile in linea di principio ed una configurazione sperimentale è effettivamente in costruzione, la quale aiuterà a far luce su effetti come la dipendenza della riduzione di resistenza dal numero di Reynolds. Allo stesso tempo, l'attuale tecnica forzante, seppur stazionaria, non è ancora considerata utilizzabile per applicazioni pratiche. Tuttavia, il successo nel design di una legge di controllo costante è un passo importante verso la realizzazione di un dispositivo passivo di riduzione della resistenza. In questo quadro, la sensibilità del ciclo turbolento vicino alla parete ad una ben definita scala di lunghezza longitudinale è un risultato fondamentale, che apre la strada alla ricerca di un tipo di rugosità efficiente per la riduzione della resistenza [13].

Attuatori normali alla parete

Andando a vedere, dopo gli studi sulle pareti oscillanti, un altro studio in merito ad un'altra soluzione cosiddetta attiva, di riduzione della resistenza, troviamo la soluzione degli attuatori normali alla parete, applicati allo strato limite di un flusso turbolento. Questa soluzione è descritta nell'articolo *The Effect of Wall Normal Actuation on a Turbulent Boundary Layer* di Schlanderer, Hutchins & Sandber [14]. In questo articolo, vengono riportate una serie di DNS che sono state svolte per studiare l'effetto di pareti attuate in direzione normale a parete, distribuite in modo omogeneo trasversalmente al flusso. Il movimento del bordo è rappresentato da una tecnica immersiva del bordo. Viene effettuato uno studio parametrico, variando la lunghezza dell'attuatore, l'ampiezza dell'attuazione normale alla parete e la frequenza di attuazione. Si trova che una localizzata attuazione basandosi solo sul movimento della parete invece di richiedere un rifornimento come nel caso di getti sintetici, genera un getto netto di flusso di quantità di moto che influenza il flusso non solo nelle immediate vicinanze dell'attuatore, ma

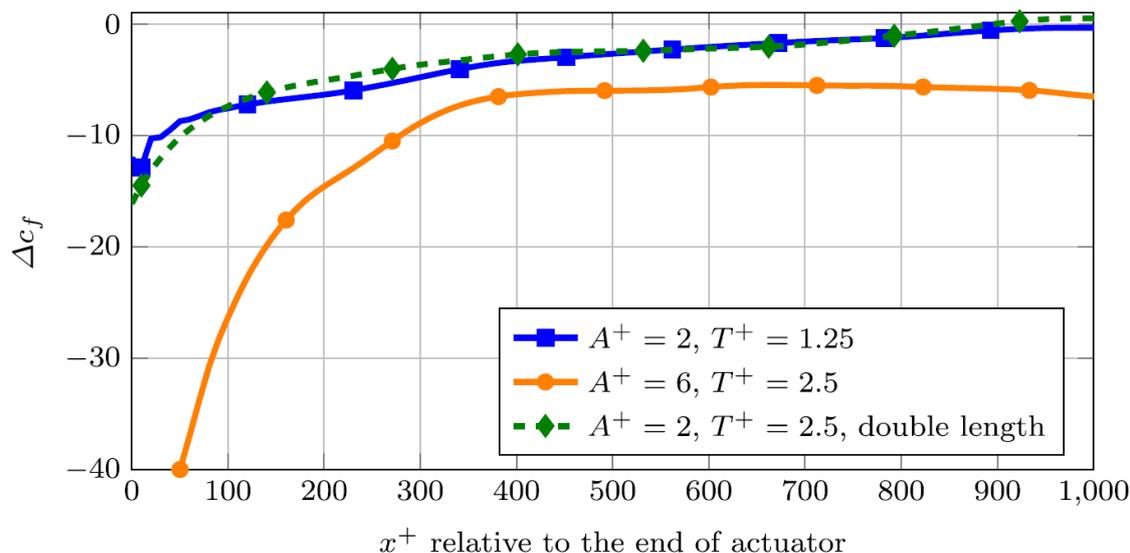


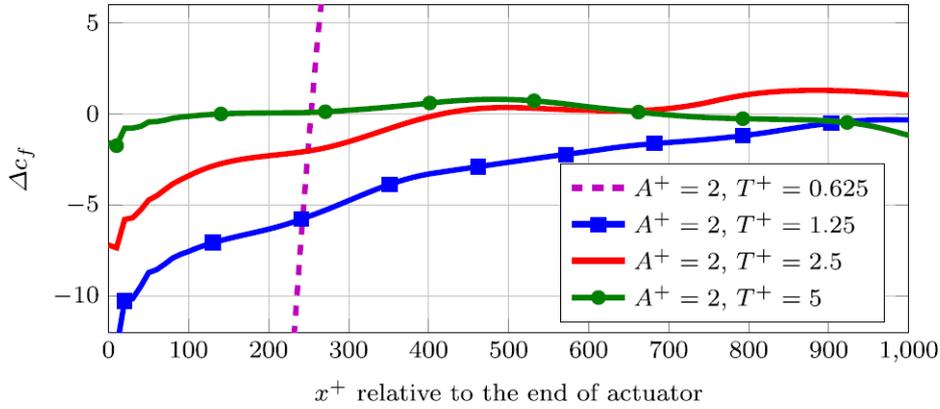
Figura 3.11: Confronto delle variazioni di attrito confrontate tra il caso non attuato e il caso con il set di dati migliori [14].

anche per una distanza significativa a valle del flusso. I casi con una velocità di attuazione di $U_{\text{act}}^+ = 20, 1$ mostrano un particolare effetto pronunciato sullo strato limite e risultano in una regione di ricircolo.

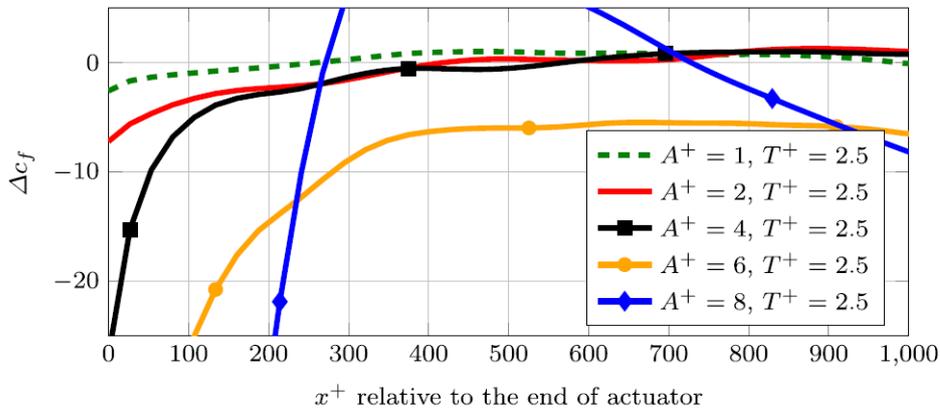
Nel campo del controllo del flusso per la riduzione della resistenza, le ricerche nelle deformazioni attive delle pareti hanno teso a concentrarsi verso le oscillazioni trasversali delle pareti e verso le onde viaggianti longitudinali. Tuttavia risulta essere evidente come questi controlli perdano di efficacia al crescere del numero di Reynolds, e risultano essere molto complicate da implementare in un esperimento ed ancora di più dal punto di vista applicativo. Le pareti deformabili di superfici o rivestimenti conformi sono state studiate inizialmente come metodo di controllo del flusso passivo; dopo di che si è passati ad un approccio di moto delle pareti attivo per il controllo del flusso, inizialmente per ritardare la transizione da laminare a turbolento con palette a sbalzo guidate da ceramiche piezoelettrici. Per il controllo dei flussi turbolenti, le deformazioni normali delle pareti hanno ricevuto meno attenzioni, nonostante il fatto che sono considerate molto affidabili in condizioni di utilizzo difficili e sono probabilmente anche relativamente facili da realizzare praticamente. Lo studio che si analizza considera un'attuazione trasversale uniforme, normale alla parete di uno strato limite turbolento, utilizzando DNS. L'obiettivo è analizzare la risposta del flusso ad un attuatore come questo, realizzabile con sforzi moderati, almeno per le misure sperimentali, con numero di Reynolds variante da basso a moderato. In questo lavoro viene utilizzato un codice di DNS per le equazioni di *Navier-Stokes* completamente comprimibile di alto ordine. L'attuazione normale alla parete è indicata con un corpo in movimento impiegando il *BDIM* (Boundary Data Immersion Method), il metodo di immersione dei dati al bordo, per i flussi compressibili, che è stato in precedenza validato da ricerche aeroacustiche e applicato con successo a problemi di interazioni fluido-strutture. L'obiettivo principale di questo lavoro è studiare l'effetto di un attuatore sullo strato limite turbolento. La riduzione della resistenza d'attrito è utilizzata semplicemente come misura per quanti-

ficare i cambiamenti nello strato limite a valle dell'attuatore.

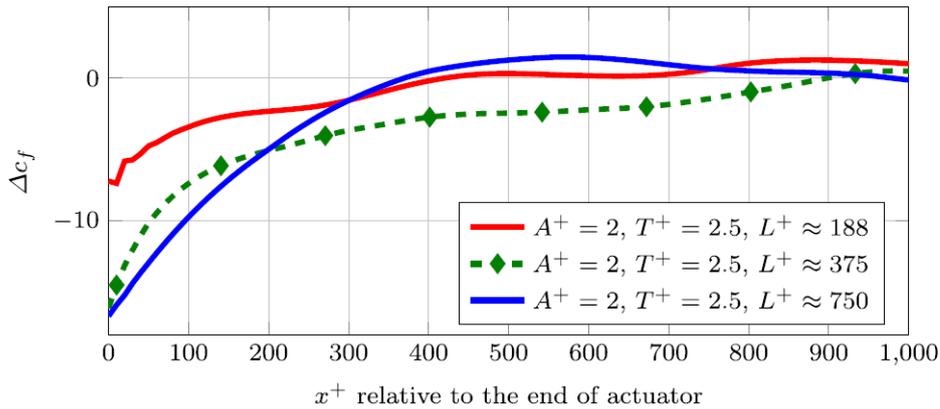
Nella figura 3.12 vengono mostrate le variazioni di attrito superficiale confrontate con lo strato limite senza attuatore, a valle dell'attuatore, per differenti parametri di attuazione. La coordinata lungo il flusso x^+ si riferisce alla stessa posizione fisica in tutti i casi, poiché x è scalata con i parametri viscosi del caso senza attuazione. Tutti i parametri variano intorno ai parametri indicati come quelli con gli effetti migliori, ovvero formalmente un'ampiezza di attuazione $A^+ \approx 2$, un periodo di attuazione $T^+ \approx 2,5$ per un attuatore di lunghezza $L^+ \approx 140$. La figura 3.12 (a) mostra l'evoluzione spaziale dell'attrito di contatto a valle dell'attuatore per differenti periodi di attuazione. L'evoluzione spaziale dell'attrito di contatto relativo al flusso di strato limite non attuato, per differenti ampiezze di attuazione, è mostrato nella figura 3.12 (b); se l'ampiezza di attuazione si riduce, per il caso base da $A^+ = 2$ ad $A^+ = 1$, con la stessa frequenza di attuazione, la riduzione di resistenza cade sotto il 5%; al contrario, se l'ampiezza aumenta, i risultati mostrano significativi aumenti di riduzione della resistenza. Infine la lunghezza di attuazione viene variata, per il caso base, tenendo costanti i parametri dell'attuazione, $A^+ = 2$ e $T^+ = 2,5$. Nella figura 3.12 (c) sono confrontati le variazioni di attrito per 3 diverse lunghezze di attuatori. I dati indicano che un attuatore con lunghezza $L_{act}^+ = 375$ sembra avere la più forte influenza sull'attrito dello strato limite. Nella figura 3.11 viene mostrato il confronto fra l'attrito del caso non attuato, ed uno con il set di parametri migliori. Per le frequenze di attuazione più elevate con un'ampiezza di $A^+ = 2$ e la massima attuazione ampiezza di $A^+ = 8$ a $T^+ = 2,5$ le velocità massime degli attuatori sono $u^+ = 20,1$, corrispondente ad $u_{act}/U_\infty = 1,1$ nella scala esterna. In entrambi i casi riduzioni elevate dell'attrito superficiale sono stati osservati fino a $x^+ = 220$ a valle dell'attuatore e, seguito poi da un forte aumentare ad un valore superiore rispetto al caso senza attuazione. Inoltre è stata effettuata una simulazione con la stessa velocità massima dell'attuatore per un'ampiezza di $A^+ = 4$ e un periodo di $T^+ = 1,25$. Dall'evoluzione dell'attrito superficiale si può concludere che l'attuazione porta alla separazione del flusso a valle dell'attuatore. L'effetto dell'attuazione periodica normale della parete omogenea trasversale su uno strato limite turbolento è stato studiato numericamente in quest'articolo, utilizzando simulazioni numeriche dirette combinate con un metodo di immersione dei dati al contorno. I risultati indicano che l'effetto dell'attuazione aumenta con l'aumentare della frequenza e dell'ampiezza fino a un certo punto. Si potrebbe dimostrare che la riduzione dell'attrito superficiale osservata non è solo correlata a un inspessimento dello strato limite, ma è una conseguenza di un'interazione tra l'attuatore e lo strato limite turbolento. L'esame dei campi di velocità media ha rivelato che l'attuazione ha come risultato un getto non nullo normale a parete di flusso di quantità di moto, nonostante non vi siano plenum o cavità come negli attuatori a getto sintetico comunemente studiati. Tuttavia, per tutti i casi attuati in cui la velocità massima dell'attuazione ha superato $u_{act} = 20,1$ o $u_{act}/U_\infty = 1,1$ il flusso separa a valle dell'attuatore. Più forte è questo getto, maggiore è il suo effetto sull'attrito superficiale a valle dell'attuatore [14].



(a) Influence of different actuation frequencies on skin-friction with fixed actuation length of $L^+ = 375$ and actuation amplitude of $A^+ = 2$.



(b) Influence of different actuation amplitudes on skin-friction with fixed actuation length of $L^+ = 375$ and actuation period of $T^+ = 2.5$.



(c) Influence of actuator length on skin-friction with fixed actuation amplitude of $A^+ = 2$ and actuation frequency of $T^+ = 2.5$.

Figura 3.12: Grafici riportanti l'andamento della resistenza, espressa come C_f , lungo la distanza dall'attuatore, espressa in funzione di diversi parametri. Vengono riportate le didascalie originali delle figure, per spiegarne il contenuto [14].

Superfici superidrofobiche

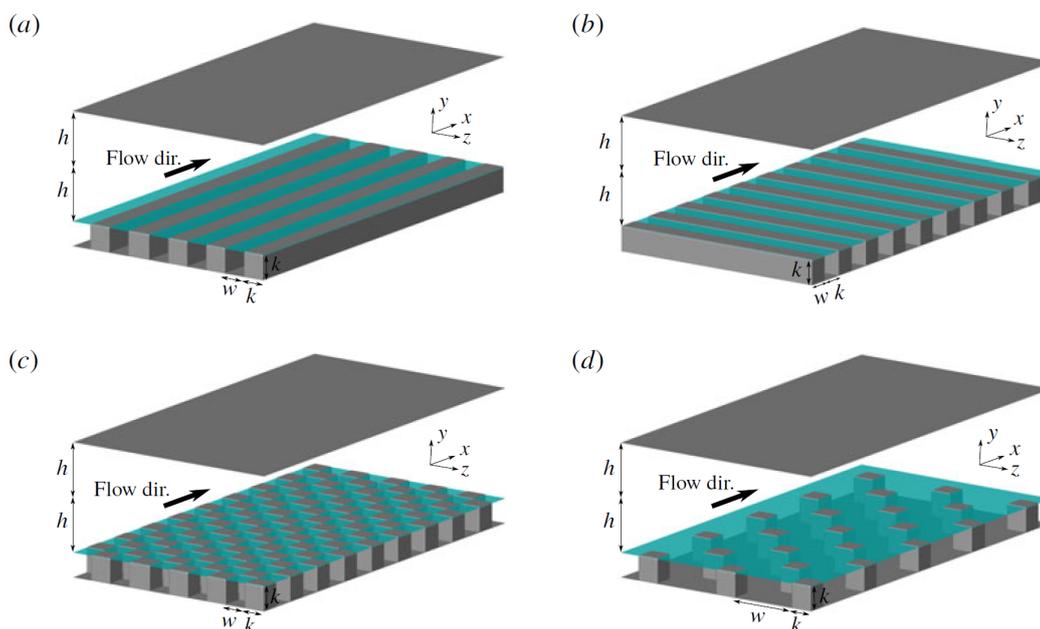


Figura 3.13: Schizzo della geometria delle superfici texturizzate: (a) barre quadrate longitudinali con ampiezza 0,5; (b) barre trasversali quadrate, con ampiezza 0,5; (c) cubi sfalsati con ampiezza 0,5; (d) cubi sfalsati con ampiezza 0,875. L'interfaccia fra i due fluidi è la superficie orizzontale in corrispondenza del piano cresta, colorato in turchese. La dimensione delle texture non è scalata rispetto all'altezza del canale, per una migliore presentazione [15].

Viene poi riportata un'altra soluzione, che può rientrare nelle soluzioni attive di riduzione della resistenza d'attrito, seppur non nel senso classico della definizione data a queste tecniche. La tecnica in questione è quella delle *superfici super idrofobiche*. Questa tecnica consiste nel realizzare su di una parete su cui scorre un flusso, delle cavità, che vengono poi riempite con dei gas, tali per cui in questi punti il fluido non è più a contatto con la parete su cui scorre, ma con un gas che non dà attrito con il fluido, causa la sua natura idrofobica. Come detto questa tecnica può rientrare nella definizione di tecnica attiva, ma non nel senso classico della definizione, in quanto si ha, che nell'uso di questa tecnica con è richiesta una spesa costante di energia per farla funzionare, come in altre tecniche attive viste fin qui, ma è necessario uno stretto controllo sul comportamento del gas che viene inserito nelle cavità, per evitare che questo trafilasse e vada sia ad miscelarsi nel fluido, e sia, che a causa della mancanza di questo, il fluido, invece di trovare una superficie con attrito minore, trovi delle cavità che vadano a generare forti effetti di mescolamento e separazione, causando quindi forti incrementi di resistenza, all'opposto di quello che la tecnica vuole realizzare. La necessità di questo controllo durante il suo uso, principalmente durante simulazioni, la rende una tecnica così detta attiva. Si vanno ora a vedere, entrando poco nel dettaglio causa la grande complessità di questa tecnica, la attuale mancanza di utilizzi pratici, e la poca attinenza con questo lavoro di tesi, alcuni articoli relativi a studi in merito questa soluzione.

Il primo articolo che consideriamo è *Comparison between super-hydrophobic, liquid*

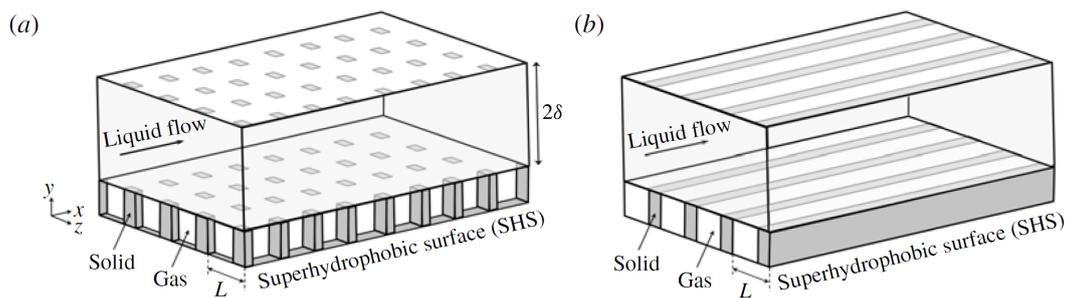


Figura 3.14: Rappresentazione schematica di un flusso in un canale con superfici superidrofobiche sulle pareti superiore ed inferiore, in particolare: (a) cubi isotropi, (b) righe longitudinali [16].

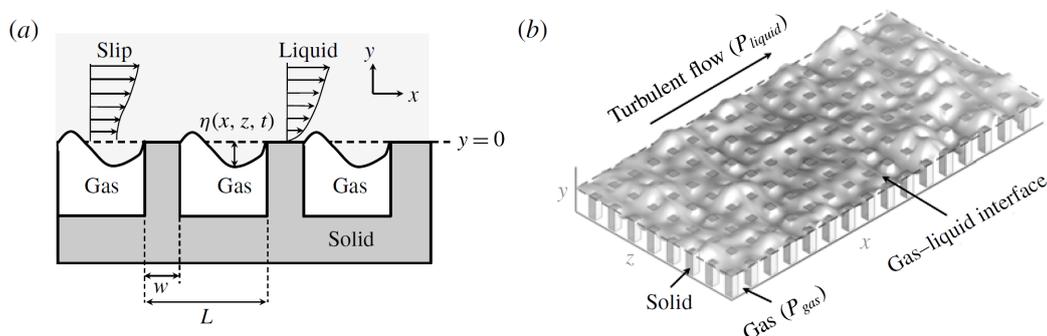


Figura 3.15: Per entrambe le figure, la posizione della linea base della parete ($y=0$) è indicata dalla linea tratteggiata. Inoltre: (a) Schematizzazione delle condizioni al contorno dei modelli di scivolamento e non scivolamento sulle superfici super idrofobiche con interfacce deformabili. (b) Interfaccia liquido-gas su cubi solidi [16].

infused and rough surfaces: a direct numerical simulation study di Isnardo Arenas, Edgardo García, Matthew K. Fu, Paolo Orlandi, Marcus Hultmark and Stefano Leonardi [15]. In questo studio sono state eseguite simulazioni numeriche dirette di due fluidi sovrapposti in un canale con una superficie con texture sulla parete inferiore. È stato condotto uno studio parametrico che varia il rapporto di viscosità tra i due fluidi per imitare le superfici superidrofobiche ideali ed infuse di liquido per valutare il suo effetto sulla resistenza d'attrito, di forma e totale per tre diverse geometrie texturizzate: barre quadrate longitudinali, barre quadrate trasversali e cubi sfalsati. Si presume che l'interfaccia tra i due fluidi sia scivolosa (ovvero senza attrito) nelle direzioni longitudinale e trasversale al fluido e non deformabile nella direzione verticale, corrispondente al caso ideale di infinita tensione superficiale. Per identificare l'importanza del ruolo dell'interfaccia fluido-fluido, è stata effettuata una serie aggiuntiva di simulazioni con un singolo fluido. Il confronto con i casi con due fluidi rivela il ruolo dell'interfaccia nel sopprimere il trasporto turbolento tra lo strato lubrificante e il flusso sovrastante diminuendo la resistenza generale. Inoltre, si è riscontrato che la resistenza e le fluttuazioni della velocità massima normale della parete sono altamente correlate a tutte le configurazioni di superficie, sia che riducano o aumentino la resistenza. Ciò implica che la struttura della turbolenza vicino alla parete è dominata dallo sforzo di taglio totale e non dalle condizioni al contorno locali delle superfici super-idrofobiche, sia infuse

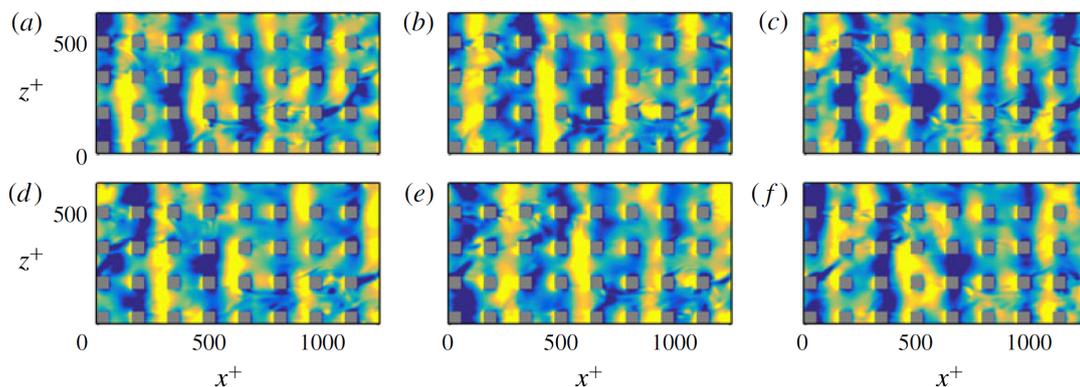


Figura 3.16: Istantanee successive di screen delle fluttuazioni della pressione a parete. La direzione del flusso è da sinistra a destra. Il range di fluttuazioni va da -10 a 10 unità di parete, dal blu al giallo. I tempi delle istantanee sono: (a) 0, (b) 2, (c) 4, (d) 6, (e) 8, (f) 10. Dati tratti da [16].

con liquido che ruvide. Nelle configurazioni di problema analizzate, si ha che la parete superiore del canale è liscia mentre la parete inferiore è formata da barre quadrate longitudinali, trasversali o cubi sfalsati, di diverse dimensioni, come si può vedere in figura 3.13. L'interfaccia tra i due fluidi è posizionata sul piano della cresta (superficie turchese nella figura 3.13), separando il flusso principale dal fluido nelle cavità. Per minimizzare il numero di parametri, invece di considerare le due viscosità dei fluidi, si va a prendere come parametro il rapporto fra le due.

Riassumendo in fine i risultati trovati in questo studio si ha che sono state eseguite simulazioni numeriche dirette di due fluidi sovrapposti (immiscibili) in un canale turbolento in cui uno dei fluidi è completamente racchiuso nelle strutture texurizzate periodiche del substrato. Sono stati valutati vari rapporti di viscosità per motivi di substrati bidimensionali (barre longitudinali e trasversali) e tridimensionali (pilastro cubico). I cubi sfalsati con una grande frazione di area fluida ($a = 0,875$) e barre quadrate longitudinali presentano la più alta riduzione della resistenza la quale aumenta al diminuire della viscosità del fluido all'interno del substrato [15]. Barre quadrate trasversali e cubi sfalsati con ($a = 0,5$), per questo particolare valore del rapporto tra le altezze, aumentano la resistenza complessiva poiché la resistenza di forma annulla gli effetti positivi della riduzione dello sforzo di taglio sopra le cavità. I risultati DNS hanno mostrato che lunghezza di scorrimento, velocità di scorrimento, attrito e trasciamento della forma e quantità di riduzione della resistenza variano in modo uniforme con il rapporto delle viscosità: il flusso all'interno del substrato riduce lo sforzo di taglio del flusso principale sopra il cavità attraverso una velocità di scorrimento. Inoltre confrontando i risultati per un singolo fluido e due fluidi separati da un'interfaccia, con lo stesso substrato, è stato dimostrato che l'interfaccia gioca un ruolo chiave nel ridurre la resistenza. L'interfaccia smorza le fluttuazioni di velocità normali alla parete e quindi limita il flusso di quantità di moto all'interno delle cavità. La deviazione standard delle fluttuazioni di velocità è strettamente connessa allo sforzo di taglio a parete con una forte correlazione tra la quantità di riduzione della resistenza e il massimo delle fluttuazioni di velocità normali alla parete. Anche se va oltre lo scopo dello studio riportato, mentre in questo è stata considerata l'interfaccia scivolosa ma non deforma-

bile nella direzione verticale, in realtà l'interfaccia può deformarsi. In questo caso, in base al numero di Weber e al modo in cui la deformazione dell'interfaccia è correlata alle fluttuazioni della velocità del flusso, il trasferimento del momento all'interno delle texture può essere ridotto solo parzialmente, con un conseguente effetto dannoso sulla resistenza. La quantità di riduzione della resistenza è inferiore a quella ottenuta con un'interfaccia piana e scivolosa. Inoltre, sono necessari ulteriori studi per ottimizzare la morfologia delle texture e le dimensioni delle cavità per evitare la fuoriuscita del fluido e quindi aumentare la durabilità della soluzione nel tempo [15].

Il secondo articolo che consideriamo è invece *Inner and outer layer turbulence over a superhydrophobic surface with low roughness level at low Reynolds number* di Abu Rowin, W., Hou, J., Ghaemia, S. [17]. In questo abbiamo che gli strati limite interno ed esterno di un flusso turbolento in un canale su di una superficie super idrofobica (SHS) sono caratterizzati utilizzando simultaneamente sia misure della velocità con tracciamento a lungo raggio di particelle microscopiche e misure della velocità attraverso immagini con particelle. Il flusso nel canale opera ad un determinato range. La SHS viene realizzata attraverso spray rivestente, e la radice quadratica media della rugosità è normalizzata dalle unità di parete. Si ottiene una riduzione di resistenza di circa il 19% in base alla pendenza del sottostrato viscoso lineare ed una del 22% in base all'espressione analitica di *Rastegari and Akhavan*. Il Re_τ si riduce sulla SHS, in base alla relativa velocità d'attrito. I risultati mostrano inoltre un incremento degli sforzi di Reynolds lungo il flusso, per le SHS, nello strato limite viscoso lineare a causa della condizione di scivolamento a parete. Se le sollecitazioni di Reynolds sono normalizzate in base alla corrispondente velocità di attrito, le sollecitazioni non dimensionali mostrano un grande aumento delle fluttuazioni di velocità quadratiche nella direzione longitudinale del flusso. La riduzione della resistenza è principalmente associata alla riduzione degli sforzi viscosi di taglio a parete, mentre la variazione degli sforzi di taglio di Reynolds alla parete è trascurabile [17].

Infine, l'ultimo articolo che si va a considerare è *Turbulent flows over superhydrophobic surfaces: flow-induced capillary waves, and robustness of air-water interfaces* di Jongmin Seo, Ricardo García-Mayoral and Ali Mani [16]. Qui abbiamo che le superfici superidrofobiche possono trattenere sacche di gas all'interno delle loro texture di microscala quando sono sommerse dall'acqua. Questa proprietà riduce il contatto diretto tra acqua e superfici solide e offre opportunità per migliorare le prestazioni idrodinamiche, riducendo la resistenza viscosa. Nella maggior parte delle applicazioni reali, tuttavia, il regime di flusso è turbolento e trattenere le tasche di gas risulta essere una difficile sfida. Per superarla, è fondamentale sviluppare una comprensione totale dei meccanismi fisici che possono portare al fallimento delle superfici superidrofobiche nel trattenere le sacche di gas quando il flusso del liquido sovrastante è turbolento. Viene qui presentato uno studio sull'insorgere di fallimenti nel trattenere i gas, analizzando simulazioni numeriche dirette (DNS) di flussi turbolenti su superfici superidrofobiche accoppiate con la deformazione delle interfacce aria-acqua, le quali trattengono le sacche di gas. Le superfici superidrofobiche sono realizzate come texture periodiche con condizioni modello di scorrimento e non scorrimento sul flusso d'acqua sovrastante. L'interfaccia liquido-gas è modellata tramite un'equazione linearizzata di Young-Laplace, che viene risolta accoppiata al flusso turbolento sovrastante. In questo studio viene considerata una vasta gamma di dimensioni delle trame e numeri Weber interfacciali. Questa analisi identifica le onde capillari che viaggiano a monte indotte dal flusso che sono coerenti nella direzione trasversale come un meccanismo per il fallimento nella ritenzione di

sacche di gas. Da ciò si evince la necessità, oltre che di ulteriori studi necessari al miglioramento della tecnica, anche di sistemi di controllo nel suo utilizzo, classificandola di fatto come una tecnica cosiddetta attiva. Per confermare la comprensione fisica di queste onde, viene sviluppata un'analisi semianalitica lineare inviscida; le velocità d'onda ottenute dalle correlazioni spazio-tempo nei dati DNS risultano corrispondenti alle previsioni del modello semianalitico. Si scopre che l'entità delle fluttuazioni di pressione dovute a queste onde aumenta con la diminuzione della tensione superficiale e aumenta con una dipendenza molto più forte dalla velocità di scorrimento, quando tutti i numeri sono riportati in unità a parete. Viene inoltre sviluppato un criterio di soglia per il fallimento di superfici superidrofobiche basato su stime delle condizioni di insorgenza richieste per il movimento delle linee di contatto. Il secondo contributo di questo lavoro è lo sviluppo di mappe dello strato limite che identificano zone stabili e instabili in uno spazio di parametri costituito da parametri di lavoro e di progettazione comprendenti dimensioni delle texture e materiali degli angoli di contatto. Viene riportata anche una breve descrizione delle modalità di fallimento precedentemente identificate delle superfici superidrofobiche, ossia la pressione di ristagno e i meccanismi di drenaggio legati agli sforzi di taglio. In una mappa sovrapposta, vengono presentate le zone stabili e instabili dovute a ciascun meccanismo. Per varie condizioni di input, sono riportate le leggi di ridimensionamento che identificano il meccanismo più critico che limita la stabilità della ritenzione di gas da parte delle superfici superidrofobiche [16]. In questo caso si va ad analizzare un flusso liquido turbolento, che scorre in un canale periodico, racchiuso da SHS, le cui pareti racchiudono sacche di gas, come viene riportato nella figura 3.14, in cui sono riportate due diverse configurazioni di SHS. Si ha inoltre, che queste configurazioni portano alla creazione di fittizie rugosità superficiali, date dalla presenza delle sacche di gas, come si può osservare nella figura 3.15(a). Qui, inoltre, nella 3.15(b) si può osservare un esempio di comportamento complessivo di una parete caratterizzata da SHS, su cui scorre un flusso turbolento. A causa delle deformazioni, causate da quest'ultimo, si ha che la posizione delle interfacce è definita dalla distanza dalle superfici piane.

Osservando, in breve, i risultati delle DNS, eseguite per le SHS, riportate in quest'articolo [16], si va a vedere cosa si è trovato. Come prima cosa si riportano i dati statistici della turbolenza misurata nelle DNS. Grazie a questa si è potuto sia investigare il comportamento del flusso nel canale con le SHS, e sia l'impatto della deformabilità dell'interfaccia liquido-gas sul flusso. Dopo di che, si vanno a vedere proprio le deformazioni delle interfacce, causate dal flusso turbolento, ed anche le fluttuazioni di pressione che si hanno su queste interfacce. Le fluttuazioni di pressione sull'interfaccia deformabile sui cubi isotropi hanno strutture distinte e coerenti trasversalmente, come si può osservare nella figura 3.16. Inoltre la pressione di ristagno indotta dai flussi di scorrimento, come nel caso di un'interfaccia piatta, rimane anche nel caso di un'interfaccia deformabile, mentre la pressione coerente trasversale risulta essere sovrapposta alla pressione di ristagno. Da questo è possibile vedere e valutare anche le caratteristiche delle onde di pressione nel tempo e nello spazio. Dai dati si osserva anche una dipendenza dai numeri di Reynolds, prendendo i dati a 2 differenti valori, $Re_\tau = 200$ e $Re_\tau = 400$. Viene anche valutato l'effetto della tensione superficiale, la quale è strettamente legata a quello delle fluttuazioni turbolente ed alle deformazioni delle superfici. Si vanno a vedere anche gli effetti dovuti alla dimensione delle texture. Come si osserva in fig 3.17, il variare della dimensione dei cubi isotropi della texture, incide molto sulla distribuzione delle onde di pressione, e sulla loro intensità, in quanto

cambia l'impatto che il flusso ha sulle superfici SHS. Si va infine a riassumere quanto effettuato e riportato nello studio presentato [16].

Si riporta un'indagine sul comportamento dinamico delle interfacce gas-liquido su una superficie superidrofobica in risposta alla turbolenza idrodinamica in un flusso sovrastante. Le simulazioni DNS del flusso turbolento in un canale su di una vasta gamma di parametri sono state sviluppate per eseguire questa indagine. Le DNS tengono conto della fisica della superficie superidrofobica tramite condizioni al contorno di modelli di scorrimento / non scorrimento sul flusso sovrastante e deformabilità dell'interfaccia aria-acqua tramite una linearizzazione dell'equazione di Young – Laplace. Le nostre ricerche hanno identificato le onde capillari indotte dal flusso come un meccanismo in cui la turbolenza può eccitare le modalità capillari sotto forma di onde coerenti che risalgono all'inizio del flusso. Mentre le statistiche cinematiche, come la lunghezza di scorrimento, non sono sensibili alla presenza di onde capillari, i campi di pressione sono fortemente influenzati da questi modi. Tramite varie analisi, inclusa l'identificazione dei rapporti di scala per il collasso dei dati e l'analisi semianalitica lineare inviscida, sono stati sviluppati approfondimenti sul comportamento delle onde capillari indotte dal flusso. Le conoscenze acquisite dall'indagine sui dati DNS hanno portato allo sviluppo di criteri di soglia per il fallimento di superfici superidrofobiche in condizioni realistiche. A tal fine, l'inizio del movimento della linea di contatto è stato utilizzato per quantificare una condizione di errore, portando alle mappe dei confini tra zone stabili e instabili nei parametri spaziali delle lunghezze caratteristiche espresse in unità di parete, considerando vari scenari dell'angolo di contatto fra le superfici e per i flussi.

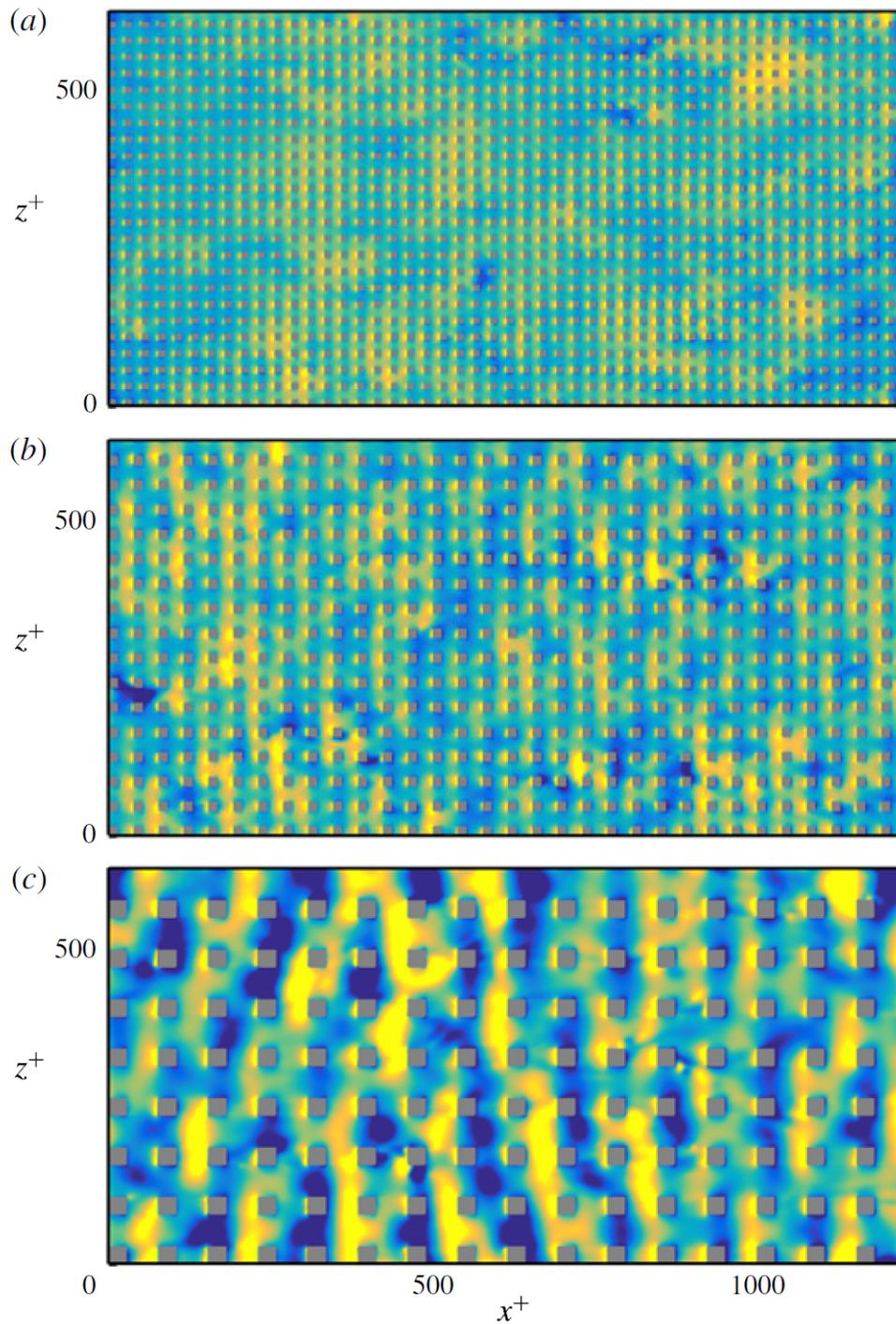


Figura 3.17: Istantanee delle linee di pressione a parete, per diverse dimensioni delle texture, in particolare si ha: (a) $L^+ \approx 26$, (b) $L^+ \approx 38$ e (c) $L^+ \approx 77$. Il range di fluttuazioni va da -10 a 10 unità di parete, dal blu al giallo. Dati tratti da [16].

3.4 Tecniche passive di riduzione della resistenza

Si vanno ora a vedere alcuni metodi, tra i più studiati ed utilizzati, di riduzione della resistenza passivi. Questi, a differenza di quelli attivi, non richiedono né energia, né sistemi di controllo, durante il loro utilizzo, rendendoli meno complicati da realizzare ed utilizzare, seppur al costo di un impatto minore sulla resistenza. Prima di entrare nello specifico di alcuni di questi metodi, si va a vedere un insieme di questi, descritti nell'articolo *Turbulent drag reduction by passive mechanisms* di L. Sirovich & S. Karlsson [18]. Viene qui riportato come in molte situazioni che coinvolgono flussi di elevato numero di Reynolds (ovvero laddove le forze d'inerzia dominano su quelle viscosi), come il volo degli aerei e il trasporto di combustibili mediante condotte, la resistenza turbolenta è un fattore importante che limita le prestazioni. Ciò ha portato ad una approfondita ricerca di metodi attivi e passivi per la riduzione della resistenza. Nell'articolo sono riportati i risultati di una serie di esperimenti in galleria del vento che dimostrano un metodo passivo per controllare efficacemente la turbolenza nel flusso del canale. L'approccio prevede l'introduzione di specifici modelli di sporgenze sulle pareti confinate, che interagiscono con le strutture vorticosi coerenti e portanti energia nella regione della parete, influenzando quindi la velocità con cui l'energia viene dissipata nel flusso turbolento. Si mostra che cambiamenti relativamente piccoli nella disposizione di queste sporgenze possono alterare la risposta del sistema da una riduzione della resistenza ad un aumento del rimescolamento (con aumento della resistenza). Si riporta la realizzazione dell'esperimento, mediante un flusso completamente sviluppato turbolento in un canale, di cui si riportano i dati. Dato il tipo di tecnica impiegata, riveste un ruolo importante la comprensione del comportamento delle *strisce* di turbolenza, responsabili del rimescolamento. Dagli esperimenti emerge che queste sono strutture tubolari, di vorticità positiva e negativa rispetto al flusso, la quale si estende in direzione del flusso per oltre 1000 unità di lunghezza d'attrito adimensionali. Lo scalare semplicemente gli argomenti suggerisce che la distanza tra i vortici dovrebbe crescere linearmente con la distanza normale dalla parete, e ciò è confermato sperimentalmente. Si sa inoltre come queste strutture, andando incontro ad un rapido ciclo di eventi, terminano la loro cascata nella regione di parete, culminando nella dissipazione, il cui risultato è quello di rallentare tutto il flusso a contatto con la parete. Le onde, che sono orientate obliquamente rispetto alla direzione del flusso, subiscono interazioni tridimensionali con i modelli di rimescolamento. La più intensa di queste interazioni ha luogo con onde che si propagano in un intervallo ben definito di angoli, e che hanno scale di lunghezza paragonabili alla dimensione dei vortici. Una recente indagine numerica, fornisce ulteriori prove dell'importanza dell'interazione tra le modalità di propagazione selezionate e le modalità di rimescolamento portanti energia per il ciclo di eventi che si verificano nel flusso. L'importanza di queste interazioni è stata ulteriormente sottolineata in recenti studi nei quali le onde propaganti rilevanti ricevono una randomizzazione della fase regolare, durante lo svolgimento delle simulazioni. Questa serie di osservazioni ha suggerito che la gestione della turbolenza delle pareti potrebbe essere raggiunta con sporgenze sulle pareti, scelte per stimolare, in modo mirato, appropriate modalità di propagazione, ed è questo che si è poi andati a testare nel canale. Nello specifico il canale utilizzato è presente una camera di sedimentazione in cui il flusso entra verticalmente attraverso fori da 4 mm , sia nel pavimento che nel soffitto del canale. Dopo di che, questo è seguito da una superficie ruvida di $0,9\text{ m}$. Il resto del canale, poi, è coperto con i motivi delle sporgenze, o lasciato liscio. L'altezza dal pavimento al

soffitto è di 5,63 cm, indipendentemente dal fatto che il pavimento sia coperto o meno da sporgenze. All'interno della camera è ospitato un sistema a filo caldo che fornisce misurazioni a parete normale e trasversale. Vengono in questo modo testati una gran varietà di sporgenze di parete; nello studio, comunque, ci si va a concentrare sugli esperimenti realizzati su due di queste configurazioni, una con le sporgenze disposte in modo casuale ed una con le sporgenze allineate in direzione del flusso. Per le altre testate, vengono fatti solo dei brevi cenni. Lo schema viene scelto per produrre onde che interagiscono al massimo con i modelli di vorticità trasportanti energia. L'obiettivo è prevenire la formazione dei vortici, e romperli laddove si formassero. Viene effettuata una stima della dissipazione ottenibile, ma se invece i vortici dovessero seguire la via della cascata energetica verso la dissipazione, in maniera forzata dalle sporgenze, la stima cambia. Pertanto, all'interno di questa approssimativa rappresentazione dei fenomeni, la dissipazione energetica viene ritardata e quindi la velocità di dissipazione dovuta ai soli vortici viene sostanzialmente ridotta. Da ciò si può dedurre che le modalità di rimescolamento ora trasportano meno energia rispetto al caso standard a parete liscia. La premessa principale di questa semplice modellizzazione degli eventi è che impedendo la creazione delle *strutture coerenti* dei vortici ritardiamo il tempo impiegato per raggiungere lo stadio finale di dissipazione che causa perdita di energia. Vengono inoltre effettuate ulteriori misure per supportare questa visione. Dalle misure effettuate, facendo dei confronti emerge anche che, le pareti con sporgenze hanno una frequenza di dissipazione significativamente più bassa, ed anche una rilevante riduzione nella frequenza di rimescolamento.

È stato inoltre recentemente dimostrato che se si separano i contributi agli sforzi di taglio di Reynolds medi, in rimescolamento e propagazione, allora i modi di rimescolamento danno il contributo principale vicino alla parete, mentre i modi di propagazione contribuiscono principalmente nel cuore del flusso. Interferendo con la formazione dei modi di rimescolamento, come le sporgenze fanno, si può diminuire il contributo dato da questi modi. Un'argomentazione diretta indica che se sia il pavimento che il soffitto del canale sono coperti con il materiale con sporgenze, l'effetto ottenibile si dovrebbe almeno raddoppiare. Sfortunatamente, il design del canale non lo ha permesso e non è stato possibile realizzare e testare questa idea nello studio riportato [18]. Tuttavia, comunque, quasi senza eccezione, ogni esperimento con il pavimento a motivi casuali, effettuato nello studio, ha mostrato una riduzione della resistenza.

Effetto di attenuazione del rimescolamento utilizzando micro-cavità nello strato limite turbolento

Si va a vedere ora l'articolo *Mechanism of sweep event attenuation using micro-cavities in a turbulent boundary layer* di A. Silvestri, F. Ghanadi, M. Arjomandi, B. Cazzolato, A. Zander & R. Chin [19], in merito all'utilizzo di micro cavità nello strato limite, sfruttanti l'effetto di risuonatori di Helmholtz, per ridurre la resistenza dello strato limite turbolento, mediante l'attenuazione dei fenomeni di rimescolamento. In questo articolo abbiamo che le schiere di cavità sono state identificate come un potenziale dispositivo passivo per interrompere e catturare eventi di rimescolamento, che sono responsabili degli sforzi di Reynolds elevati nello strato limite. Nel presente studio, è stato analizzato il meccanismo dell'attenuazione degli eventi di rimescolamento che vengono catturati, nonché la relazione non lineare tra il volume della cavità sottostante e la riduzione dell'intensità del rimescolamento. È stata analizzata l'influenza della schiera di cavità sullo strato limite turbolento, con un totale di 6 diverse disposizioni di

cavità sottostanti con volumi variabili. 3 delle cavità sottostanti sono state utilizzate per determinare la relazione non lineare tra l'efficacia della schiera di cavità nel ridurre l'intensità di rimescolamento e il volume della cavità sottostanti. Gli altri 3 sono stati usati per determinare il meccanismo mediante il quale le schiere manipolano gli eventi di rimescolamento catturati. Gli spettri di energia pre-moltiplicata di diverse storie di velocità vengono significativamente ridotti, fino al 12,5%, nei valori di lunghezza d'onda, bassa e media, associati alle strutture coerenti. I risultati mostrano che la riduzione massima dell'intensità di rimescolamento di circa il 7% può essere ottenuta quando $Re_\theta = 3771$. È stato dimostrato che la relazione non lineare tra la riduzione dell'intensità del fenomeno di rimescolamento ed il volume delle cavità raggiunge un limite superiore nell'analisi riportata. I risultati di questo studio hanno anche rivelato che la schiera di cavità indebolisce l'intensità di rimescolamento di questi fenomeni catturati, smorzando l'energia dei fenomeni attraverso le perdite d'attrito nella schiera di cavità e anche nel grande volume sottostante le cavità.

Entrando maggiormente nel dettaglio si ha che la resistenza d'attrito superficiale è un importante problema per molte applicazioni a livello mondiale, come per esempio l'industria aeronautica, nella quale la resistenza d'attrito superficiale per gli aerei è stimata essere il 50% della resistenza totale. Di conseguenza, molte ricerche significative sono state portate avanti per ridurre l'influenza della resistenza d'attrito superficiale negli strati limite turbolenti. Due di questi metodi passivi molto analizzati di recente sono le riblet (che verranno meglio analizzate più avanti, vedi 3.4.1) e i risuonatori di Helmholtz. Un risuonatore di Helmholtz eccita un flusso in modo simile ad un getto sintetico, ma a differenza di questo, si tratta di una tecnica e di un sistema di controllo completamente passivo, e il meccanismo di funzionamento è dato dal flusso passante. Gli obiettivi dei risuonatori di Helmholtz sono le strutture coerenti che si hanno nei flussi turbolenti, e i risultati di diversi studi, ci dicono che è possibile una riduzione di intensità e durata di questi del 11% e del 5%, rispettivamente. Anche quando poi i risuonatori non sono attivati correttamente dal flusso, si hanno comunque riduzioni del 5% e del 8% in intensità e durata, senza comunque la necessità di avere aspirazione e soffiaggio. Oltre al lavoro sull'ottimizzazione della geometria esterna della schiera di cavità (cioè il diametro dell'orifizio), è stata anche condotta una significativa ricerca sulla determinazione dell'effetto della geometria interna (ovvero la lunghezza dell'orifizio ed il volume della cavità sottostante). Tuttavia, uno dei maggiori limiti allo studio di questa tecnica è il numero limitato di casi sperimentali con cui identificare la relazione non lineare tra l'aumento del volume della cavità e la riduzione dell'intensità di rimescolamento. Mentre era ipotizzabile che la relazione avrebbe raggiunto un limite superiore appiattendosi, non ci sono stati riscontri sperimentali a riguardo. Grazie a questo studio sono state formulate due ipotesi supportate dalla letteratura pertinente in merito a questa soluzione. L'ipotesi iniziale degli autori [19] suggerisce che la schiera di cavità va ad indebolire la struttura dei fenomeni di rimescolamento catturati, smorzando l'energia di questi fenomeni attraverso le perdite d'attrito che si hanno nei fori e nel grande volume della cavità sottostante situato sotto la schiera. L'ipotesi aggiuntiva fornita dagli autori suggerisce che anche una volta che un fenomeno di rimescolamento entra nella schiera di cavità, l'energia del singolo fenomeno viene ridistribuita in tutte le cavità della schiera di queste. Ciò comporterebbe che un fenomeno di rimescolamento viene rimosso dallo strato limite e distrutto in un processo di dissipazione. Ciò richiederebbe ulteriori studi per provare ciò, ma questo fenomeno rimane senza risposta nell'articolo considerato. In questo, in particolare, ci si va a concentrare sulla relazione

non lineare fra l'aumento del volume delle cavità e il flusso di controllo, per trovare un limite massimo al volume delle cavità, ottenendo comunque il massimo possibile di energia turbolenta e rimescolamento dissipato.

In tutti gli esperimenti svolti riguardanti questa tecnica, la schiera di cavità si trova su

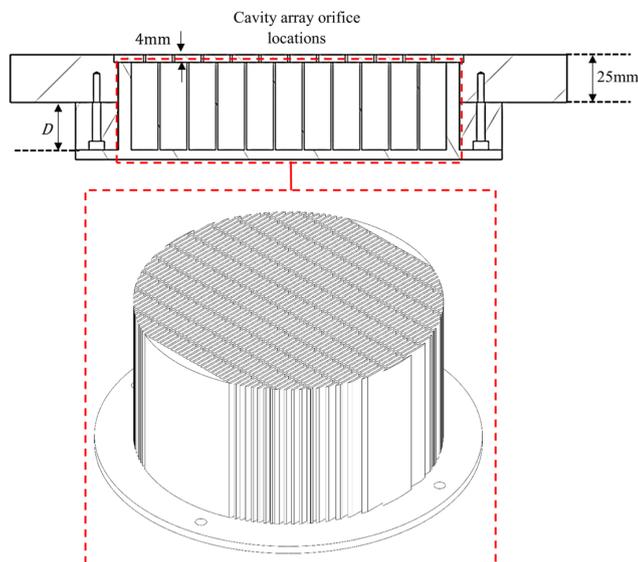


Figura 3.18: Schema della schiera di cavità con cavità sottostante individuale con uso di mesh, con visione isometrica della mesh sottostante [19].

di una lamina piana, posta in una galleria del vento. La lamina ha delle caratteristiche particolari per evitare il più possibile la separazione del flusso. Nella figura 3.18 viene riportato uno degli schemi di cavità che è stato analizzato. In questo ogni singola cavità possiede un suo volume sottostante individuale, a differenza della configurazione con un volume comune unico. La velocità lungo il flusso e i profili di intensità turbolenta dello strato limite turbolento sono misurati a valle della schiera di cavità, mediante l'uso di un anemometro a filo caldo. Si misurano anche gli stessi dati per uno strato limite inalterato dalle cavità (queste si trovano solamente su uno dei due lati della piastra, per lasciare l'altro appunto indisturbato e poter effettuare confronti). Dal confronto delle due misurazioni si ha che, mentre il profilo di velocità media lungo il flusso dello strato limite non viene influenzato dalla schiera di cavità, l'intensità di turbolenza cambia. Questa è infatti una grandezza statistica che misura l'intensità della turbolenza nello strato limite. Nella figura 3.19 sono riportati gli andamenti di alcuni profili di intensità turbolenta che si hanno a valle della schiera, sia per un caso senza cavità, e sia per alcune configurazioni differenti di cavità, in particolare sia nel numero, che nel tipo di volume sottostante. Tutte le configurazioni di schiere di cavità danno riduzioni significative dell'intensità turbolenta nella regione logaritmica, indicante una significativa riduzione delle fluttuazioni di velocità. Nella figura 3.19 quest'area viene ingrandita per poter meglio apprezzare questo effetto. La schiera di cavità è quindi una tecnica con successo nel controllo dello strato limite, indipendentemente dalla singola configurazione, in quanto si ha che le schiere indeboliscono le strutture dei fenomeni di rimescolamento catturati attraverso lo smorzamento di energia dei fenomeni attraverso le perdite d'attrito che si hanno nelle singole cavità e nel volume sottostante. Si ha che quindi le micro cavità modificano le strutture coerenti trovate nelle regioni

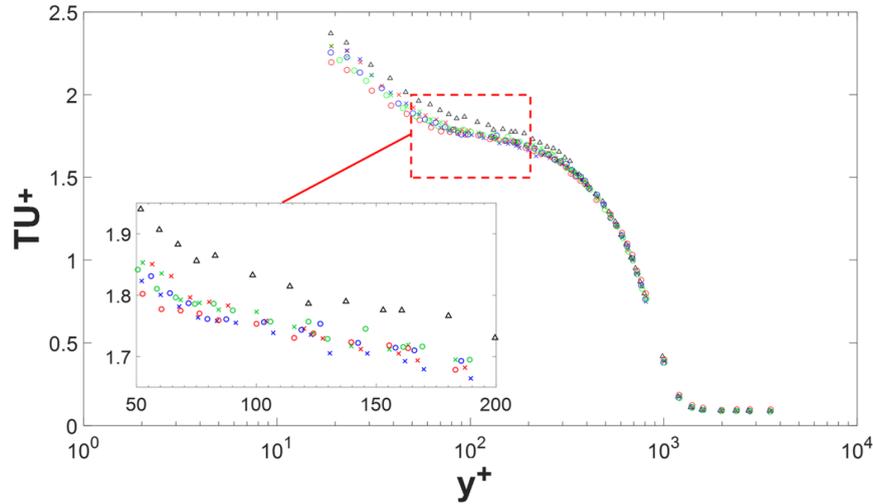


Figura 3.19: Andamento dell'intensità di turbolenza per $Re_\theta = 2939$ a valle della schiera. I simboli triangolari indicano nessuna cavità, mentre gli altri riportano diverse configurazioni di cavità [19].

vicino a parete dello strato limite. Questo come aggiunta all'ipotesi che le cavità riducessero i fenomeni di rimescolamento attraverso la cattura delle strutture per ottenerne la distruzione nei processi dissipativi. Si trova inoltre che entrambi i tipi di configurazione di cavità esaminati, sia quello con il volume sottostante comune, e sia quello con il volume sottostante individuale diviso in mesh, svolgono molto bene il compito di controllo del flusso. Questo è associato all'indebolimento delle strutture dei fenomeni catturati.

Inoltre, un ulteriore obiettivo dello studio riportato è stato comprendere meglio la relazione non lineare tra l'aumento del volume della cavità sottostante e la riduzione dei fenomeni di rimescolamento, e per poter fare ciò si è andati ad analizzare un'ulteriore configurazione di cavità, oltre a quelle inizialmente analizzate. Si trova che, mentre il valore massimo di dissipazione è superiore al valore precedentemente registrato, per il caso base, la riduzione che si ottiene varia molto poco tra le due configurazioni con i maggiori volumi sottostanti le cavità, per entrambi i numeri di Reynolds studiati. I due maggiori volumi sottostanti le cavità, per ciascun numero di Reynolds, mostrano una differenza trascurabile dello 0,2% nella riduzione dell'intensità di rimescolamento ottenuta, illustrando appiattimento della riduzione ottenuta.

È stato dimostrato che le micro-cavità hanno un effetto significativo nella modifica delle strutture coerenti che si hanno negli strati limite turbolenti e colpendo queste strutture provoca l'interruzione del ciclo di dissipazione, che è responsabile degli sforzi di taglio nella regione vicino a parete. Tuttavia, nonostante le misurazioni quantitative mostrino una riduzione dell'intensità di turbolenza, dell'intensità di rimescolamento e degli spettri energetici, l'esatto meccanismo che regola la riduzione è ancora sconosciuto. Nello studio presentato [19] è stata analizzata una schiera di cavità su piastra singola con una varietà di diverse cavità sottostanti con volumi e disposizioni diversi. L'anemometria a filo caldo a valle delle schiere esaminate è stata utilizzata per misurare lo strato limite,

dove i risultati sono stati quindi utilizzati per valutare il profilo di velocità e intensità della turbolenza, le strutture coerenti e lo spettro di energia. Si è stato dimostrato che una schiera di cavità indebolisce la struttura dei fenomeni di rimescolamento catturati smorzando l'energia di questi attraverso le perdite per attrito nel collo di ciascun orifizio ed attraverso l'elevata dissipazione di energia data dal grande volume sottostante le cavità, sotto la schiera. Inoltre, è stata studiata la vera relazione tra il volume della cavità sottostante e la riduzione dell'intensità del rimescolamento. Si è trovata essere una relazione non lineare con un limite superiore e un appiattimento per volumi di cavità sottostanti più grandi. Le conclusioni trovate si basano sull'alterazione del volume e della geometria della cavità sottostanti. Si è fatto un significativo passo avanti nel futuro del controllo dello strato limite turbolento attraverso questa tecnica. È stata stabilita una chiara comprensione della capacità delle schiere di cavità di catturare le strutture di rimescolamento e ridurre l'impatto delle successive strutture coerenti, consentendo a lavori futuri di visualizzare il flusso dello strato limite sopra e nella schiera di cavità.

Parete porosa anisotropa

Si riporta un breve estratto di un articolo riguardante un'altra tecnica passiva di riduzione della resistenza d'attrito dello strato limite turbolento. La tecnica in particolare è quella dell'utilizzo di una parete porosa anisotropa, e viene descritta nell'articolo *Turbulent channel flow over an anisotropic porous wall - Drag increase and reduction* di M. E. Rosti, L. Brandt & A. Pinelli [20]. Qui abbiamo che l'effetto delle variazioni del tensore di permeabilità sul comportamento vicino alla parete di un flusso turbolento in un canale, delimitato da pareti porose, viene studiato usando una serie di simulazioni numeriche dirette. Si riscontra che la resistenza totale può essere ridotta o aumentata di oltre il 20% regolando le proprietà direzionali della permeabilità. La riduzione della resistenza è ottenuta per il caso di materiali con permeabilità, nella direzione verticale, inferiore a quella dei piani paralleli alla parete. Questa configurazione limita la velocità normale alla parete all'interfaccia, promuovendo al contempo un aumento della velocità di scorrimento tangenziale, che porta a una turbolenza quasi "univoca" in cui la coerenza delle strisce a bassa e alta velocità è fortemente migliorata. D'altra parte, un forte aumento della resistenza si riscontra quando sono utilizzate permeabilità alte alle pareti normali e basse alle pareti parallele. In questa condizione, il miglioramento delle fluttuazioni normali alle pareti, a causa del ridotto effetto di blocco delle pareti, innesca l'insorgenza di strutture fortemente correlate nella direzione trasversale, un fenomeno osservato anche nei flussi su strati porosi isotropi o su pareti con riblet con altezze di sporgenza troppo elevata. L'uso di pareti porose anisotrope per la riduzione della resistenza è particolarmente attraente poiché è possibile ottenere uguali guadagni a numeri di Reynolds diversi solo riscaldando l'entità della permeabilità [20].

3.4.1 Riblet

Come ultima tra le tecniche passive presentate di riduzione della resistenza d'attrito in un flusso turbolento, troviamo quella delle riblet. Questa tecnica consiste nel realizzare sulle pareti su cui scorre il flusso turbolento delle microscalanature, che andando ad interagire con le strutture che si trovano nello strato limite, le vanno ad alterare causando una riduzione di resistenza. Data la grande importanza di questa tecnica per questo lavoro di tesi, essendo che questo si basa proprio sullo studio di alcune configu-

razioni di questa soluzione, verranno riportati ora alcuni articoli riguardanti studi che sono stati effettuati in merito a questa soluzione, per poter avere una migliore comprensione delle conoscenze acquisite in merito fin ora, avere una buona base di partenza per gli studi che verranno poi effettuati in questa tesi, e avere anche dati con cui, in caso di studi molto simili, potercisi confrontare.

Drag reduction with riblets in nature and engineering di D.W. Bechert & W. Hage

Il primo articolo in merito che viene riportato è *Drag reduction with riblets in nature and engineering* di *D.W. Bechert & W. Hage* [21]. Qui viene detto che la riduzione dell'attrito a parete dello strato limite turbolento può essere ottenuta utilizzando superfici con proprietà correlate alle strutture superficiali presenti in natura. Per esempio le squame di diverse specie di squali a nuoto veloce mostrano una minuscola struttura a scanalature. Questa struttura nervata ostacola lo scambio di quantità di moto a parete, che è l'origine degli sforzi di taglio turbolenti. Il meccanismo di riduzione della resistenza delle superfici nervate (o riblet) è spiegato usando un modello analitico che definisce la regione viscosa del flusso, dove le riblet sono completamente immerse nel sottostrato viscoso dello strato limite turbolento. La riduzione degli sforzi di taglio a parete è stata studiata sperimentalmente per varie superfici con riblet, tra cui una replica di pelle di squalo composta da 800 modelli in scala di plastica con ancoraggi conformi. Si tratta anche l'applicazione di questi tipi di strutture agli aeromobili commerciali a lungo raggio, compreso l'effetto delle riduzioni della resistenza sui costi operativi dell'aeromobile. Entrando maggiormente nel dettaglio, soprattutto per quanto riguarda la parte sull'origine di questa tecnica, ispirata alla pelle degli squali, abbiamo che la riduzione di resistenza per un animale o di un veicolo in movimento può essere ottenuta mediante una riduzione degli sforzi di taglio a parete e, in alcune situazioni, mediante il controllo della separazione del flusso. Qui ci si concentra sulla riduzione degli sforzi di taglio turbolenti a parete.

A differenza degli alianti che hanno flussi laminari e quindi il minor sforzo possibile, sui

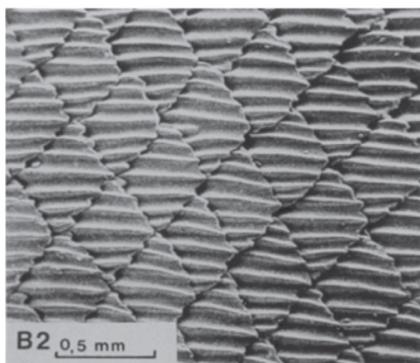


Figura 3.20: Microstrutture sulla pelle dello squalo delle Galapagos. Fotografato da W.E: Reif [21].

velivoli commerciali, i problemi associati all'implementazione del flusso laminare su ali transoniche non sono attualmente ancora completamente risolti. Anche quando il flusso laminare viene mantenuto su una percentuale maggiore dell'ala, la maggior parte della superficie dell'aeromobile avrà comunque uno strato limite turbolento. La riduzione

turbolenta degli sforzi di taglio a parete, pertanto, rimane importante in tutti gli scenari della futura progettazione degli aeromobili. Quindi è importante capire come ridurre gli sforzi per un flusso turbolento. Le superfici con riblet sono state ideate osservando le microstrutture presenti sulla pelle degli squali, come si può osservare nella figura 3.20; qui si sono osservate interessanti microstrutture, di scala molto piccola, tipicamente di un millesimo di millimetro, le quali sono allineate nella direzione del flusso, con una relativa distanza trasversale. Queste osservazioni hanno portato ad effettuare i primi studi sulle riblet, ovvero su tecniche che andassero a copiare queste microstrutture. Inizialmente queste presentavano problemi di realizzazione, viste le loro estremamente ridotte dimensioni. Il flusso turbolento vicino a una parete liscia piana presenta deviazioni istantanee molto significative dalla direzione media del flusso. Si hanno grandi scambi di quantità di moto nello strato limite turbolento. Per l'ottimizzazione della

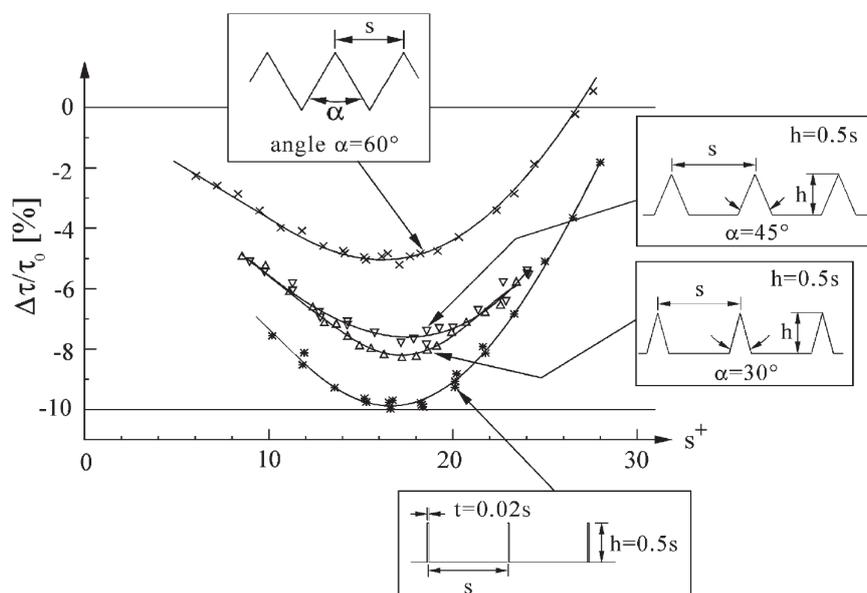


Figura 3.21: Prestazioni di riduzione di resistenza per diverse geometrie di riblet [21].

riduzione di sforzo di taglio, sono state selezionate ed analizzate diverse superfici con riblet. Si trova che le massime prestazioni si hanno per riblet formate da lame verticali molto sottili. Tuttavia questa soluzione presenta grandi problemi realizzativi ed applicativi. Oltre a questa, nella figura 3.21, si possono osservare le prestazioni di alcune delle configurazioni analizzate. Quello che si osserva è che le prestazioni migliori si hanno per le soluzioni di più difficile realizzazione ed uso pratico, mentre al contrario, le schiere di riblet consecutive, di più facile realizzazione, hanno quelle inferiori. Si è provato inoltre ad emulare in diversi modi la costituzione delle superfici della pelle degli squali, ma tutti i tentativi hanno mostrato una diminuzione della resistenza minore. Si ci è quindi posti il dubbio se ci fosse anche altri meccanismi alla base della pelle degli squali a rendere questa superiore alle superfici plastiche fabbricate. Un possibile meccanismo potrebbe derivare dall'ancoraggio conforme delle squame che si verifica sulle pelli reali, consentendo loro di muoversi liberamente, a differenza di quanto accade per quelle costruite. In merito a ciò si prospettano diversi sviluppi futuri con potenzialità interessanti. Nella figura 3.22 si possono osservare quelle che sono le differenze fra la reale composizione della pelle dello squalo martello in natura, e come la si è andati a

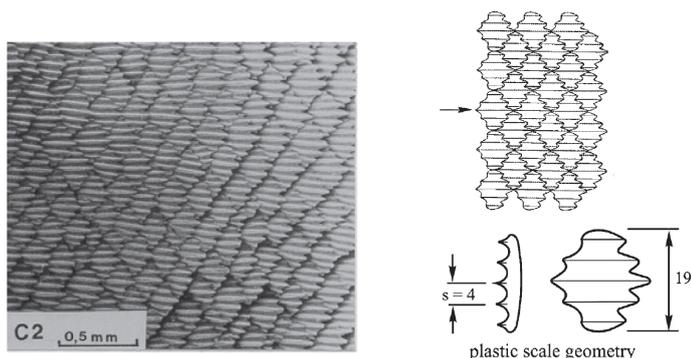


Figura 3.22: Pelle di squalo martello in natura e riprodotta artificialmente per simulazioni [21].

simulare in laboratorio.

Dopo aver analizzato le riblet su di una configurazione piana, ovvero su di una super-

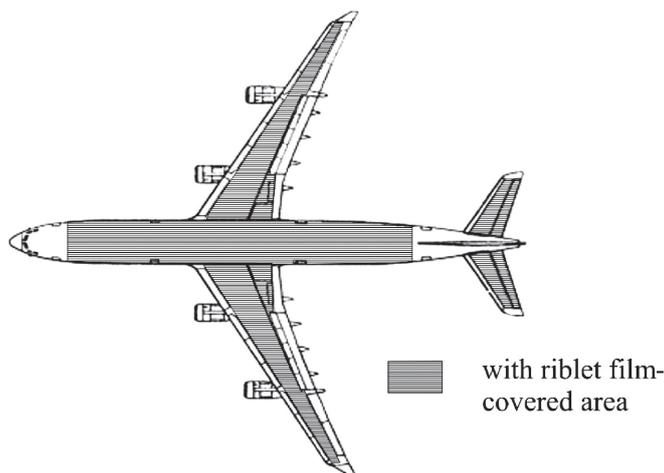


Figura 3.23: Zone su cui sarebbe possibile applicare il film con riblet su di un aereo a lungo raggio commerciale. Il film non sarebbe applicato sotto le ali e sulla metà inferiore della fusoliera [21].

fice senza inclinazione, si è andati a vedere quali possono essere le effettive applicazioni aeronautiche, mediante l'applicazione di queste su superfici portati, o comunque con incidenza rispetto al flusso. Oltre alla prevista riduzione della resistenza misurata, si è registrato anche un leggero aumento del coefficiente di portanza. L'ordine di grandezza di questo aumento di portanza è dell'1%. Le riblet riducono anche lo spessore di spostamento dello strato limite e ciò aumenta efficacemente la campanatura del profilo aerodinamico e quindi produce un profilo alare con portanza più elevata. Attualmente, tuttavia, le superfici con riblet non sono ancora ampiamente utilizzate negli aerei commerciali, e i motivi di ciò coinvolgono sia motivi razionali che non. I motivi razionali riguardano preoccupazioni tecniche, come la qualità dell'adesione delle superfici, possibili danni dai raggi UV e possibile riduzione della riduzione di resistenza causata dalla sporcizia nelle riblet. Attualmente comunque questi motivi non sono più validi, in quanto ci sono stati notevoli miglioramenti nei materiali plastici ed adesivi, i quali sono

anche resistenti ai raggi UV, e repellenti alla sporcizia. Inoltre la prospettiva di avere un velivolo rivestito con una superficie di film repellente allo sporco può offrire un'ulteriore attrattiva per quanto riguarda i costi di manutenzione. Oltre a questi argomenti, c'è la questione dei costi di applicazione, in quanto si ritiene, erroneamente, che nel tempo che ci vuole per applicare il film con riblet su un aereo, circa una settimana, l'aereo non vola e quindi non guadagna. Sarebbe comunque possibile fare ciò durante le manutenzioni programmate, insieme ad altri lavori, in quanto applicare riblet non richiede uso di vernici o fumi pericolosi per la salute di altri operatori. Il motivo irrazionale consiste nel pregiudizio dei dipartimenti di marketing e dei produttori di aeromobili di alcune compagnie aeree secondo cui il luccicante film di riblet di plastica può sembrare insolito e quindi poco attraente per i clienti. Questo, ovviamente, potrebbe essere facilmente modificato da un'intelligente campagna di marketing, considerando invece i notevoli risparmi che si avrebbero dall'uso di questa soluzione. Infatti il vantaggio dell'utilizzo di nervature sui velivoli risiede sia nel minor consumo di carburante che nella conseguente riduzione del peso. Sugli aeromobili a lungo raggio si ha che sia i costi del carburante contribuiscono in modo significativo ai costi operativi diretti e sia che il peso del carburante supera di gran lunga il carico utile. Inoltre alcune delle rugosità sulla superficie dell'aeromobile verrebbero così coperte e levigate dal film plastico delle riblet. Inoltre, la riduzione dell'attrito della parete garantisce uno strato limite leggermente più sottile che, a sua volta, provocherebbe una riduzione della resistenza della forma sulla parte posteriore della fusoliera. Sarebbe quindi possibile ottenere una riduzione del 2% della resistenza totale dell'aeromobile. Nella figura 3.23 è possibile osservare le zone su cui si potrebbe applicare questa soluzione.

Infine, si vanno a vedere le conclusioni riportate nello studio esaminato [21]. Le proprietà di riduzione della resistenza delle superfici con riblet sono state ben comprese. Si è andati a stabilire una teoria viscosa che copre il regime di flusso in cui le riblet sono completamente coperte dal sottostrato viscoso dello strato limite turbolento. Questa teoria porta alla scelta di una geometria delle riblet per fornire la massima riduzione della resistenza, che poi è stato confermato sperimentalmente. Sebbene sia stata fatta un'attenta indagine per esplorare le fluide proprietà meccaniche della pelle degli squali veloci, non sono state trovate ulteriori informazioni sui segreti della pelle di squalo. L'idea che la flessibilità della pelle potesse mostrare un'ulteriore riduzione della resistenza non è stata supportata dai risultati. D'altra parte, si dovrebbe tenere presente che la pelle deve essere necessariamente flessibile. Altrimenti, gli squali avrebbero probabilmente sviluppato una struttura più rigida, che sarebbe più vicina alla struttura "tecnica" della riblet. È stato inoltre più volte dimostrato che le riblet possono essere utilizzati per ridurre il consumo di carburante degli aeromobili.

Drag reduction by riblets di R. Garcia-Mayoral & J. Jimenez

Il secondo articolo riportato è *Drag reduction by riblets* di R. Garcia-Mayoral & J. Jimenez [22]. In questo troviamo che vengono analizzate l'interazione del flusso turbolento sovrastante con le riblet e il loro impatto sulle proprietà di riduzione della resistenza. Nel cosiddetto spazio del regime viscoso occupato dalle riblet, la riduzione della resistenza è proporzionale alla dimensione della riblet, ma per riblet più grandi la proporzionalità si interrompe e la riduzione della resistenza diventa un aumento. Si è riscontrato che la sezione trasversale della riblet A_g^+ è una caratterizzazione migliore di questa suddivisione rispetto alla spaziatura della riblet, con un $A_g^{+1/2} \approx 11$ ottimale.

Si è anche scoperto che il punto di rottura non è associato alla sistemazione di vortici quasi-longitudinali all'interno delle scanalature della riblet, o all'inapplicabilità dell'ipotesi di Stokes al flusso lungo le scanalature, ma alla comparsa di vortici trasversali quasi bidimensionali sotto $y^+ \approx 30$, con lunghezze d'onda tipiche del flusso longitudinale $\lambda_x^+ \approx 150$. Questi sono collegati con l'instabilità del tipo Kelvin-Helmholtz del profilo di velocità medio. Un modello di stabilità semplificato per la superficie con riblet spiega approssimativamente il ridimensionamento del punto di rottura viscoso rispetto A_g^+ .

Entrando nel dettaglio troviamo che le riblet sono piccole sporgenze allineate con la direzione del flusso, che conferiscono una rugosità anisotropa alla superficie. Sono una delle poche tecniche che sono state applicate con successo per la riduzione dell'attrito di contatto nello strato limite turbolento, sia in laboratorio che in una configurazione aerodinamica completa. Sono state testate riblet di molte geometrie differenti, dimostrando una diminuzione di resistenza di circa il 10% su di una piastra piana. Un tema ricorrente nello studio delle riblet, come si è già visto in precedenza, è quello dello studio di superfici biologiche già esistenti, da cui questa tecnica trae ispirazione, come la pelle degli squali. Si devono considerare anche le condizioni fuori progetto, le cui conseguenze devono essere attentamente analizzate, che possono essere causate da fenomeni come erosione o gradienti di pressione avversi. Le riblet sono state utilizzate con successo anche per ridurre la resistenza totale di aeromobili, con una distribuzione ottimale di riblet.

Il meccanismo fisico della riduzione della resistenza da parte delle riblet è stato stu-

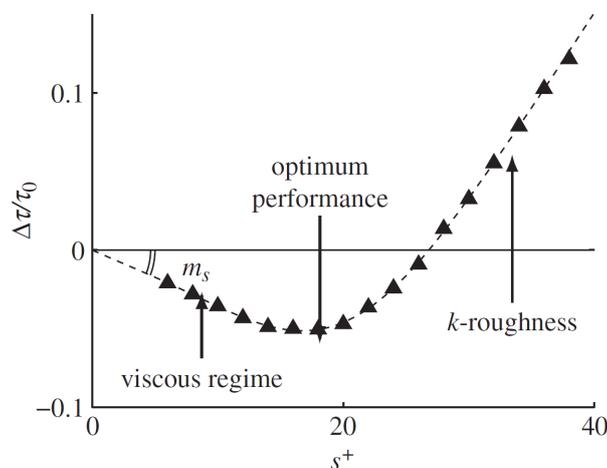


Figura 3.24: Definizione del regime di riduzione di resistenza osservato su riblet triangolari con angolo di punta di 60° , in funzione della distanza tra i picchi s^+ [22].

diato in dettaglio, anche se alcuni aspetti rimangono controversi. In particolare, sono stati riportati profili di velocità medi e locali e statistiche turbolente all'interno e al di sopra delle scanalature della riblet per diversi esperimenti di DNS, in galleria del vento e in canali con acqua. Da ciò si scopre come la dipendenza dal numero di Reynolds dell'effetto di riduzione della resistenza, può essere espresso in funzione della lunghezza in unità di parete. Nella figura 3.24 viene mostrata una curva tipica della riduzione di resistenza in funzione della distanza tra le riblet, in cui diversi regimi di resistenza possono essere definiti, in funzione di come questa dipende da s^+ . Si ha il regime viscoso, formalmente per $s^+ \ll 1$, ma praticamente a $s^+ \lesssim 10 - 15$, il contributo dei termini

non lineari al flusso all'interno e nelle immediate vicinanze delle scanalature della riblet è trascurabile e la resistenza è proporzionale alla distanza adimensionale s^+ . Il regime viscoso termina vicino ad $s^+ = s_{\text{opt}}^+$, la distanza ottimale per la quale la riduzione di resistenza è massima e dopo cui la riduzione di resistenza diventa un incremento, adottando il comportamento tipico di una rugosità κ . I parametri che determinano le performance ottimali di una determinata riblet sono la dimensione ottimale e la pendenza della curva di resistenza nel regime viscoso; entrambi dipendono dalla geometria, ma la curva qualitativa della resistenza è sempre come appena descritto.

Si è quindi visto che la riduzione di resistenza delle riblet è funzione della dimensione

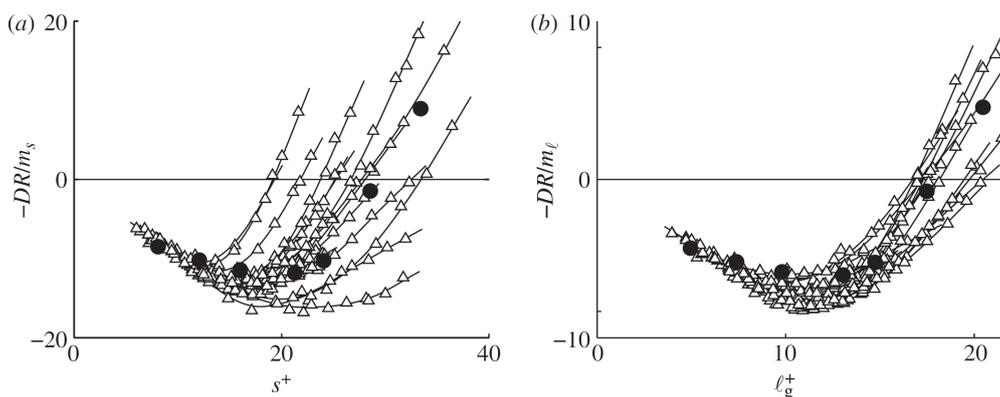


Figura 3.25: Curve di riduzione della resistenza di diverse riblet, scalate per una comune viscosità. Riduzione della resistenza in funzione di: (a) spaziatura delle riblet s^+ e (b) radice quadrata della sezione trasversale delle riblet l^+ [22].

delle riblet, espressa in unità di parete, ed è comune indicare quella grandezza con la spaziatura s^+ . Il valore ottimale di questa grandezza non è fisso, ma varia in funzione di quale forma di riblet si prende in considerazione. Attraverso molti dati sperimentali è stato possibile esprimere la riduzione della resistenza in termini di parametri geometrici, che racchiudono sia l'influenza della spaziatura tra le riblet e sia la forma. I risultati migliori si trovano per la radice quadrata della sezione trasversale delle riblet, $l_g^+ = (A_g^+)^{1/2}$. Nella figura 3.25 sono confrontate curve di resistenza sperimentale, per una grande varietà di geometrie di riblet, riscalate con la relativa viscosità per compensare le differenti performance viscoso. La figura mostra come ci sia una buona sovrapposizione dei dati per l_g^+ . Va comunque sottolineato che nonostante le curve si allineino meglio per l^+ che non per s^+ o h^+ , per geometrie convenzionali, non è detto lo facciano anche per geometrie non convenzionali.

Alla base del comportamento delle riblet nel regime viscoso troviamo il concetto di altezza sporgente. C'è una sottile regione vicino alla parete nei flussi turbolenti sulle pareti lisce, dove gli effetti viscosi dominano, gli effetti inerziali non lineari possono essere trascurati e il profilo di velocità medio è lineare. Si ha uno spessore di circa 5 – 10 unità di parete. Le riblet vanno a distruggere l'uniformità che si ha vicino alla parete, ma se la spaziatura è sufficientemente piccola, il flusso assume un comportamento uniforme a distanza da parete maggiore dell'altezza delle riblet. Lontano da parete gli effetti delle riblet riducono l'origine virtuale di Δ_u e Δ_w , mentre proprio Δ_u è la spiegazione della riduzione della resistenza, essenzialmente perché sposta la turbolenza lontano dalla parete; si trova comunque che l'altezza di sporgenza, Δh viene definita

come $\Delta h = \Delta_w - \Delta_u$. Intuitivamente, se il flusso incrociato ha un'origine virtuale più elevata di quella longitudinale ($\Delta h > 0$), il flusso trasversale indotto dai vortici longitudinali sovrastanti è impedito più severamente che su una parete liscia. I vortici fluidi vengono spostati lontano dal muro e il rimescolamento turbolento della quantità di moto lungo il flusso viene ridotta. Poiché questo rimescolamento è responsabile dell'alto sforzo di taglio locale vicino alla parete, la sua riduzione provoca un minore attrito di contatto. La relazione tra l'altezza di sporgenza e la resistenza viene modificata dall'effetto del resto profilo di velocità dello strato limite. La teoria classica della turbolenza ci dice anche che se la dimensione delle riblet è molto più piccola dello strato limite, cosa che avviene facilmente se si rimane nel campo della riduzione di resistenza, l'effetto delle riblet dovrebbe essere limitato allo strato vicino alla parete e che qualsiasi cambiamento in B (dove B è la costante della legge logaritmica di velocità nello strato limite turbolento) dovrebbe dipendere solo dalla geometria delle riblet ridimensionate in unità di parete. A causa della linearità del regime viscoso, la variazione di B dovrebbe essere proporzionale all'altezza di sporgenza $\Delta B = \mu_0 \Delta h^+$ mediante un coefficiente universale μ_0 . Ciò è stato testato da *Jiménez*, mediante DNS di un canale con strato limite piccolo, e si è trovato un valore di $\mu_0 \approx 0,66$.

I risultati che si trovano poi, possono essere utilizzati per prevedere le performance delle riblet, fino al limite della riduzione ottimale di resistenza. Inoltre possono essere utilizzati per comprendere il comportamento delle riblet in condizioni fuori progetto. Un primo esempio è un semplice caso in cui l'effetto di un angolo di disallineamento θ tra le riblet e il flusso, che deve essere misurato sperimentalmente; questo fino ad un angolo di $\theta = 15^\circ$ può essere trascurato, mentre si ha che l'effetto delle riblet scompare per angoli tra $\theta = 25 - 35^\circ$, fino ad arrivare al caso estremo opposto, $\theta = 90^\circ$ in cui si ha il massimo incremento di resistenza. Considerando invece il regime viscoso, poiché il problema è lineare, le velocità longitudinali e trasversali rispetto al flusso indisturbato possono essere proiettate sulla struttura delle riblet, con il risultato che si hanno offset rispetto al flusso, che sono la combinazione lineare degli offset di riblet perfettamente allineate con il flusso, e che l'altezza di sporgenza decade con l'imbardata, come confermato qualitativamente da esperimenti. D'altra parte, questa semplice dipendenza non si estende fuori dal regime viscoso. Si riporta di effetti di imbardata sulle riblet vicino al valore di spaziatura ottimale, e si trova una dipendenza geometrica della diminuzione tipicamente più grande che nel caso viscoso. Dato che il punto di rottura viscoso è un'indicazione dell'effetto della non linearità sulle riblet, non sorprende che le predizioni lineari non si applichino in quella zona. Un altro effetto da analizzare è l'erosione delle riblet, che è di maggiore importanza per le applicazioni pratiche. Misurando la riduzione di resistenza per riblet triangolari con punte arrotondate, si trova una perdita di performance fino al 40%, mentre invece non si hanno perdite per valli arrotondate. Il dettaglio della punta delle riblet non influenza significativamente il fenomeno del punto di rottura viscoso, invece, in quanto l'arrotondamento a malapena influenza la sezione trasversale delle riblet. Da esperimenti si trova infatti che assottigliando le punte cambia la massima riduzione di resistenza, ma non il valore ottimale di spaziatura, indicando che le differenze di performance sono dovute al cambio di pendenza della curva di riduzione di resistenza nel regime viscoso. Nella figura 3.26 sono riportati gli andamenti per diverse geometrie con punte arrotondate. Per le riblet triangolari si osservano diminuzioni di performance drastiche, mentre per quelle piane cambia molto poco l'altezza di sporgenza. Per le applicazioni pratiche, dove l'erosione è un problema, è probabilmente preferibile utilizzare riblet che non dipendono inizialmente troppo

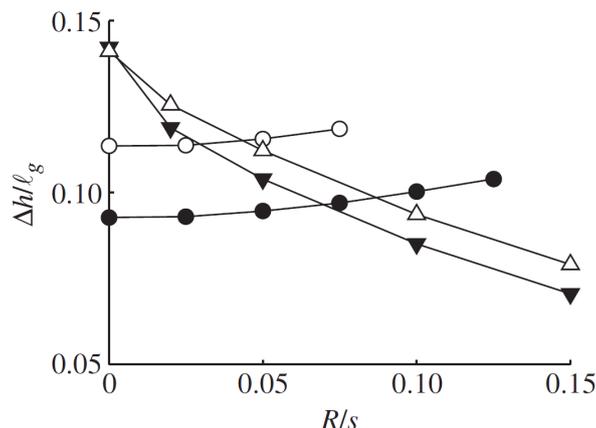


Figura 3.26: Altezza sporgente di punte arrotondate di riblet, scalate con la lunghezza di scala caratteristica, in funzione del raggio delle punte [22].

(triangoli rovesciati pieni: riblet triangolari con apertura di 60° ; triangoli vuoti, riblet triangolari con apertura di 45° ; cerchi vuoti, riblet lama con spessore 0,15 dell'apertura; cerchi pieni, riblet lama con spessore 0,25 dell'apertura).

dalla forma delle punte.

Il meccanismo fisico che porta al crollo del regime viscoso è un problema chiave nella progettazione delle riblet, poiché si è visto che la riduzione ottimale della resistenza è proporzionale alla dimensione della riblet in caso di crollo e l'utilizzo di riblet più grandi potrebbe portare a prestazioni ottimali migliori. In merito le teorie proposte in letteratura, da studi effettuati, si dividono in due ampi gruppi. Il primo è che gli effetti delle riblet sui flussi trasversali perdono efficacia una volta che si va al di là del regime di Stokes. Il secondo gruppo di teorie, invece, asserisce che l'ottimale lunghezza d'onda trovata, $s^+ \approx 10 - 20$, è relativa alla scala delle strutture turbolente nella regione di parete. *R. Garcia-Mayoral & J. Jimenez* hanno recentemente documentato uno scenario differente in una serie di DNS, con riblet che coprono l'intero range di dimensioni dalla riduzione della resistenza fino all'incremento. Si è osservato che per la dimensione delle riblet vicino l'ottimale, la formazione di vortici cilindrici trasversali vicino alla parete la cui intensità cresce rapidamente con la dimensione delle riblet. I cilindri sono centrati su $y^+ \approx 10 \sim 15$ e penetrano leggermente nelle scanalature della riblet. Queste nuove strutture spiegano la maggior parte del degrado delle prestazioni delle riblet con il crescere delle dimensioni. La formazione di strutture perpendicolari, piuttosto che parallele, alle riblet può sembrare sorprendente, ma non è del tutto inaspettata. Vortici trasversali simili sono stati riscontrati su tettoie vegetali e su pareti permeabili e porose. La scala di lunghezza delle strutture varia a seconda del particolare problema, ma, sebbene esistano poche analisi quantitative in letteratura, il fenomeno è sempre stato attribuito a un'instabilità di tipo Kelvin-Helmholtz. In sostanza, il profilo medio di uno strato limite ha quasi un punto di flesso alla parete e la ragione per cui rimane invisibilmente stabile è che la condizione di impermeabilità, $\nu = 0$, preclude le autofunzioni instabili antisimmetriche caratteristiche di Kelvin - Helmholtz. Una volta che qualsiasi modifica della parete consente la traspirazione locale, ricompare l'instabilità del punto di flesso. *R. Garcia-Mayoral & J. Jimenez* hanno adattato questa idea generale alle superfici con riblet. Il modello concettuale è che il flusso di Stokes longitudinali lungo le riblet è guidato da variazioni di pressione del flusso turbolento soprastante, e che

le variazioni longitudinali risultanti della velocità all'interno delle scanalature creano una traspirazione normale alla parete che agisce come una condizione al contorno per l'equazione di Rayleigh inviscida per perturbazioni linearizzate intorno al profilo medio di velocità.

In conclusione si è andati a rivedere i regimi per la riduzione della resistenza sulle superfici con riblet, con particolare enfasi sulle informazioni pratiche che possono essere estratte dal regime viscoso, e su altre condizioni sotto le quali quel regime termina. Si è mostrato che gli esperimenti esistenti per la posizione della rottura collassano meglio con una nuova scala di lunghezza, $l_g^+ = A_g^{+1/2}$ basata sull'area della scanalatura, che con scelte più classiche come la spaziatura o la profondità della riblet. La migliore stima per l'ottimale riduzione di velocità è $l_g^+ \simeq 10,7$, che, insieme con la pendenza di resistenza nel limite viscoso, possono essere usati per predire le performance delle riblet all'incirca del 20%. Si è mostrato come questo predice probabilmente l'effetto dell'erosione delle punte, che, in modo sorprendente, non è sempre deleterio, ma che l'effetto dell'imbardata per riblet ottimizzate è solo qualitativamente previsto dal regime viscoso.

Utilizzando DNS di canali con riblet comprendenti un range che va dalla riduzione viscosa di velocità fino all'incremento di resistenza, si è mostrato che la degradazione per riblet larghe del regime lineare della riduzione di resistenza non è collegato con la rottura del comportamento di Stokes della velocità longitudinale all'interno le scanalature delle riblet. Anche quando la resistenza si incrementa già, la velocità di scivolamento sul piano delle punte delle riblet rimane proporzionale alla dimensione di queste. La resistenza extra nasce da un sistema di vortici trasversali sotto $y^+ \approx 30$, con dimensioni scalate in unità di parete, indipendentemente dalla dimensione delle riblet. Questi vortici sono stati collegati ad un'instabilità di tipo Kelvin-Helmholtz comune in altri sistemi con superfici traspiranti, (tettoie, superfici permeabili e porose), e si è descritto un modello per superfici con riblet che fornisce alcune giustificazioni teoriche per lo scalare sperimentale del punto di rottura con l'area trasversale delle riblet [22].

On the boundary layer/riblets interaction mechanisms and the prediction of turbulent drag reduction di A. Baron, M. Quadrio & L. Vigevano

Il terzo articolo che si va a valutare è *On the boundary layer/riblets interaction mechanisms and the prediction of turbulent drag reduction* di A. Baron, M. Quadrio & L. Vigevano [23]. Qui si ha che la riduzione turbolenta della resistenza sperimentata dalle superfici nervate è il risultato sia dell'interazione tra le punte delle riblet e le strutture coerenti che caratterizzano i flussi turbolenti vicino alla parete, e sia delle modifiche del flusso del sottostrato laminare causate dalla forma della riblet, che può bilanciare, in condizioni appropriate, la penalità in resistenza dovuta all'aumento della superficie bagnata. Quest'ultimo meccanismo "viscoso" è studiato per mezzo di un modello analitico del sottostrato laminare, che rimuove le restrizioni geometriche e ci permette di tenere conto delle forme *reali* dei contorni della riblet, eventualmente affette da inesattezze di fabbricazione, e di calcolare anche per tali casi un parametro, chiamato altezza di sporgenza, correlato al flusso medio longitudinale. Considerando le geometrie reali, l'efficacia delle riblet si mostra chiaramente correlata alla differenza tra le altezze di sporgenza longitudinali e trasversali. Viene quindi ideato un metodo semplice per la previsione delle prestazioni delle superfici con riblet. I dati di riduzione della resistenza previsti e misurati, per le diverse geometrie della nervatura e le caratteristiche di flusso, sono in stretto accordo tra loro. La solidità dell'interpretazione fisica

alla base di questo metodo di predizione è di conseguenza confermata.

Entriamo poi maggiormente nel dettaglio di alcuni aspetti di questo studio. Si ha che le riblet rappresentano una delle più attraenti tecniche, tra tutte quelle sviluppate, con lo scopo della riduzione della resistenza d'attrito per un flusso turbolento. Il principale vantaggio di questa tecnica è di non richiedere energia per il funzionamento, mentre il principale problema può essere quello di un eventuale disallineamento con il flusso. Possono esserci diverse configurazioni di riblet, sia in forma, che in parametri geometrici, ed ognuna ha una sua configurazione in cui si raggiunge il massimo di riduzione della resistenza. Tra le varie ipotesi in merito all'interazione tra le riblet e il flusso si ha la possibilità che il parametro chiave che determina l'efficacia delle riblet potrebbe essere la differenza tra le due altezze di sporgenza per il flusso longitudinale e per il flusso incrociato sulla superficie con riblet, anziché la semplice altezza di sporgenza. Vengono quindi analizzati i meccanismi ipotizzati sulle interazioni tra riblet e flusso turbolento. I risultati dell'analisi, inoltre, possono essere anche utilizzati in un metodo predittivo di riduzione della resistenza, che si basa sui due meccanismi di interazione principali.

A questo punto, non si entra nel dettaglio degli esperimenti svolti, trattandosi di risultati abbastanza datati (1993), ma ci si va a concentrare maggiormente su quelli che sono stati i ragionamenti, le ipotesi e i risultati che sono stati trovati in questo studio. Sulla base delle analisi fenomenologiche e teoriche dell'interazione riblet-flusso, è possibile escogitare un semplice metodo di predizione in grado di valutare le prestazioni di riduzione della resistenza di una data superficie riblettata. L'obiettivo principale è quello di confermare la solidità dell'interpretazione fisica dei meccanismi di interazione, piuttosto che fornire una previsione quantitativa accurata. Tuttavia, tale capacità predittiva, sebbene approssimativa, può essere utilizzata come linea guida per selezionare le forme più promettenti tra le molte possibili geometrie della riblet. Il metodo utilizza le informazioni sperimentali fornite dalle misurazioni dirette della forza di attrito. L'efficacia di riduzione della resistenza dipende fortemente dalla geometria effettiva della superficie con riblet. Viene mostrata la possibilità di analizzare la geometria "vera" della riblet, derivata dalle microfotografie, ed è un punto chiave per correlare le misurazioni della forza. Per questi motivi, il database di dati da utilizzare è stato assemblato prendendo in considerazione solo esperimenti per i quali sia le misurazioni della resistenza sia le geometrie effettive delle riblet sono ben documentate. L'asserzione principale del metodo è che la variazione della resistenza d'attrito turbolenta dovuta alle riblet, rispetto ad una superficie liscia, è prodotta da soltanto due meccanismi:

1. L'interazione tra i picchi delle riblet ed il moto trasversale istantaneo, che può essere espresso come funzione della spaziatura trasversale adimensionale;
2. La modifica del sottostrato viscoso, dove la sporgenza adimensionale quantifica quanto davvero le riblet entrino nel sottostrato.

La scelta della spaziatura trasversale adimensionale come variabile indipendente viene suggerita dagli esperimenti, mentre è ragionevole supporre che nel caso di valore nullo, la superficie con riblet diventi piatta. Si può inoltre supporre che per i più alti valori di altezza di sporgenza, le riblet escano fuori completamente dal sottostrato viscoso, e si ha, sperimentalmente, che in questa condizione, un incremento della resistenza proporzionale all'aumento di area bagnata. Si è visto come l'altezza di sporgenza longitudinale è in qualche modo correlata alla massima riduzione di resistenza, per particolari geometrie di riblet, ma non si è ancora riuscito a quantificare in maniera esatta il massimo

beneficio ottenibile. Si è andato inoltre a validare il metodo di previsione della riduzione di resistenza ipotizzato, confrontando i dati previsti con le misure ottenute. L'affidabilità del metodo di previsione è ben evidenziata dalla sua capacità di prevedere il massimo valore di riduzione della resistenza, il punto di cross-over di riduzione zero e il trend di incremento della resistenza per valori più alti di s^+ . In generale, si può concludere che le informazioni sostanziali sono ben previste, in tutti i set di dati considerati, nonostante l'estrema semplicità del metodo predittivo considerato [23].

Come conclusione, si ha che le riblet riducono l'attrito turbolento attraverso due meccanismi base. Il primo è semplicemente un'interazione viscosa, che permette di compensare completamente l'aumento di resistenza associato all'aumento della superficie bagnata, permettendo alle superfici ribletate di comportarsi come superfici "aerodinamicamente lisce" (cioè quando le riblet si trovano completamente all'interno del sottostrato laminare). Il secondo riguarda l'influenza delle riblet sulle strutture vicino a parete, ed implica alcune azioni smorzanti sui flussi trasversali istantanei nello strato limite turbolento. Infine viene trovato un semplice metodo per la previsione della riduzione di resistenza prodotto dalle superfici con riblet, basato sull'analisi dei due meccanismi considerati. I dati numerici confermano i valori delle previsioni fatte con questo metodo [23].

On the relationship between drag modification and vertical velocity fluctuations in flow over riblets di S. Di Giorgio, S. Pirozzoli, S. Leonardi & P. Orlandi

Il quarto articolo che si va a valutare è *On the relationship between drag modification and vertical velocity fluctuations in flow over riblets* di S. Di Giorgio, S. Pirozzoli, S. Leonardi & P. Orlandi [24]. Vengono eseguite DNS di flusso su riblet triangolari e rettangolari per una vasta gamma di dimensioni e numero di Reynolds, con l'obiettivo di stabilire i meccanismi che controllano la riduzione della resistenza. Si riscontra che la riduzione della resistenza è principalmente associata alla capacità di inibire le fluttuazioni della velocità verticale sul piano delle riblet. Ciò si ottiene, in modo efficiente nelle DNS, con la soppressione artificiale di v , che produce una forte riduzione della resistenza, proporzionale alla dimensione delle riblet. Una parametrizzazione dell'effetto di riduzione della resistenza in termini di variazione della velocità verticale risulta essere abbastanza efficace per spiegare anche la variazione dei parametri di controllo. Si riporta un diagramma simile a Moddy che incorpora l'effetto della velocità di scorrimento ed un singolo parametro dipendente dalla geometria. Si riscontra una minore efficienza di riduzione della resistenza delle riblet rispetto ai casi con velocità di scorrimento imposta artificialmente.

Entrando nel dettaglio, andiamo ad analizzare soprattutto la parte che riguarda gli studi effettuati in merito alla componente v della velocità, ed al suo ruolo nella riduzione della resistenza. Diverse geometrie e forme di riblet sono state studiate numericamente e sperimentalmente, è la massima riduzione di resistenza trovata è stata di circa il 9%; i risultati migliori sono stati, generalmente ottenuti, per riblet affilate a forma di lama, che tuttavia sono difficili da realizzare ed utilizzare a numeri di Reynolds che abbiano un qualche interesse pratico. Si mostra che dare una velocità di scorrimento nelle direzioni longitudinali e trasversali ha effetti abbastanza differenti sulla turbolenza vicina alla parete, che risulta in una grande riduzione di resistenza nei casi in alcuni casi, ed in un grande incremento in altri.

Nello studio riportato si sono svolte DNS in un canale con riblet triangolari e rettangolari sulla parete inferiore. Si sono svolte simulazioni numeriche cambiando le dimensioni

delle riblet e/o il numero di Reynolds del flusso, anche attraverso la soppressione della velocità verticale sul piano delle punte. Oltre ad uno studio generale delle modifiche della turbolenza dovute alla presenza delle riblet, l'obiettivo dello studio è quello di stabilire se esiste una relazione tra la velocità verticale e le variazioni di resistenza. Nella figura 3.27 sono riportati due esempi di profili di riblet analizzati, uno triangolare

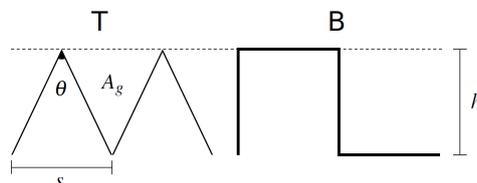


Figura 3.27: Schizzo dei profili trasversali delle riblet triangolari (T, sinistra) e rettangolari (R, destra). Le linee tratteggiate segnano il piano delle punte [24].

ed uno rettangolare. Le simulazioni di riblet rettangolari, aventi la stessa spaziatura e altezza (quindi la stessa frazione di gas) delle nervature T (triangolari) sono state eseguite utilizzando lo stesso dominio e griglia, e vengono indicate come B nel seguito.

Andando poi a vedere i risultati delle DNS si ha che nella figura 3.28 sono mostrati i

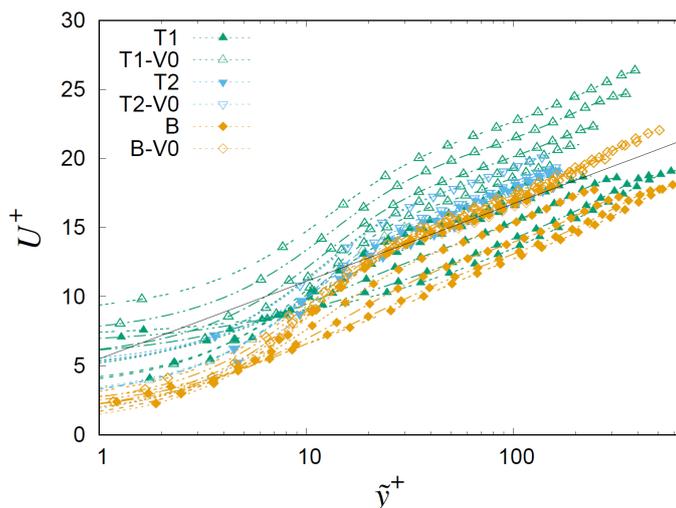


Figura 3.28: Profili di velocità scalati con le grandezze interne, vicino le pareti, in funzione della distanza dal piano delle punte, per varie simulazioni. La linea nera sottile indica la legge logaritmica per flussi a basso Reynolds ($U^+ = 1/k \log(y^+) + C$) [24].

profili di velocità media, scalati con le grandezze interne, vicino alle pareti pure, dove y^+ indica la distanza verticale dal piano delle punte, e l'attrito è stimato dal bilancio medio di quantità di moto. I dati delle DNS con condizioni al contorno standard di non scivolamento sono mostrati sempre nella stessa figura, 3.28, con i simboli pieni, mentre i casi con soppressione delle velocità verticali alle punte delle riblet sono mostrati con simboli vuoti. Sia i casi con riduzione di resistenza (velocità media sopra la linea di riferimento della legge logaritmica, mostrata come linea piena sottile nera) che quelli con incremento di resistenza (velocità media sotto la linea logaritmica) sono riportati nella figura 3.28. Questa mostra chiaramente la proporzionalità diretta tra la velocità media sul piano delle punte e l'effetto di riduzione di resistenza. Viene anche analizzata

la relazione tra la percentuale di resistenza ridotta e la lunghezza di scala tipica della sezione, ed i risultati sono coerenti con quanto visto negli studi di *R. Garcia-Mayoral & J. Jimenez* [22]. Si è visto in precedenza che impostando la velocità verticale, sul piano delle punte, a 0, si previene il punto di rottura delle performance delle riblet. In effetti, la soppressione delle velocità verticali equivale alla creazione di un piano scivoloso, sostituendo così la condizione al contorno di non scivolamento sulla parete inferiore con uno scorrimento assiale di velocità. Tuttavia, la velocità di scorrimento non è uniforme nel caso di riblet allineate longitudinalmente alla corrente, ma varia piuttosto lungo la direzione trasversale, a seconda della forma delle riblet e del numero di Reynolds. Simulazioni con queste condizioni imposte forzatamente, portano a replicare in modo più vicino la distribuzione di velocità media osservata nelle riblet triangolari. Si ritiene quindi che le deviazioni da una legge universale osservate, con conseguente riduzione dell'efficienza di riduzione della resistenza, siano controllate dall'effettiva geometria della parete, sebbene non vi sia comunicazione di massa tra il flusso di massa e il flusso all'interno delle scanalature. Ugualmente importante, l'influenza della geometria di parete è racchiusa nel singolo parametro α , che può essere una volta per tutte.

Per le riblet tradizionali si trova che raggiungono performance ottimali per uno specifico numero di Reynolds, superato il quale mostrano incremento e saturazione del coefficiente d'attrito in un regime completamente approssimativo.

Concludendo si è effettuato uno studio di riblet triangolari e rettangolari, in un'ampia gamma di dimensioni e numeri di Reynolds, con anche la soppressione della velocità verticale sul piano delle punte. In modo coerente con studi precedenti si trova che si ottiene una riduzione della resistenza fino al 9% con riblet tradizionali, rendendo difficile costruirle e mantenerle ai numeri di Reynolds di interesse ingegneristico. La riduzione di resistenza si trova essere strettamente collegata con la soppressione degli sforzi turbolenti nelle scanalature delle riblet, il che è effettivamente ottenuto dalla soppressione delle velocità verticali, come nel caso di superfici superidrofobiche o interfacce liquido-gas. Si può ottenere una grande riduzione di resistenza, poiché si trova essere proporzionale al cambio nella massiva variazione verticale di velocità, ed associata con lo spostamento verso l'alto dei profili di velocità scalati con le grandezze interne rispetto al caso di parete liscia. Globalmente si trova che la riduzione di resistenza può essere ottenuta solamente attraverso sistemi di controllo capaci di sopprimere la v , nei cui casi in cui le dinamiche dello strato vicino a parete non sono modificate. Si trova anche una chiara associazione tra l'effetto di riduzione di resistenza e la presenza di velocità di scorrimento [24].

Geometric Optimization of Riblet-Textured Surfaces for Drag Reduction in Laminar Boundary Layer Flows di S. Raayai-Ardakani & G. H. McKinley

L'ultimo articolo che si va infine a valutare, in merito alle riblet, è *Geometric Optimization of Riblet-Textured Surfaces for Drag Reduction in Laminar Boundary Layer Flows* di *S. Raayai-Ardakani & G. H. McKinley* [25]. Abbiamo che è stato dimostrato che le riblet su microscala modificano sistematicamente l'attrito viscoso superficiale nei flussi laminari con un elevato numero di Reynolds. I dentelli texturizzati di pelle di squalo nativa sono ampiamente citati come un esempio naturale di questo meccanismo passivo di riduzione della resistenza. Poiché la struttura di uno strato limite viscoso si evolve lungo la piastra, i cambiamenti di resistenza d'attrito locale sono noti empiricamente come una funzione della lunghezza della piastra nella direzione del flusso, così come la spaziatura delle riblet e il rapporto tra l'altezza e spaziatura di queste.

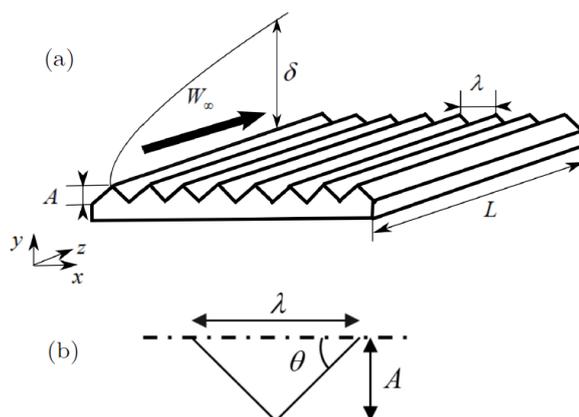


Figura 3.29: (a) Schematizzazione tridimensionale del flusso sulla superficie con riblet e delle riblet con scanalatura a V. La lunghezza della piastra è L . (b) Sezione trasversale di ogni riblet, con spaziatura λ e ampiezza A [25].

Nello studio si mira a stabilire una teoria canonica per il flusso laminare ad un alto numero di Reynolds su riblet con scanalatura a V per esplorare l'auto-somiglianza dei profili di velocità e l'evoluzione della resistenza di attrito totale esercitata su piastre di diverse lunghezze. Vengono combinati analisi di ridimensionamento, mappatura conforme e calcoli numerici per mostrare che la potenziale riduzione della resistenza ottenuta usando le superfici con riblet dipende da una forma opportunamente ridimensionata col numero di Reynolds e dall'aspect ratio delle riblet (definito come il rapporto tra altezza e spaziatura della riblet). Viene dimostrato che le superfici con riblet richiedono un ridimensionato numero di Reynolds inferiore a una soglia massima per ridurre la resistenza e che la modifica della resistenza è una funzione non monotona delle proporzioni della struttura della riblet. Il ridimensionamento fisico e i risultati computazionali presentati possono essere utilizzati per spiegare il meccanismo fisico sottostante di questa modalità di riduzione passiva della resistenza, per razionalizzare le dimensioni geometriche dei dentelli di squalo, nonché i risultati di esperimenti con repliche di dentelli di squalo di varie dimensioni, e fare da guida ai progetti per l'ottimizzazione dei parametri strutturali che si traducono in superfici che riducono l'attrito. Entrando nel dettaglio dello studio riportato si ha i metodi passivi di riduzione della resistenza utilizzando superfici con riblet longitudinali sono parzialmente ispirati da una grande varietà di superfici riblettate naturali osservate su pesci a nuoto veloce e squali; tra gli esempi più importanti ci sono le pelli di squali veloci. Le superfici bagnate di questi squali sono costituite con dentelli simil pelle, dove ognuno di questi dentelli contiene su di esso tra le 3 e le 7 *punte* (in media). La spaziatura e l'altezza di queste punte sono circa $35 - 105 \mu\text{m}$ e $20 - 30 \mu\text{m}$ rispettivamente. Alcuni di questi squali (quelli con questi tipi di pelle) possono raggiungere le velocità di nuoto fino a $10 - 20 \text{ m/s}$. Queste superfici texturizzate naturali sono pensate per essere in grado di modificare il flusso intorno al corpo dello squalo controllando la resistenza d'attrito, la transizione al turbolento, le strutture vicino a parete e la distribuzione di vorticità in prossimità della parete.

Inspirati dalla forma delle punte dei dentelli di squalo, sono stati ideati e testati diversi approcci utilizzando le riblet per ridurre la resistenza, con diverse forme di riblet, come scanalature a V, scanalature a U, rettangolari, strutture a lama di coltello, tutte sviluppate in flussi laminari, flusso in canale turbolento e strati limite. *Nastri di riblet bio*

inspirati sono stati anche applicati allo scafo dello yacht Stelle & Strisce nella Coppa America nel 1987, risultato vincitore. Negli anni recenti sono state studiate nuove varianti di riblet, come le riblet frattali, le superfici super idrofobiche riblet, così come le superfici riblet convergenti-divergenti; inoltre sono stati fatti anche studi sull'uso di superfici con riblet nei rotori di turbine radiali, o sulle pale di turbine a vento. Studi precedenti su riblet di forma semicircolare (ovvero simili ai dentelli degli squali) avevano dato una riduzione della resistenza fino al 7%, rispetto al flusso su parete liscia. Dalle DNS di superfici con riblet sinusoidali si trova che la riduzione totale di resistenza ha un comportamento non monotono al variare del rapporto tra l'altezza e la spaziatura delle riblet. Questo andamento non monotono è dovuto all'interazione di due trend opposti che si hanno quando l'aspect ratio delle riblet cresce:

1. Una riduzione locale degli sforzi di taglio medi per ogni sezione trasversale;
2. Un incremento della superficie bagnata sulla parete con riblet.

In aggiunta, si mostra che una piastra con riblet sinusoidali in flusso di strato limite ad alto numero di Reynolds riduce la resistenza purché la lunghezza della piastra sia maggiore di una lunghezza critica minima (ovvero che il numero di Reynolds del flusso, basato sulla lunghezza della piastra, sia maggiore del Reynolds critico). Il flusso di strato limite laminare ad alto numero di Reynolds isotermico, stazionario, incomprimibile di un fluido newtoniano è un problema di flusso canonico. Esplorare sistematicamente gli effetti del cambiamento della geometria delle superfici della riblet può aiutare a comprendere i parametri geometrici e dinamici che definiscono questa modalità di riduzione della resistenza passiva (prima di introdurre ulteriori fattori complicanti come la turbolenza e la miscelazione). Sono stati osservati vari aspetti del flusso di strato limite laminare di alto numero di Reynolds sulle superfici della riblet usando la simulazione numerica delle equazioni tridimensionali complete di Navier-Stokes; utilizzando modelli teorici dello strato limite laminare di un flusso su riblet periodiche, si analizza il ruolo chiave della geometria del flusso sulla resistenza totale sulla superficie. Utilizzando un nuovo ridimensionamento per lo strato limite tridimensionale laminare su superfici periodiche della riblet ed impiegando quindi una mappa conforme per risolvere analiticamente le equazioni dello strato limite sulle superfici con riblet, ci si sposta verso un più conveniente sistema di coordinate ortogonali curvilinee. Utilizzando il caso più semplice di scanalature a V con spaziatura λ e ampiezza A , calcoliamo la riduzione della resistenza in funzione della spaziatura e dell'altezza delle riblet, nonché della lunghezza della placca. Si mostra come i risultati della riduzione di resistenza possono essere confrontati in modo compatto come una funzione di tra variabili non dimensionali:

1. il numero di Reynolds del flusso sulla piastra, definito come $Re_L = \rho W_\infty L / \mu$;
2. la spaziatura adimensionale λ/L , che è il rapporto tra la spaziatura delle riblet rispetto alla lunghezza della piastra;
3. l'aspect ratio adimensionale della riblet $AR = 2A/\lambda$, che indica la misura di forma della riblet nella direzione trasversale.

Infine si confrontano i risultati con quelli disponibili in letteratura, come le dimensioni geometriche caratteristiche dei dentelli di squalo naturali ed artificiali. Invece della teoria classica dello strato limite bidimensionale, si va ad utilizzare quella tridimensionale nello studio delle riblet, per analizzare anche gli effetti delle periodicità di queste. Si

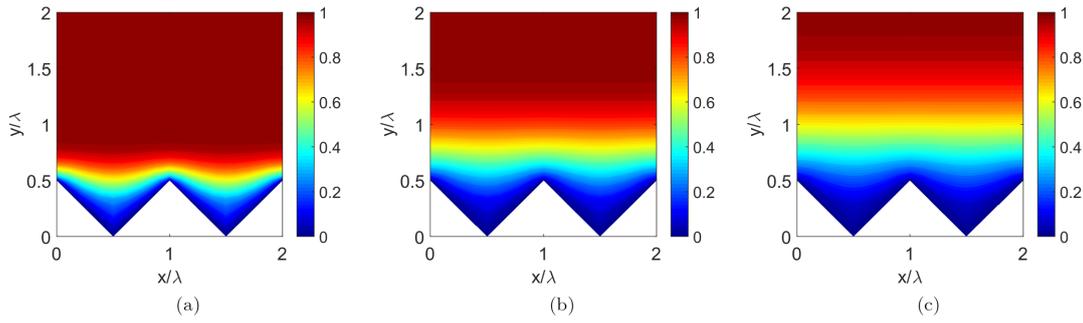


Figura 3.30: Isolinee di velocità per lo strato limite sulle riblet, con $AR = 1$ e $Re_\lambda = 100$, per:

(a) $Re_z = 100$ e $z/\lambda = 1.0$ e $Re_z(\lambda/z)^2 = 100$; (b) $Re_z = 500$ e $z/\lambda = 5.0$ e $Re_z(\lambda/z)^2 = 20$; (c) $Re_z = 1000$ e $z/\lambda = 10.0$ e $Re_z(\lambda/z)^2 = 10$. [25]

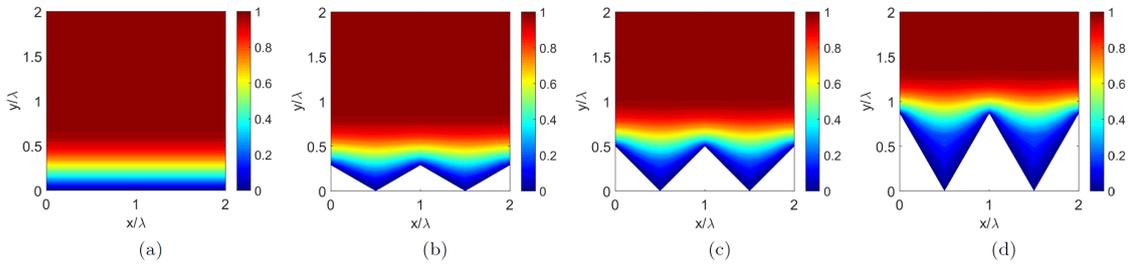


Figura 3.31: Isolinee di velocità per lo strato limite sulle riblet con scanalature a V, a condizioni dinamiche fissate ($Re_z = 200$, $z/\lambda = 2.0$) per:

(a) $AR = 0$; (b) $AR = 0, 58$; (c) $AR = 1, 0$; (d) $AR = 1, 73$. [25]

considera inizialmente una geometria a scanalature a V. Questa può essere osservata nella figura 3.29, dove sono riportate anche le dimensioni caratteristiche considerate. Si assume che la pressione sia costante e che quindi il gradiente di pressione sia nullo.

L'importanza fisica delle variazioni laterali o trasversali nella struttura dello strato limite e come questo evolve lungo la piastra la si comprende dall'osservazione dei profili di velocità alle punte ed alle valli delle riblet. Nel flusso di strato limite si assume che la diffusione della vorticità sia sempre perpendicolare alla superficie della parete a cui sono applicate le condizioni di non scivolamento.

Dalle DNS si ha che quando lo strato limite evolve sulle superfici delle riblet, inizialmente i profili di velocità seguono la forma delle superfici; in questa zona lo spessore di strato limite è più piccolo della scala delle riblet. Dopo una certa distanza critica lungo la piastra, lo spessore di strato limite inizia a diventare più grande della scala delle riblet. Da questo punto in poi l'evoluzione dello strato limite fuori dalle scanalature è leggermente influenzata dalle strutture locali di riblet e cresce similmente allo strato limite classico di Blasius. In studi precedenti si è visto come a Re costante ed a costante valore di L/λ i profili di velocità, la media locale della distribuzione degli sforzi di taglio, lo spessore di strato limite e il grado totale di riduzione della resistenza siano tutte funzioni dell'aspect ratio AR delle riblet. Ci si aspetta quindi che lo strato limite che si va a valutare sia una funzione di Re_L , di L/λ ed anche di AR . In caso di scanalature a V, questo si esprime come $AR = \tan \theta$. Nello studio considerato [25], dato che il set di equazioni dello strato limite tridimensionale considerato è molto più complesso da risolvere rispetto al bidimensionale, comprendendo anche derivate parziali in

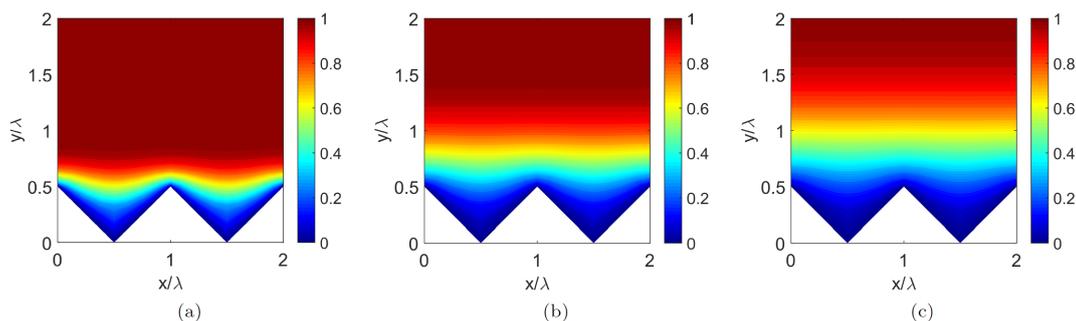


Figura 3.32: Isolinee di velocità per l'evoluzione nello strato limite sulle riblet con lunghezza d'onda delle riblet fisicamente doppia rispetto a quella della figura 3.30, cioè $Re_\lambda = 200$. I casi sono a: (a) $Re_z = 400$ e $z/\lambda = 2.0$ e $Re_z(\lambda/z)^2 = 100$; (b) $Re_z = 2000$ e $z/\lambda = 10.0$ e $Re_z(\lambda/z)^2 = 20$; (c) $Re_z = 4000$ e $z/\lambda = 20.0$ e $Re_z(\lambda/z)^2 = 10$. [25]

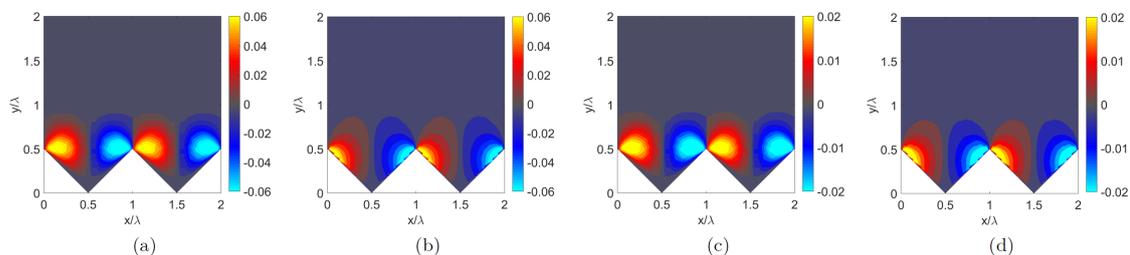


Figura 3.33: Isolinee di vorticità longitudinale per lo strato limite viscoso su piastre con riblet a scanalatura a V, a $d AR = 1, 0$, trovate utilizzando: (a) previsioni teoriche e (b) simulazioni numeriche per condizioni dinamiche di $Re_z = 17350$, $z/\lambda = 75$ e $Re_z(\lambda/z)^2 = 3, 08$; (c) previsioni teoriche e (d) simulazioni numeriche per condizioni dinamiche di $Re_z = 34700$, $z/\lambda = 150$ e $Re_z(\lambda/z)^2 = 1, 54$ [25]

tre direzioni, si vanno a considerare modelli di semplificazioni e cambi di variabili, per risolvere e/o semplificare il problema. (Tuttavia essendo questa parte non di interesse per questo lavoro di tesi, non viene qui riportata). Le equazioni così ottenute sono risolte, per le riblet, in funzione del parametro spaziale (λ) e dell'aspect ratio (AR), utilizzando Matlab, ed i risultati sono riportati in forma di evoluzione nelle isolee di velocità e dello spessore dello strato limite lungo la piastra, in aggiunta alla variazione di resistenza totale sulla superficie. L'evoluzione dei profili di velocità lungo le piastre con le riblet viene rappresentata in modo conveniente attraverso grafici isolee, in funzione di Re_z , λ/z e AR . Come prima cosa si tengono costanti la forma e la dimensione delle riblet, e si presentano le isolee di velocità a diverse coordinate locali z (avendo quindi diversi Re_z e λ/z). Questi primi grafici isolee sono riportati in figura 3.30. Così come evolve lo strato limite su di una piastra piana, si osserva un incremento di spessore di strato limite al crescere della distanza. Inoltre ad una determinata distanza si osserva che quando lo strato limite supera le scanalature, la sua crescita diventa molto più lenta. Dopo di che, per osservare il ruolo chiave della geometria, si va a considerare il caso in cui il numero di Reynolds e λ/z sono tenuti costanti, e cambia soltanto la forma trasversale delle riblet, attraverso il variare dell'aspect ratio AR . In figura 3.31 sono riportate le isolee di velocità, per scanalature a V, per diversi AR , in cui si osserva che l'introduzione ed il crescere delle riblet causano in incremento dello

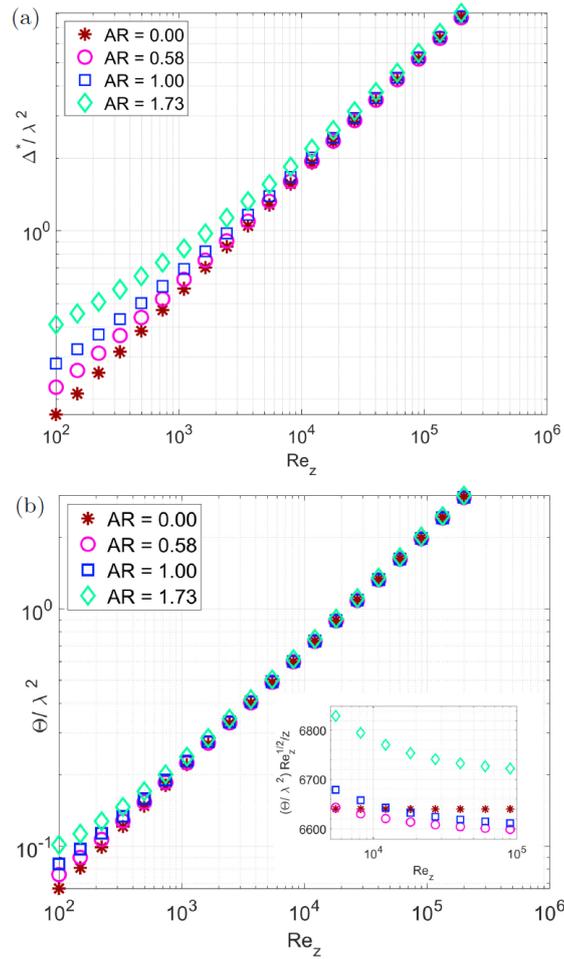


Figura 3.34: Variazioni in: (a) area di spostamento (Δ^*) e (b) area di quantità di moto (Θ) del flusso di strato limite sulle riblet in funzione del numero di Reynolds e diversi AR . La spaziatura delle riblet è costante per tutti i casi, mentre A varia progressivamente [25].

spessore dello strato limite, e del sottostrato viscoso; inoltre si ha un'area maggiore di flusso rallentato all'interno delle scanalature. Cambiare la spaziatura laterale delle riblet cambia le variazioni trasversali nella forma dello strato limite, e così lo spessore di strato limite ha strutture spaziali differenti e gradienti dipendenti dalla spaziatura. Ci si aspetta, con una nuova spaziatura, che lo strato limite viscoso che si sviluppa sulle riblet con lo stesso AR , ma spaziatura fisica differente, sia auto simile. Dalle figure 3.30 e 3.32, dove si ha che la seconda ha una spaziatura doppia rispetto alla prima, si può osservare, nei corrispondenti casi (quelli indicati con la stessa lettera), che i profili di velocità corrispondono strettamente l'un l'altro. Infine, nella figura 3.33, si possono osservare isolinee della vorticità sul piano centrale e su quello esterno ($z = L/2$ e $z = L$ rispettivamente) corrispondenti a valori di Reynolds locali il secondo il doppio del primo. In particolare, in questa figura, abbiamo che le immagini (a) e (c) sono calcolate da modelli teorici, mentre le immagini (b) e (d) si trovano utilizzando simulazioni ai volumi finiti delle equazioni di Navier-Stokes tridimensionali. Si osserva che le differenze tra le previsioni teoriche e le simulazioni derivano dalle ipotesi semplificative utilizzate per

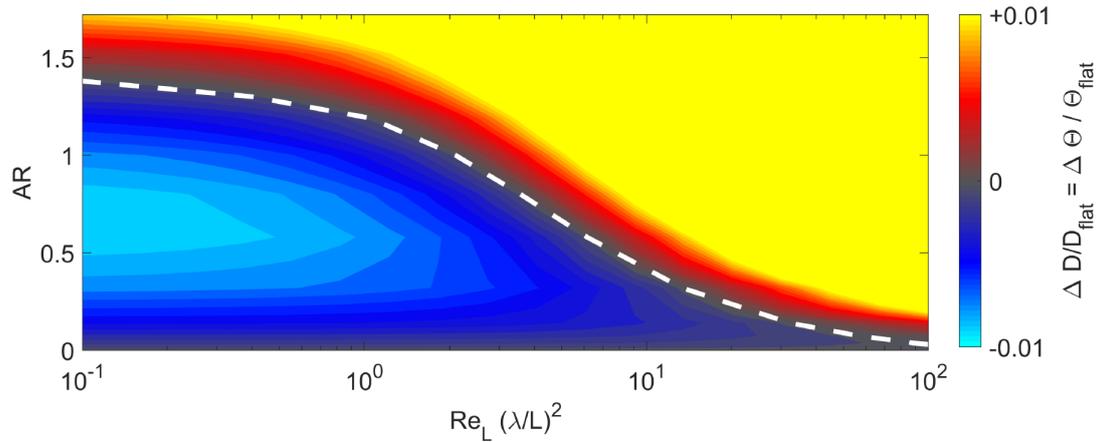


Figura 3.35: Variazioni nella resistenza totale esercitata su una piastra con riblet a scanalatura a V, mostrata in forma di cambiamenti nell'area di quantità di moto della piastra in funzione dell'aspect ratio AR delle riblet e del numero di Reynolds adimensionale $Re_L(\lambda/L)^2$. La linea tratteggiata indica l'isolinea per cui si ha $\Delta\Theta = \Delta D = 0$; la riduzione di resistenza si ha per $\Delta\Theta < 0$ [25].

risolvere le equazioni dello strato limite fino al primo ordine, oltre al trascurare l'effetto del bordo d'attacco e il gradiente di pressione risultante. Si osserva come al diminuire del numero di Reynolds locale, anche il massimo valore di vorticità (corrispondente al centro di ogni vortice) diminuisce. Nella figura 3.31 si è mostrato come per proprietà costanti, al variare dell' AR delle riblet si abbiano cambiamenti qualitativi nello spessore di strato limite. L'aumentare dell' AR della sezione trasversale incrementa lo spessore di strato limite rispetto al caso piano. Le isolinee di velocità della figura 3.31 visualizzano questi cambiamenti nella struttura dello strato limite, in modo qualitativo. Misure appropriate dello spessore di strato limite vengono eseguite utilizzando una definizione simil integrale allo spessore di spostamento ed allo spessore di quantità di moto, utilizzati nell'analisi dello strato limite. L'estensione tridimensionale di queste grandezze ci dà quindi due nuovi valori, che tengono conto della dimensione trasversale del problema, che sono, rispettivamente *l'area di spostamento* e *l'area di quantità di moto*. Queste sono definite dalle seguenti equazioni:

$$\Delta^* = \int_S \left(1 - \frac{w}{W_\infty}\right) dS$$

$$\Theta = \int_S \frac{w}{W_\infty} \left(1 - \frac{w}{W_\infty}\right) dS$$

dove si ha che S è l'area normale alla sezione trasversale alla direzione del flusso a z costante. Queste grandezze vengono adimensionalizzate attraverso la lunghezza delle riblet (λ), al quadrato; nella figura 3.34 vengono riportate in forma adimensionale in funzione del numero di Reynolds e dell' AR . Nella 3.34(a) si vede come la presenza delle riblet incrementa lo spessore di strato limite (o l'area di spostamento), rispetto al caso piano ($AR = 0$), mentre il crescere dell' AR risulta in un'ulteriore crescita dell'area di spostamento. A bassi numeri di Reynolds il trend di incremento è chiaramente

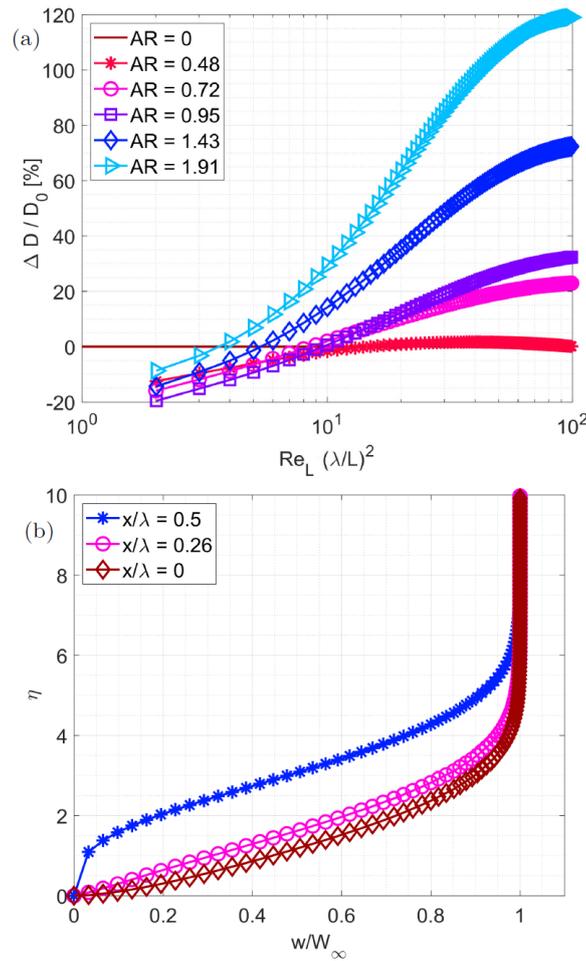


Figura 3.36: (a) Simulazioni numeriche dell'incremento ($\Delta D > 0$) e della riduzione ($\Delta D < 0$) della resistenza totale (D) per piastre con riblet di lunghezza totale differente, in funzione del numero di Reynolds, per diversi AR delle riblet, a lunghezza costante. (b) Profili di velocità calcolati per il flusso di strato limite su superfici con riblet a scanalatura a V, con $Re_L(\lambda/L)^2 = 6,05$ e $AR = 1,72$ [25].

visibile, mentre al crescere del numero di Reynolds, questo diventa meno significativo. Nella 3.34(b) si vede invece che l'area di quantità di moto, segue un comportamento differente. A bassi numeri di Reynolds l'aumento dell' AR incrementa l'area di quantità di moto. Esaminando meglio i dati si vede come per $Re_z \approx 10^4$ l'area di quantità di moto per le riblet è leggermente inferiore rispetto al caso piano.

Si trova come calcolare lo spessore di quantità di moto $\Theta(L)$ sia sufficiente per calcolare la resistenza totale sulla superficie, invece di calcolare e poi integrare la distribuzione di sforzi di taglio sulla superficie. Per valutare l'effetto delle riblet sulla forza di resistenza totale, la riduzione di resistenza è calcolata come

$$\frac{D - D_{\text{flat}}}{D_{\text{flat}}} = \frac{\Theta - \Theta_{\text{flat}}}{\Theta_{\text{flat}}}$$

dove D_{flat} è la forza di resistenza sulla superficie piatta (corrispondente ad $AR = 0$) e Θ_{flat} è l'area di quantità di moto della superficie per il numero di Reynolds globale.

Come si è visto il flusso di strato limite sulle riblet presenta un livello di autosimilarità per il numero di Reynolds scalato. Grazie a questa similarità nei profili sulle riblet di differenti dimensioni e posizioni longitudinali si può riportare l'area di quantità di moto in funzione di due parametri, il numero di Reynolds scalato e l'aspect ratio AR ; questo viene mostrato nella figura 3.35. La prima osservazione chiave che può essere fatta da questa figura è che le riblet a scanalatura a V possono sia diminuire che incrementare la resistenza totale d'attrito, in base alla forma delle riblet, la dinamica del flusso e l'aspect ratio AR delle riblet. Questa visualizzazione aiuta a razionalizzare le spesso conflittuali osservazioni numeriche e sperimentali riportate in letteratura, in cui si trova sia incremento che riduzione di resistenza. Il secondo punto da notare è che a AR costante, diminuendo il numero di Reynolds, si ottiene una diminuzione della resistenza totale sulla superficie. Per avere una riduzione di resistenza con superfici con riblet a scanalatura a V, il numero di Reynolds scalato deve essere minore di un valore critico. Questo valore di soglia dipende dall'aspect ratio delle riblet, AR e il valore di soglia per tutti i casi calcolati è mostrato dalla linea tratteggiata bianca nella figura 3.35. Da questo diagramma isolivello è chiaro che per una data velocità di flusso, con proprietà del fluido definite e geometria delle riblet fissata, per ottenere la riduzione della resistenza, la lunghezza della piastra deve essere maggiore di una lunghezza minima corrispondente al numero di Reynolds scalato di soglia. D'altra parte, con una lunghezza della piastra fissa, proprietà del fluido costanti e dimensioni delle riblet costanti, la velocità deve essere inferiore a una velocità massima (corrispondente nuovamente al numero di Reynolds scalato di soglia) affinché le riblet riducano la resistenza. Inoltre a numero di Reynolds scalato costante, la possibilità o meno di ottenere riduzione di resistenza con le riblet dipende dall' AR di queste; per ogni valore del Re , infatti, la resistenza d'attrito totale ha un comportamento non monotono, in funzione dell' AR , poiché al crescere di questo (muovendoci verticalmente dal basso verso l'alto sulla figura 3.35, ad un Re fissato) la resistenza prima si riduce al crescere dell' AR , per poi, dopo esser passata per un valore minimo, ricomincia a crescere, fino a che, dopo aver nuovamente raggiunto il valore iniziale (quello di piastra senza riblet o $AR = 0$) continua ad aumentare. Si ha così che per ogni valore di Re si trova un AR ottimale per avere la minima resistenza. Si vede anche come per riblet con AR maggiore di un valore soglia, queste non danno mai riduzione di resistenza (in questo caso, figura 3.35, è $AR \gtrsim 1,4$).

I risultati di riduzione della resistenza sono stati presentati in forma di riduzione della resistenza totale ($\Delta D/D_0$) come funzione del numero di Reynolds globale del flusso sulla lamina. Si possono riplottare i risultati in funzione del numero di Reynolds scalato, come si osserva nella figura 3.36 (a). Questi risultati sono coerenti con quelli riportati nella figura 3.35, mostrando come per un AR costante, il Re deve essere minore di un valore critico per avere riduzione di resistenza. Comunque la totalità delle simulazioni numeriche sono in grado di catturare i valori totali di riduzione della resistenza (fino al 20%) che sono più grandi delle predizioni teoriche asintotiche in assenza di gradiente di pressione. In aggiunta, le simulazioni mostrano che anche le riblet con $AR > 1,5$ sono in grado di ridurre moderatamente la forza di resistenza viscosa, mentre dall'analisi asintotica si aveva che si aveva riduzione di resistenza solo per $AR \lesssim 1,4$. Nella figura 3.36 (b) sono mostrati i profili di velocità in 3 posizioni trasversali differenti; confrontando queste velocità si può vedere qualitativamente che i profili di velocità sono molto simili fra loro, specialmente muovendoci lontano dalla superficie con le riblet.

La teoria dello strato limite esteso introdotta in questo studio può essere anche utilizzata per dare uno sguardo alle caratteristiche geometriche delle riblet sulla pelle di

squalo e il loro potenziale per essere in grado di ridurre la resistenza d'attrito superficiale. Le analisi effettuate supportano le ipotesi che le riblet sui dentelli di squalo naturali ed artificiali offrono il potenziale di riduzione della resistenza e dei loro parametri geometrici, in particolare le proporzioni delle riblet e i valori stimati di un numero di Reynolds adeguatamente ridimensionato si trovano nella regione che calcoliamo per offrire la massima riduzione della resistenza [25].

Si è mostrato, nello studio riportato [25] che lo strato limite viscoso laminare, che si instaura su riblet tridimensionali periodiche, può essere modellato utilizzando una forma estesa della teoria classica dello strato limite. Le equazioni dello strato limite esteso su una piastra strutturata (con riblet) possono essere risolte, utilizzando una mappatura conforme ed in assenza di gradienti di pressione, come funzioni di 3 variabili adimensionali: il numero di Reynolds globale del flusso, il rapporto di spaziatura delle riblet (λ/L) e l'aspect ratio delle riblet ($AR = 2A/\lambda$). Si è mostrato che i risultati dei calcoli sullo strato limite possono essere presentati in modo autosimilare, utilizzando una forma scalata del numero di Reynolds e l' AR delle riblet. Si ha che lo strato limite che si sviluppa sulle riblet è più spesso rispetto al caso senza, ed all'aumentare dell' AR delle riblet, questo spessore (quantificato nella forma di un'area di spostamento appropriata) si incrementa. Aumentando l' AR delle riblet si ottiene una più ampia regione di rallentamento del flusso all'interno delle scanalature; tuttavia, questo rallentamento è compensato dalla crescente area bagnata della sezione trasversale. Utilizzando le equazioni integrali della quantità di moto, è possibile utilizzare l'area della quantità di moto appropriata, catturata per ciascuna forma delle riblet per calcolare la resistenza di attrito totale sull'intera piastra. I risultati sono stati presentati sotto forma di un diagramma di isolinee della riduzione della resistenza $\frac{\Delta D}{D_0}$ in funzione del numero di Reynolds scalato e dell' AR delle riblet, come mostrato in figura 3.35. La scoperta più importante di questa analisi è che ad ogni valore costante dell' AR , il numero di Reynolds scalato del flusso deve essere inferiore a un valore critico massimo affinché la superficie strutturata con le riblet riduca la resistenza. I calcoli mostrano anche che ad ogni valore costante del numero di Reynolds scalato esiste un AR ottimale (dell'ordine dell'unità) corrispondente alla massima riduzione di resistenza. Questa analisi asintotica può anche essere utilizzata come struttura per aiutare a progettare texture artificiali e per interpretare osservazioni sui dentelli di squalo reali. Per un numero di squali di grandi dimensioni, le caratteristiche texture dei loro dentelli sembrano trovarsi nella regione di riduzione ottimale della resistenza. Infine, le tendenze presentate da questa teoria aiutano a chiarire molte delle tendenze sperimentali contrastanti precedentemente riportate in letteratura. Il diagramma di riduzione della resistenza mostrato in figura 3.35 (ampliato da opportuni calcoli numerici) può quindi essere usato come una guida di progettazione conservativa per la selezione di superfici con riblet e condizioni di flusso appropriate che porteranno a una riduzione della resistenza in applicazioni pratiche. Infine, questo quadro d'insieme può essere espanso a flussi di strati limite più complicati, in particolare quelli che coinvolgono gradienti di pressione (avversi e favorevoli) per prevedere le conseguenze idrodinamiche delle riblet in una gamma più ampia di condizioni applicabili a diverse situazioni industriali [25].

Capitolo 4

Simulazione numerica diretta di un flusso in un canale con riblet triangolari

4.1 Svolgimento dello studio

Obiettivo dello studio

Dopo esserci soffermati, prima, sulla teoria che sta alla base sia della fluidodinamica in generale e sia del problema studiato e, poi, sugli studi delle tecniche di *riduzione della resistenza d'attrito viscosa di un flusso turbolento* noti e riportati in letteratura, si va ora a descrivere e riportare quello che è stato il procedimento effettivamente seguito e svolto per andare a risolvere il problema che ci siamo posti nello studio oggetto di questo lavoro di tesi. Si è come prima cosa compreso quale fosse il problema da risolvere, in questo caso, trovare una tecnica, qui scelta passiva, che andasse a ridurre la resistenza d'attrito di un flusso turbolento in un canale. Dopo di che, anche grazie agli articoli presenti in letteratura, si è scelto un percorso da seguire, per raggiungere la soluzione, ovvero, si è scelto di utilizzare e studiare la tecnica delle *riblet*. Nello specifico l'obiettivo dello studio è stato quello di studiare diverse configurazioni di riblet, al fine di trovare quella che minimizzasse la resistenza d'attrito di un flusso turbolento in un canale piano.

Si è andati quindi a realizzare, attraverso il programma *OpenFOAM*, un caso base che simulasse il problema che si doveva analizzare, e dopo averne appurato il funzionamento, lo si è utilizzato per svolgere simulazioni numeriche dirette per tutti i casi che si reputato essere di interesse ed utili, per capire la fenomenologia che si è voluti andare a studiare. Da questi, una volta ottenuti i dati *grezzi* delle simulazioni numeriche dirette, si è andati ad estrarre i dati di vero e proprio interesse fisico, ai fini del lavoro di tesi. In particolare si è andati a calcolare la velocità media del flusso, le fluttuazioni di velocità mediate nel tempo, da questi si è trovati gli sforzi di Reynolds, gli sforzi viscosi, la resistenza totale del flusso, i profili di velocità in scala logaritmica, ed altre grandezze che verranno poi riportate e descritte, utili. Si è inoltre andati sia a fare visualizzazioni del flusso turbolento nel canale che si era simulato attraverso le simulazioni numeriche dirette, e sia a riportare mediante grafici, le grandezze risultanti che si sono estratte dalla simulazione. Infine, con i dati ottenuti, si è andati a constatare se l'obiettivo prefissato fosse stato raggiunto, e nello specifico a quantificare i risultati ottenuti. Si

entra ora maggiormente nel dettaglio del *percorso pratico* che si è svolto per ottenere i risultati.

4.1.1 Inquadramento del problema

Come prima cosa si è andati ad inquadrare qual era il problema che si voleva affrontare, e quali erano i possibili modi per risolverlo. Il problema che si voleva risolvere era, come già detto, studiare un metodo, ed in particolare una tecnica fluidodinamica, per ridurre la *resistenza d'attrito di un flusso turbolento* in un canale piano. Si è poi scelto, tra le tecniche presenti in letteratura, ovvero tra quelle di cui sono già note, informazioni sulle configurazioni ottimali, risultati numerici ottenuti con simulazioni o esperimenti, andamento delle curve di performance, possibili criticità o limiti, ed altre informazioni utili sia allo svolgimento dello studio, che ad eventuali confronti successivi. Si è scelto di studiare una tecnica specifica, per capire come questa potesse portare alla *riduzione della resistenza d'attrito di un flusso turbolento*. La tecnica scelta è quella delle *riblet*, ovvero piccole scanalature longitudinali che vengono realizzate e/o applicate a una o più pareti su cui scorre un flusso, di cui si vuole ridurre appunto la resistenza d'attrito. Questa tecnica presenta moltissime configurazioni possibili, alcune con performance migliori, ma di più difficile applicazione, realizzazione e mantenimento, come la forma a lama di coltello, ed altre, che invece, pur fornendo performance minori, hanno meno problemi applicativi, e quindi si prestano meglio ad applicazioni pratiche. In questo studio si è scelto di andare a studiare riblet con scanalature triangolari, o anche dette a scanalatura a V, come quelle che si sono viste nello studio (vedi 3.4.1) di *S. Raayai-Ardakani & G. H. McKinley* [25].

Si è scelto di studiare le *riblet* per alcuni motivi. Il primo è che le *riblet* sono una tecnica passiva, ovvero una tecnica che nel suo utilizzo non richiede né energia per funzionare e né un sistema di controllo. Ciò la rende, così come altre tecniche passive, di più facile realizzazione, produzione ed utilizzo, per fini pratici, e quindi di maggiore interesse nel migliorarne le performance. Inoltre come detto le *riblet* appartengono alla classe delle soluzioni passive, ma tra queste, sono attualmente, la soluzione migliore e più studiata. La scelta di una soluzione passiva ci ha permesso inoltre di poter valutare direttamente le performance pure della tecnica, in termini di riduzione della resistenza, senza dover tenere conto di bilanci energetici, come nel caso di tecniche attive, che rendessero una soluzione, seppur ben funzionante, poco performante a causa dell'elevato costo energetico, rispetto al risparmio fornito. Per quanto riguarda la scelta delle *riblet*, si è fatta questa scelta anche per motivi pratici, in quanto, data l'organizzazione, il funzionamento del programma utilizzato, *OpenFOAM*, come si vedrà di seguito e nelle appendici A e B, risultava di più facile realizzazione (seppur più lungo computazionalmente) realizzare una mesh che avesse una elevata complicazione geometrica (come è nel caso delle *riblet*, dovendo discretizzare ogni singola semiriblet), ma condizioni al contorno semplici (di fatto si devono solamente dare le condizioni del flusso che scorre, le pareti sono ferme), che non il contrario. Per quanto riguarda la scelta della forma delle *riblet*, invece, questa non è stata dettata da nessuna motivazione particolare, ma si è scelto di studiare la forma triangolare, o a scanalature a forma di V, semplicemente perché si tratta di una delle forme più utilizzate praticamente.

Dopo aver scelto la soluzione da analizzare/utilizzare per risolvere il problema postoci, e di questa averne scelto la particolare configurazione, si è dovuti andare a decidere quali sarebbero stati i parametri dell'analisi che si sarebbe andati a svolgere. Infatti, come ogni problema fluidodinamico, anche questo studio, non avendo soluzioni ana-

litiche esatte, necessità che determinati parametri del problema siano tenuti costanti, mentre altri siano fatti variare. Nello studio che si è andato a svolgere, volendoci soffermare maggiormente sulle riblet, si è, come prima cosa, scelto di tenere costanti le proprietà del flusso, ed anche la geometria esterna del canale in cui questo scorre, e di far cambiare la forma delle riblet. In particolare, si è scelto di utilizzare come parametro delle simulazioni l'altezza delle riblet, espressa in termini di unità di parete. Con questa scelta, si ottengono due vantaggi, dal punto di vista della similitudine fluidodinamica dei dati, sia per eventuali confronti, che per applicazioni pratiche. Il primo, dal punto di vista geometrico, è che, decidendo di tenere invece fissa la base delle riblet, facendo variare l'altezza di queste ¹ cambiano anche l'angolo di apertura delle riblet e soprattutto il loro *aspect ratio*, il quale esprime, in modo adimensionale, il fattore di forma delle riblet. Questo ci dice, che i risultati trovati per una delle configurazioni analizzate, possono essere trasportati ad altre riblet con lo stesso *aspect ratio*; inoltre è possibile confrontare i dati delle simulazioni, con alcuni di quelli presenti in letteratura, dove la riduzione di resistenza è riportata in funzione proprio dell'*aspect ratio* delle riblet, soprattutto per i casi di riblet a forma triangolare. Il secondo vantaggio, invece, lo si ha dal punto di vista fluidodinamico, in quanto si ha i risultati che si trovano per le altezze espresse in unità di parete per una determinata combinazione di altezza e numero di Reynolds d'attrito, sono le stesse per riblet ad altezza differente, a patto che si abbia un numero di Reynolds d'attrito tale che si abbia la stessa altezza in unità di parete.

Si è inoltre scelto di studiare il comportamento delle riblet per un flusso turbolento che scorre in un canale piano, e non per uno strato limite libero su di una lamina piana, per alcuni motivi. Il primo è legato alle applicazioni pratiche che questa tecnica ha. Infatti attualmente, le tecniche passive, ed in particolare le riblet risultano essere molto utilizzate per problemi di fluidodinamica interna, come il trasporto di fluidi in condotti, rispetto ad applicazioni esterne, se non in qualche caso, perché generalmente le superfici esterne in questione hanno anche altre funzioni, come quella di generare portanza, nel caso delle ali di un aereo, e quindi è necessario studiare le interazioni che le riblet hanno in questo caso, mentre nel caso di condotti, si vuole solamente ridurre la resistenza. Il secondo motivo, invece, è fluidodinamico e computazionale allo stesso tempo. Infatti nel caso di un flusso in un condotto, non si hanno problemi di effetti esterni, in quanto il fluido è confinato nel canale, e quindi computazionalmente si deve discretizzare e simulare solamente quello che accade nel canale, avendo così un dominio di calcolo ben definito, chiaro e non eccessivamente esteso. Nel caso di flusso esterno, invece, è necessario considerare l'influenza che la superficie investita dal flusso ha su questo, anche a notevoli distanze, in quanto si possono avere leggere perturbazioni a monte dell'oggetto, lontano da esso a causa dello strato limite verticale, e soprattutto a valle a causa dell'effetto di scia. Ciò richiede quindi, dal punto di vista computazionale, di avere un dominio di calcolo molto esteso, al fine di non trascurare nessuno di questi effetti, per evitare che le simulazioni risultino alterate da queste mancanze; ciò si traduce quindi in domini di calcolo molto più pesanti e complessi da discretizzare.

A questo punto, dopo aver deciso quale tecnica studiare ed utilizzare, quali parametri far variare nello studio, e come questo sarebbe stato condotto, si sono definite le condizioni esterne del flusso e la geometria del problema da analizzare. Queste grandezze vengono riportate, di seguito, nella sezione dedicata alla realizzazione della mesh, vedi

¹perché si, si è scelto come parametro l'altezza espressa in termini di unità di parete, ma avendo il numero di Reynolds e le proprietà del fluido fisse, questa varia proporzionalmente all'altezza geometrica

4.2.1. Ora invece, si vanno a riportare, gli ultimi cenni di teoria utilizzati in questo studio di tesi; in particolare abbiamo la resistenza d'attrito che si ha in un flusso turbolento, che è il problema che si è affrontato e le tecniche di discretizzazione numeriche che vengono utilizzate da *OpenFOAM*, nel realizzare le simulazioni numeriche dirette, che sono state svolte per questa tesi.

4.1.2 Resistenza di un Flusso Turbolento

La resistenza totale di un flusso turbolento che scorre in un canale chiuso, o su di una superficie libera, formando uno strato limite, è costituita da due contributi: la resistenza d'attrito viscoso (o sforzi di taglio) e gli sforzi di Reynolds. In particolare si ha che questi due contributi contribuiscono in modo complementare, lungo lo strato limite, a determinare la resistenza totale del flusso. Infatti nella zona molto vicina alla parete dello strato limite si ha che sono dominanti gli sforzi di taglio, dovuti alla viscosità, con gli sforzi di Reynolds che hanno valori trascurabili. Allontanandoci dalla parete, si ha invece, che mentre la resistenza viscosa crolla rapidamente, gli sforzi di Reynolds, detti anche resistenza inerziale, crescono e diventano prevalenti per tutto il resto dello strato limite. Come si è visto nelle figure 2.10 e 2.11, infatti, dopo il punto in cui i due valori di resistenza sono confrontabili, la resistenza viscosa diventa pressappoco nulla, mentre quella inerziale cresce fino a raggiungere il suo valore massimo, che si ha in corrispondenza del massimo delle fluttuazioni di velocità, generalmente per $y^+ \simeq 15 - 20$, per poi iniziare anch'essa a diminuire, seppur in modo più lento. Si ha infatti, che allontanandoci dalla parete, la resistenza globale diminuisce. Dai grafici, si osserva inoltre, come il massimo della resistenza lo si abbia a parete, dove è massima e dominante la resistenza viscosa. Entrando nello specifico del nostro caso di interesse, un flusso turbolento in un canale piano, si vanno a fare ulteriori considerazioni sulla resistenza. Come si è visto, per un flusso turbolento in un canale piano, vedi 2.3.8, si ha che, andando a considerare l'equazione della quantità di moto in direzione parallela a quella del flusso, e considerando il flusso turbolento già completamente sviluppata, questa si riduce nell'eguaglianza tra i termini di pressione ed i termini viscosi. Infatti, nel caso di un flusso in un canale, il flusso risulta essere completamente sviluppato e congelato, e non sono presenti termini inerziali, in quanto non ci sono variazioni di velocità lungo la direzione del flusso, e quindi l'equazione considerata ci dice che, per mantenere inalterata la configurazione di flusso che si instaura nel canale, è necessario che le forze di pressione (il gradiente di pressione fra monte e valle del condotto considerato) siano uguali a quelle d'attrito (ovvero la resistenza che il flusso incontra nell'avanzare nel condotto), come si è visto nell'equazione (2.50a). Andando ad esplicitare questi termini nel caso di un canale piano abbiamo:

$$-\langle \rho u'v' \rangle + \mu \frac{d\langle u \rangle}{dy} = -Gy \quad (4.1)$$

Nell'equazione riportata si può osservare come a sinistra siano riportati i termini che costituiscono la resistenza del flusso, e nello specifico il primo rappresenta gli sforzi di inerzia e il secondo quelli viscosi, mentre a destra abbiamo il gradiente di pressione, dove G indica le perdite di pressione che si hanno lungo il canale a causa dell'attrito, ed rappresenta anche la pendenza della curva della resistenza totale del flusso. Le perdite di pressione sono date, inoltre, dall'integrale degli sforzi d'attrito. Nel nostro caso, dati i valori delle variabili che si sono scelte, andando ad esplicitare i termini di questa

equazione si trova:

$$\tau_w = -G \quad (4.2)$$

Le equazioni riportate sono state poi utilizzate per andare a calcolare la resistenza che si aveva nelle simulazioni svolte, in quanto si ha che la resistenza totale è data dal valore delle perdite di pressione, calcolate a parete, come si vedrà meglio in 4.3.2.

4.1.3 Metodi di calcolo e discretizzazione della turbolenza utilizzati

Si vanno ora a riportare, alcuni brevi richiami teorici, in merito agli strumenti di fluidodinamica computazionale utilizzati in *OpenFOAM*, per svolgere le simulazioni. In particolare ci si concentra solamente sui metodi di discretizzazione che sono stati effettivamente utilizzati. Andremo quindi a parlare di:

- DNS
- Metodo ai volumi finiti
- Metodo di Interpolazione
- Metodo di Discretizzazione Temporale
- Metodo di Discretizzazione dei Gradienti
- Metodo di Discretizzazione delle Divergenze
- Metodo di Discretizzazione dei Laplaciani

In particolare andremo a vedere un po' più nel dettaglio i primi due, verrà fatto un breve cenno sul metodo di interpolazione utilizzato, che è quello lineare. Per gli altri, diciamo soltanto, che nelle simulazioni che si sono svolte su *OpenFOAM* si sono utilizzati, nell'ordine: la discretizzazione temporale con il metodo *backward*, i gradienti, le divergenze ed i laplaciani, invece, sono stati discretizzati con il metodo *Gauss lineare*. Per questi ultimi metodi appena riportati non si entra nel dettaglio, ma è possibile trovare ulteriori informazioni in merito, sia nelle guide in merito all'uso di *OpenFOAM* e sia in [26].

Simulazioni numeriche dirette

Le simulazioni numeriche dirette, o DNS consistono nel risolvere le equazioni di Navier-Stokes, risolvendo tutte le scale di grandezza, con condizioni iniziali ed al contorno appropriate per il flusso considerato. Ogni simulazione produce una singola realizzazione del flusso. L'approccio delle DNS era inizialmente infattibile fino a quando non sono diventati disponibili calcolatori con potenza tale da supportarlo. Anche se a causa di ciò si tratta di un approccio sviluppatosi tardi, risulta essere il più importante, ed anche il primo dal punto di vista logico. Concettualmente, infatti, è l'approccio più semplice e intuitivo, e, quando risulta essere anche applicabile, a causa delle elevate risorse richieste, è inarrivabile per quanto riguarda accuratezza e nel livello di descrizione fornito [6]. Tuttavia si deve considerare che il suo costo computazionale è estremamente elevato, e le risorse di calcolo necessarie crescono molto rapidamente al crescere del numero di Reynolds, così che l'applicabilità di questo approccio sia limitata a flussi di basso o moderato numero di Reynolds. L'elevato costo di questo metodo è dato dalla

necessità di andare a simulare, in modo accurato, ogni singola struttura fisica presente nel problema di interesse, in particolare, nel caso di flussi turbolenti, i vortici di più piccola scala. Si ha infatti, che per riuscire a simulare efficacemente, senza trascurare nulla, una struttura fluidodinamica, è necessario che la griglia di discretizzazione che si va ad utilizzare, sia costituita da celle di dimensioni minori la più piccola struttura fisica che si vuole calcolare. Ciò genera un primo elevato costo computazionale, in termini di potenza di calcolo richiesta, in quanto avere celle di dimensioni molto piccole, significa avere moltissime celle per discretizzare un problema di dimensioni ridotte, ed un numero esorbitante per un problema di grandi dimensioni. Ciò è collegato al numero di Reynolds in quanto si ha che, al crescere di questo, nel flusso si vanno a creare strutture fisiche di dimensioni sempre più piccole. Si ha quindi che se il numero di Reynolds cresce, si formano strutture più piccole, le quali, per essere catturate, richiedono celle sempre più piccole, che si traducono in un numero maggiore di punti, ovvero potenza di calcolo richiesta più elevata. Oltre a ciò, la dimensione estremamente ridotta delle celle necessarie per il calcolo, si traduce anche in passi temporali ridotti, nel caso di metodi risolutivi espliciti. In questo caso si ha infatti che, per assicurare la stabilità del metodo risolutivo, il rapporto tra il passo temporale moltiplicato la velocità del flusso e quello spaziale rappresentano un parametro, detto *numero di Courant*, che deve essere minore di 1, per avere simulazioni stabili. Riportando in termini di equazione abbiamo:

$$\frac{\Delta t a}{\Delta x} = C < 1$$

da cui troviamo che, per assicurare la stabilità del metodo, si deve avere:

$$\Delta t < a \Delta x C$$

Da ciò si capisce come, a pari valore di a , la velocità del flusso considerata, mantenendo C al di sotto del valore di stabilità, se il passo temporale (Δx) diminuisce, è necessario diminuire anche il passo temporale (Δt) al fine di mantenere stabile la simulazione; passo temporale maggiore, si traduce, a parità di intervallo di tempo da simulare, in un maggior numero di iterazioni da svolgere, e quindi tempi e risorse di calcolo maggiori. Tuttavia, seppur con queste limitazioni in termini di elevati tempi e risorse di calcolo richiesti, laddove utilizzabili, le DNS sono il metodo di simulazione più accurato, che permette di cogliere tutte le microstrutture di un flusso turbolento.

Metodo ai Volumi Finiti

Il metodo ai volumi finiti utilizza la forma integrale dell'equazione di conservazione (vedi [26]), come punto di partenza, ovvero:

$$\int_S \rho \phi \mathbf{v} \cdot \mathbf{n} dS = \int_s \Gamma \nabla \phi \cdot \mathbf{n} dS + \int_{\Omega} q_{\phi} d\Omega \quad (4.3)$$

Il dominio della soluzione è suddiviso in un numero finito di piccoli volumi di controllo (CV) da una griglia che, in contrasto con il metodo delle differenze finite (FD), definisce i confini del volume di controllo, e non i nodi computazionali. L'approccio usuale è definire i CV mediante una griglia adatta e assegnare il nodo computazionale al centro del CV. Tuttavia, si potrebbe anche (per le griglie strutturate) definire prima le posizioni nodali e costruire i CV attorno ad esse, in modo che le facce dei CV si trovino a metà strada tra i nodi. Il vantaggio del primo approccio è che il valore

nodale rappresenta la media sul volume di controllo con una precisione maggiore (al secondo ordine) rispetto al secondo approccio, poiché, in questo caso, il nodo si trova al centroide del CV. Il vantaggio del secondo approccio è che le approssimazioni delle derivate sulle facce dei CV sono più precise quando la faccia si trova a metà strada tra due nodi. Esistono diverse altre varianti specializzate di metodi ai volumi finiti (schemi cellvertex, schemi a doppia griglia, ecc.); i principi di discretizzazione sono gli stessi per tutte le varianti: bisogna solo tenere conto della relazione tra le varie posizioni all'interno del volume di integrazione. L'equazione integrale di conservazione (4.3) si applica a ciascun CV, nonché al dominio della soluzione nel suo insieme. Se sommiamo le equazioni per tutti i CV, otteniamo l'equazione di conservazione globale, poiché gli integrali di superficie sulle facce dei CV interni vengono annullati. Pertanto la conservazione globale è intrinseca nel metodo e ciò costituisce uno dei suoi principali vantaggi. Per ottenere un'equazione algebrica per un CV particolare, gli integrali di superficie e volume devono essere approssimati usando le formule di quadratura. A seconda delle approssimazioni utilizzate, le equazioni risultanti possono essere o meno quelle ottenute dal metodo FD [26].

Approssimazione delle superfici

Le superfici dei volumi di controllo sono formate da 4 (nei casi 2D) o 6 (nei casi 3D) facce piane, che sono generalmente indicate con la lettera corrispondente alla loro direzione (e, w, n, s, t, and b), rispetto al nodo centrale (P). Il flusso netto attraverso il contorno di un CV è la somma degli integrali sulle sue facce (4 o 6):

$$\int_S f dS = \sum_k \int_{S_k} f dS \quad (4.4)$$

dove f è la componente convettiva ($\rho\phi \cdot \mathbf{n}$) o diffusiva ($\Gamma\nabla\phi \cdot \mathbf{n}$) del vettore flusso nella direzione normale alla faccia del CV [26]. Poiché il campo di velocità e le proprietà del fluido sono note, solo ϕ è incognita. Per il mantenimento della condizione di continuità, è importante che i CV non si sovrappongano; ogni faccia del CV è unica per i due CV che si trovano ai suoi lati.

Considerando una generica faccia, indicata ora con il pedice e (ma il ragionamento vale per tutte, basta solo sostituire i pedici opportuni), per calcolare l'integrale di superficie dell'eq. (4.4) esattamente, si dovrebbe conoscere l'integrando f ovunque sulla superficie S_e . Questa informazione non è disponibile, poiché vengono calcolati solo i valori nodali (al centro del CV) di ϕ , pertanto è necessario introdurre un'approssimazione. È meglio farlo usando due livelli di approssimazione [26] :

- l'integrale è approssimato in termini dei valori delle variabili in una o più posizioni sulla faccia della cella;
- i valori sulla faccia della cella sono approssimati in termini di valori nodali (centro CV).

L'approssimazione più semplice dell'integrale è la regola del punto centrale: l'integrale è approssimato come prodotto dell'integrando al centro della faccia della cella (che è lei stessa un'approssimazione del valore medio sulla superficie) per l'area della faccia della cella:

$$F_e = \int_{S_e} f dS = \bar{f}_e S_e \approx f_e S_e \quad (4.5)$$

Questa approssimazione dell'integrale (purché sia noto il valore di f nella posizione 'e') è al secondo ordine di accuratezza. Ma poiché il valore di f non è disponibile al centro della faccia della cella, deve essere ottenuto tramite interpolazione. Per preservare il secondo ordine di accuratezza della regola di approssimazione del punto centrale sulla superficie integrale, il valore di f_e deve essere calcolato almeno con il secondo ordine di accuratezza.

Un'altra approssimazione al secondo ordine degli integrali di superficie in 2D è la regola trapezoidale, che dice [26] :

$$F_e = \int_{S_e} f dS \approx \frac{S_e}{2} (f_{ne} + f_{se}) \quad (4.6)$$

In questo caso è necessario calcolare il flusso agli angoli del CV. Per approssimazioni di ordine superiore degli integrali di superficie, il flusso deve essere valutato in più di due punti. Un'approssimazione del quarto ordine è la *regola di Simpson*, che calcola l'integrale su S_e come:

$$F_e = \int_{S_e} f dS \approx \frac{S_e}{6} (f_{ne} + 4f_e + f_{se}) \quad (4.7)$$

Qui servono i valori di f in 3 posizioni: il centro della faccia della cella 'e', e i due angoli, 'ne' e 'se'. Per ottenere accuratezza al quarto ordine questi valori devono essere ottenuti da interpolazioni dei valori nodali almeno accurate come la *regola di Simpson*. Sono adatti, per esempio, i polinomi cubici.

In 3D, la regola del punto medio è comunque l'approssimazione del secondo ordine più semplice. Sono possibili approssimazioni di ordine superiore, che richiedono l'integrazione in punti diversi dal centro della faccia della cella (ad es. angoli e centri dei bordi), ma sono più difficili da implementare. Se si presume che la variazione di f abbia una particolare forma semplice (ad esempio un'interpolazione polinomiale), l'integrazione è semplice. La precisione dell'approssimazione dipende quindi dall'ordine delle funzioni della forma.

Approssimazione dei volumi integrali

Alcuni termini nelle equazioni di trasporto richiedono l'integrazione sul volume di un CV. L'approssimazione accurata del secondo ordine più semplice è quella di sostituire l'integrale di volume con il prodotto del valore medio dell'integrando e del CV e approssimando il primo come il valore nel centro del CV [26] :

$$Q_P = \int_{\Omega} q d\Omega = \bar{q} \Delta\Omega \approx q_P \Delta\Omega \quad (4.8)$$

dove q_P sta per il valore di q al centro del CV. Questa quantità è facilmente calcolata; poiché tutte le variabili sono disponibili sul nodo P , non è necessaria interpolazione. L'approssimazione diventa esatta se q è costante o varia linearmente all'interno del CV; in caso contrario, contiene un errore del secondo ordine, come è facilmente mostrato. Un'approssimazione di ordine superiore richiede i valori di q in più posizioni rispetto al solo centro. Questi valori devono essere ottenuti interpolando i valori nodali o, equivalentemente, usando le funzioni di forma. In 2D l'integrale del volume diventa un integrale di area. Un'approssimazione del quarto ordine può essere ottenuta usando la funzione di forma bi-quadratica [26] :

$$q(x, y) = a_0 + a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy + a_6x^2y + a_7xy^2 + a_8x^2y^2$$

I 9 coefficienti sono ottenuti dal fitting della funzione con il valore di q su 9 posizioni ('nw', 'w', 'sw', 'n', 'P', 's', 'ne', 'e' e 'se'). L'integrale può essere così calcolato. In 2D l'integrale, per la griglia Cartesiana, diventa:

$$Q_P = \int_{\Omega} q d\Omega \approx \Delta x \Delta y \left[a_0 + \frac{a_3}{12} (\Delta x)^2 + \frac{a_4}{12} (\Delta y)^2 + \frac{a_8}{144} (\Delta x)^2 (\Delta y)^2 \right] \quad (4.9)$$

Solo 4 coefficienti devono essere determinati, ma loro dipendono dal valore di q in tutti i 9 punti indicati. Su di una griglia Cartesiana uniforme si ottiene:

$$Q_P = \frac{\Delta x \Delta y}{36} (16q_P + 4q_s + 4q_n + 4q_w + 4q_e + q_{se} + q_{sw} + q_{ne} + q_{nw}) \quad (4.10)$$

Dato che solo a P il valore è disponibile, l'interpolazione deve essere utilizzata per ottenere q negli altri punti. Ciò deve essere fatto almeno al quarto ordine di accuratezza per mantenere l'accuratezza dell'approssimazione integrale.

L'approssimazione al quarto ordine appena descritta dell'integrale di volume 2D può essere utilizzata per approssimare gli integrali di superficie in 3D. Approssimazione di ordine superiore di integrali di volume in 3D sono più complessi, ma possono essere trovati utilizzando le stesse tecniche [26].

Metodi di Interpolazione

Le approssimazioni agli integrali richiedono i valori delle variabili in posizioni diverse dai nodi computazionali (i centri dei CV). L'integrando, indicato nelle sezioni precedenti da f , coinvolge il prodotto di più variabili e / o gradienti di variabili in tali posizioni: $f^C = \rho \phi \mathbf{v} \cdot \mathbf{n}$ per il flusso convettivo e $f^D = \Gamma \nabla \phi \cdot \mathbf{n}$ per il flusso diffusivo. Si parte dal presupposto che il campo di velocità e le proprietà del fluido ρ e Γ sono note in tutte le posizioni. Per calcolare i flussi convettivi e diffusivi, sono necessari il valore di ϕ e il suo gradiente normale alla faccia della cella in una o più posizioni sulla superficie del CV. Anche gli integrali di volume dei termini sorgente possono richiedere questi valori. Devono essere espressi in termini di valori nodali per interpolazione [26]. Si va ora a riportare, tra i possibili modi in cui ciò può essere fatto, il metodo delle *interpolazioni lineari*, in quanto è quello che viene utilizzato nelle simulazioni con *OpenFOAM*. L'interpolazione lineare tra due nodi vicini è un metodo di approssimazione semplice per i valori al centro delle facce dei CV. Con questo metodo, considerando la posizione 'e' abbiamo:

$$\phi_e = \phi_E \lambda_e + \phi_P (1 - \lambda_e)$$

dove il fattore di interpolazione lineare λ_e è definito come:

$$\lambda_e = \frac{x_e - x_P}{x_E - x_P}$$

L'equazione di ϕ_e è al secondo ordine di accuratezza, come si ottiene dall'espansione in serie di Taylor (vedi [26]).

4.2 Realizzazione delle Simulazioni

Si va ora, finalmente a vedere, come sono state effettivamente realizzate le simulazioni numeriche, che sono state utilizzate per studiare le configurazioni di riblet che si sono scelte. In questo capitolo, il codice computazionale realizzato verrà solamente nominato in alcuni punti, specificando però quali valori sono stati inseriti e per quale motivo; il codice di calcolo effettivamente realizzato, per *OpenFOAM*, interamente, è riportato nelle appendici, dove abbiamo, nell'appendice A la parte di codice che è stata necessaria allo svolgimento delle simulazioni per ottenere i dati, e nell'appendice B la parte di codice che invece si è utilizzata per rielaborare i dati nel post processing ed estrarre le grandezze di nostro interesse. In questa appendice (la B) inoltre sono riportati anche altri codici, che sono stati realizzati ed utilizzati in questa simulazione, sempre per il post processing, ma che non servivano ad *OpenFOAM*, ma sono serviti per riordinare i dati ottenuti dal post processing e per calcolare effettivamente lo sforzo d'attrito che si aveva per ogni configurazione di riblet simulata. Inoltre, nell'appendice A, abbiamo anche un altro codice, non per *OpenFOAM*, che è stato realizzato ed utilizzato, basandoci sulle formule di funzionamento di *OpenFOAM*, per conoscere i valori esatti della posizione delle celle nella direzione y , che era necessario fornire al programma, per avere l'esatta spaziatura voluta, per poter meglio discretizzare le dove vicino alle pareti (vedi 4.2.1). Inoltre nelle 2 appendici, uno per ciascuna, sono riportati i 2 codici di *submit*, che sono stati utilizzati, rispettivamente, per eseguire la simulazione e per eseguire il post processing dei dati, sul sistema di calcolo *HACTAR*.

Oltre alla parte relativa alla realizzazione computazionale delle simulazioni (la mesh, in 4.2.1) vengono riportati anche i dati in merito ad i tempi di calcolo su cui le simulazioni hanno girato (vedi 4.2.2), i parametri del post processing (vedi 4.3), ed altri dati e/o passaggi in merito a tutta la parte effettiva riguardante le simulazioni, il loro run, e la loro rielaborazione in post processing, il riordino dei dati ottenuti (vedi 4.3.1), fino alla realizzazione di visualizzazioni, grafici e dati risultato (vedi 4.3.2).

4.2.1 Creazione della mesh

Dopo aver deciso di utilizzare, come tecnica di riduzione della resistenza di un flusso turbolento, quella delle *riblet*, e di voler simulare questo flusso per il caso di un flusso in un *canale piano*, si è andati a realizzare questo problema, *canale piano sulle cui superfici sono applicate le riblet*, dal punto di vista computazionale.

Come prima cosa si sono dovute scegliere le dimensioni del canale. Tuttavia, prima di fare ciò, si è deciso di impostare tutto il problema, in modo adimensionale, in modo tale da non avere problemi in seguito, nel confronto tra le grandezze ottenute. Si sono quindi scelti i parametri di adimensionalizzazione. In particolare abbiamo:

- Per la velocità, si è presa la U_b , ovvero la velocità media del flusso nel canale;
- Per lo spazio, si è presa la semialtezza h del canale piano;
- Per il tempo, si è andati a prendere un tempo caratteristico dato da $T = h/U_b$

Entrambi questi parametri, gli unici a cui si assegna quindi un valore assoluto, sono stati presi in valore unitario. Si ha quindi $U_b = 1$ e $h = 1$.² Una volta scelti i parametri

²da qui in poi tutti i valori di grandezze riportate saranno già espressi nel loro valore adimensionale; inoltre essendo tutte le grandezze di adimensionalizzazione di valore unitario, i valori riportati coin-

Tabella 4.1: Abbreviazioni e valori adimensionali delle dimensioni del canale simulato.

Dimensione	Abbr.	Valore adimensionale
Altezza canale	H	2
Lunghezza canale	L_x	9
Larghezza canale	L_z	4,5

di adimensionalizzazione, siamo andati quindi a definire tutte le altre grandezze del problema, in funzione di queste. Per quanto riguarda le dimensioni del canale si ha: altezza del canale = 2, lunghezza del canale = 9 e larghezza del canale = 4,5. Nella

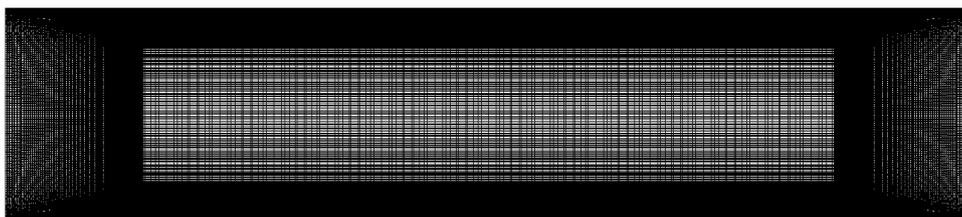


Figura 4.1: Vista del canale sul piano $x - y$, dove si possono osservare la lunghezza del canale, $L_x = 9$ e l'altezza del canale, $H = 2$. Sezione della simulazione $H7$

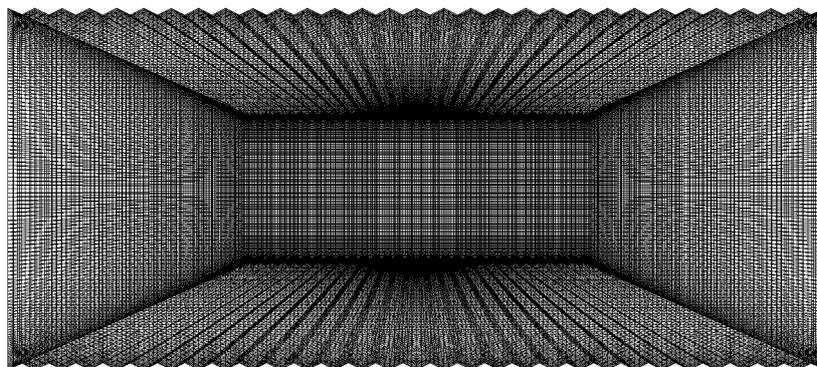


Figura 4.2: Vista del canale sul piano $z - y$, dove si possono osservare la larghezza del canale, $L_z = 4,5$ e l'altezza del canale, $H = 2$. Sezione della simulazione $H7$

tabella 4.1 sono riportati i valori scelti, ed anche le abbreviazioni per il nome di queste grandezze, che verranno da qui in poi utilizzate, per maggiore semplicità. Si possono osservare le dimensioni del canale scelte, nelle figure 4.1, 4.2 e 4.3. In queste abbiamo, rispettivamente, nella 4.1 la vista sul piano $x - y$, in cui si hanno lunghezza ed altezza del canale, nella 4.2 la vista sul piano $z - y$, in cui si hanno larghezza ed altezza del canale e nella 4.3 la vista sul piano $x - z$, in cui si hanno lunghezza ed larghezza del canale. In queste figure è inoltre possibile osservare anche alcuni cenni in merito alla discretizzazione della griglia ed alle riblet; questi vengono ora definiti, ognuno secondo i suoi criteri di scelta.

Oltre alle dimensioni del canale, si sono andati a stabilire anche i parametri che

cidono anche con il rapporto che si ha tra il valore della grandezza considerata e quello della relativa grandezza adimensionalizzante, che è quello che è stato utilizzato nel codice.

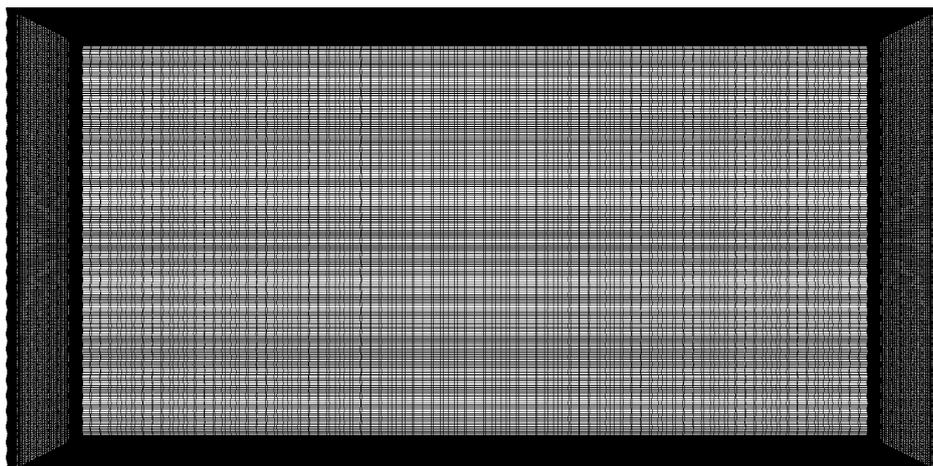


Figura 4.3: Vista del canale sul piano $x - z$, dove si possono osservare la lunghezza del canale, $L_x = 9$ e la larghezza del canale, $L_z = 4,5$. Sezione della simulazione $H7$

Tabella 4.2: Valori di altezza delle riblet scelti per le DNS, e relativi parametri derivati

Nome caso	h^+	h_r	$AR = \frac{h_r}{b_{r/2}}$	$\theta [rad]$	$\theta [^\circ]$	$\frac{\theta}{2} [^\circ]$
H0	0	0	0	n.d.	n.d.	n.d.
H3	3	0,015	0,2	2,746	157,38	78,69
H5	5	0,025	1/3	2,497	143,12	71,56
H6	6	0,03	0,4	2,38	136,39	68,43
H7	7,5	0,0375	1/2	2,214	126,87	63,43
H9	9	0,045	0,6	2,06	118,07	59,03
H10	10	0,05	2/3	1,965	112,61	56,30
H12	12	0,06	0,8	1,792	102,68	51,34
H15	15	0,075	1	1,57	90	45
H18	18	0,09	1,2	1,39	79,61	39,80
H20	20	0,1	4/3	1,286	73,73	36,86
H22	22,5	0,1125	1,5	1,176	67,38	33,69

caratterizzano le riblet e la discretizzazione della mesh. Per quanto riguarda le riblet, come si è detto, si è scelta l'altezza in termini di unità d'attrito, h^+ , come variabile del sistema. Quindi per questa grandezza non si ha un unico valore, ma diversi, in funzione dei diversi casi esaminati. I valori scelti sono riportati nella tabella 4.2, in cui troviamo le altezze h^+ scelte, le corrispettive altezze geometriche, h_r , corrispondenti per il $Re_\tau = 200$ scelto (si spiegherà dopo il perché di questo valore), il rapporto rispetto alla semiapertura delle riblet (che si spiegherà a breve), ovvero l'aspect ratio (AR) delle riblet e l'angolo di apertura delle riblet, ovvero l'angolo alla punta di queste (anche questa grandezza verrà tra poco descritta e spiegata) oltre a specificare di quale caso analizzato si tratta (anche se qui i casi vengono semplicemente identificati con la rispettiva h^+ , per semplicità, tranne in due casi, in cui si tronca la cifra decimale). Oltre all'altezza, le riblet sono definite dalla dimensione della base, e dal numero di riblet che sono presenti lungo la direzione trasversale del canale. Queste due grandezze, se la dimensione trasversale (la larghezza L_z) del canale è definita, sono strettamente legate,

se si vuole avere un numero intero di riblet, ovvero far sì che il canale inizi lateralmente con l'inizio della prima riblet e termini lateralmente con la fine dell'ultima riblet. In questo caso, dopo aver scelto la larghezza del canale, $L_z = 4,5$, si è deciso di avere 30 riblet. Ciò ha portato ad avere, come dimensione della base delle riblet $b_r = 0.15$. Questo valore, risulta comunque buono, ed accettabile sia dal punto di vista fisico, in quanto si è, dal punto di vista delle grandezze d'attrito ($b_r^+ = 30$), nelle dimensioni del sottostrato viscoso, e sia dal punto di vista di altri studi consultati in merito. Si va a calcolare anche, come parametro importante per la definizione delle grandezze caratterizzanti le riblet e per la realizzazione della griglia, la semiapertura della base; abbiamo che la semibase è definita come $b_{r/2} = \frac{b_r}{2} = 0,075$, (ovvero $b_{r/2}^+ = 15$). Inoltre, come si è visto, questa serve anche per definire l'aspect ratio delle riblet, ovvero il rapporto tra l'altezza e la semibase delle riblet. L'aspect ratio AR delle riblet, infatti, è definito come $AR = \frac{h_r}{b_{r/2}}$. Questo valore è utile per poter eventualmente fare confronti con dati presenti in letteratura. Infine, per finire di ben caratterizzare le riblet, si è calcolato anche l'angolo di apertura, ovvero l'angolo che si ha alla punta delle riblet, θ , definito come

$$\theta = 2 \tan^{-1} \left(\frac{1}{AR} \right)$$

Tutte le grandezze fin qui descritte, che caratterizzano le riblet, come detto, sono riportate nella tabella 4.2. Inoltre è possibile osservare la forma di una di delle configurazioni di riblet scelte, in particolare quella del caso *H22*, nella figura 4.4, dove si ha uno zoom sul flusso calcolato all'interno del canale su cui sono applicate quelle riblet, concentrandoci sulla parte vicino alla parete superiore, dove si può osservare, per confronto, la forma di quella configurazione di riblet.

Per la mesh, invece, è stato necessario definire il numero di celle voluto nelle 3 direzio-



Figura 4.4: Vista del flusso all'interno del canale su cui sono applicate le riblet, per la simulazione *H22*, con zoom sulla parte in cui si osserva meglio la forma delle riblet.

ni. Per ognuna di queste, è stato necessario fare un ragionamento differente, e quindi si riportano i valori distintamente. Per la direzione x , quella della lunghezza del canale, ovvero quella in cui scorre il flusso, si è semplicemente scelta la dimensione delle celle che si voleva, e si è andati a dividere il canale in un numero tale di celle, per ottenere quella dimensione. Nello specifico abbiamo, nella direzione x :

- $\Delta x = 0,06$
- $N_x = 150$
- $\Delta x^+ = 12$

Per quanto riguarda il valore riportato Δx^+ , questo verrà poi spiegato.

Nella direzione z , quella trasversale, la base del canale, invece, il ragionamento è stato

leggermente più complesso. Si è inizialmente deciso di avere, come nel caso della direzione x , un numero di celle in questa direzione che fossero tutte della stessa dimensione; tuttavia, in questo caso, si è dovuto tenere presenti le riblet, e far sì, che ogni riblet e semiriblet fosse discretizzata in modo esatto dalle celle, ovvero che una cella iniziasse all'inizio della riblet ed un'altra terminasse alla semiriblet, ed anche al contrario per la seconda parte della stessa riblet. Volendo fare ciò, si è quindi andati a definire la dimensione delle celle in questa direzione, basandoci sulla dimensione di una riblet, sapendo che poi sarebbe bastato moltiplicare il numero di celle costituenti una riblet, per il numero totale di riblet, e si sarebbe ottenuto il numero di celle totale in direzione z . Si è quindi definita, come lunghezza delle celle in direzione z , $\Delta z = 0,01875$ da cui si ottiene $N_z = 240$. In questa configurazione si ottiene che ogni riblet viene discretizzata con 8 celle, ed ogni semi riblet con 4. Dopo aver fatto ciò, avendo definito gli intervalli di discretizzazione per ogni semiriblet, e non in modo totale su tutto il canale, si è deciso, per avere una maggiore precisione di calcolo sulla punta delle riblet, che risulta essere la parte della riblet un po' più critica per il calcolo, di andare a scalare la dimensione delle celle, che non sarebbero state più uniformi, ma leggermente più grandi vicino alle *valli* delle riblet, e leggermente più piccole vicino alle *punte*. In particolare si sono scelti, come rapporti di *grading*, ovvero il rapporto tra la dimensione dell'ultima e della prima cella dell'intervallo per cui si dà questo valore, rispettivamente:

- $Rz_{dx} = 1,25$, il rapporto di *grading* per i blocchi *destri* delle riblet, quelli discendenti, ovvero dalla punta alla valle;
- $Rz_{sx} = 0,8$, il rapporto di *grading* per i blocchi *sinistri* delle riblet, quelli ascendenti, ovvero dalla valle alla punta.

Infine, nella direzione y , quella dell'altezza del canale, il ragionamento effettuato, è stato ancora più complesso. Infatti questa è la direzione che va a discretizzare le celle in direzione normale alla parete, ed è quindi la direzione che meglio cattura lo strato limite; tuttavia per fare ciò, soprattutto per quanto riguarda il sottostrato viscoso, è necessario avere celle di dimensioni molto ridotte, vicino alla parete. Si è quindi stabilito, di avere come dimensione della prima ed ultima cella, quelle vicino alla parete, $\Delta y = 0,005$, con il quale si ha $\Delta y^+ = 1$ (anche questo valore viene poi spiegato). Tuttavia, se si fosse andati a discretizzare tutto il canale in direzione y con celle di questa dimensione, il risultato sarebbe stato quello di un numero tale di celle da rendere troppo pesante la mesh, e quindi le simulazioni non eseguibili. Per ovviare a ciò, senza perdere la precisione a parete, si è decisi, anche in direzione y , di andare a scalare la dimensione delle celle, avendo quelle più piccole vicino alle due pareti, con la dimensione che cresce muovendoci verso il centro del canale (dove, man mano che ci si allontana dalla parete la dimensione delle strutture vorticoso cresce, e quindi non sono più necessarie celle molto piccole per catturarle bene), e celle un pò più grandi, ed uniformi al centro del canale. Per fare ciò si è diviso il canale, in direzione y , in 6 blocchi di discretizzazione (come si è fatto anche in direzione z , dove si è diviso il canale in 60 blocchi, ovvero 1 per ogni semiriblet che abbiamo), ed a ognuno di questi blocchi, di differenti dimensioni, è stato assegnato un diverso rapporto di *grading*. Volendo però avere, che le celle aumentassero di dimensione allontanandoci dalla parete, in maniera uniforme e continuativa, è stato necessario entrare nelle formule di calcolo di *OpenFOAM*, e capire, come si potesse avere ciò, utilizzando i valori che possono essere forniti al codice, ovvero la dimensione del blocco (Δy_{Bl}), il rapporto di *grading* (R_G) e il numero di celle per quel blocco ($N_{y_{Bl}}$). Ciò è stato possibile andando a consultare le guide che vengono fornite con

sul sito del programma *OpenFOAM*, ed in particolare i file in cui le formule utilizzate sono riportate, sono *TutorialGuide* e *UserGuide*. Fatto ciò si è realizzato un codice in *Fortran*, nel quale, dando come parametri di input quelli scelti da noi, ovvero la dimensione del blocco (Δy_{Bl}), il rapporto di grading (R_G) scelto da noi, e la dimensione che si vuole che abbia la prima cella del blocco (Δy_1), questo ci restituisce in output, il numero di celle ($N_{y_{Bl}}$) che si deve imporre a quel blocco per avere i risultati voluti, e la dimensione dell'ultima cella del blocco (che poi sarebbe la dimensione della prima, moltiplicata per il valore di grading), in modo tale da poter sapere quale dimensione si deve imporre alla prima cella del blocco successivo, per continuare la sequenza ordinata di celle crescenti. Il suddetto codice è riportato nell'appendice A, ed è chiamato *Newton*, in quanto nel suo funzionamento sfrutta il metodo di Newton per al risoluzione delle equazioni differenziali, vedi A.6.1.

Tornando quindi alla discretizzazione del problema in direzione y , si è diviso in canale

Tabella 4.3: Parametri di discretizzazione in direzione y

Numero Blocco	N. punti	R. Grading	Dim. Blocco	y_{iniz}	y_{fine}
B1	27	3	0,25	0	0,25
B2	14	1,5	0,25	0,25	0,5
B3	23	1	0,5	0,5	1
B4	23	1	0,5	1	1,5
B5	14	2/3	0,25	1,5	1,75
B6	27	1/3	0,25	1,75	2

in 6 blocchi, numerati dal basso verso l'alto (quindi 1 e 6 sono quelli vicino alle pareti, 3 e 4 quelli centrali, e 2 e 5 quelli intermedi), attraverso l'utilizzo del codice *Newton*, e dai valori che noi vogliamo imporre alla griglia, si ottengono i parametri della mesh, per ogni blocco. Questi sono riportati, per una maggiore chiarezza espositiva, nella tabella 4.3.

Con i numeri di celle scelti per le 3 direzioni, si trova che la mesh che si andrà ad utilizzare, è composta da 4608000 punti, quindi più di 4 milioni e mezzo di punti. Nella figura 4.5 è possibile osservare il canale nella sua quasi totalità, riportato mediante l'uso di *ParaFOAM*. Tuttavia si deve ricordare, che pur con la mesh scelta, i dati che si sono ottenuti, come si è osservato sia da visualizzazioni del flusso con *ParaFOAM* (vedi 5.4) e sia dai grafici riportati (vedi 5), sono comunque affetti da errore numerico, causato sia dalla discretizzazione numerica della mesh (che si annullerebbe solo per una mesh infinita, con celle infinitesimali) e soprattutto, per le configurazioni di riblet con altezza grande, dalla distorsione che si ha della griglia, a causa di queste. È importante tenere a mente ciò, nel valutare i risultati ottenuti, per non prenderli come valori assoluti ed esatti, ma considerando sempre un margine di incertezza, che come detto, cresce al crescere dell'altezza delle riblet.

Per quando riguarda la realizzazione geometrica della mesh, sono stati riportati fin qui, tutti i valori che sono necessari a ciò. Mancano ancora, invece, le condizioni al contorno e le condizioni iniziali delle simulazioni, così come, la parte relativa alle proprietà del fluido simulato, e quella in merito al tempo di simulazione. Tuttavia, prima di aggiungere queste informazioni, si va sia a riportare le proprietà del flusso, ed anche, attraverso queste, a giustificare alcuni dei valori geometrici scelti, dal punto di vista fisico (come si è accennato in precedenza, parlando di Δx^+ e Δy^+).

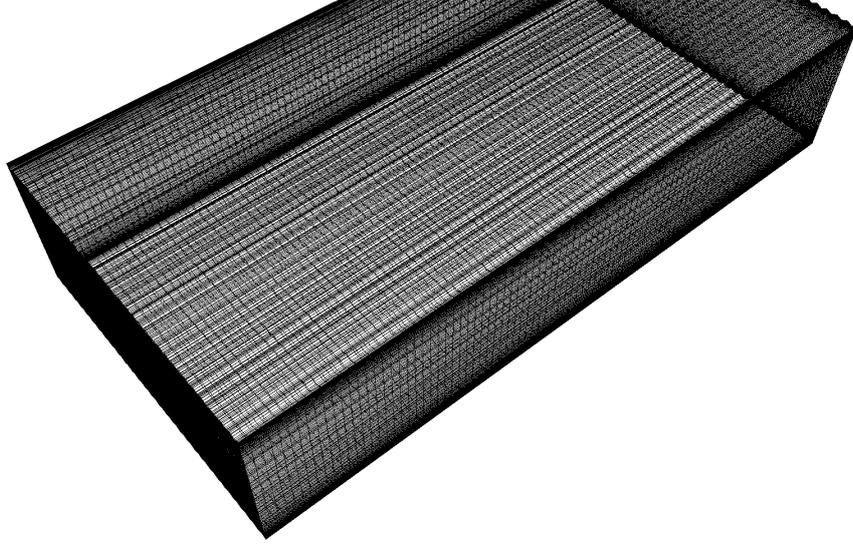


Figura 4.5: Vista totale del canale che si va a simulare, con riportata anche la discretizzazione della mesh. Questa è la griglia per il caso $H7$. Visualizzazione effettuata con *ParaFOAM*.

Proprietà del flusso e giustificazione fisica delle dimensioni scelte

Per quanto riguarda le proprietà del flusso, si è andati a stabilire un punto di partenza, e da questo si sono ricavate tutte le altre grandezze ad esso collegato. In particolare, nel nostro caso, si è andati a scegliere, come punto di partenza il Re_τ , ovvero il numero di Reynolds, calcolato per le grandezze d'attrito. Da questo si deve andare poi a calcolare il numero di Reynolds per il flusso, il Re_b . Per fare ciò si considera l'equazione per la velocità nello strato limite, $u^+ = \frac{1}{\kappa} \log y^+ + A^+$, dove abbiamo $\kappa = 0,41$, $A^+ = 5,2$ e $u^+ = \frac{u}{u_\tau}$. Dopo di che si considera la velocità media nel canale, come:

$$\bar{u} = \frac{1}{h} \int_0^h u \, dy$$

in cui andiamo a sostituire $u = u_\tau u^+$ e $y = y^+ \frac{\nu}{u_\tau}$, cambiando anche gli estremi di integrazione, da $0 - h$ ad $0 - Re_\tau$, otteniamo:

$$\bar{u} = \frac{1}{h} \int_0^{Re_\tau} u_\tau u^+(y^+) \frac{\nu}{u_\tau} \, dy^+$$

semplificando la u_τ , e portando fuori dall'integrale i termini costanti, si ottiene:

$$\bar{u} = \frac{\nu}{h} \int_0^{Re_\tau} u^+(y^+) \, dy^+$$

Ora sostituendo ad u^+ l'equazione logaritmica per lo strato limite e tenendo conto che $\int \log y = y(\log y - 1)$ si ha che:

$$\begin{aligned} \bar{u} &\simeq \frac{\nu}{h} \left[\int_0^{Re_\tau} \left(\frac{1}{\kappa} \log y^+ + A^+ \right) \, dy^+ \right] = \\ &= \frac{\nu}{h} \left[\frac{1}{\kappa} y^+ \left(\log y^+ - 1 \right) + A^+ y^+ \right]_0^{Re_\tau} = \end{aligned}$$

$$\bar{u} = \frac{\nu}{h} \left[\frac{1}{\kappa} \left(Re_\tau (\log Re_\tau - 1) \right) + A^+ Re_\tau \right]$$

A questo punto, dalla definizione di $Re_b = \frac{\bar{u}h}{\nu}$, otteniamo:

$$Re_b = \frac{\bar{u}h}{\nu} = \frac{h}{\nu} \frac{\nu}{h} \left[\frac{1}{\kappa} Re_\tau (\log Re_\tau - 1) + A^+ Re_\tau \right]$$

da cui, semplificando, otteniamo:

$$Re_b = \frac{1}{\kappa} Re_\tau (\log Re_\tau - 1) + A^+ Re_\tau \quad (4.11)$$

Si è così trovata la relazione che lega il numero di Reynolds del flusso, Re_b , con quello espresso con le grandezze d'attrito, Re_τ .

Avendo trovato la relazione che lega Re_τ a Re_b , la (4.11), possiamo, una volta scelto il valore di uno, trovare il corrispondente valore dell'altro. In particolare nel nostro caso, noi scegliamo di prendere come valore di Re_τ , $Re_\tau = 200$, e dall'equazione (4.11) abbiamo che $Re_b = 3136,74$. Definito quindi quale sia il valore del numero di Reynolds voluto, per averlo è necessario prendere la giusta viscosità cinematica ν . In particolare, avendo già definito come grandezze di adimensionalizzazione, sia $U_b = \bar{u} = 1$ ed $h = 1$, abbiamo che il valore di ν è dato da:

$$\nu = \frac{1}{Re_b} = \nu^* = 3,18 \cdot 10^{-4}$$

avendo inoltre scelto, come valore di densità ρ , $\rho = 1$. Abbiamo così definito tutte le proprietà necessarie del fluido. Tutte le proprietà del flusso, per quanto riguarda il codice di calcolo, sono riportate nei codici *transportProperties* e *turbulenceProperties* (vedi appendice A, ed in particolare la sezione A.4. Inoltre, avendo definito il Re_τ , si possono andare a giustificare anche alcuni dei valori delle grandezze geometriche prima riportati (in realtà si è appunto prima definito il valore di Re_τ , e poi da questo, come ora si riporta, si sono trovati i valori indicati sopra). Come si è detto in merito alle DNS, (vedi 4.1.3), in queste, per far sì che tutto il campo di moto sia correttamente simulato, senza trascurare nessuna struttura vorticoso, è necessario che il dominio di calcolo, abbia delle dimensioni minime, mentre le celle di calcolo, abbiano dimensioni inferiori a determinati valori.

Per quanto riguarda il dominio di calcolo, nel nostro caso, un canale piano turbolento, le indicazioni in merito alle dimensioni minime, riguardano i grandi vortici che si formano nel canale. In particolare, per avere una simulazione accurata, è necessario che il dominio permetta un completo sviluppo dei vortici, in direzione longitudinale, e permetta di ospitarne un numero sufficiente, in direzione trasversale. Per quanto riguarda la direzione longitudinale, abbiamo che è necessario che il dominio, abbia, per permettere un corretto sviluppo dei vortici, una lunghezza adimensionale, L_x^+ minima di almeno di 1200. Essendo L_x^+ definita come $L_x^+ = \frac{L_x}{h} Re_\tau \gtrsim 1200$, nel nostro caso abbiamo una $L_x^+ = 9 \cdot 200 = 1800$ che è ben maggiore di 1200. Si sarebbe quindi potuto prendere anche un canale più corto, ma si è preferito fare così per essere sicuri del corretto sviluppo dei vortici. Un discorso analogo, vale per la direzione trasversale. In questo caso, però, è necessario permettere lo sviluppo di un numero minimo di vortici. Data la dimensione di un vortice, $\lambda^+ = 120$, ed il numero minimo di vortici che si vuole affinché la simulazione sia sufficientemente accurata, almeno 5, si ha che la lunghezza

trasversale minima deve essere $L_z^+ \simeq \frac{5\lambda^+}{U_b} \simeq 600$, visto che nel nostro caso si è considerata una velocità $U_b = 1$. Essendo definita $L_z^+ = \frac{L_z}{h} Re_\tau$, nel nostro caso abbiamo una $L_x^+ = 4,5 \cdot 200 = 900$ che è ben maggiore di 600. Anche in questo caso, quindi, siamo molto conservativi per quanto riguarda lo sviluppo di un numero sufficiente di vortici (anche se per quanto riguarda L_z si è trattata più di una verifica che si rispettassero le condizioni minime, in quanto qui la larghezza del canale è stata data dalle dimensioni delle riblet, e dal numero di queste, consigliati secondo quanto riportato in letteratura). Passando invece alla dimensione delle celle del dominio di calcolo, anche qui, come si è visto in merito alle DNS, le celle devono essere di dimensioni minori rispetto alle strutture fisiche più piccole, che si vogliono catturare. In particolare si ha, che per ogni direzione, in funzione di come le strutture fisiche si sviluppano in quella direzione, è necessaria una dimensione minima. Per quanto riguarda la direzione x , abbiamo, che per essere sufficientemente accurati, è necessario avere un $\Delta x^+ \leq 12 - 15$. Nel nostro caso, essendo Δx^+ definito come $\Delta x^+ = \Delta x Re_\tau$, abbiamo $\Delta x^+ = 0,06 \cdot 200 = 12$ e quindi rientriamo nel valore minimo necessario. Per calcoli estremamente accurati si potrebbe prendere $\Delta x^+ = 10$, ma nel nostro caso, ciò avrebbe reso le simulazioni ancora più pesanti, creando problemi nelle run. Per quanto riguarda la direzione z , il discorso è molto simile a quello della direzione x , con la differenza che qui per essere sufficientemente accurati, è necessario avere un $\Delta z^+ \leq 4 - 5$. Nel nostro caso, considerando la divisione inizialmente fatta, ovvero quella senza *grading* lungo le riblet, abbiamo un Δz^+ , dato da $\Delta z^+ = \Delta z Re_\tau$, di $\Delta z^+ = 0,01875 \cdot 200 = 3,75$. Quindi si è già sotto i valori ideali per avere un'ottima accuratezza. Si ha poi che, con il *grading*, le celle vicino alla punta, diventano ancora leggermente più piccole, migliorando la precisione, mentre quelle vicino alla valle, diventano un pò più grandi, ma essendo il *grading* comunque ridotto, ed essendo nel caso generale ben al di sotto del valore di soglia, questa condizione è rispettata ovunque. Nella direzione z si è scelti di essere più conservativi rispetto a quella x , in quanto qui si è sotto al valore di soglia più accurato, perché, essendoci le riblet in questa direzione, si è voluto simulare queste con maggior accuratezza. Infine, nella direzione y , come si è già riportato in precedenza, è necessario che le celle vicino la parete, rispettino la condizione di $\Delta y^+ \leq 1$. Nel nostro caso, abbiamo appositamente scelto di avere la cella vicino alla parete $\Delta y = 0,005$, proprio per avere $\Delta y^+ = \Delta y Re_\tau = 0,005 \cdot 200 = 1$. In questo modo si è rispettata la condizione di accuratezza a parete, ma come già detto, farlo per tutto il canale avrebbe reso la mesh estremamente pesante per i calcoli, e quindi attraverso la divisione in blocchi ed il *grading*, si è andati ad aumentare progressivamente la dimensione delle celle verso il centro del canale, dove invece l'accuratezza richiesta è minore (dato che qui troviamo vortici di dimensioni ben maggiori), e quindi si possono avere celle più grandi, al fine di averne un numero totale minore.

Condizioni al contorno, casi preparati e mappatura delle simulazioni

Dai valori delle grandezze geometriche fin qui riportati è stato possibile andare a definire e realizzare la mesh dei casi da studiare. Per fare ciò è stato necessario, dopo aver dato appunto tutti i valori delle grandezze che caratterizzano il canale andare a definire, nell'ordine:

1. Le coordinate delle posizioni di tutti i punti caratterizzanti la mesh, in particolare quelli relativi all'inizio e fine di ogni semiriblet, oltre alla posizione di tutte le punte e le valli;

2. La posizione e la definizione esatta di tutti i punti caratterizzanti la mesh, con le 3 coordinate nello spazio;
3. Tutti i blocchi di discretizzazione, definiti mediante la loro forma, i punti che li delimitano, il numero di celle in cui discretizzarli nelle 3 direzioni e gli eventuali *grading*;
4. Tutte le facce esterne caratterizzanti la mesh, definiti mediante i punti che le costituiscono, e l'appartenenza alla relativa condizione al contorno;
5. Le condizioni al contorno caratterizzanti la mesh.

Tutto quello appena riportato può essere visto, nel dettaglio, nel codice `blockMeshDict`, riportato nell'appendice A.1. Invece, per quanto riguarda le condizioni al contorno, è necessario fare ulteriori precisazioni. Infatti, nello studio che si è andati ad effettuare, per alcuni dei casi analizzati, sono state preparate più di una versione della simulazione, con differenti condizioni al contorno. Il motivo verrà di seguito spiegato, nella parte relativa ai tempi di simulazione (vedi 4.2.2). Qui, invece, ci limiteremo a riportarne, soltanto le diverse condizioni al contorno applicate. Dal punto di vista delle condizioni al contorno, vengono preparate 2 versioni differenti della mesh (o 3 se si considera che in un caso la mesh va a copiare le condizioni al contorno da un altro caso, vedi sempre 4.2.2). Nello specifico, prima di indicare i due diversi casi, andiamo ad riportare quali sono le condizioni al contorno e le condizioni iniziali che è necessario fornire al programma. Si devono dare le condizioni al contorno per 6 superfici:

- Inlet
- Outlet
- BottomWall
- TopWall
- Left
- Right

Per 4 di queste superfici, BottomWall, TopWall, Left e Right, le condizioni al contorno sono sempre le stesse, in entrambi i tipi di simulazione, e sono:

- BottomWall = wall (parete solida, con attrito)
- TopWall = wall (parete solida, con attrito)
- Left = periodic (associata alla superficie Right)
- Right = periodic (associata alla superficie Left)

La condizione *wall* indica che si tratta di una superficie solida, che quindi non permette passaggio di flusso in direzione normale, e nel nostro caso è in condizione di non scivolamento, ovvero con attrito. La condizione *periodic*, invece, deve essere necessariamente associata ad un'altra superficie (come nel nostro caso, Left e Right) e rende il caso periodico o infinito, ovvero quello che entra/esce da questa superficie rispettivamente esce/entra da quella associata. Per le altre 2 superfici rimanenti, Inlet ed Outlet, ci sono 2 possibili configurazioni, in base ai 2 tipi di simulazioni svolte:

1. Per la prima simulazione abbiamo che entrambe hanno come condizione *patch*, ovvero può essere associato a questa superficie una qualsiasi condizione fisica (vedi dopo);
2. Per la seconda simulazione, invece, entrambe le superfici hanno come condizione *periodic*, associate una all'altra, e quindi si comportano nello stesso modo di Left e Right.

Per quanto riguarda il primo caso, quello con *patch*, entrando nello specifico, abbiamo che Inlet ha una velocità fissa, di modulo 1, in ingresso e un gradiente di pressione nullo, mentre Outlet ha gradiente di velocità nullo e pressione fissa, di intensità 0, equivalente alla condizione di ambiente esterno, come condizione di uscita. Tutte queste condizioni, sia quelle per i due casi di Inlet ed Outlet, e sia quelle associate alle altre superfici, devono essere indicate, sia nel codice *blockMeshDict*, riportato nell'appendice A.1, in cui si specifica il tipo di superficie che abbiamo, e poi definite maggiormente nel dettaglio (come per esempio il modulo di velocità di Inlet), negli specifici codici delle condizioni al contorno/iniziali, che si trovano in una cartella apposita (vedi appendice A.5, ed in particolare A.5.1). Per quanto riguarda le condizioni iniziali delle simulazioni, si devono specificare i campi iniziali di velocità e pressione. Questo viene fatto nei file che si trovano nella cartella del primo istante temporale, (vedi appendice A.5, ed in particolare A.5.1), in cui in un caso abbiamo campi uniformi, di valore nullo, mentre in un altro il campo delle condizioni iniziali viene copiato da un'altra simulazione (come verrà tra poco spiegato). Entrando nello specifico, nel nostro caso, per quasi tutte le simulazioni, sono necessarie, come condizioni al contorno/iniziali, 2 file, uno per la pressione, chiamato p , ed uno per la velocità, chiamato U . Questi si trovano nella cartella chiamata con il tempo dell'istante iniziale, se presente.

Infatti, come detto anche in precedenza, che questi file non sono sempre presenti, in alcuni casi, queste condizioni al contorno, non sono presenti nella simulazione. Si verifica ciò quando si vuole che la simulazione vada a *mappare* le proprie condizioni iniziali/al contorno, da un'altra simulazione. Ciò può avvenire in due modi differenti. Il primo caso, quello in cui i file con le condizioni al contorno/iniziali non sono presenti, è quello in cui si vanno a mappare esattamente le stesse condizioni da una simulazione all'altra, e ciò viene fatto, per esempio, quando si passa da una mesh meno fitta ad una più fitta (o viceversa) oppure da un dominio ad un altro più o meno grande, a patto che i punti che lo caratterizzano siano gli stessi (anche se in posizioni leggermente differenti) si abbiano lo stesso numero di superfici e che appunto le condizioni al contorno siano le stesse (quelle specificate nel codice *blockMeshDict*). In questo caso la mappatura, che serve anche a copiare le stesse condizioni di flusso da un caso all'altro, viene effettuata con il comando:

```
1 mapFields ../C_per_h0_transitorio -consistent
```

dove *../C_per_h0_transitorio* indica la cartella contenente la simulazione da cui mappare le condizioni al contorno/iniziali, e l'opzione *-consistent* specifica che bisogna fare proprio questo tipo di mappatura.

Il secondo caso, invece, è quello in cui si vuole andare a mappare il campo di moto da un'altra simulazione (che deve essere quanto meno simile a quella che si sta mappando), ma a differenza del caso precedente, si vuole andare ad imporre alcune condizioni al contorno differenti (come nel nostro caso riportato sopra, in cui in uno Inlet ed Outlet sono *patch*, mentre nell'altro sono *periodic*). In questo caso, si procede in modo leggermente differente. Come prima cosa, in questo caso, è necessario che siano presenti

sia la cartella che i file con, appunto, le nuove condizioni al contorno e/o iniziali, da imporre al nuovo caso. Inoltre è necessario inserire nella cartella *system* della simulazione (vedi A.3), il relativo file per la mappatura (come riportato in A.3.4). un codice chiamato *mapFieldsDict*, che serve a fare la mappatura, in cui si specificano eventuali condizioni particolari da rispettare (ma nel nostro caso se ne è usata la versione base, senza nessuna aggiunta, in quanto non erano presenti condizioni particolari). Inoltre, in questo caso il comando da utilizzare, per effettuare la mappatura, diventa:

```
1 mapFields ../C_iniz_h0
```

dove *../C_iniz_h0* indica sempre la cartella contenente la simulazione da cui mappare le condizioni al contorno/iniziali, mentre in questo caso *non* si deve aggiungere l'opzione *-consistent*.

4.2.2 Svolgimento simulazioni (run simulazioni)

Dopo aver definito tutti i parametri relativi alla parte geometrica del problema, la discretizzazione della mesh, le condizioni al contorno, le proprietà del fluido, prima di poter effettivamente lanciare le simulazioni, è necessario andare a definire i parametri temporali delle simulazioni, e quelli relativi ai solutori utilizzati. Qui vengono appunto riportate tutte queste informazioni per le simulazioni svolte. Come prima cosa, vengono riportate le altre impostazioni da fornire al codice di calcolo non ancora riportate, come i solutori utilizzati e/o altre proprietà. Dopo di che, invece, si riportano le impostazioni puramente temporali, con tutti i relativi commenti ed osservazioni. Infine si spiega il procedimento logico che si è fatto nell'andare a preparare le simulazioni, e secondo quale criterio ed ordine queste sono state lanciate.

Proprietà della simulazione e solutori utilizzati

Tra le opzioni che si devono fornire al codice di calcolo, per poter svolgere le simulazioni, ci sono i solutori, sia per quanto riguarda gli schemi numerici, e sia per il calcolo delle grandezze fisiche di nostro interesse. Tutte le impostazioni che si sono utilizzate in queste simulazioni (che sono le stesse per tutte le simulazioni) e sia tutte quelle che in generale è necessario fornire al codice per il suo funzionamento, sono riportate nei file *fvSchemes* e *fvSolution*, in cui troviamo rispettivamente le impostazioni per gli schemi dei solutori e le impostazioni per il calcolo delle grandezze fisiche. Laddove si dovesse fare simulazioni che richiedono altri solutori, o il calcolo di altre grandezze fisiche, queste vanno inserite, rispettivamente, in questi 2 file, con le relative opzioni, in base a ciò che si vuole. I codici dei due file utilizzati in queste simulazioni sono riportati nell'appendice A, e nello specifico, in A.3, dove sono riportati singolarmente quelli da noi utilizzati, *fvSchemes* in A.3.2 e *fvSolution* in A.3.3. Oltre a questi due file, per le simulazioni che abbiamo svolto, sono necessari altri 2 file. Il primo file necessario è quello chiamato *decomposeParDict*, nel quale è contenuto il codice con le impostazioni da fornire al programma, per poter dividere il dominio in vari sottodomini, per poter effettuare una simulazione in parallelo (che ci permette, al costo di utilizzare più processori, di svolgere le simulazioni in un tempo inferiore). In questo codice, prendendone la versione base, è necessario specificare quanti sottodomini si vogliono, e come questi devono essere ricavati, ovvero quante divisioni si devono fare per ognuna delle 3 direzioni. Nel nostro caso, si sono utilizzate due configurazioni differenti per le divisioni:

1. Una configurazione in cui si è diviso il dominio in 24 parti, rispettivamente, 6 lungo x , 2 lungo y e 2 lungo z ;
2. Una configurazione in cui si è diviso il dominio in 18 parti, rispettivamente, 3 lungo x , 2 lungo y e 3 lungo z .

La prima configurazione è stata utilizzata per simulare i primi due step delle simulazioni che fungono da base per tutte le altre (vedi dopo, 4.2.2), per le quali, dovendo simulare tempi molto lunghi, era necessario ridurre maggiormente questi tempi (utilizzando il massimo numero di processori per un singolo nodo di calcolo, 24, come dopo si spiegherà meglio, relativamente al sistema di calcolo utilizzato, vedi 4.2.2); la seconda configurazione è stata utilizzata per simulare il terzo step delle simulazioni, quello finale per ognuna, dove avendo un tempo da simulare minore, ci si è potuti permettere di dilatare un pò di più i tempi, per cercare di ottenere una priorità maggiore nel lancio delle simulazioni (vedi sempre dopo, in merito al sistema di calcolo), dovendone lanciare insieme, un numero maggiore.

Per quanto riguarda poi, dal punto di vista esecutivo, la divisione del dominio in sottodomini, e la ricostruzione di questo, una volta terminata la simulazione in parallelo, per avere i dati simulati su tutto il dominio intero, questo viene fatto, mediante i comandi:

```
1 decomposePar
2 reconstructPar
```

dove con `decomposePar` si da il comando di decomporre il dominio nel numero indicato di sottodomini, e con `reconstructPar` si da il comando di ricostruirlo.

Il penultimo file di codice che è servito per svolgere le simulazioni (anche se non per tutte) è il file `fvOptions`. Questo contiene il codice che serve per imporre alla simulazione opzioni di funzionamento aggiuntive, laddove necessarie. Come si è detto, nel nostro caso non è servito sempre; infatti nel caso iniziale, quello con velocità in ingresso sull'*Inlet* ed uscita sull'*Outlet*, non serve; al contrario, nell'altra configurazione simulata, quella con *Inlet* ed *Outlet* periodici e collegati, se non ci fosse altro, il flusso non avrebbe forze che ne porterebbero il moto. Nel nostro caso, tramite proprio il file `fvOptions` forniamo il codice al programma per imporre un gradiente di pressione medio, relativo ad un flusso di velocità media unitaria (quello che si aveva anche nella I configurazione, ma ora in modo periodico), che mantiene in movimento il flusso e permette di studiarne l'evoluzione. Tutta la parte relativa alle diverse configurazioni di simulazioni verrà tra poco spiegata (vedi dopo, 4.2.2), mentre il codice utilizzato `fvOptions` è riportato nell'appendice A, e nello specifico, nella sezione A.4.3.

Impostazioni temporali e controlDict

L'ultimo file che è necessario e fondamentale al programma, per lo svolgimento delle simulazioni è il file `controlDict`. Nel codice riportato in questo file, sono riportate sia il tipo di simulazione che si vuole svolgere, tutte le impostazioni relative al tempo di simulazione ed altre impostazioni relative alla precisione numerica dei calcoli, oltre ad eventuali aggiunte facoltative per opzioni o funzioni aggiuntive che sono necessarie (vedi parte relativa la post processing, 4.3). In merito alle impostazioni relative alla precisione numerica dei calcoli, nel nostro caso, sono state utilizzate tutte le impostazioni predefinite. Per quanto riguarda il tipo di simulazione da svolgere, si è utilizzata la funzione:

```
1 application      pisoFoam;
```

con la quale si va a simulare un flusso turbolento, incompressibile, non stazionario, tramite simulazione numerica diretta, con il modello di turbolenza indicato nel file *turbulenceProperties* (anche se nel nostro caso, impostando come modello `laminar`, si indica di non utilizzare nessun modello di turbolenza, ed eseguire quindi DNS).

Infine, si vanno a specificare tutte le impostazioni temporali. In particolare si devono fornire al codice il time iniziale delle simulazioni, il time finale delle simulazioni (il quale può essere espresso, invece che come time, come numero di iterazioni da svolgere), il passo temporale e il numero di ogni quanti istanti temporali simulati, si vuole salvare i valori calcolati. Tutte queste impostazioni verranno di seguito riportate, quando per ognuna delle simulazioni effettuate, si andrà ad indicare come questa è stata simulata. Qui invece, facciamo un'altra osservazione in merito al passo temporale. Come si è già detto, le DNS che vengono qui eseguite, per essere stabili, devono rispettare la condizione di stabilità di Courant, ovvero si deve far sì che il numero di Courant, ovvero il rapporto tra il passo temporale, moltiplicato per la velocità del flusso, e il passo spaziale, non sia superiore all'unità, altrimenti le simulazioni diventano instabili, e quindi oltre a fornire risultati non esatti, rischiano anche di *esplodere* durante le run. Nel nostro caso, come riportato in precedenza nella parte geometrica (vedi 4.2.1) si hanno diverse dimensioni delle celle, nelle 3 direzioni; tuttavia risulta che le dimensioni minori, quindi quelle che devono essere rapportate al passo temporale, sono quelle in direzione y . Qui si ha, come detto, che a parete, si hanno le celle più piccole di tutto il dominio, per poter catturare bene i più piccoli vortici che si formano a parete; ma poi, muovendoci verso il centro del canale, le dimensioni di queste aumentano. Il ragionamento più conservativo che si potrebbe fare sarebbe quello di utilizzare il valore di queste celle, nel calcolo del Δt ; tuttavia ciò causerebbe un passo temporale estremamente ridotto, allungando moltissimo i tempi di calcolo. Inoltre si deve tenere presente, che nel calcolo del numero di Courant è presente anche la velocità, ma questa è relativa alla singola cella che si sta considerando; ciò significa che per le celle a parete, abbiamo sì un $\Delta y = 0,005$ molto ridotto, ma essendo molto vicini alla parete, c'è anche una velocità molto bassa, quasi prossima allo 0. Ciò significa che quindi non è necessario prendere un Δt estremamente ridotto. Nonostante questo ragionamenti, inizialmente, come primo Δt di prova, si era comunque preso $\Delta t = 0,005$, per evitare eventuali problemi; ma dopo aver appurato da una simulazione di prova che il massimo numero di Courant era molto lontano da 1, si è deciso di raddoppiare il Δt , prendendo quindi $\Delta t = 0,01$, per diminuire i tempi di calcolo, senza avere comunque nessun problema di simulazione. Lanciando poi queste simulazioni si è appurato come, anche in questo caso, il numero di Courant fosse circa 0,5 quindi ancora ben distante da 1, ma si è preferito non aumentare ulteriormente il Δt , per non rischiare eventuali problemi, avendo comunque ridotto a sufficienza il tempo di calcolo necessario ad eseguire le simulazioni.

Tutte le impostazioni utilizzate nel codice del *controlDict* nelle simulazioni, sono riportate nel codice riportato nell'appendice A, e nello specifico, in A.2, che riporta il file di *controlDict* che si deve utilizzare per far girare le simulazioni; qui sarà chiamato *ControlDictRun* per distinguerlo da quello (che verrà tra poco descritto) che serve per il post processing, ma al momento di lanciare la simulazione, è comunque necessario rinominarlo in *controlDict*. Questo passaggio sarà spiegato nell'appendice A, nella parte in merito al codice di *submit* per l'esecuzione delle simulazioni sul sistema di calcolo utilizzato, riportato in `subsec:CodiciSubmit`. In merito allo svolgimento delle simulazioni, sul sistema di calcolo utilizzato, si andrà nella sezione successiva, a fare maggiore chiarezza.

Divisione, organizzazione e tempi delle simulazioni

Si va ora a descrivere il procedimento logico, già accennato alcune volte in precedenza, che è stato seguito, nella decisione dei casi da simulare. In particolare, come si è visto nella sezione relativa alla mesh (vedi 4.2.1), in questo studio si è deciso di analizzare 12 diverse configurazioni di riblet, in cui a variare è l'altezza espressa in termini di unità di parete (h^+). Tuttavia lo studio dell'effetto delle riblet deve essere fatto su di un flusso che sia turbolento, che abbia superato la fase di transizione, e che abbia quindi raggiunto la condizione di mediamente stazionario. Per avere un flusso del genere, è necessario, inizialmente simulare la condizione di un flusso esterno che entra in un canale, per simulare l'imbocco del canale (e questo avviene nella I configurazione di condizioni al contorno, quella in cui si ha una velocità in ingresso sull'Inlet, e una condizione di uscita sull'Outlet, vedi 4.2.1). Dopo di che, invece, per evitare di simulare un canale in tutta la sua interezza, che richiederebbe, in direzione x , dimensioni tali da portare ad una griglia spropositata, si utilizza la condizione di flusso periodico, simulando che ciò che esce dal canale, rientra dall'ingresso, in modo da seguire sempre lo stesso volume di fluido (ed è ciò che avviene nella seconda configurazione di condizioni al contorno, con Inlet ed Outlet che sono collegati dalla condizione *periodic*, vedi sempre 4.2.1). Tuttavia, anche in questo caso, per passare dalla condizione di flusso laminare, a quella di flusso turbolento prima, ed a quella di flusso turbolento mediamente stazionario poi, è necessario un tempo di calcolo comunque abbastanza alto, in funzione del numero di Reynolds scelto per lo studio (nel nostro caso, si è impiegato un tempo adimensionale, di 700, circa, per raggiungere le condizioni di flusso turbolento quasi stazionario). Si è deciso, per evitare di dover fare per tutti i casi ideati l'intera simulazione del transitorio del flusso, di prendere alcuni casi tra quelli preparati, ed effettuare la simulazione del transitorio, solamente per questi. Una volta poi raggiunta la condizione di flusso turbolento, gli altri casi, avrebbero interpolato sul loro dominio di calcolo, il campo di moto che si sarebbe venuto a trovare in questi casi *base* (mediante i comandi visti in 4.2.1, in merito alla parte relativa alla mappatura). I casi scelti come base sono stati:

- $H0$, da cui poi si sarebbero simulati anche $H3$, $H5$ e $H6$;
- $H7$, da cui poi si sarebbero simulati anche $H9$, $H10$ e $H12$;
- $H15$, da cui poi si sarebbero simulati anche $H18$, $H20$ e $H22$;

Dopo aver effettuato questa scelta, si è quindi proceduti nel seguente modo:

1. Per i casi $H0$, $H7$ e $H15$, si sono lanciate le simulazioni con le condizioni al contorno di flusso entrante nel canale (e queste simulazioni sono state denominate *iniziali*);
2. Per gli stessi casi $H0$, $H7$ e $H15$, si sono lanciate poi le simulazioni con le condizioni periodiche per il flusso che scorre nel canale, interpolando come condizione di partenza, quella finale delle rispettive *simulazioni iniziali* (e queste simulazioni sono state denominate *transitorio*);
3. Infine, sempre per i casi $H0$, $H7$ e $H15$, si sono lanciate nuovamente le simulazioni con le condizioni periodiche per il flusso che scorre nel canale, interpolando ora come condizione di partenza, quella finale delle rispettive *simulazioni transitorio*, al fine di avere, per questo step finale, delle simulazioni che fossero (circa

equivalenti in lunghezza, per tutti e 12 i casi che si sono voluti studiare (e queste simulazioni sono state denominate *periodiche*);

4. Si sono poi lanciati i casi $H3$, $H9$ e $H18$, questi già nella configurazione di caso periodico, interpolando come condizione di partenza, quella finale delle simulazioni, rispettivamente, $H0$, $H7$ e $H15$, periodiche (tutte le simulazioni, da qui in poi, vengono comunque denominate *periodiche*);
5. Si sono poi lanciati i casi $H5$, $H6$, $H10$, $H12$ e $H20$, sempre nella configurazione periodica, con le prime 2 ($H5$ e $H6$) che interpolano, come condizione di partenza, quella finale della simulazione $H3$, le successive 2 ($H10$ e $H12$) che interpolano, come condizione di partenza, quella finale della simulazione $H9$, ed $H18$, che interpola, come condizione di partenza, quella finale della simulazione $H15$;
6. Infine si è lanciato il caso $H22$, in configurazione periodica, che interpola, come condizione di partenza, quella finale della simulazione $H20$;
7. Dopo di che, per le simulazioni che lo richiedevano, ovvero quelle che non hanno ancora raggiunto la configurazione di flusso turbolento mediamente stazionario, sono state fatte proseguire per un altro T opportuno (come si è stabilito se una simulazione avesse raggiunto o meno la configurazione stazionaria, verrà tra poco spiegato).

Dopo aver riportato il processo logico che si è fatto, nel definire l'ordine con cui si sono concatenate le simulazioni, si riportano ulteriori dettagli in merito a questo procedimento. Inizialmente si sono andate a simulare i casi $H0$, $H7$ e $H15$, in configurazione iniziale, per un tempo di calcolo adimensionale ³ di $25T$. Dopo di che, mediante la mappatura del campo di moto, quella che prevede il cambio di condizioni al contorno, si è andati a mappare, rispettivamente per ogni caso, il T finale dei casi iniziali, come T dei rispettivi casi periodici, cambiando in questo modo le condizioni al contorno. A questo punto, i casi $H0$, $H7$ e $H15$, in configurazione periodica (in termini di condizioni al contorno, ma detti transitori), sono stati simulati per un T totale di 675 , raggiungendo come istante finale $700T$. A questo punto, volendo avere per tutti i casi, un intervallo utile (per il postprocessing, vedi 4.3 e 4.3.1), si è deciso, di fare per questi 3 casi, un ulteriore step simulativo, il cui istante di partenza, $700T$, fosse approssimativamente quello di partenza dell'intervallo utile. Si è quindi andati, utilizzando questa volta il comando di mappatura *consistente*, quello che permette di mappare una simulazione su di un'altra, senza cambiare le condizioni al contorno, ad interpolare il $700T$ delle rispettive simulazioni transitorie, come punto di partenza per le simulazioni, degli stessi casi, definite ora periodiche (quelle che sarebbero poi effettivamente state utilizzate per il postprocessing). Queste poi sono state simulate per un T di 150 , andando quindi da $700T$ a $850T$. Terminata (temporaneamente) l'esecuzione di questi primi 3 casi iniziali, si è andati a svolgere gli altri, nell'ordine sopra riportato, in cui, ogni caso, andava a mappare, sempre in modo *consistente*, l'ultimo T della simulazione a cui si collegava, prendendolo come punto di partenza. Ognuna di queste simulazioni, è stata poi simulata sempre per un T totale di 150 , ottenendo differenti intervalli utili; ciò non costituisce un problema, in quanto, avendo flussi stazionari, non è di interesse il T a cui ci si trova, ma l'intervallo di T considerato.

Per tutte le simulazioni svolte, in funzione dell'utilizzo che quei dati avrebbero avuto,

³da qui in poi, dovendo riportare diversi di questi valori, si abbrevierà tempo di calcolo adimensionale semplicemente come T .

Tabella 4.4: Parametri temporali importanti delle simulazioni effettuate

Nome caso	T iniz.	T fin.	I step sim.	T tot. sim.	ΔT_S	T utile	Caso interp.
H0 iniz.	0	25	0-25	0-25	5	25	n.d.
H0 tran.	25	700	25-700	25-700	5	700	H0 iniz.
H0 per.	700	950	700-850	700-950	2	850-950	H0 tran.
H3 per.	850	1100	850-1000	850-1100	2,5	1000-1100	H0 per.
H5 per.	1000	1200	1000-1150	1000-1200	2,5	1050-1200	H3 per.
H6 per.	1000	1200	1000-1150	1000-1200	2,5	1050-1200	H3 per.
H7 iniz.	0	25	0-25	0-25	5	25	n.d.
H7 tran.	25	700	25-700	25-700	5	700	H7 iniz.
H7 per.	700	1150	700-850	700-1150	2	1050-1150	H7 tran.
H9 per.	850	1050	850-1000	850-1050	2,5	950-1050	H7 per.
H10 per.	1000	1150	1000-1150	1000-1150	2,5	1090-1150	H9 per.
H12 per.	1000	1150	1000-1150	1000-1150	2,5	1090-1150	H9 per.
H15 iniz.	0	25	0-25	0-25	5	25	n.d.
H15 tran.	25	700	25-700	25-700	5	700	H15 iniz.
H15 per.	700	900	700-850	700-900	2	780-900	H15 tran.
H18 per.	850	1000	850-1000	850-1000	2,5	930-1000	H15 per.
H20 per.	1000	1150	1000-1150	1000-1150	2,5	1080-1150	H18 per.
H22 per.	1150	1350	1150-1300	1150-1350	2,5	1250-1350	H20 per.

si è andato ad utilizzare un passo di salvataggio dei dati ⁴, differente. Per i 3 casi *H0*, *H7* e *H15*, sia in configurazione iniziale, che in configurazione transitoria, si è utilizzato come ΔT_S , $5T$, in quanto non interessava la storia temporale, ma solamente arrivare all'ultimo istante temporale per continuare. Per questi 3 casi, ma in configurazione periodica, quella finale, si è utilizzato come ΔT_S , invece, $2T$, in quanto in questo caso si, è di interesse la storia temporale, per capire se si fosse arrivati o meno in condizione di mediamente stazionario. Infine, per tutti gli altri 9 casi, in configurazione periodica, il ΔT_S è stato di $2,5T$, in quanto, dopo aver osservato i casi precedenti, si è capito che questo livello di dettaglio era accettabile, ma salvando meno dati, le simulazioni risultavano essere più veloci e meno pesanti.

Come detto, inoltre, tutte le simulazioni *periodiche*, sono state simulate per $150T$. Tuttavia, una volta fatto ciò, non si era certi che queste avessero effettivamente raggiunto la condizione di mediamente stazionarie (ed infatti per buona parte di esse non la si era ancora raggiunta); tramite le visualizzazioni che è possibile fare con *ParaFOAM*, infatti, è si possibile osservare la transizione del flusso, da laminare a turbolento, ma non è possibile capire se il flusso turbolento sia diventato mediamente stazionario. Per capire ciò, allora, si lancia una prima volta il codice di postprocessing (che sarà spiegato in 4.3 e riportato nell'appendice B) per ottenere l'andamento temporale, per ogni simulazioni, della media delle fluttuazioni $u'u'$ e $v'v'$ (oltre a tutte le altre grandezze di postprocessing, che però non interessano qui). Osservando quindi gli andamenti temporali di queste fluttuazioni, soprattutto per i valori di y^+ cui si hanno le massime fluttuazioni ($y^+ = 15$ e $y^+ = 20$), si è potuto capire quando si fosse raggiunta la condizione di stazionarietà, ovvero quando queste curve fossero diventate lineari (come si può osservare nei grafici riportati nella sezione 5.2).

In particolare si è scelto di andare a valutare l'andamento delle curve delle fluttua-

⁴che verrà ora abbreviato con ΔT_S

zioni $u'u'$ e $v'v'$, rispetto a quello della velocità media \bar{U} , per esempio, perché si ha che *i momenti di ordine superiore, quali queste 2 fluttuazioni sono, convergono più lentamente rispetto a quelli di ordine inferiore, quale la velocità media; quindi, laddove questi vanno a convergere, si ha una maggiore sicurezza di essere arrivati nella condizione di mediamente stazionaria.* In questo modo, osservando i grafici ottenuti, si è potuto capire se si fosse raggiunti o meno la stazionarietà; in caso negativo, le simulazioni sono state fatte continuare per step simulativi di $50T$ o $100T$ (in funzione di quanto si ipotizzava essere ancora lontani dalla stazionarietà) alla volta, fino a che tutte non hanno raggiunto questa condizione. Inoltre, sempre osservando queste curve, si è potuti andare a valutare intervalli di dati che fossero tutti mediamente stazionari, e sono stati questi intervalli, quelli ad essere davvero presi in considerazione, per le elaborazioni successive del postprocessing, la realizzazione di grafici esplicativi e il calcolo delle grandezze obiettivo dello studio (nella sezione 5.2 viene spiegato come sono stati presi gli intervalli utili, mentre nella sezione 4.3.1 si spiega a cosa serve in particolare l'intervallo che è stato scelto). Tutte le informazioni importanti in merito ai tempi utilizzati per le simulazioni, come tempi di simulazione, intervalli utili stazionari, passo temporale di salvataggio dati (ΔT_S), quale simulazione interpolano come punto di partenza, ecc.. sono riportate nella tabella 4.4.

Lancio delle simulazioni sul sistema di calcolo

Avendo tutti i parametri necessari allo svolgimento delle simulazioni, si è andati ad eseguirle. In particolare l'esecuzione delle DNS di questo lavoro di tesi, è stato fatto sul calcolatore:

HACTAR [“Risorse di calcolo fornite da `hpc@polito` (<http://www.hpc.polito.it>)], un calcolatore costituito da 29 nodi di calcolo, ognuno a sua volta formato da 24 core computazionali, con circa 3,7TB di RAM, (per altre specifiche fare riferimento al sito appena riportato). Questo, come detto è costituito da nodi di calcolo i quali sono formati da 24 core computazionali, ciascuno. Quando si è parlato dell'esecuzione delle simulazioni (vedi 4.2.2) si è detto che per i casi più lunghi, quelli con le condizioni iniziali e quelli con le condizioni transitorio, si è utilizzato il massimo numero di processori (o core) di un nodo di calcolo del calcolatore (*HACTAR*). Questo perché, nell'andare a dividere il dominio in pezzi, per svolgere le simulazioni in parallelo, si deve tenere conto della perdita temporale, causata dalla necessità di comunicazione tra i vari sotto-domini; questa diventa più lenta, se i domini che devono comunicare, si trovano su nodi computazionali differenti. Negli altri casi, avendo simulazioni su T minori, si è scelto di utilizzare solamente 18 core di calcolo per avere una priorità più alta nella coda del sistema.

Al di là degli aspetti tecnici del sistema di calcolatori utilizzato, per quanto riguarda la vera e propria esecuzione delle simulazioni, queste, richiedono prima del vero e proprio lancio, la creazione della mesh; questa avviene, dopo averne fornito tutti i dati nel file *blockMeshDict*, attraverso il comando:

```
1 blockMesh
```

che deve essere lanciato da linea di comando, trovandoci nella cartella principale della simulazione. Dopo di che, si può lanciare la simulazione, utilizzando il comando che ha il nome del tipo di solutore indicato nel *controlDict*, nel nostro caso:

```
1 pisoFoam
```

Questo però, se si vuole fare una run (ovvero l'esecuzione della simulazione) in modo diretto e senza dividere il dominio in pezzi. Nel nostro caso, invece, che si è proceduti mediante run in parallelo, dopo aver creato la mesh, si è andati a dividere il dominio in pezzi, mediante l'opportuno comando (come riportato nella relativa sezione) e solamente dopo si è lanciata la simulazione, con il comando che ora diventa:

```
1 mpirun pisoFoam -parallel &> log
```

Dopo di che, si va a ricostruire il dominio, utilizzando il comando appropriato. Tuttavia la sequenza di operazioni riportata si utilizza per lanciare in maniera istantanea la simulazione; nel caso, invece, di *HACTAR*, è necessario inserire tutte queste operazioni all'interno di un file di *submit*, da dare al sistema di controllo della macchina, che provvede poi, tramite questo, a lanciare la simulazione. Un esempio di un codice di *submit* utilizzato in questa simulazione, è riportato nell'appendice A, nella parte in merito al codice di *submit* per l'esecuzione delle simulazioni sul sistema di calcolo utilizzato, riportato in subsec:CodiciSubmit. Si specifica, nuovamente che le "Risorse di calcolo sono fornite da *hpc@polito* (<http://www.hpc.polito.it>).

Come informazione complementare si va riportare che, per quanto riguarda il lancio della simulazione in parallelo, qui si è riportato il comando:

```
1 mpirun pisoFoam -parallel &> log
```

Tuttavia questo va bene nel caso che la simulazione in parallelo si svolga su *HACTAR*, e venga lanciata mediante di codice di *submit*, in quanto questa è la forma del comando che viene accettata. Ma, laddove si voglia lanciare una simulazione in parallelo, in maniera diretta, senza il file di *submit*, il comando da utilizzare diventa:

```
1 mpirun -np 4 pisoFoam -parallel > log &
```

In cui, come si osserva è necessario specificare tramite l'opzione `-np 4` il numero di processori su cui si vuole fare la simulazione, che corrispondono ai pezzi in cui si è diviso il dominio. Su *HACTAR*, invece, questo viene specificato nel codice di *submit*, e quindi non lo si deve dire anche nel comando di run.

Si è quindi andati ad eseguire le DNS, su *HACTAR*, per ottenere i campi di moto che si instaurano all'interno di un canale piano, sulle cui pareti sono presenti delle *riblet*, e al cui interno scorre un flusso turbolento. Tuttavia, ottenere i campi di moto per le configurazioni che si è voluti studiare, non ci fornisce tutti i dati di cui abbiamo bisogno per il nostro studio, ma sono necessari altri step di calcolo, come riportato di seguito.

4.3 Codice di postprocessing

Dopo aver lanciato tutte le simulazioni che sono state preparate per questo studio, si sono ottenuti tutti i dati che servono in merito ai campi di moto che si hanno all'interno di canali piani, con diverse configurazioni di riblet. Tuttavia i soli campi di moto, ovvero le 3 componenti di velocità e quella di pressione, in ogni punto del dominio, per ogni istante temporale simulato, così come sono state ottenute, non servono ad altro che a permetterci di fare visualizzazioni del flusso, mediante il programma *ParaFOAM*; tuttavia alcune di queste visualizzazioni sono comunque utili e ci permettono di capire le differenti evoluzioni del flusso, e sono riportate nel relativo paragrafo, vedi 5.4. Comunque, per poter proseguire lo studio, e trovare tutte le grandezze di nostro interesse, è necessario estrarre, per ogni simulazione, da tutto il campo di moto che si è simulato, le informazioni che servono. Per poter fare ciò si va ad utilizzare un altro codice per *OpenFOAM*, una variante del *controlDict*, che ci permette di estrarre i dati. In particolare in questo file, troviamo, oltre agli stessi input che ha il *controlDict* utilizzato per l'esecuzione delle simulazioni, una funzione, aggiunta alla fine del codice, che permette di fare la media temporale, per le grandezze che interessano. Nello specifico, in questo studio, serve la media della velocità \bar{U} , e l'andamento di tutte le componenti delle fluttuazioni di velocità, in particolare: $u'u'$ $v'v'$ $w'w'$ e $u'v'$. Per fare ciò si specifica nella funzione del *controlDict*, che le grandezze di cui fare la media temporale, e che poi devono essere salvate in un file, sono queste, la U , e le fluttuazioni di velocità della U . Tuttavia la sola funzione definizione della funzione serve solamente ad indicare cosa si vuole, ma per ottenere effettivamente dei risultati, è necessario indicare anche da dove questi devono essere presi; nello specifico, nel nostro caso, si vanno a fare delle medie temporali su piani, normali all'asse y . Si deve quindi indicare, nel codice, ogni piano, su cui si vuole che la funzione vada a prendere i dati, specificando per ogni piano, il punto di origine, e la direzione a cui questo piano deve essere normale. Dovendo quindi andare, in questo caso, ad indicare dei valori di y (i piani normali all'asse, che lo intersecano il quella coordinata, per essere precisi), si è scelto di utilizzare gli stessi valori (e tutti) che sono stati ottenuti in precedenza, mediante il codice Fortran chiamato *Newton*, che era stato utilizzato per definire la spaziatura delle celle lungo l'asse y , appunto, e che come output, oltre a fornire i numeri da dare a *OpenFOAM* per ottenere la griglia voluta, ci dà anche la posizione dei valori y , che si ottengono utilizzando questi valori. Si è quindi andati a realizzare il codice *controlDict* per il postprocessing, indicando le coordinate di 245 piani da cui estrarre la media temporale della velocità e delle sue fluttuazioni. Il codice di postprocessing utilizzato è riportato interamente nell'appendice B, nella sezione B.1, dove viene chiamato *controlDictReynolds*, come si era fatto anche per quello utilizzato per il run delle simulazioni, per non confonderli, anche se, al momento di utilizzarlo, deve essere rinominato semplicemente *controlDict*, per essere accettato da *OpenFOAM*. Una volta che questo codice è stato realizzato, rinominato, ed inserito nella cartella della simulazione (in realtà i codici *controlDict* vanno inseriti nella cartella *system* delle simulazioni, vedi appendici A e B), questo viene lanciato mediante il comando:

```
1 pisoFoam -postProcess
```

Anche questo comando, su *HACTAR*, viene inserito in un apposito file di *submit*, riportato anche questo, con tutti i suoi input, nell'appendice B, nella sezione B.3.

4.3.1 Riordino dei dati di postprocessing

Una volta lanciato il codice di postprocessing, si ottengono, per ogni piano indicato nella function, e per ogni istante temporale precedentemente simulato, i valori delle grandezze richieste. Però, per come il codice di postprocessing funziona, i dati vengono forniti in un ordine che risulta di difficile utilizzo pratico; infatti abbiamo una cartella per ognuno dei piani che si è indicato nella funzione, ed all'interno di queste, in una sottocartella avente come nome il primo istante temporale della simulazione su cui si è fatto il postprocessing, un file con le grandezze volute, ordinate per colonne, dove la prima riporta gli istanti temporali della simulazione e per cui si è fatto il postprocessing, mentre le altre, sono ognuna, uno dei dati indicati, nel nostro caso: \bar{U} $u'u'$ $u'v'$ $u'w'$ $v'v'$ $v'w'$ $w'w'$. Con questo ordine dei dati di postprocessing, è possibile avere gli andamenti temporali delle grandezze estratte per i diversi piani (o per le diverse coordinate y , relative ad ogni piano), ma non avere un profilo di velocità, o di un'altra grandezza, al variare della y . Comunque è con questi andamenti temporali, quelli di $u'u'$ e $v'v'$, che si è andati a determinare se una simulazione fosse arrivata o meno nella condizione di flusso mediamente stazionario, e si è stabilito l'intervallo utile dei dati, ovvero l'intervallo di stazionarietà (come visto in 4.2.2).

Diventa necessario quindi, andare a riordinare in modo diverso, i dati ottenuti dal post processing, per poterli utilizzare in tutti i modi che è necessario per lo studio che si sta effettuando. Per fare ciò, si è andati a realizzare un altro codice Fortran, chiamato `ricostruisci_umedial_v2bis_ureynolds_RE_all.f90`, che poi, dopo compilato, come eseguibile, diventa `riordina_Um_URey_RE_all.x`, che realizza ciò. In particolare, questo codice va a leggere tutti i file prodotti dal post processing, e poi li riorganizza nel seguente modo: crea un file per ognuna delle grandezze calcolate dal postprocessing, e in questi inserisci, per ogni grandezza, i relativi dati, ordinandoli in modo tale che ogni colonna sia un diverso istante temporale, riportante tutti i dati per tutte le coordinate y , per poter quindi fare i profili delle grandezze in funzione di y ; inoltre, per ognuno di questi file, aggiunge come prima colonna l'intero vettore delle coordinate y , in modo da facilitare la realizzazione di grafici utilizzando questi dati, oltre ad aggiungere, in testa ad ogni colonna, sotto forma di commento, sia a quale istante temporale corrispondono, e sia il numero della colonna, per facilitare l'utilizzo di questi dati con programmi di grafici, come *gnuplot*. Inoltre il programma crea un altro file, in cui va a fare la media temporale, per ogni grandezza calcolata, e per ogni coordinata y , dei dati letti, previa richiesta di un preciso T , da cui partire, in modo tale, da escludere nella media, i dati corrispondenti alla condizione di flusso non ancora stazionario. Infine, in questo file, oltre a riportare anche qui il vettore delle y , il programma va a calcolare ed inserire una colonna riportante la derivata di U rispetto a y , valore che è fondamentale nel calcolo della resistenza d'attrito viscosa, prendendo come valori di U , per il calcolo della derivata, solamente quelli ottenuti nel file media. L'intero codice Fortran realizzato è riportato nell'appendice B, nella sezione *Codice riordino grandezze Reynolds* in B.2. Inoltre questo codice, per funzionare correttamente richiede, oltre alla presenza della cartella contenente i dati di postprocessing, che deve essere chiamata `postProcessing_RE_all`, la presenza di un file di testo riportante tutte le coordinate y per cui si è fatto il postprocessing, che deve chiamarsi `coordinate_y.txt`, ed alcuni input. Questi input possono essere forniti anche tramite un opportuno file di testo, indicandolo poi al momento del lancio del codice, che avviene mediante il comando:

```
1 ./riordina_Um_URey_RE_all.x < input_media.riordina_rey
```

dove `input_media.riordina_rey` è il nome del file contenente gli input da fornire al codice di riordino. Anche una versione del file di input del codice di riordino è riportata nell'appendice B, nella sezione *Input codice Riordino* in B.2.1; in questo file viene specificato anche quali input si devono fornire al codice; invece, se il codice viene lanciato senza file di input, li va a chiedere a schermo, specificando anche di che input si tratta. Inoltre nell'appendice B, nella sezione *Coordinate Y* in B.2.2, è riportato anche un file `coordinate_y.txt`, in cui sono contenuti tutti i valori di y per cui si è fatto il post processing, e che serve anche per il lancio del codice di riordino.

4.3.2 Realizzazione Immagini

Dopo aver ottenuto i dati di post processing, riordinati, mediati, ed aver calcolato anche la derivata rispetto ad y della velocità U , è possibile andare a realizzare dei grafici che mostrano l'andamento di diverse grandezze di interesse per lo studio. Questi grafici vengono realizzati mediante l'utilizzo congiunto di un codice *gnuplot*, che serve a realizzare effettivamente l'immagine, e di un codice \LaTeX , che invece serve a realizzare in modo più chiaro ed ordinato, tutte le scritte che compaiono sul grafico, come le etichette degli assi, quelle delle curve, ed altre che eventualmente possono servire. Non si entrerà nel dettaglio di come tutti i codici utilizzati sono fatti, trattandosi appunto di semplici codici *gnuplot* e \LaTeX , personalizzati in base alle proprie esigenze. Tuttavia, uno di questi codici *gnuplot*, risulta essere di particolare interesse per quello che è lo scopo di questa tesi, perché, attraverso questo, si è andati a calcolare lo *sforzo d'attrito a parete*, τ_w . In particolare, il codice di nostro interesse è quello utilizzato per diagrammare l'andamento, in funzione della distanza da parete, delle componenti della resistenza che si hanno nel canale, quella di inerzia e quella viscosa, ed anche la loro somma (vedi 5.5.3). Questo codice, risulta interessante perché, al suo interno, oltre a diagrammare gli sforzi, viene calcolato lo *sforzo d'attrito a parete*. Ciò risulta essere interessante, in quanto, generalmente, nel caso di un canale piano turbolento, lo *sforzo di taglio a parete* viene calcolato mediante il legame di questo con le perdite di pressione che si hanno lungo il canale. Tuttavia, nel nostro caso, dato l'unico codice funzionante per applicare il gradiente di pressione al flusso nel caso periodico, e come questo va ad intervenire sul campo di moto, generando un flusso di valore medio pari a quello imposto da noi, nel file *fvOption* (ovvero velocità $U = 1$), non è possibile calcolare un gradiente di pressione tra ingresso ed uscita del canale, in quanto questo risulta nullo. Si è quindi dovuti procedere in modo differente. Riprendendo l'equazione (4.1), è possibile interpretare questa, conoscendo anche l'andamento della resistenza in funzione della distanza da parete, come l'uguaglianza tra 2 rette, con lo stesso coefficiente angolare, espresso in termini differenti; andando a riscrivere l'equazione, infatti, otteniamo:

$$\left(-\langle \rho u'v' \rangle\right)(y) + \left(\mu \frac{d\langle u \rangle}{dy}\right)(y) = (-G)(y)$$

da cui otteniamo:

$$\left[-\langle \rho u'v' \rangle + \mu \frac{d\langle u \rangle}{dy}\right](y) = [-G](y) \quad (4.12)$$

Vista in questo modo, l'equazione, considerando i 2 termini come 2 rette, si traduce in un'uguaglianza dei coefficienti, con cui otteniamo:

$$-\langle \rho u'v' \rangle + \mu \frac{d\langle u \rangle}{dy} = -G$$

Ora, richiamando l'equazione dell'equilibrio delle forze in un canale piano turbolento, abbiamo:

$$-\Delta P H = 2\tau_w \Delta x$$

da cui otteniamo:

$$\tau_w = -\frac{\Delta P}{\Delta x} \frac{H}{2} = -G \frac{H}{2}$$

che nel nostro caso, avendo preso $H = 2$, diventa:

$$\tau_w = -G$$

Andando a sostituire quanto ottenuto nella relazione precedente otteniamo che τ_w è:

$$\tau_w = -G = -\langle \rho u'v' \rangle + \mu \frac{d\langle u \rangle}{dy}$$

Si è così trovata una relazione per determinare lo *sforzo d'attrito a parete*, τ_w . Infatti, andando a fare un'interpolazione lineare della curva (che poi come si vede è effettivamente una retta) della somma dei due coefficienti di resistenza, $-\langle \rho u'v' \rangle$ e $\mu \frac{d\langle u \rangle}{dy}$, il coefficiente di interpolazione, chiamato a , sarà proprio il valore di τ_w . Riassumendo abbiamo quindi, che per trovare il valore di τ_w , si è seguito questo procedimento:

1. Si è andati a diagrammare le componenti della resistenza, $-\langle \rho u'v' \rangle$ e $\mu \frac{d\langle u \rangle}{dy}$, e la loro somma;
2. Si è andati ad interpolare linearmente, mediante un coefficiente chiamato a , l'equazione della somma delle 2 componenti;
3. Per il procedimento appena riportato, si ha che a corrisponde proprio al valore di τ_w , ovvero lo *sforzo d'attrito a parete*;
4. Mediante il calcolo di questa grandezza, si è poi potuti andare a calcolare prima la velocità d'attrito u_τ , e poi tutti gli altri valori di nostro interesse, ad essa collegata, come u^+ , y^+ , ecc;
5. Dal valore trovato di τ_w , di è calcolato anche C_f , il coefficiente d'attrito adimensionale, definito come $C_f = \frac{\tau_w}{1/2\rho U^2}$, che nel nostro caso, avendo $\rho = 1$ e $U = 1$, diventa $C_f = 2\tau_w$.

Con il procedimento qui riportato, in questo studio, si è andati a calcolare lo *sforzo d'attrito a parete*, τ_w , con il quale si è trovata u_τ , e si è potuto realizzare tutti i grafici con le grandezze espresse in unità di parete. Inoltre si è potuti andare a graficare i dati ottenuti sulla resistenza che si ha per ogni configurazione, che sono riportati nella sezione 5.6. Bisogna però tenere presente, come si vedrà poi quando si riporteranno i relativi grafici (vedi 5.5.3), come l'interpolazione che si va ad effettuare, al crescere dell'altezza delle riblet delle configurazioni analizzate, sia affetta da un errore crescente, causato dallo spostamento dalla posizione classica, delle curve, sia degli sforzi viscosi e sia della resistenza totale. Questo errore dovuto all'interpolazione, si va quindi ad aggiungere a quello dovuto alla discretizzazione della mesh, ed a quello causato dalla distorsione della griglia che si ha al crescere dell'altezza delle riblet.

Solamente la piccola parte di codice *gnuplot*, utilizzata appunto per l'interpolazione ed il calcolo quindi di τ_w , è riportata nell'appendice B, nella sezione *Interpolazione Gnuplot* in B.2.3. I restanti codici *gnuplot* e \LaTeX , utilizzati per la realizzazione delle immagini, non sono invece riportati.

Capitolo 5

Dati e Risultati

5.1 Risultati ottenuti dalle simulazioni

Dopo aver eseguito tutte le simulazioni numeriche, aver ottenuto i campi di moto per ogni configurazione di campo che si è voluta studiare, caratterizzata da una diversa altezza delle riblet, ed aver estratto da queste i dati di post processing utili alla comprensione del fenomeno fisico, si hanno tutti gli strumenti per poter finalmente raggiungere l'obiettivo postoci in questo lavoro di tesi, ovvero trovare quale configurazione di riblet vada a ridurre maggiormente la resistenza d'attrito, per un flusso turbolento che scorre in un canale piano. Per fare ciò, utilizzando i dati ottenuti dal post processing, sono stati realizzati diversi grafici, che riportando le grandezze calcolate offrono una comprensione chiara ed immediata del fenomeno che si ha, e ci permettono di capire cosa cambia tra una configurazione di riblet ed un'altra. Inoltre, durante la realizzazione di questi grafici, si sono andati ad estrarre ulteriormente altri dati fondamentali per lo studio effettuato, ed anche questi, sono stati tabellati prima e graficati poi, per mostrare in modo chiaro e netto il risultato finale del lavoro svolto. In aggiunta, come già detto in precedenza, prima di andare ad effettuare il post processing per estrarre i dati utili, si è come prima cosa utilizzato *ParaFOAM*, per visualizzare il campo di moto simulato, ed avere una prima idea di come il campo di moto simulato risultasse e di come si fosse evoluto a causa della presenza delle diverse configurazioni di riblet. Da queste visualizzazioni sono state estratte delle sezioni di piani del campo di moto, che possono essere interessanti per ciò che mostrano, in quanto, grazie a queste sezioni, è possibile confrontare direttamente il campo di moto che si ha, nello stesso punto, per simulazioni con altezza di riblet differente, ed osservare così come questo sia cambiato. In quest'ultima parte del lavoro, andremo ora a riportare, nell'ordine:

1. I grafici riportanti l'evoluzione temporale delle grandezze calcolate, per capire come le simulazioni si sono sviluppate nel tempo, durante la loro esecuzione;
2. Un grafico che riporta il confronto tra i dati del nostro canale piano, senza riblet, e i dati presenti in letteratura, in merito agli studi di *Kim, Moser & Mansour* [8];
3. Le visualizzazioni del campo di moto, fatte per 3 diversi tipi di piani, ovvero 3 piani perpendicolari ai 3 differenti assi, per diverse posizioni e per diverse configurazioni di riblet;

4. I grafici che riportano gli andamenti, per le diverse configurazioni analizzate, delle diverse grandezze che caratterizzano la turbolenza, per capire come questa cambia, al cambiare dell'altezza delle riblet;
5. Alcuni grafici riportanti l'andamento degli sforzi d'attrito, mediante i quali si è andati a calcolare la resistenza d'attrito per ogni configurazione analizzata;
6. I grafici riportanti l'andamento dei profili di velocità per i diversi casi, confrontando le differenze, e cosa queste significano a livello di resistenza;
7. I dati ottenuti, in merito alla resistenza delle singole configurazioni, indicanti in quale caso si ha la resistenza minore;
8. Le conclusioni finali in merito al lavoro svolto, con il commento ai risultati ottenuti.

In merito a tutte le immagini che si andranno a riportare da qui in poi, si deve tenere conto, che nell'andare a realizzare le immagini, nella direzione y , si è andati ad effettuare un cambio di intervallo, passando da $[0; 2]$ che si è usato nelle simulazioni, a $[-1; 1]$ che si è utilizzato nelle immagini, sia visualizzazione che grafici. Ciò è stato fatto, in quanto l'intervallo $[0; 2]$ risultava essere più comodo da utilizzare nella programmazione e realizzazione della mesh, in quanto non prevede cifre negative, che creano problemi su *OpenFOAM*, perché richiedono una scrittura particolare ed articolata; mentre poi si utilizza l'intervallo $[-1; 1]$ per le immagini, per uniformarsi a quelle presenti in letteratura.

I primi grafici che andiamo ora a riportare ed osservare, come detto sono:

- Gli andamenti temporali di alcune delle grandezze analizzate, sia per comprendere quando si ha la condizione di flusso turbolento mediamente stazionario, sia per osservare la diversa convergenza tra grandezze differenti;
- Il confronto tra i dati del canale piano senza riblet analizzato (caso $H0$), e i dati delle simulazioni di un canale piano turbolento, svolte da *Kim, Moser & Mansour* [8], per appurare la validità e la qualità delle simulazioni svolte.

Dopo di che, si riporteranno le visualizzazioni effettuate per il flusso, ed i restanti grafici, come sopra indicato.

5.2 Scelta intervallo temporale

Per prima cosa, si è determinato l'intervallo temporale in cui il flusso possa ritenersi mediamente stazionario e che quindi possa essere utilizzato per calcolare le statistiche del flusso. A tal fine, si sono considerate i primi due momenti delle velocità, ovvero medie e varianze della componenti di velocità. Per ogni istante, si sono determinate la velocità media e la varianza delle componenti di velocità mediando su piani paralleli alla parete. Si è quindi individuato un intervallo di tempo in cui queste medie, calcolate su un singolo istante, oscillassero intorno ad un valore medio costante, ovvero in cui la velocità media e la varianza non si discostassero di più del 5%, soprattutto per quanto riguarda la varianza, la quale essendo un momento di ordine superiore, e convergendo quindi più lentamente, ci dà maggiore certezza sulla convergenza della velocità. In questo intervallo di tempo si possono considerare conclusi gli effetti delle condizioni iniziali ed il flusso è rappresentativo del flusso mediamente stazionario che s'instaura in un canale turbolento, sicché può essere utilizzato per determinare le statistiche del flusso. Le simulazioni sono state eventualmente proseguite tutte in modo che l'ampiezza di questo intervallo fosse almeno pari a $100T$, ovvero a 100 tempi caratteristici del problema studiato, definito nel nostro caso come $\frac{h}{U_b} = 1$; per alcuni casi simulazioni, si è trovato che questo intervallo, dai grafici, risultava essere più ampio, e lo si è preso di $150T$, mentre per altre, che convergevano molto velocemente, lo si è preso di $80T$. Comunque per ogni simulazione il numero esatto di tempi caratteristici, essendo questo uguale ad 1, è riportato nella tabella 4.4, in particolare considerando l'intervallo riportato T utile, ovvero la settima e penultima colonna. Il tempo caratteristico di una simulazione può anche essere definito come $\frac{h}{u_\tau}$, ma nel nostro caso, essendo u_τ calcolabile solamente dopo che si fosse arrivati in condizione di mediamente stazionario, decidere il tempo di simulazione in funzione di u_τ , si sarebbe trasformato in un problema implicito, iterativo, che avrebbe notevolmente complicato ed allungato il tempo necessario allo svolgimento delle simulazioni. Si è quindi preferiti utilizzare $\frac{h}{U_b}$. Nel caso di $\frac{h}{u_\tau}$, comunque, il numero minimo di *tempi caratteristici* da considerare, sarebbe stato differente, circa 10.

Le figure 5.1, 5.2, 5.3, 5.4 e 5.5 mostrano l'andamento temporale della velocità media U e della varianze $\langle u'u' \rangle$ e $\langle v'v' \rangle$ in alcuni piani, scelti tra quelli in cui la U è massima e quelli nella regione in cui la varianza della velocità longitudinale è massima. Sono infatti riportati, alcuni grafici relativi agli andamenti temporali veri e propri delle grandezze, di U , $u'u'$ e $v'v'$, prese per alcune coordinate y , con le quali si è andati a determinare quali fossero gli intervalli utili dei dati, ovvero quelli per cui si è nella condizione di flusso turbolento mediamente stazionario. Oltre a questi, abbiamo, invece, dei profili di velocità di U e di $u'u'$, in funzione della distanza da parete y , ma che confrontano questi profili, per diversi tempi differenti, potendo così osservare l'evoluzione nel tempo di questi profili. Grazie a questi andamenti, in particolare i primi, si è quindi potuto caratterizzare meglio l'evoluzione temporale delle componenti di velocità.

Nei grafici 5.1, 5.2, 5.3, 5.4 e 5.5 sono riportati gli andamenti nel tempo delle medie di U , $u'u'$ e $v'v'$, per 2 diversi valori di y , $y = 0,075$ e $y = 1$. Da questi è possibile osservare, per ogni configurazione di riblet, da quale istante temporale inizi la zona di flusso turbolento mediamente stazionario, osservando gli andamenti di $u'u'$ e $v'v'$, in particolare, in quanto si ha che *i momenti di ordine superiore, quali queste 2 fluttuazioni sono, convergono più lentamente rispetto a quelli di ordine inferiore, quale la velocità media; quindi, laddove questi vanno a convergere, si ha una maggiore sicu-*

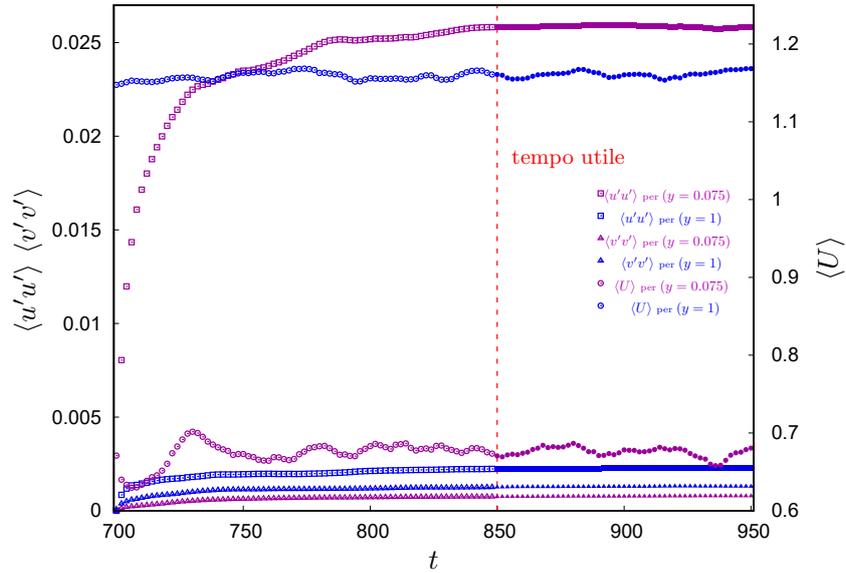


Figura 5.1: Grafico riportante l'andamento di $\langle u'u' \rangle$ e $\langle v'v' \rangle$ su di un asse, e di $\langle U \rangle$ sull'altro, tutte in funzione del tempo t , per la configurazione $H0$. Sul grafico inoltre è evidenziato come, da $t = 850$ inizi la zona di dati utili, in quanto il flusso diventa mediamente stazionario (lo si osserva dalla convergenza delle curve). Inoltre, come riportato nella legenda, i dati sono presi per 2 valori differenti di y , $y = 0,075$ e $y = 1$.

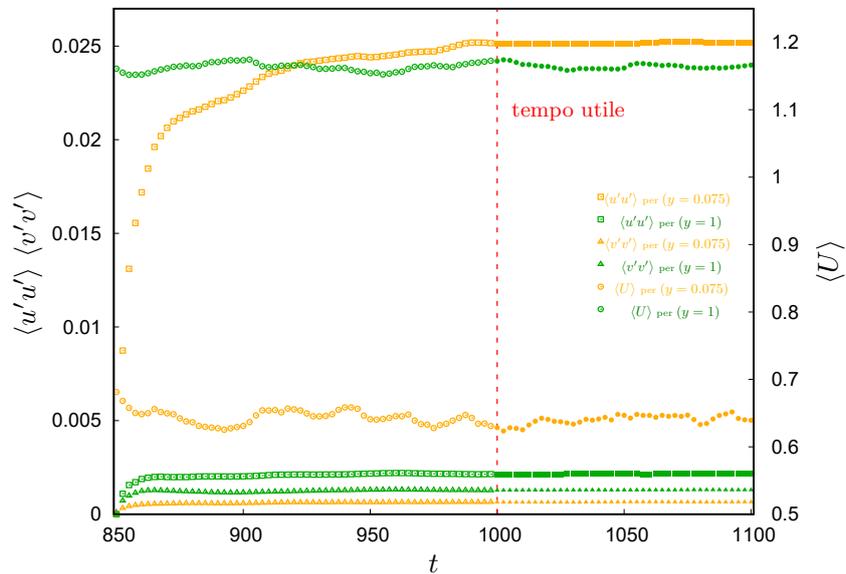


Figura 5.2: Grafico riportante l'andamento di $\langle u'u' \rangle$ e $\langle v'v' \rangle$ su di un asse, e di $\langle U \rangle$ sull'altro, tutte in funzione del tempo t , per la configurazione $H3$. Sul grafico inoltre è evidenziato come, da $t = 1000$ inizi la zona di dati utili, in quanto il flusso diventa mediamente stazionario (lo si osserva dalla convergenza delle curve). Inoltre, come riportato nella legenda, i dati sono presi per 2 valori differenti di y , $y = 0,075$ e $y = 1$.

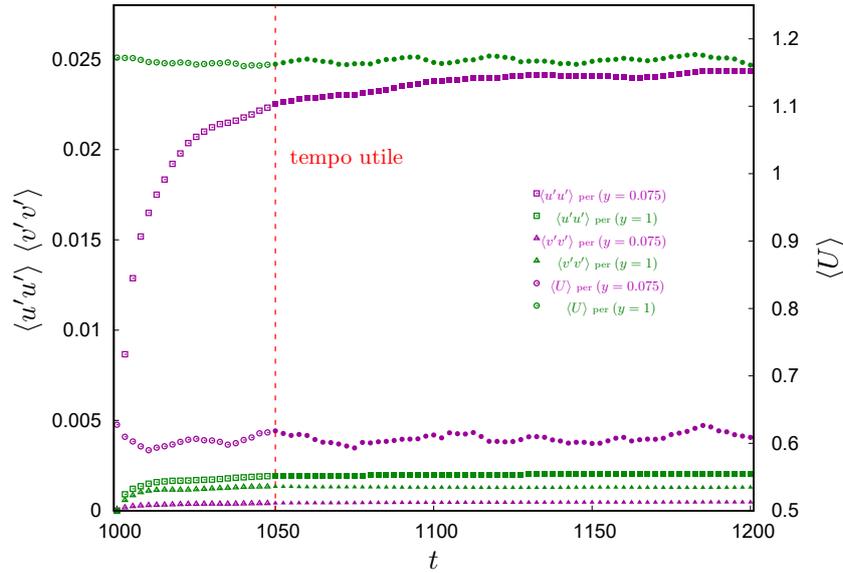


Figura 5.3: Grafico riportante l'andamento di $\langle u'u' \rangle$ e $\langle v'v' \rangle$ su di un asse, e di $\langle U \rangle$ sull'altro, tutte in funzione del tempo t , per la configurazione $H3$. Sul grafico inoltre è evidenziato come, da $t = 1050$ inizi la zona di dati utili, in quanto il flusso diventa mediamente stazionario (lo si osserva dalla convergenza delle curve). Inoltre, come riportato nella legenda, i dati sono presi per 2 valori differenti di y , $y = 0,075$ e $y = 1$.

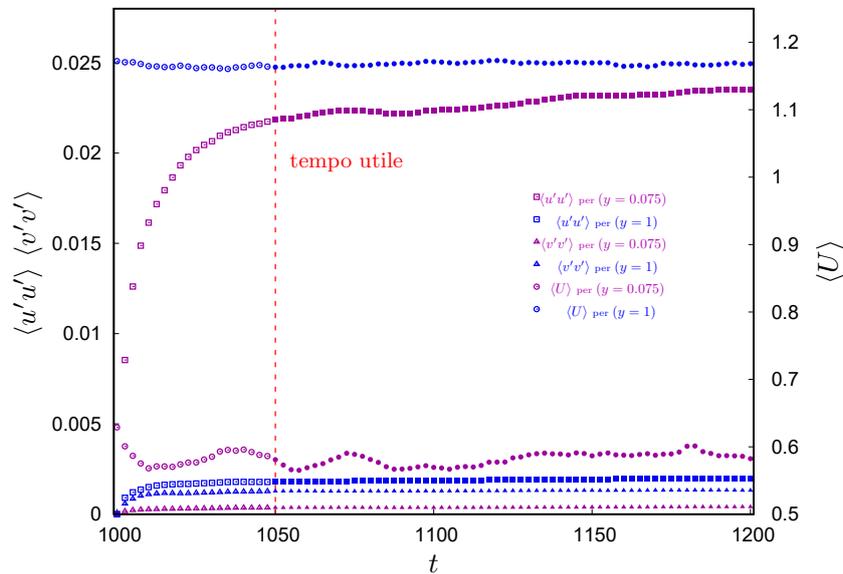


Figura 5.4: Grafico riportante l'andamento di $\langle u'u' \rangle$ e $\langle v'v' \rangle$ su di un asse, e di $\langle U \rangle$ sull'altro, tutte in funzione del tempo t , per la configurazione $H6$. Sul grafico inoltre è evidenziato come, da $t = 1050$ inizi la zona di dati utili, in quanto il flusso diventa mediamente stazionario (lo si osserva dalla convergenza delle curve). Inoltre, come riportato nella legenda, i dati sono presi per 2 valori differenti di y , $y = 0,075$ e $y = 1$.

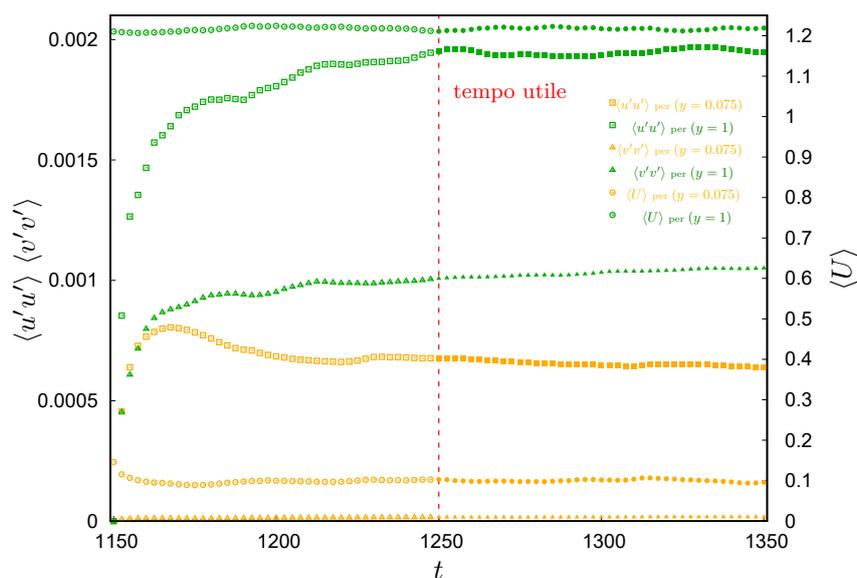


Figura 5.5: Grafico riportante l'andamento di $\langle u'u' \rangle$ e $\langle v'v' \rangle$ su di un asse, e di $\langle U \rangle$ sull'altro, tutte in funzione del tempo t , per la configurazione $H22$. Sul grafico inoltre è evidenziato come, da $t = 1250$ inizi la zona di dati utili, in quanto il flusso diventa mediamente stazionario (lo si osserva dalla convergenza delle curve). Inoltre, come riportato nella legenda, i dati sono presi per 2 valori differenti di y , $y = 0,075$ e $y = 1$.

rezza di essere arrivati nella condizione di mediamente stazionaria. L'andamento della velocità U , sul secondo asse, viene riportato solamente per completezza dell'immagine. Osservando queste figure, inoltre è possibile fare alcune osservazioni sul comportamento delle simulazioni. Come prima cosa si osserva che, le configurazioni successive (ovvero quelle che sono interpolazioni dei casi precedenti, ovvero $H5$, $H6$ e $H22$) raggiungono prima la condizione di mediamente stazionario, rispetto agli altri casi. Ciò si spiega in quanto queste simulazioni, avendo come punto di partenza altri casi, già quasi stabilizzati, devono solamente adattare il campo di moto interpolato alla loro nuova configurazione di riblet; invece, al contrario, il caso $H0$, avendo come punto di partenza il caso transitorio, impiega più tempo, perché in questo caso il flusso deve stabilizzarsi temporalmente e non spazialmente (cioè non si hanno nuove condizioni esterne, quali riblet di altezza diversa). Inoltre si osserva che i 2 casi che hanno lo stesso punto di partenza, $H5$ e $H6$, convergono allo stesso istante temporale, e le loro curve hanno andamenti molto simili; ciò ci dice che, se il flusso di partenza è già sufficientemente stabile, per adattarsi a due diverse nuove configurazioni del campo di moto, si impiega lo stesso tempo. Inoltre osservando i nuovamente i casi $H5$, $H6$ e $H22$, si osserva come, al crescere dell'altezza delle riblet, aumentando la velocità massima nel canale, il flusso tenda a stabilizzarsi ancora prima. Infine si osserva come, le fluttuazioni $\langle v'v' \rangle$ convergano all'incirca nello stesso istante temporale, in tutte le configurazioni riportate, in cui lo fanno le fluttuazioni $\langle u'u' \rangle$. Ciò ci mostra come il comportamento delle fluttuazioni di velocità sia collegato.

Dopo aver osservato il comportamento sia della velocità media $\langle U \rangle$ e sia delle fluttuazioni, $\langle u'u' \rangle$ e $\langle v'v' \rangle$, per 2 coordinate y , $y = 0,075$ e $y = 1$, si va a vedere, se cambia qualcosa nel comportamento, l'andamento temporale di $\langle u'u' \rangle$, per 5 coordinate y . Nello specifico, si vanno a riportare questi andamenti per 2 configurazioni, $H0$ e $H6$.

Nei grafici 5.6 e 5.7 si osservano gli andamenti di $\langle u'u' \rangle$, in funzione del tempo t , per 5

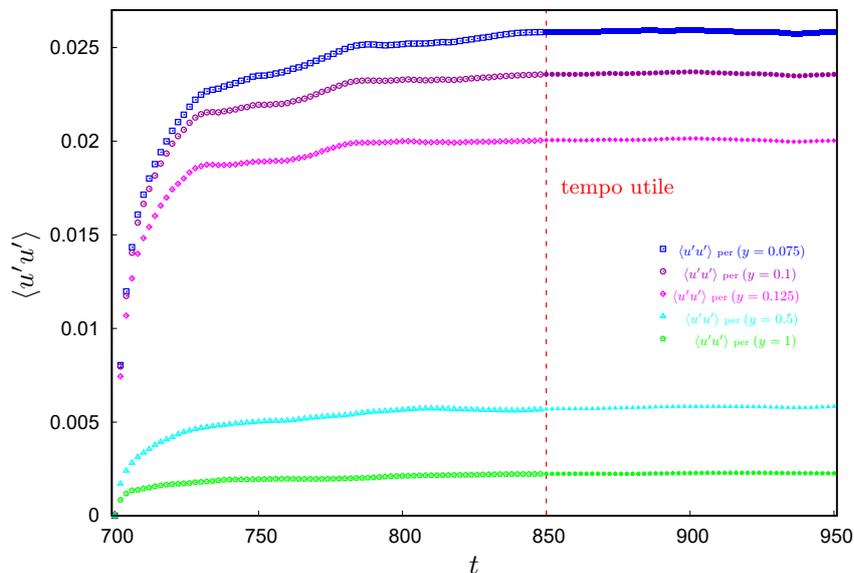


Figura 5.6: Grafico riportante l'andamento di $\langle u'u' \rangle$, in funzione del tempo t , per la configurazione $H0$, per 5 valori differenti di y , $y = 0,075$, $y = 0,1$, $y = 0,125$, $y = 0,5$ e $y = 1$. Sul grafico è sempre evidenziato come, da $t = 850$ inizi la zona di dati utili.

valori differenti di y . Da questi si osserva semplicemente come l'andamento di $\langle u'u' \rangle$ sia sempre lo stesso, per entrambe le configurazioni, e per tutte le coordinate y , cambiando solamente il modulo dell'intensità, che è funzione della y , appunto.

Si vanno a riportare ora, i profili di velocità di U e $u'u'$, delle configurazioni $H0$, $H7$ e $H22$, per le quali si riportano i profili in funzione della distanza da parete y , ma ne vengono riportati diversi profili, per differenti istanti temporali, per osservare la variazione di queste grandezze nel tempo.

Nelle figure 5.8 e 5.9 sono riportati gli andamenti descritti. In queste è possibile osservare, come pur cambiando le configurazioni, il fenomeno di convergenza temporale dei dati, per il flusso che da turbolento instazionario, diventa mediamente stazionario, dipende poco dalla presenza o meno delle riblet, e della loro configurazione; infatti solamente l'ispessimento del sottostato viscoso, con conseguente aumento della velocità massima al centro del canale, velocizza leggermente la convergenza, ma in questi grafici è difficilmente apprezzabile. Le uniche osservazioni fattibili sono in merito al comportamento anomalo delle fluttuazioni $u'u'$ per la configurazione $H7$, dove al di là del valore iniziale che ha un andamento molto diverso dalle altre (ma ciò può essere giustificato dal fatto che si tratta dei dati ottenuti dall'interpolazione del caso transitorio, e quindi è possibile che l'interpolazione abbia lasciato qualche anomalia sui dati) la vera differenza è che in questo caso, forse come conseguenza del punto di partenza, le fluttuazioni, invece di crescere di intensità, come negli altri casi, tendono a decrescere. Oltre a ciò non si nota nulla di rilevante in questi grafici.

Dai grafici fin qui riportati, in particolare 5.1, 5.2, 5.3, 5.4 e 5.5, si sono quindi trovati gli istanti temporali per ogni configurazione analizzata (anche per quelle non riportate) da cui il flusso diventa mediamente stazionario, e quindi da dove i dati sono validi e possono essere utilizzati. Si prendono quindi questi come intervalli per fare la media temporale dei dati di post processing estratti (come spiegato in 4.3.1). Una volta

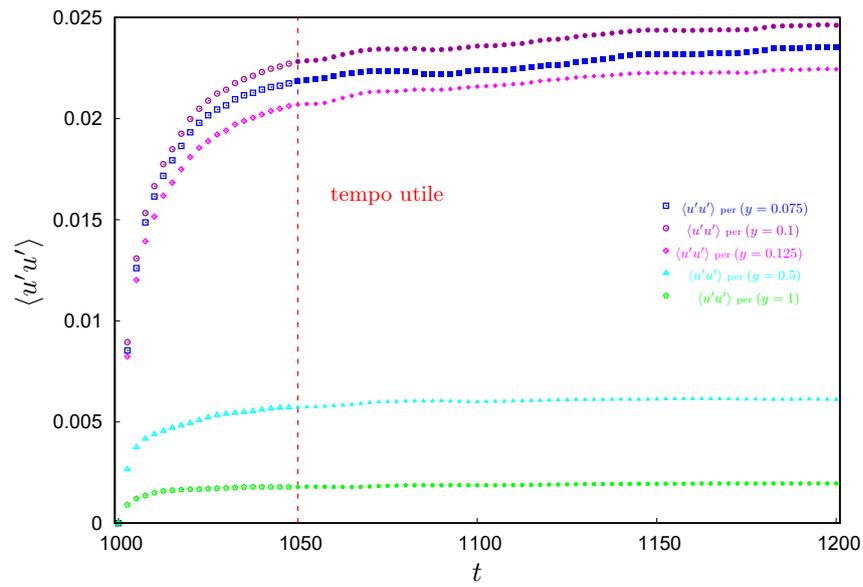
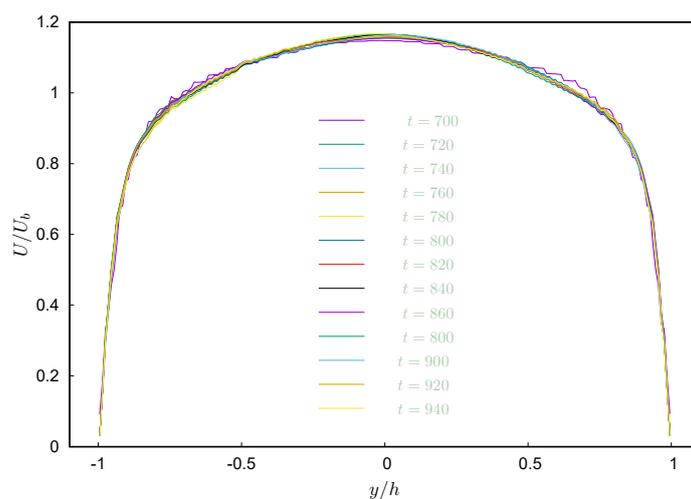
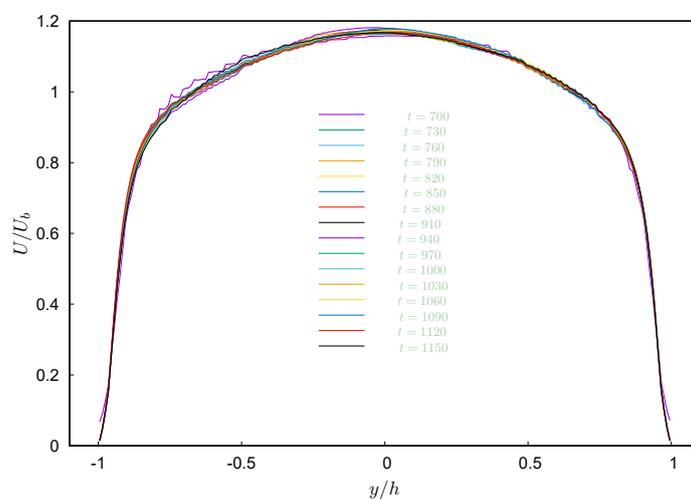
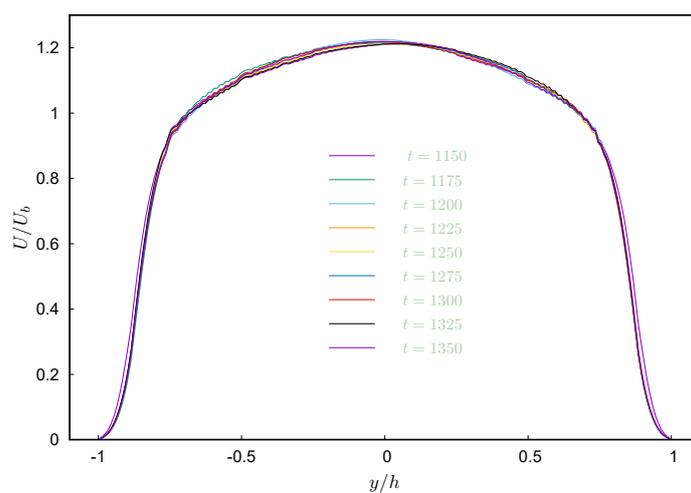


Figura 5.7: Grafico riportante l'andamento di $\langle u'u' \rangle$, in funzione del tempo t , per la configurazione $H6$, per 5 valori differenti di y , $y = 0,075$, $y = 0,1$, $y = 0,125$, $y = 0,5$ e $y = 1$. Sul grafico è sempre evidenziato come, da $t = 1050$ inizi la zona di dati utili.

fatto ciò si hanno i dati per poter realizzare grafici significativi in merito allo studio effettuato.

(a) Configurazione H0, profilo di U (b) Configurazione H7, profilo di U (c) Configurazione H22, profilo di U Figura 5.8: Grafici di U , per 3 configurazioni, H0, H6 e H22, riportanti diversi profili ad istanti temporali differenti.

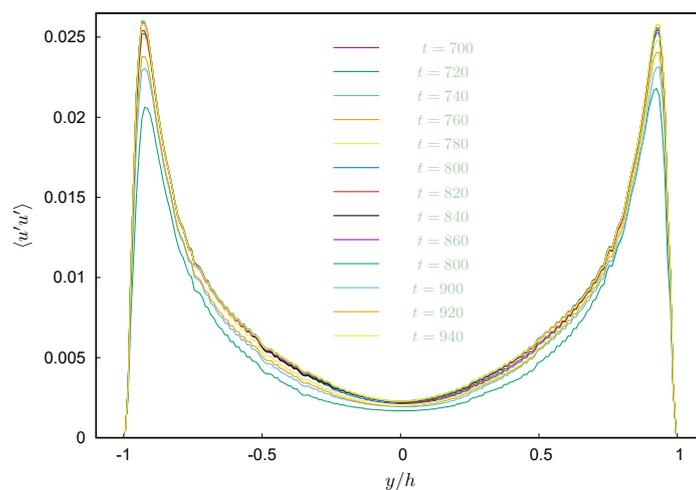
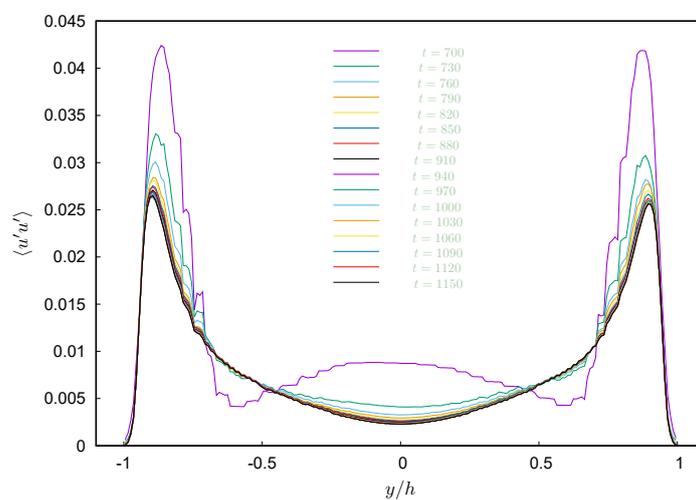
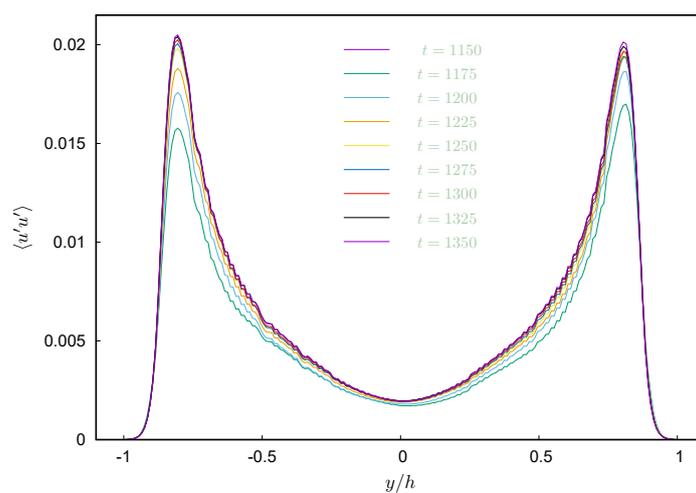
(a) Configurazione H0, profilo di $u'u'$ (b) Configurazione H7, profilo di uu (c) Configurazione H22, profilo di uu

Figura 5.9: Grafici di $u'u'$, per 3 configurazioni, H0, H6 e H22, riportanti diversi profili ad istanti temporali differenti.

5.3 Validazione dei risultati: canale liscio

Dopo aver riportato gli andamenti temporali con cui si sono trovati gli istanti temporali da cui iniziare a considerare validi i dati ottenuti, in quanto ci si trova in condizione di flusso mediamente stazionario, è possibile utilizzare le grandezze ottenute, mediate nello spazio (per piani xz) e nel tempo (a partire dal primo istante utile fino all'ultimo calcolato) per graficare i dati di nostro interesse. Però, prima di fare ciò, ci si è assicurati che le simulazioni svolte fossero effettivamente valide, ovvero rispecchiassero i dati ottenuti per altri studi significativi. Poiché andare a confrontare i dati con le riblet presenti diventa estremamente complicato a causa dell'elevatissimo numero di configurazioni e tipi di riblet studiate in letteratura, si è pensati di andare a validare una sola soluzione, che appurasse la bontà e la qualità dello studio effettuato. Si è allora scelto di andare a confrontare i dati del canale piano turbolento simulato, senza riblet (ovvero la configurazione $H0$), con dei dati molto utilizzati in letteratura in merito al canale piano turbolento, ovvero le DNS di *Kim, Moser & Mansour* [8]. Nel loro articolo *Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$* , infatti, vengono svolte DNS per un canale piano a diversi numeri di Reynolds, Re_τ . In particolare, tra quelli analizzati ce ne sono 2, che racchiudono quello da noi scelto (200), che sono 180 e 395. In questo modo, analizzando soprattutto la curva a $Re_\tau = 180$, ed il suo andamento, rispetto a quello del nostro caso, è possibile capire se i dati da noi ottenuti sono validi. Si vanno quindi a riportare gli andamenti della velocità, in unità di parete e semiscala

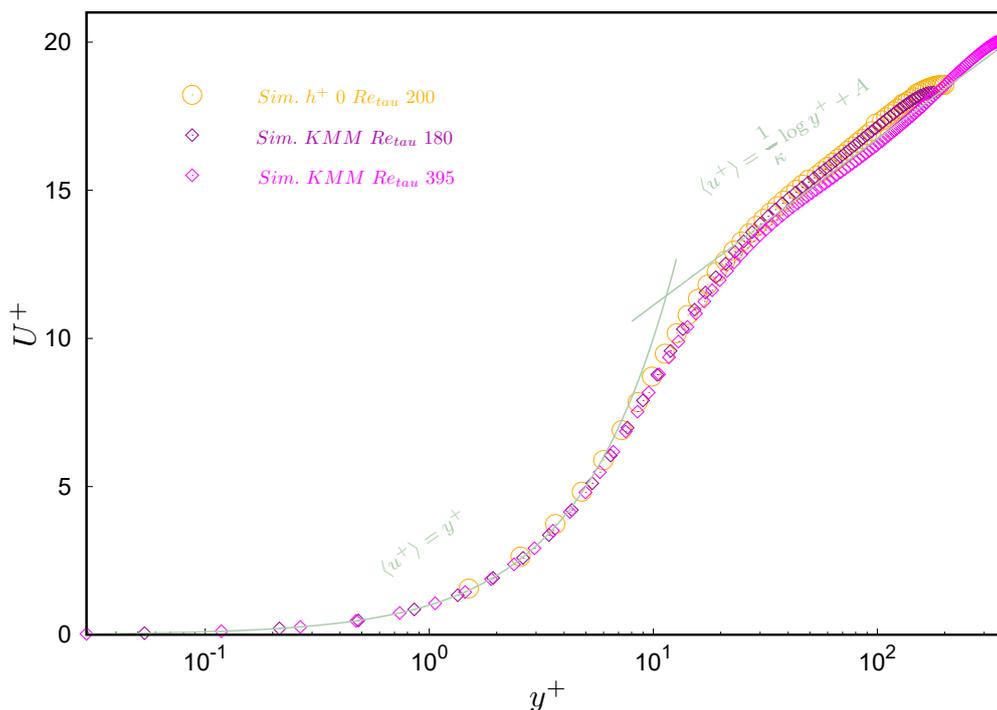


Figura 5.10: Grafico di $y^+ - U^+$ in scala semilogaritmica, riportante il confronto tra i dati della configurazione $H0$ (canale piano) ed i dati, per il canale piano, di *Kim, Moser & Mansour* [8], per $Re_\tau = 180$ e $Re_\tau = 395$; è inoltre riportata anche la legge empirica per un flusso turbolento.

logaritmica, per confrontare l'andamento delle 3 curve tra di loro, e rispetto la legge

empirica dello strato limite turbolento. Fatto ciò, dall'immagine 5.10 si osserva come i nostri dati coincidano perfettamente con quelli di *Kim, Moser & Mansour*, in quanto la leggera differenza che si ha con la curva a $Re_\tau = 180$ è data dal fatto che i nostri dati sono a $Re_\tau = 200$, e quindi una leggera variazione, funzione proprio del Re_τ è prevista, come ci dice anche la curva a $Re_\tau = 395$. Inoltre le curve corrispondono molto bene anche alla legge empirica riportata nel grafico. Queste 2 osservazioni ci permettono di affermare che le simulazioni DNS da noi svolte sono sufficientemente accurate, precise e valide, e che quindi si possono usare i dati ottenute da queste per trovare i valori di nostro interesse, in merito al comportamento del flusso turbolento, in un canale con riblet.

5.4 Visualizzazioni del Flusso

Sono ora riportate alcune visualizzazioni del flusso; queste sono state effettuate con tagli su determinati piani. Inoltre vanno a riportare solamente alcune simulazioni, per capire come la presenza delle riblet sia andata ad influenzare il flusso nel canale, ma senza essere eccessivamente ridondanti e ripetitivi. Mediante queste, infatti, è possibile andare ad osservare, in forma assoluta, le strutture turbolente che si formano all'interno di un canale piano, in cui scorre un flusso turbolento. In particolare le visualizzazioni, ovvero i tagli, sono state effettuate, su diversi piani, i quali sono sempre normali ad un asse. Nello specifico si sono effettuati tagli con piani normali a tutti i 3 assi. Inoltre, nel caso dei piani normali all'asse y , si è andati a fare tagli a diverse posizioni lungo l'asse y , in quanto si ha, che cambiando la distanza dalla parete, cambiano le strutture che si osservano nel flusso. Specificando nel dettaglio quali sono stati i tagli effettuati, abbiamo:

- 2 tagli sul piano xy , normale all'asse z , effettuati per la configurazione $H22$, in cui si è preso un taglio per $z = 2,25$ ed un altro per $z = 1,125$;
- 10 tagli sul piano zy , normale all'asse x , effettuati per $x = 4,5$, ovvero a metà del canale simulato, ed effettuati per le configurazioni: $H0, H3, H5, H7, H9, H12, H15, H18, H20$ e $H22$;
- 3 tagli sul piano xz , normale all'asse y , effettuati per $y = 1$, ovvero a centro canale, ed effettuati per le configurazioni: $H0, H7$ e $H15$;
- 5 tagli sul piano xz , normale all'asse y , effettuati per $y = 0,1$, ovvero dove si ha la massima turbolenza, ed effettuati per le configurazioni: $H0, H6, H7, H10$ e $H15$.

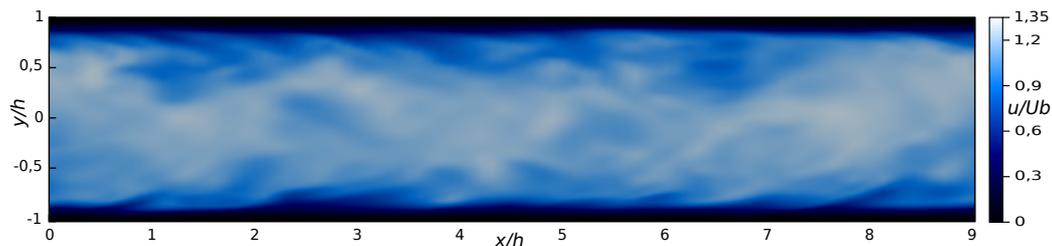
Vengono ora riportati, secondo l'ordine indicato, i tagli che sono stati effettuati. Per ognuno di questi tagli, si andrà prima ad indicare cosa si va ad osservare in queste sezioni, poi se ne riportano tutte le immagini, per poterle confrontare, ed infine si vanno a riportare commenti ed osservazioni in merito. Tutte le immagini delle visualizzazioni sono riportate con affiancate la legenda dei colori, che ci permette di comprendere l'intensità della velocità. Abbiamo infatti che il campo di moto che si osserva in tutte le visualizzazioni che si riportano, è il modulo della velocità U . Da queste visualizzazioni, inoltre, è possibile osservare, come sui dati ci sia un leggero errore numerico, che cresce al crescere dell'altezza delle riblet, causato dall'effetto di deformazione della griglia, che ci ha quando il blocco di parete, si deforma per la presenza sempre maggiore delle riblet (si entra maggiormente nel dettaglio nella sezione 6). In queste visualizzazioni si può quindi iniziare ad osservare la presenza di questo errore di distorsione della griglia, mentre poi, nei grafici riportati di seguito, verrà maggiormente analizzato. Infine si deve anche tener presente, l'errore dato dalla discretizzazione numerica che si ha per la mesh.

Piani XY

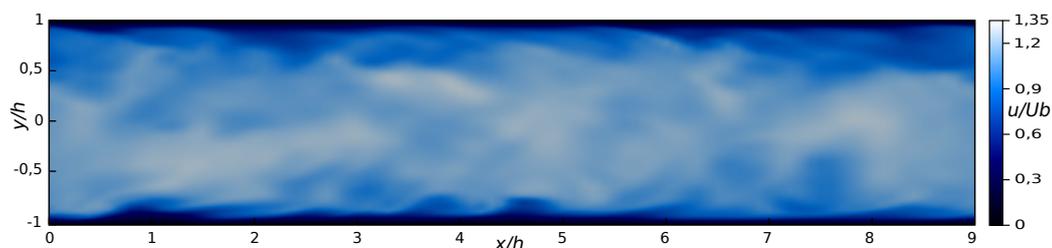
Le sezioni che vengono effettuate sul piano xy , normali quindi alla direzione z , ci permettono di osservare come cambia il comportamento del flusso in direzione trasversale. Tuttavia, nel nostro caso, avendo imposto in direzione trasversale le condizioni di periodicità, il flusso non presenta differenze notevoli muovendoci lungo z . In queste

2 sezioni, si è, invece, andati a prendere 2 posizioni tali che ci si trova, in un caso in corrispondenza della punta di una riblet, e nell'altro in corrispondenza della valle di una riblet. Così facendo si cerca di cogliere il maggior numero di differenze tra le 2 sezioni. Inoltre, dato che la simulazione anche in direzione longitudinale è periodica, con questo tipo di simulazione non è possibile cogliere differenze tra una configurazione di riblet ed un'altra; per questo motivo, se ne riporta solamente una, quella *H22*, ovvero quella con la massima altezza delle riblet, per cercare, come detto prima, di cogliere quante più differenze possibili, tra la sezione alla punta della riblet e quella alla valle.

Nelle figure 5.11 riportate, si possono osservare le differenze di comportamento del



(a) Sezione a $z = 2,25$, valle della riblet



(b) Sezione a $z = 1,125$, punta della riblet

Figura 5.11: Sezioni sul piano xy , normali a z , per la configurazione *H22*

flusso tra la punta di una riblet e la valle. Bisogna però fare subito una precisazione importante. Nell'immagine 5.11b, quella che riporta la punta della riblet, sono indicati come estremi dell'asse y -1 e 1 , che sarebbero gli estremi del canale; tuttavia questi sono riportati automaticamente quando si va a generare l'immagine, ma si deve tenere conto, che *ParaFOAM*, visualizzando solo i dati simulati da *OpenFOAM*, visualizza solamente il campo di moto del flusso, e non le pareti esterne; ciò significa che dove sono presenti le riblet, il programma va a riportare uno spazio vuoto, non visibile nell'immagine, come si vedrà poi nei piani zy , vedi immagini 5.12; quindi i veri estremi in direzione y della figura 5.11b sono rispettivamente $-0,8875$ e $0,8875$. Questa precisazione deve essere fatta, sia perché non è stato possibile correggere questo problema, nella realizzazione delle immagini, ed anche perché questa è necessaria per poter capire la differenza fondamentale che si ha tra le due immagini, che viene ora riportata.

Tenendo quindi conto che le 2 immagini non iniziano per le stesse coordinate y , si va ad osservare lo strato limite che si ha per entrambe le figure; si osserva come nella figura 5.11a, quella a valle, la zona a bassa velocità, il sottostrato viscoso, sia molto più spesso rispetto alla figura 5.11b, a valle; questo perché, quando ci si trova a valle della riblet, tutta la riblet è immersa nel sottostrato viscoso, dove la velocità risulta esser molto bassa; al contrario, alla punta della riblet, si è ormai fuori da questo sottostrato, e quindi la velocità cresce rapidamente. Ciò ci indica come, a causa della presenza delle riblet, in questo caso molto importante, lo strato limite turbolento, vada a *diluirsi* nello

spazio, riducendo di fatto il forte gradiente di velocità che lo caratterizza, responsabile dell'alta resistenza d'attrito, che quindi cala (come si vedrà quando si riporterà l'andamento della derivata della velocità, $\frac{dU}{dy}$, e l'andamento degli sforzi d'attrito viscosi, strettamente collegati a questa). Oltre a questa grande differenza, si osserva invece, come le strutture turbolente che è possibile osservare in entrambe le figure, siano le medesime; ciò indica come il flusso, nella zona in cui le riblet non lo influenzano più, terminato lo strato limite, assuma la medesima configurazione al centro del canale, sia in corrispondenza della valle di una riblet, che in corrispondenza di una punta.

Piani ZY

Le sezioni che vengono effettuate sul piano zy , normali quindi alla direzione x , ci permettono di osservare come cambia la struttura del flusso, al cambiare della dimensione delle riblet, andando da una parete solida all'altra. In questa vista, infatti, si osserva la presenza di queste, nella zona che il programma non riesce a visualizzare, causa mancanza di campo di moto, poiché *OpenFOAM*, e quindi anche *ParaFOAM*, non riportano le pareti esterne nel campo di moto. Attraverso queste visualizzazioni, osservando l'aumento dell'altezza delle riblet per le configurazioni riportate, è possibile fare osservazioni sia sullo strato limite che si ha per le diverse configurazioni, e sia eventuali cambiamenti nella parte centrale del flusso. Di questa visualizzazione vengono riportate le configurazioni: $H0$, $H3$, $H5$, $H7$, $H9$, $H12$, $H15$, $H18$, $H20$ e $H22$.

Nelle figure 5.12 riportate, si possono osservare alcuni cambiamenti nel campo di moto, dovuti alla presenza delle riblet. Come prima cosa, si nota, come già osservato in precedenza, che all'aumentare dell'altezza delle riblet, il sottostrato viscoso cresce, ma *stirandosi*, ovvero presentando gradiente di velocità meno accentuato; questo aspetto è positivo, per quanto riguarda la resistenza, in quanto si ha che così, la componente viscosa della velocità, diminuisce. Inoltre si osserva come il flusso, vada ad avere un incremento di velocità massima al centro del canale, all'aumentare dell'altezza delle riblet; questo è collegato all'aumento di dimensione della zona di strato limite, in quanto si ha, che crescendo questo, ma avendo una velocità media imposta dal gradiente di pressione, per avere la risultante media di velocità costante, questa aumenta al centro del canale. Infine si osserva come, all'aumentare dell'altezza delle riblet, le fluttuazioni di velocità, e quindi anche i fenomeni di rimescolamento, vadano a diminuire di intensità. In queste immagini, si osserva inoltre, come già accennato in precedenza, come la parte di dominio occupata dalle riblet, non venga visualizzata dal programma; tuttavia, per contrasto, ovvero osservando la zona bianca vicino alle pareti del canale, si possono comunque osservare le riblet, e valutarne la forma all'aumentare della loro altezza.

Piani XZ

Le sezioni che vengono effettuate sul piano xz , normali quindi alla direzione y , ci permettono, infine, di osservare come cambia la struttura del flusso, al cambiare della dimensione delle riblet, questa volta osservandolo lungo il suo sviluppo longitudinale e trasversale. Inoltre, queste visualizzazioni sono effettuate per 2 differenti coordinate y , a $y = 1$, ovvero al centro del canale, ed a $y = 0, 1$, ovvero nella zona di massima turbolenza. Per la prima posizione, $y = 1$ è possibile osservare il comportamento della velocità del flusso, come questa cambia, al centro del canale, ovvero alla sua massima intensità, al variare dell'altezza delle riblet. Nella seconda posizione, $y = 0, 1$, invece, è possibile concentrarsi sulle strutture vorticosi di rimescolamento che si formano nel

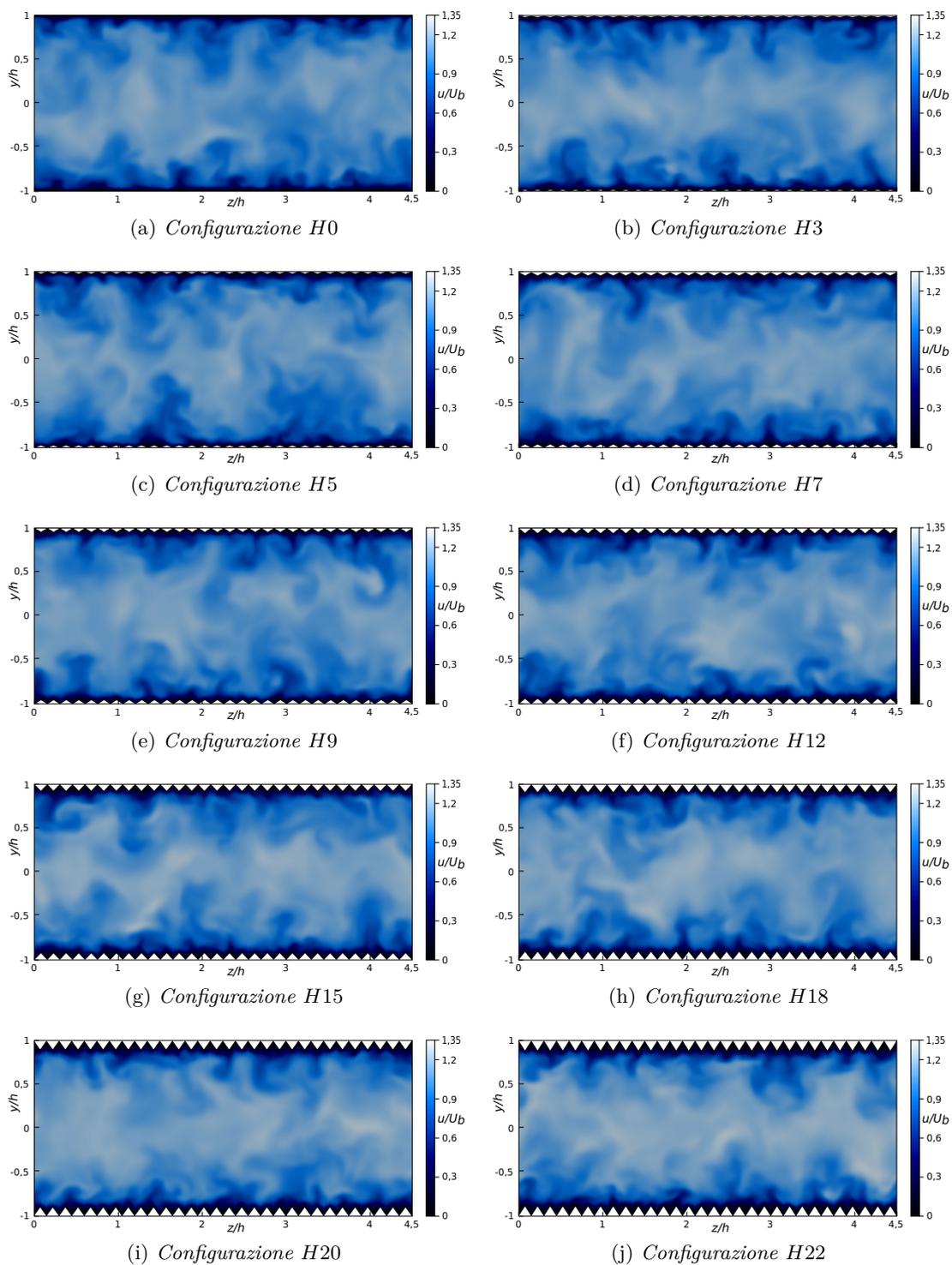
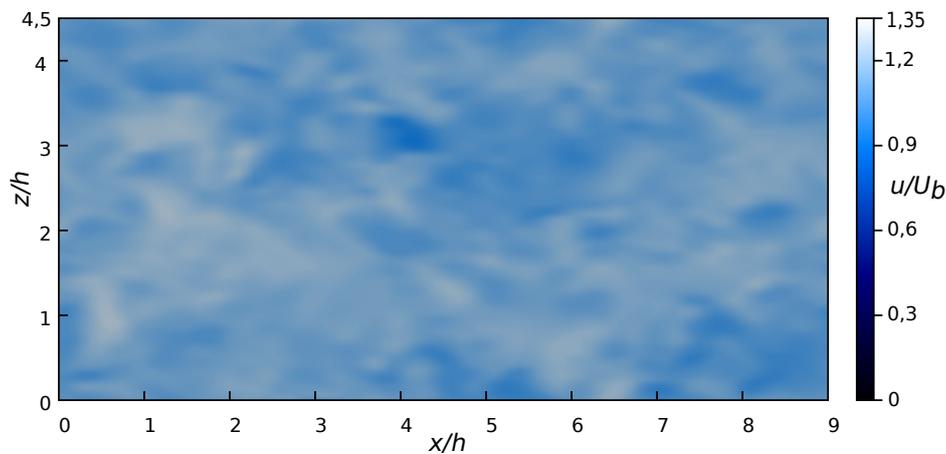


Figura 5.12: Sezioni sul piano zy , normali a x , per $x = 4, 5$, ovvero a metà del canale simulato. È possibile osservare le riblet come la zona bianca che il programma non va a riportare.

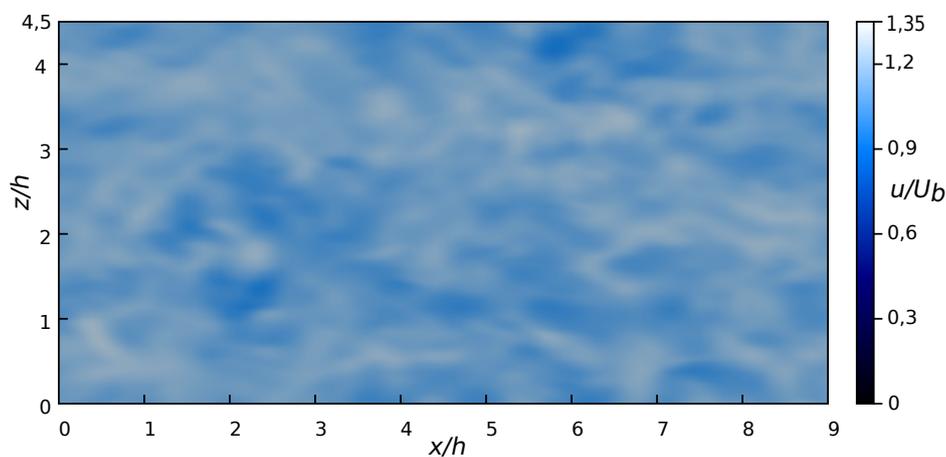
flusso, e vedere, come queste cambiano, in presenza di riblet di altezze differenti. Delle visualizzazioni per $y = 1$, vengono riportate le configurazioni: $H0$, $H7$ e $H15$. Invece, per $y = 0, 1$, vengono riportate le configurazioni: $H0$, $H6$, $H7$, $H10$ e $H15$.

Nelle figure 5.13 riportate, trovandoci al centro del canale, si osservano poche differenze sostanziali tra i vari casi. Infatti, pur sapendo che la velocità massima, aumenta leggermente al crescere dell'altezza delle riblet, ciò risulta difficilmente osservabile in queste visualizzazioni, dati i rimescolamenti del flusso turbolento, che non rendono chiaro ed assoluto il valore della velocità. Tuttavia si può osservare qui, ma è molto più chiaro nelle immagini 5.14 e 5.15, come al crescere dell'altezza delle riblet, le macrostrutture vorticose di rimescolamento vadano a rompersi in strutture più piccole; ciò è osservabile dalla dimensione delle zone che alternano colori (e quindi valori di velocità) differenti; dove si hanno zone più grandi, le strutture di rimescolamento sono più grandi; dove invece se ne alternano molte, ma di dimensioni ridotte, è perché anche le strutture sono di dimensioni minori.

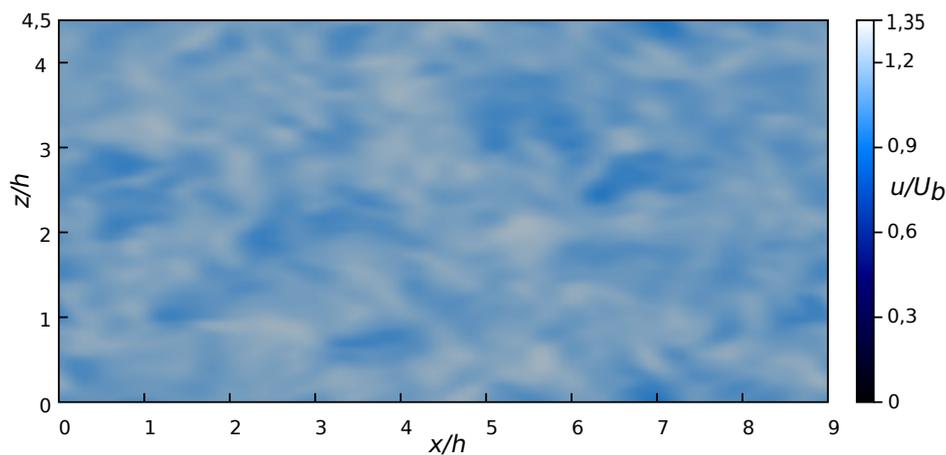
Nelle immagini 5.14 e 5.15 riportate, riportanti la zona di massima turbolenza, e quindi quella dove le fluttuazioni di velocità hanno il valore massimo, è possibile osservare meglio, rispetto a quelle 5.13, come la presenza delle riblet va a modificare la struttura delle strutture turbolente. Osservando le immagini riportate, si osserva infatti, che all'aumentare dell'altezza delle riblet, le strutture turbolente, le *strisce vorticose*, diventano più sottili, in quanto quelle grandi che si osservano per la configurazione $H0$ (ovvero nel caso senza riblet, vedi 5.14a), man mano che l'altezza delle riblet aumenta, si rompono in strutture sempre più piccole, come è possibile osservare bene nelle figure 5.15. In particolare nella figura 5.15b, che è il caso di riblet $H15$, si osserva come, a differenza di tutti gli altri, queste strisce diventano molto più piccole, ma numerose, ed anche come la velocità media della sezione, sia più bassa, ad indicare che ci si trova ancora in prossimità del sottostrato viscoso, il quale, come si è visto nelle figure 5.12, aumenta di spessore al crescere delle riblet. Si nota infatti, osservando tutte le immagini, che aumentando l'altezza delle riblet, il flusso diventa mediamente più lento, anche se, come si era visto in precedenza, la velocità massima nel canale cresce all'aumentare dell'altezza delle riblet; tuttavia ciò si verifica, perché le immagini riportate sono in prossimità del sottostrato viscoso, e quindi, al crescere delle riblet, questo cresce in spessore, rendendo il flusso nelle sue prossimità, mediamente più lento (mentre invece la velocità massima più alta la si ha al centro del canale). Si può osservare infine, anche il fenomeno del *vortex stretching*, in quanto si osserva come le strutture vorticose, al crescere delle riblet, diventano più sottili, ma tendono anche ad allungarsi.



(a) Configurazione H0

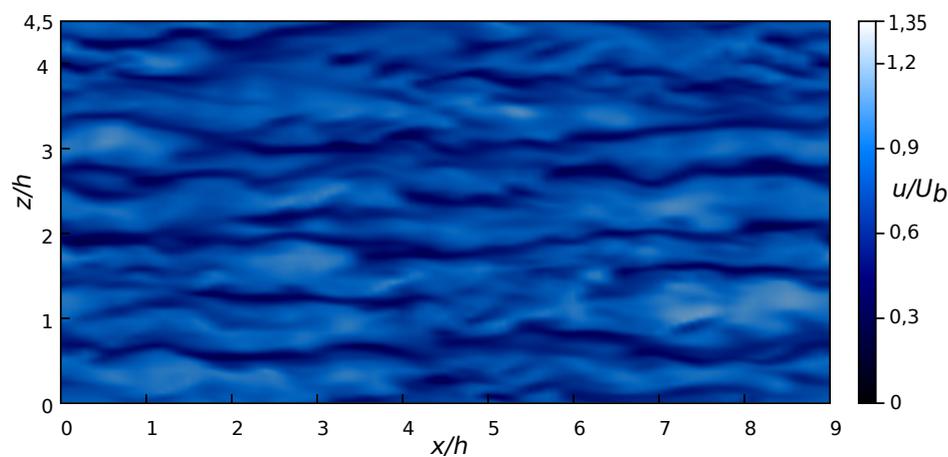


(b) Configurazione H7

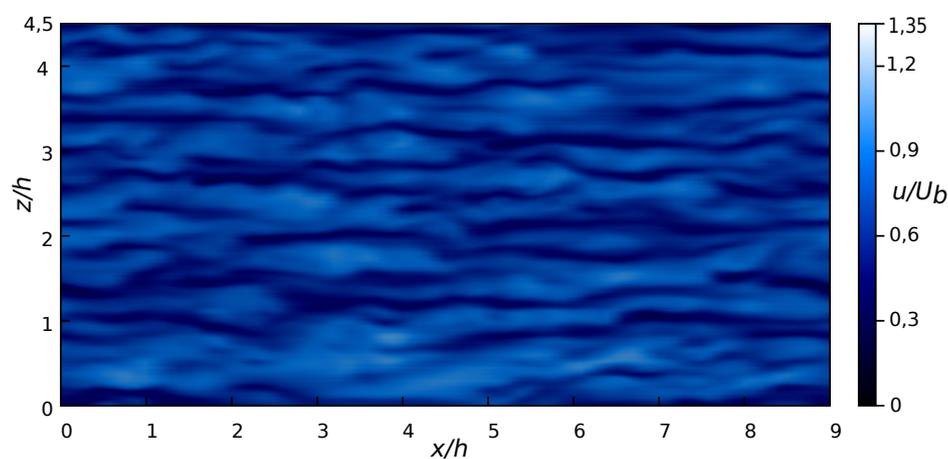


(c) Configurazione H15

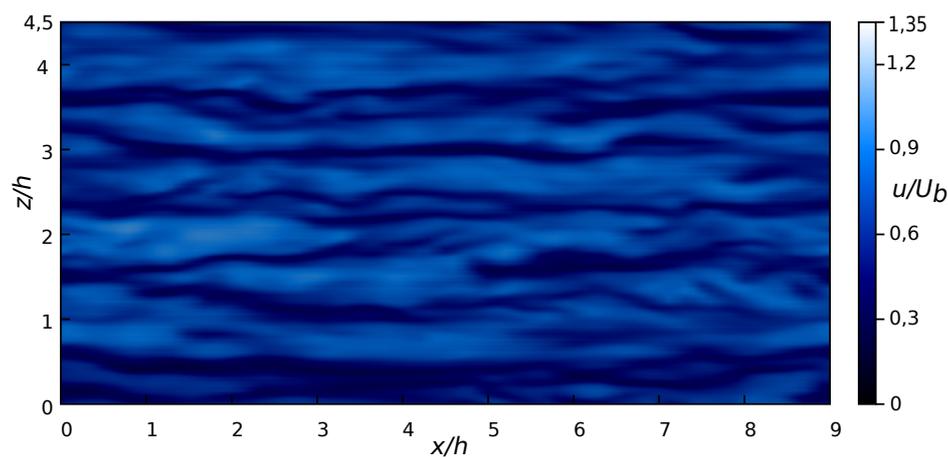
Figura 5.13: Sezioni sul piano xz , normali a y , per $y = 1$, ovvero a centro canale, dove si ha la massima velocità U .



(a) Configurazione H0

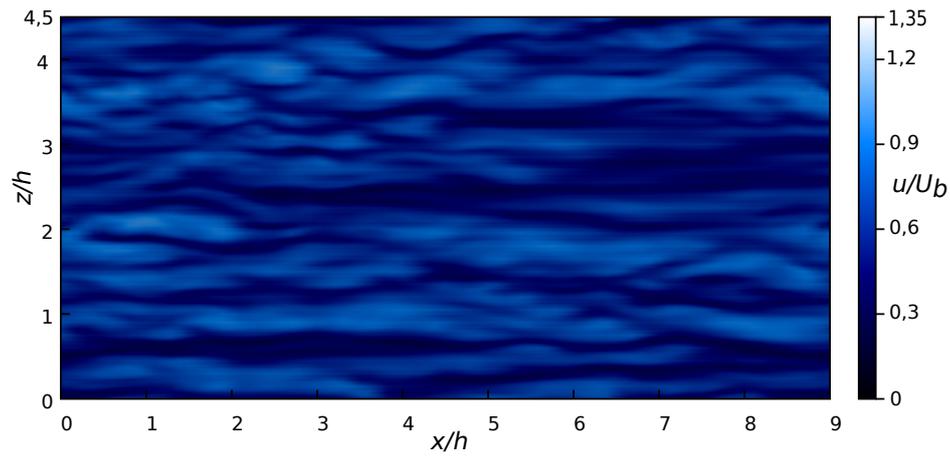


(b) Configurazione H6

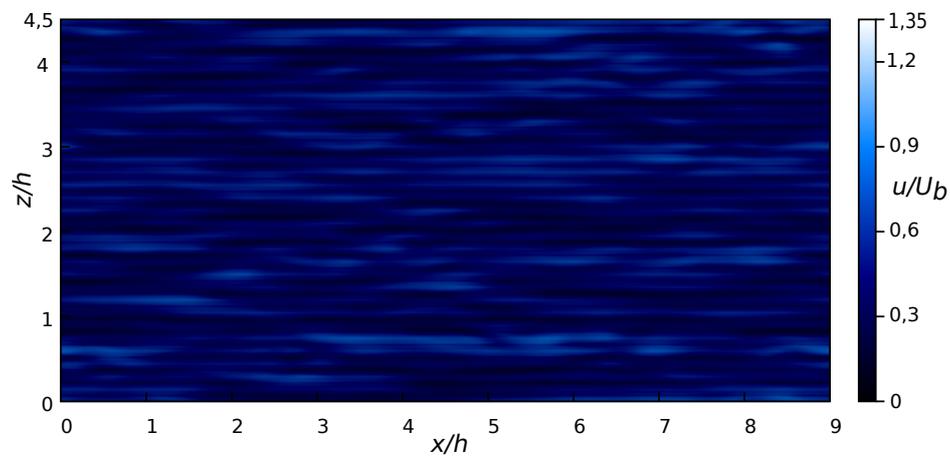


(c) Configurazione H7

Figura 5.14: Alcune sezioni sul piano xz , normali a y , per $y = 0, 1$, ovvero nella zona dove si ha le massime fluttuazioni di velocità, dovute alla turbolenza.



(a) Configurazione H10



(b) Configurazione H15

Figura 5.15: Altre sezioni sul piano xz , normali a y , per $y = 0, 1$, ovvero nella zona dove si ha le massime fluttuazioni di velocità, dovute alla turbolenza.

5.5 Analisi statistica del flusso

Dopo aver visto, prima, la scelta degli intervalli temporali, mediante i grafici degli andamenti dei momenti della velocità, e poi, le visualizzazioni, realizzate con *Para-FOAM*, che riportano sezioni del campo di moto delle configurazioni analizzate, andiamo ora a riportare, altri grafici, realizzati attraverso *gnuplot* (e personalizzati graficamente con \LaTeX), che riportano gli andamenti di alcune grandezze caratteristiche, delle simulazioni effettuate, ovvero quelle ottenute dall'analisi statistica del flusso, mediante il post processing, prima, e la media dei dati ottenuti, poi. In particolare, andremo a riportare, dei grafici relativi a:

1. Gli andamenti dei profili di velocità ottenuti per le simulazioni svolte, sia in forma assoluta, che in unità di parete, per osservarne le differenze tra le diverse configurazioni, ed anche l'andamento in funzione dell'altezza delle riblet, dell'incremento di velocità che si ha al centro del canale;
2. L'andamento, lungo l'altezza del canale, confrontato per le diverse configurazioni analizzate, delle grandezze che caratterizzano un flusso turbolento, quali le fluttuazioni di velocità, e le grandezze da esse derivate;
3. L'andamento delle diverse componenti della resistenza in un flusso turbolento, come queste cambiano per le diverse configurazioni di riblet, e come ciò incide sulla resistenza totale del flusso.

In questo modo, si avrà una visione globale e comprensiva di tutto ciò che si è calcolato, in merito alle configurazioni analizzate, e ciò ci permetterà di trarre le opportune conclusioni in merito al lavoro svolto.

5.5.1 Velocità media

Dopo aver validato i nostri dati, andiamo a riportare i profili di velocità che si ottengono per le configurazioni analizzate, sia quelli in scala semi logaritmica di $y^+ - U^+$, come quello utilizzato per il confronto con *Kim, Moser & Mansour*, ma ora per tutti i casi da noi simulati, e sia quello adimensionale, ma che riporta le grandezze in termini di valori di flusso, $\frac{y}{h} - \frac{U}{U_b}$. In particolare, per quanto riguarda i profili $y^+ - U^+$, per poterli realizzare, è necessario calcolare la u_τ , per poter ottenere la u^+ , che è data da $u^+ = \frac{u}{u_\tau}$. Come si è riportato in precedenza, vedi 4.3.2, per calcolare la u_τ , è necessario fare l'interpolazione della somma delle componenti della resistenza. Nel nostro caso, per questa curva ($y^+ - U^+$), se ne sono realizzati 2 tipi, una in cui la u_τ utilizzata è stata calcolata solamente per la configurazione *H0*, ed utilizzata per tutte le simulazioni, ed un'altra, in cui, invece, per ogni configurazione è stata calcolata ed utilizzata la propria u_τ . In queste curve è possibile osservare come le riblet vadano ad influire sulla velocità nel flusso. Nella figura 5.16 si possono osservare tutti i profili di velocità, per tutte le configurazioni analizzate. La prima cosa che si osserva è come, i profili, all'aumentare dell'altezza delle riblet, vadano a stringersi ai lati, ovvero in corrispondenza delle pareti, ed a crescere al centro, ovvero in corrispondenza del centro canale. Ciò significa, che la presenza delle riblet, fa sì che il flusso, tenda ad essere più lento in prossimità delle pareti, dove sono presenti le riblet, le quali fanno aumentare lo spessore del sottostrato viscoso, il quale rende il flusso più lento nelle sue prossimità; inoltre, proprio questo rallentamento in prossimità delle pareti, fa sì che, il flusso abbia un comportamento

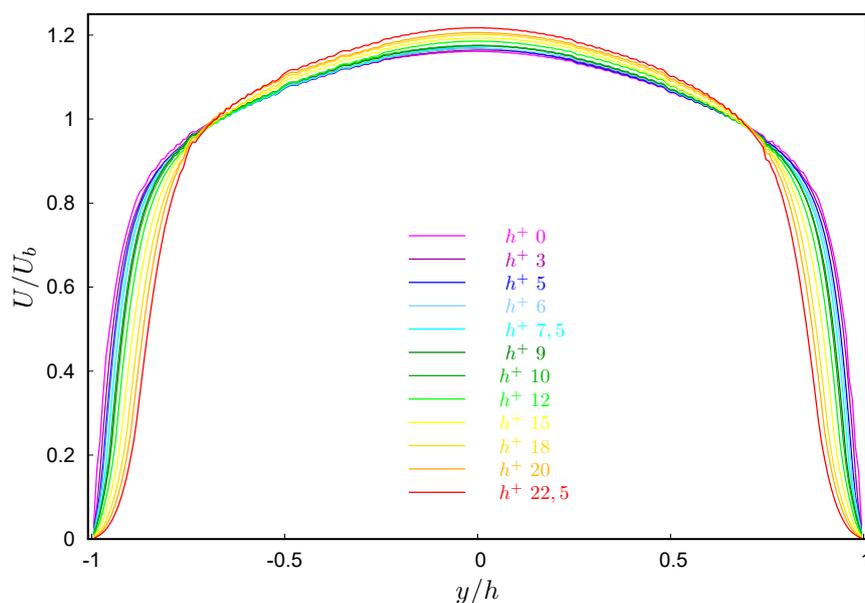


Figura 5.16: Grafico di $\frac{y}{h} - \frac{U}{U_b}$, riportante gli andamenti di tutte le configurazioni analizzate.

simil convergente, ovvero che in presenza di una zona di scorrimento libero ristretta, vada ad aumentare la propria velocità, in questo caso al centro del canale. Questo comportamento è inoltre giustificato dalla presenza del gradiente di pressione imposto, il quale fa sì che si abbia una velocità media unitaria; ciò significa, che se la velocità a parete diminuisce per l'ispessirsi del sottostrato viscoso, necessariamente in un'altra zona, in questo caso al centro del canale, questa deve aumentare, per far sì di rispettare il valore medio imposto dal gradiente di pressione.

Per quanto riguarda i grafici 5.17 e 5.18, alcune delle osservazioni che possono essere fatte, riguardano entrambi i grafici. Infatti si osserva per entrambe le figure come solo la curva corrispondente alla configurazione senza riblet ($H0$) vada ad allinearsi bene con la curva della legge empirica, come si è già visto nel confronto con le DNS di *Kim, Moser & Mansour*, vedi 5.3. Ma ciò è in linea con quello che ci si aspetta, in quanto la legge empirica vale nel caso di un canale piano, mentre la presenza delle riblet, va a modificare il comportamento del flusso, influenzando la velocità, come si è visto, sia molto, vicino alla parete, nel sottostrato viscoso, quindi nella zona lineare della legge empirica, ed anche un po' verso il centro del canale, ovvero nella zona logaritmica. Quindi il discostamento sempre più ampio al crescere dell'altezza delle riblet che si osserva per entrambe le figure è in linea con quanto ci si aspetta. Entrano maggiormente nel dettaglio di questo fenomeno, ed andando ora a considerare separatamente le due figure, si va a spiegare come mai questo accade.

Considerando prima il caso con un'unica u_τ , quella calcolata per la configurazione $H0$ (5.17), qui si osserva come si ha un forte discostamento fra le curve nella zona lineare, dove le curve con riblet hanno una velocità minore rispetto al caso senza; ciò è in linea con quanto riportato fin ora, ovvero che la presenza delle riblet, vicino la parete, va a dilatare il sottostrato viscoso, creando una zona più ampia di velocità bassa, con effetti positivi sulla resistenza. Dopo di che si osserva, come le curve vadano a convergere in un punto, ovvero dove, una volta superato il sottostrato viscoso, la velocità ritorna ad

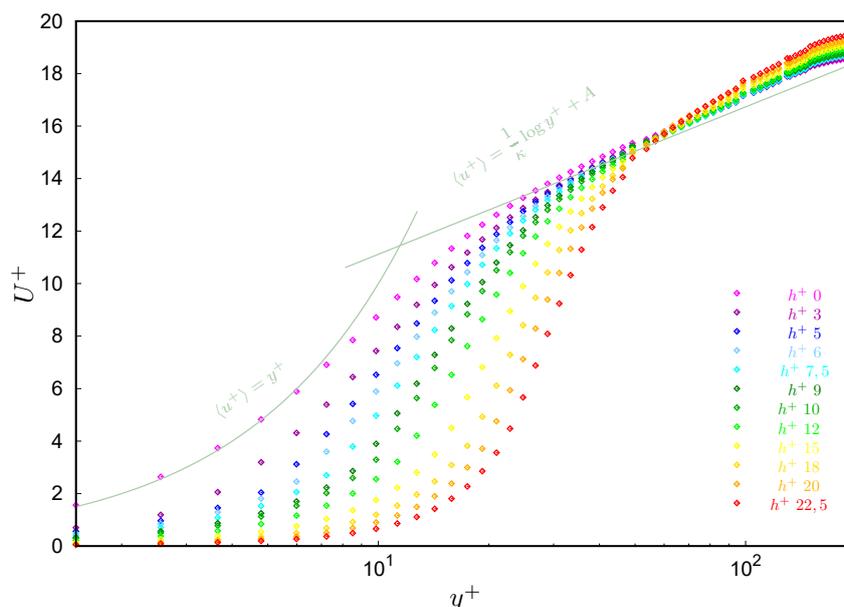


Figura 5.17: Grafico di $y^+ - u^+$, in scala logaritmica, riportante gli andamenti di tutte le configurazioni analizzate, dove la u^+ è calcolata utilizzando, per tutte le curve, la u_τ della configurazione $H0$.

essere simile per tutte le configurazioni, ed infine, andando verso il centro del canale, si osserva come ora, le curve dei casi con presenza di riblet vadano ad avere una velocità addirittura leggermente superiore al caso con riblet, in linea con quanto riportato fin ora.

Nel caso con le u_τ calcolate per ogni curva invece, si osserva lo stesso fenomeno, ma con ampiezze differenti per le due zone considerate, quella lineare e quella logaritmica. L'effetto di discostamento, in questo grafico (5.18), ora deve tenere infatti conto, della variazione della u_τ per ogni configurazione. Con la presenza delle riblet, come si è già accennato e che si vedrà a breve (vedi 5.5.3), la resistenza del flusso turbolento cala, ed essendo la u_τ direttamente proporzionale alla resistenza, cala anch'essa. Ciò provoca, che, nella zona lineare, il discostamento verso il basso delle curve, dovuto ad una minore velocità a causa del sottostrato viscoso più ampio, è mitigato dal più basso valore anche della u_τ , portando ad un discostamento meno accentuato. Al contrario, nella zona logaritmica, invece, dove si ha un discostamento positivo, a causa della maggiore velocità nel centro del canale, questo viene accentuato da una u_τ minore, rendendo il discostamento in questa zona più ampio e molto più evidente rispetto al caso con la u_τ costante.

Infine si è detto in precedenza come le curve delle configurazioni con riblet si discostassero dalla curva dell'andamento empirico, questo perché questa è stata studiata per il caso di un canale piano, senza riblet. Sarebbe possibile trovare dei coefficienti correttivi a questa legge, per ottenere una curva empirica della velocità per il flusso in un canale con riblet, ma non rientrando nell'obiettivo di questo studio, qui ciò non è stato fatto. Infine, si vanno a riportare 2 grafici che esprimono l'incremento di velocità che si ha al centro del canale, per effetto delle riblet, in funzione dell'altezza in unità di parete, di queste. In particolare nelle immagini 5.19 e 5.20, abbiamo, rispettivamente, l'incremento di velocità espresso in grandezze fisiche adimensionali ($\frac{\Delta U}{U_b}$) e quello espresso in

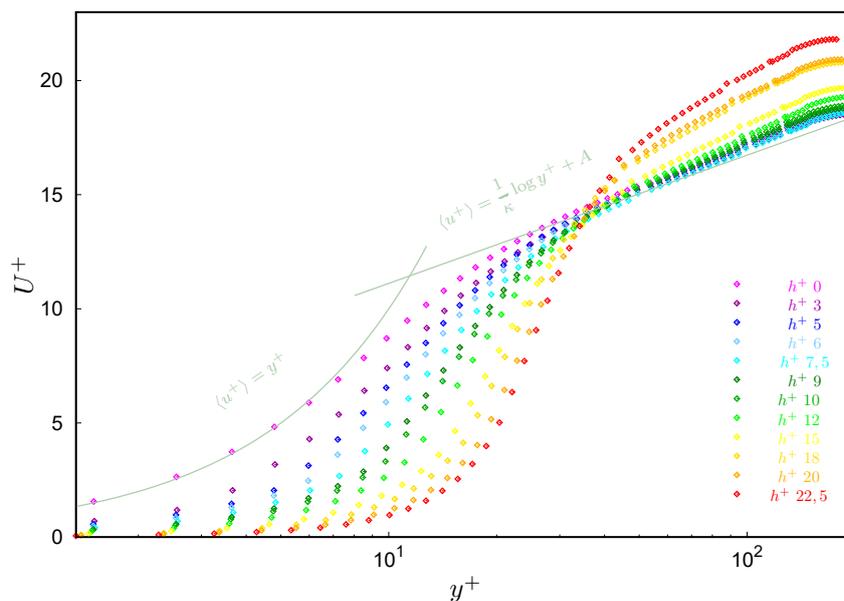


Figura 5.18: Grafico di $y^+ - u^+$, in scala logaritmica, riportante gli andamenti di tutte le configurazioni analizzate, dove la u^+ è calcolata utilizzando, per ogni curva, la propria u_τ .

unità di parete (ΔU^+). Dato che l'unica differenza tra queste 2 immagini è la u_τ , ma essendosi preso sempre la stessa, quella della configurazione $H0$, è costante, e quindi le 2 curve hanno lo stesso andamento, seppur con scala differente, e quindi per entrambe valgono le stesse osservazioni. Si può osservare, in entrambe le figure, quindi, che al crescere dell'altezza delle riblet, la velocità al centro canale cresce, come si è già osservato in precedenza in diversi casi, sia relativi a $\frac{U}{U_b}$, che relativi a ΔU^+ . Osservando invece i grafici nel dettaglio, e soffermandoci sui numeri riportati, si osserva come, nel caso di velocità $\frac{U}{U_b}$, l'incremento che si osserva sia solamente di circa il 6% per la configurazione più estrema analizzata, ovvero $H22$; invece osservando la velocità espressa in unità di parete, si ha, che per la stessa configurazione si ha un incremento di quasi il 90%, ovvero che la velocità in termini di unità di parete, rispetto al caso di canale piano, è quasi raddoppiata. Ciò ci indica ulteriormente come le riblet siano una soluzione che vada ad intervenire molto sul flusso al livello dello strato limite (in quanto le velocità in termini di unità di parete, sono le grandezze che caratterizzano questa zona), mentre poco a livello globale sul flusso, nella sua zona centrale. Inoltre si deve tener presente, per entrambe le figure, come ricordano anche le bande verticali di errore, che per queste si deve sempre tener presente la presenza di un errore nei valori ottenuti, a causa della distorsione della griglia, che è tanto maggiore, quanto maggiore è l'altezza delle riblet della configurazione riportata (e ciò spiega perché le bande d'errore crescono al crescere dell'altezza delle riblet). La presenza di questo errore, ci fa anche capire come, la posizione dei punti, ed il relativo valore, non siano da prendere in modo assoluto, ma vadano intesi come un intervallo di possibili valori possibili, per la grandezza che si ha, per una determinata configurazione.

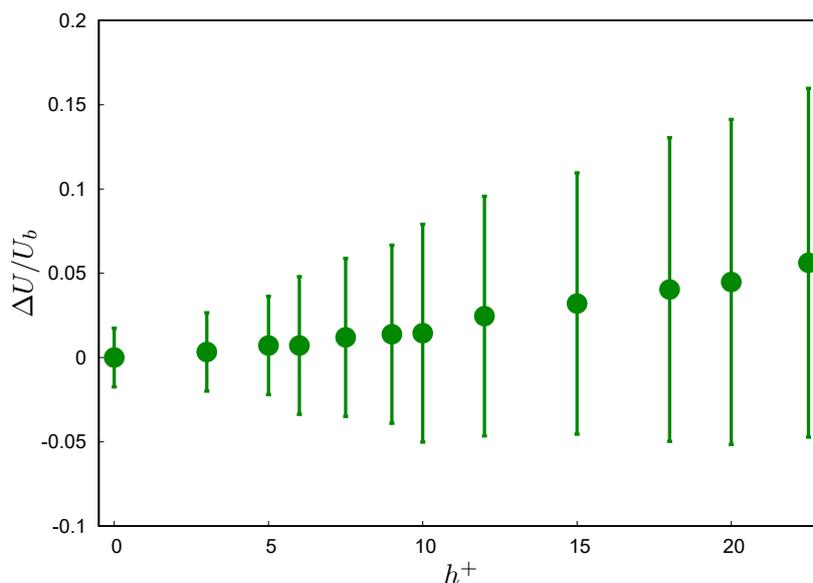


Figura 5.19: Grafico di $h^+ - \frac{\Delta U}{U_b}$, riportante l'incremento di velocità che si ha al centro del canale, a causa delle riblet, espresso in funzione dell'altezza in unità di parete di queste. Inoltre è riportata una banda d'errore verticale, che ci indica l'incertezza del dato, dovuta alla distorsione della griglia. La banda è proporzionale al valore del dato ed all'altezza della riblet della configurazione.

5.5.2 Varianza della velocità

Si vanno a vedere ora, invece, dopo esserci soffermati sulla velocità media del flusso, in alcune delle sue forme, le componenti di velocità che sono maggiormente importanti per la turbolenza, ovvero le fluttuazioni di velocità, cioè quelle che sono le componenti della varianza della velocità. Queste sono state calcolate dal codice del post processing (vedi 4.3) nelle 6 componenti fondamentali della decomposizione di Reynolds, che sono: $u'u'$ $u'v'$ $u'w'$ $v'v'$ $v'w'$ $w'w'$. Da queste, è possibile ricavare anche altre grandezze caratteristiche della turbolenza.

Come prima cosa, si vanno a riportare gli andamenti, in funzione della distanza dalla parete, delle 3 componenti fondamentali delle fluttuazioni di velocità, ovvero: u' v' w' . Ciò viene fatto in 3 grafici, uno per componente, che confrontano l'andamento di queste grandezze, per ogni configurazione analizzata. In queste figure (5.21, 5.22 e 5.23) è possibile osservare, che tutte e 3 le componenti di fluttuazioni della velocità, diminuiscono all'aumentare dell'altezza delle riblet. Ciò vale sia come valore massimo raggiunto dalle componenti, e sia come ampiezza della curva, in quanto i casi con riblet piccole o nulle, raggiungono il loro picco molto velocemente, e più vicino alla parete, mentre le curve con riblet alte, oltre a raggiungere picchi più bassi, lo fanno più lentamente e più lontano da parete. Osservando i valori numerici delle curve, si osserva che, per tutte le configurazioni analizzate, la componente u' sia molto più grande rispetto alle altre 2 (circa 3 volte la v' e più del doppio di w'). Questa risulta essere molto importante, per quanto riguarda sia l'energia cinetica della turbolenza, che l'intensità della turbolenza, in quanto, essendo la componente predominante delle fluttuazioni, è quella che determina queste grandezze (che verranno tra poco riportate). Comunque, seppur minori in modulo, anche le altre 2 componenti risultano essere

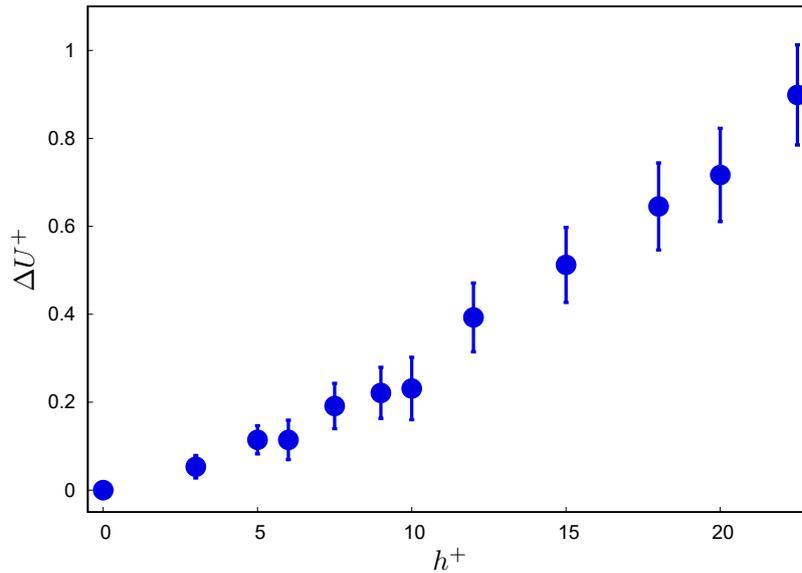


Figura 5.20: Grafico di $h^+ - \Delta U^+$, riportante l'incremento di velocità che si ha al centro del canale in unità di parete, a causa delle riblet, espresso in funzione dell'altezza in unità di parete di queste. La u_τ con cui si è calcolato il ΔU^+ , è quella della configurazione $H0$. Inoltre è riportata una banda d'errore verticale, che ci indica l'incertezza del dato, dovuta alla distorsione della griglia. La banda è proporzionale al valore del dato ed all'altezza della riblet della configurazione.

importanti, soprattutto dal punto di vista della comprensione della fenomenologia fisica alla base della turbolenza, in quanto si ha che queste 2 misurano l'intensità delle strutture turbolente, e soprattutto dei vortici di rimescolamento, in quanto la presenza di componenti istantanee di velocità in direzione y e z , ci indica che c'è movimento di flusso in quelle direzioni, ovvero che si ha un vortice che rimescola il flusso.

Per mettere in maggiore risalto, la differenza di intensità tra le 3 componenti di fluttuazione della velocità, si vanno ora a riportare alcuni grafici, i quali riportano insieme, le 3 componenti di fluttuazione, per alcune delle configurazioni analizzate, nello specifico $H7$, $H9$ e $H12$. Nelle figure 5.24, 5.25 e 5.26 si osserva come tra le componenti delle fluttuazioni, in questo caso riportate come valore quadro, la differenza sia molto accentuata, con una prevalenza della $u'u'$ per tutta l'altezza del canale H . Inoltre, confrontando le 3 figure, si osserva, come già visto in precedenza, che l'intensità di tutte e tre le componenti di fluttuazione, al crescere delle riblet, diminuisce.

Si riportano ora, invece, gli andamenti dell'intensità della turbolenza e dell'energia cinetica turbolenta, già citate in precedenza. Queste sono importanti, in quanto contribuiscono ad identificare quanto la turbolenza va ad influenzare un qualsiasi flusso turbolento che si sta studiando, sia questo il nostro flusso nel canale piano, o un qualsiasi altro problema fluidodinamico, computazionale o sperimentale. In particolare l'intensità di turbolenza, definita come:

$$I_{\text{Turb}} = \frac{\sqrt{\langle u'u' \rangle} + \sqrt{\langle v'v' \rangle} + \sqrt{\langle w'w' \rangle}}{U_{\text{max}}}$$

misura la percentuale di turbolenza presente in un flusso che si sta analizzando, ed è particolarmente importante in studi il cui obiettivo è quello di andare a ridurre il più

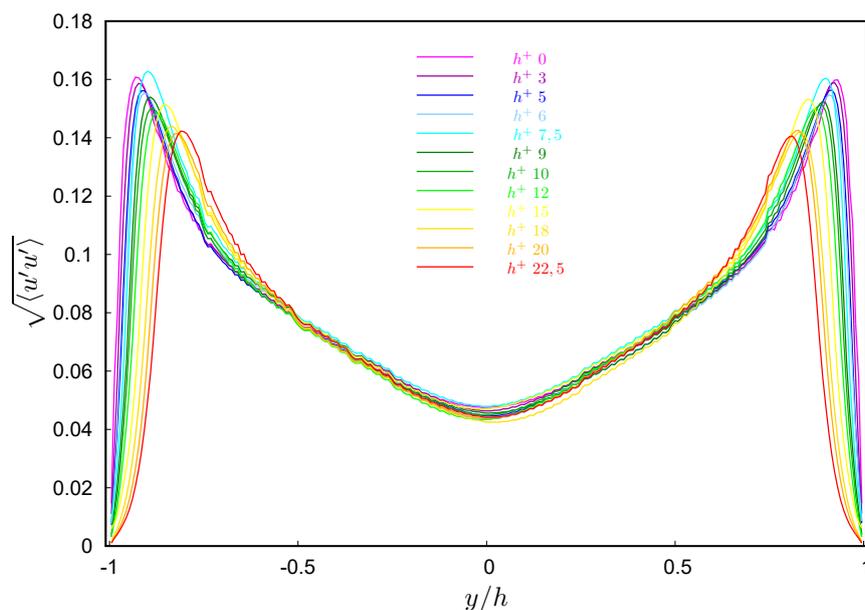


Figura 5.21: Grafico di $\frac{y}{h} - \sqrt{\langle u'u' \rangle}$, riportato per tutte le configurazioni di riblet analizzate.

possibile la turbolenza (come nel caso di progetto delle reti per le gallerie del vento a circuito chiuso). L'energia cinetica turbolenta, invece, misura quanta energia cinetica, tra tutta quella posseduta da un flusso, appartiene alle componenti turbolente della velocità, ovvero le fluttuazioni. Questa è definita come:

$$EK_{\text{Turb}} = \frac{1}{2} (\overline{u'u'} + \overline{v'v'} + \overline{w'w'})$$

, ed è importante soprattutto per quantificare le perdite che si hanno in un flusso, dovute alla velocità che divenuta turbolenta, non può essere trasformata in pressione, senza avere grandi perdite. In entrambe le figure riportate, la 5.27 e la 5.28, si osserva come, essendo la u predominante nel definire il valore di queste grandezze, come già detto in precedenza, queste 2 curve rispecchiano lo stesso andamento della curva di u per quanto riguarda l'intensità turbolenta, e della curva di $u'u'$ per quanto riguarda l'energia cinetica turbolenta, in quanto la u compare in queste 2 forme nelle equazioni definenti le grandezze. Essendo così strettamente dipendenti da u , valgono, per entrambe, le stesse osservazioni fatte per questa, ovvero che al crescere dell'altezza delle riblet, queste diminuiscono.

L'ultima grandezza caratterizzante la turbolenza di un flusso che si va a riportare, è anche una delle più importanti. Nello specifico si tratta della componente di turbolenza $u'v'$, che risulta essere particolarmente importante, in quanto questa è una delle due componenti della resistenza in un flusso turbolento, ovvero la componente inerziale, detta anche sforzo di Reynolds. Per questa grandezza si riportano 2 grafici, dove il primo (vedi 5.29) ne riporta l'andamento globale su tutta l'altezza del canale, mentre il secondo (vedi 5.30) ne riporta l'andamento solo su una metà canale, ed espressa nella forma che determina lo sforzo di Reynolds. Questo secondo grafico in particolare, risulta essere di fondamentale interesse, in quanto si va ad osservare come la presenza delle riblet incide anche sulla seconda delle 2 componenti di resistenza di un flusso turbolento (dopo aver osservato più volte indirettamente, come vada a diminuire di molto

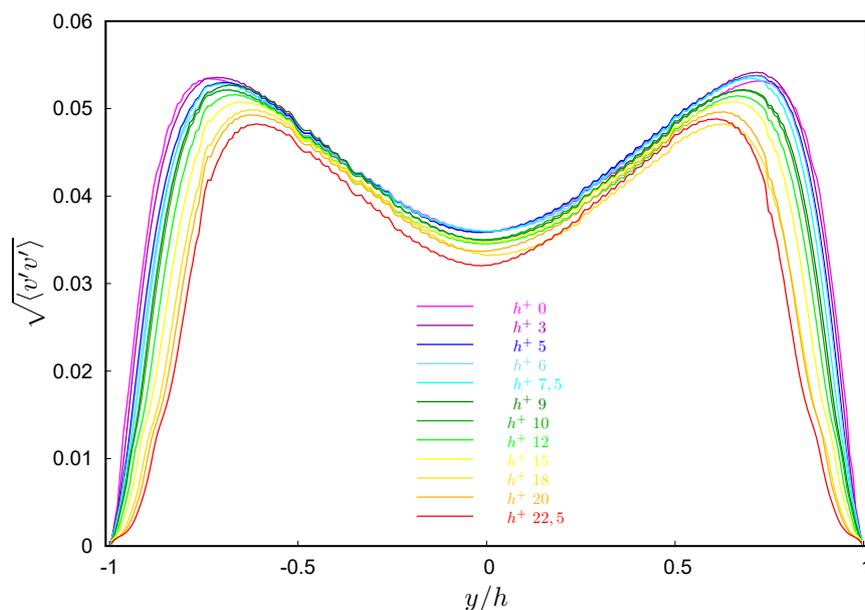


Figura 5.22: Grafico di $\frac{y}{h} - \sqrt{\langle v'v' \rangle}$, riportato per tutte le configurazioni di riblet analizzate.

la componente viscosa stirando il sottostrato viscoso). Ulteriori discorsi in merito alla resistenza sono fatti nella sezione seguente (vedi 5.5.3), mentre qui ci si concentra semplicemente ad osservare come cambia il comportamento di questa curva, a causa della presenza delle riblet. Abbiamo che, anche per questa grandezza, si osserva una diminuzione al crescere dell'altezza delle riblet, e così come per le altre grandezze della turbolenza fin qui riportate, si ha sia una diminuzione del valore massimo raggiunto dalla grandezza, e sia un raggiungimento di questo valore meno rapido, ed ad una distanza maggiore dalla parete. Ciò significa che quindi le riblet, oltre ad andare a modificare la struttura del sottostrato viscoso, riducendo la componente viscosa della resistenza, vanno a modificare e ridurre, anche la componente inerziale della resistenza. Di seguito, si entra maggiormente nel dettaglio in merito alle componenti della resistenza, e come la presenza delle riblet va ad agire su di esse.

5.5.3 Sforzi di Reynolds e sforzi viscosi

Infine, si vanno a riportare i grafici con gli andamenti delle componenti della resistenza di un flusso turbolento. Queste sono la componente viscosa, che dipende dalla derivata della velocità a parete, in particolare nel sottostrato viscoso e la componente inerziale, che dipende dal termine $u'v'$ delle fluttuazioni di velocità. Per quanto riguarda i grafici che si vanno ora a riportare, è già stato spiegato in precedenza il procedimento che si è seguito per realizzarli (vedi 4.3.2), mentre per quanto riguarda i valori di resistenza totale che si è ottenuti attraverso la realizzazione di questi grafici, sono riportati nella sezione successiva (vedi 5.6) insieme ai relativi grafici ed osservazioni. In questo punto, invece, si riporteranno i grafici ottenuti, anche se non per tutte le configurazioni analizzate, per non essere ridondanti ed appesantire inutilmente questa sezione, e si osserveranno gli effetti delle riblet sulle componenti della resistenza (che quindi, saranno gli stessi che poi si avranno sulla resistenza totale). I grafici che si è

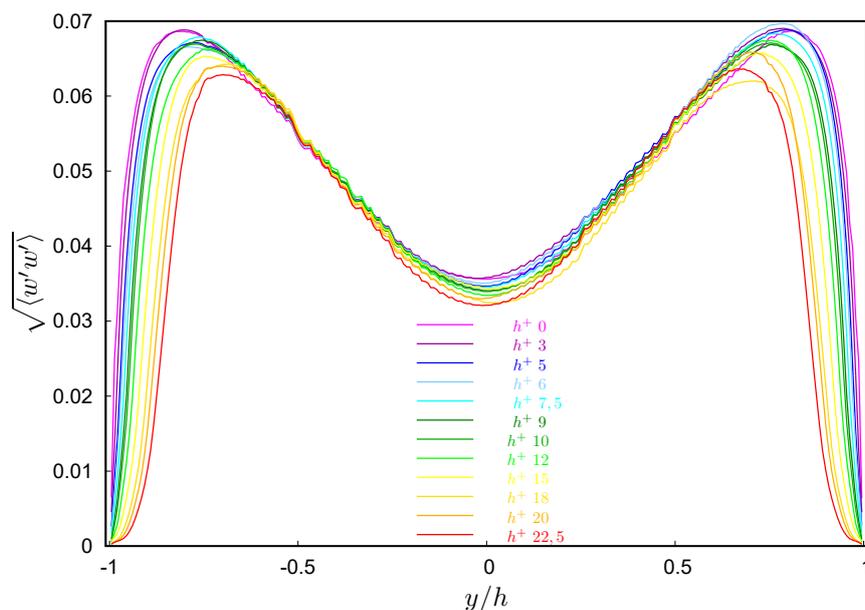


Figura 5.23: Grafico di $\frac{y}{h} - \sqrt{\langle w'w' \rangle}$, riportato per tutte le configurazioni di riblet analizzate.

scelti di riportare sono 4, e coprono, tutto il range di altezze che si è andati a studiare, in quanto si ha il caso senza riblet, $H0$, un caso con riblet piccole, $H6$, un caso con riblet medie, $H12$ ed il caso con le riblet più alte tra quelli analizzati, $H22$. Nella prima di queste figure, la 5.31, quella per la configurazione $H0$, si può osservare l'andamento delle componenti della resistenza di un flusso turbolento, nel caso di un canale piano liscio; si osserva come, in prossimità della parete, la totalità della resistenza sia data dalla componente viscosa, mentre quella inerziale sia trascurabile; allontanandoci dalla parete, si ha invece che la componente inerziale cresce, fino a raggiungere il suo picco ed a prendere il sopravvento su quella viscosa, che invece cala velocemente e rimane trascurabile per il resto del canale; in questo caso si ha che la massima resistenza si ha a parete, ed è appunto lo sforzo di taglio a parete, τ_w , qui calcolato mediante l'interpolazione della curva riportante la somma delle due componenti di resistenza. Osservando poi la configurazione successiva riportata, la 5.32, quella per la configurazione $H0$, si osserva come la componente di attrito viscoso non sia più massimo a parete, ma questo si sia spostato leggermente verso il centro del canale; inoltre anche il suo valore massimo è diminuito; questa diminuzione della componente viscosa, soprattutto nella zona di parete, si traduce ovviamente in una diminuzione della resistenza totale, come è possibile osservare anche dalla retta di fitting, che ha ora una pendenza minore, corrispondente ad uno sforzo di taglio a parete minore. Per quanto riguarda la componente inerziale della resistenza, c'è anche qui una diminuzione, ma molto minore rispetto all'altra componente, risultando quasi impercettibile. Passando poi alla terza configurazione riportata, quella per $H12$, il grafico 5.33, si osserva come qui, la riduzione della componente viscosa della resistenza, inizi a diventare davvero significativa, con conseguente diminuzione della resistenza totale. Il massimo della resistenza viscosa, per effetto delle riblet, tende a spostarsi ancor più verso il centro del canale, ed iniziano a diventare apprezzabili anche le diminuzioni della componente inerziale. Per l'ultima componente riportata, il grafico 5.34, per la configurazione $H12$ si osservano ulteriori

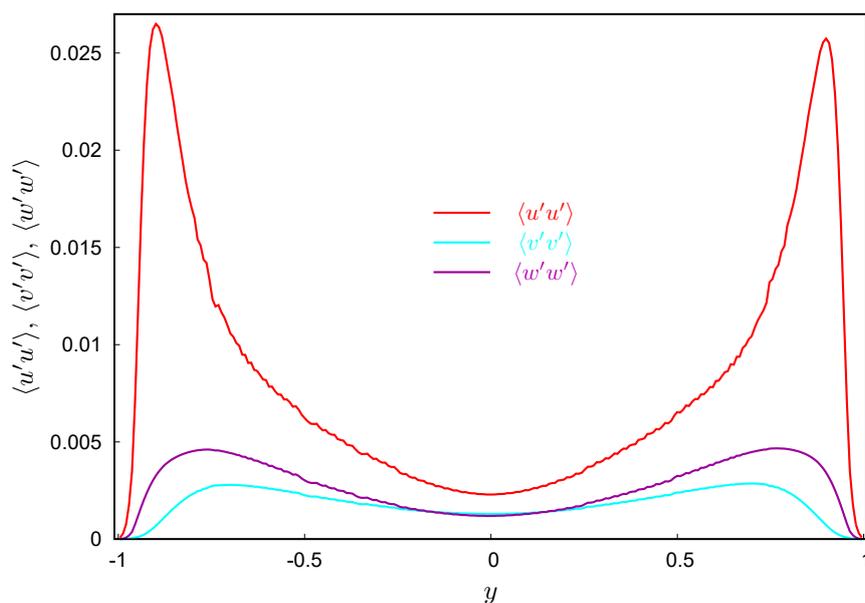


Figura 5.24: Grafico di $\sqrt{\langle u'u' \rangle}$, $\sqrt{\langle v'v' \rangle}$ e $\sqrt{\langle w'w' \rangle}$ in funzione di y , per la configurazione $H7$.

diminuzioni, sia della componente inerziale, ma soprattutto di quella viscosa, che ora anche al suo massimo, non è superiore rispetto al massimo della componente inerziale. Anche la resistenza totale, di conseguenza, cala visibilmente, così come si riduce la pendenza della retta interpolante. Per gli altri casi non riportati, il comportamento è il medesimo dei casi riportati, in funzione ognuno della sua altezza delle riblet, andando dalla configurazione di $H0$ a quella di $H22$.

Grazie a queste curve, come già detto, oltre a poter osservare graficamente l'andamento delle componenti della resistenza in un flusso turbolento, si è potuti, grazie all'interpolazione della somma delle componenti di resistenza, trovare il coefficiente angolare della retta interpolante, che corrisponde, attraverso l'uguaglianza delle aree di integrazione, o l'uguaglianza dei coefficienti delle rette, allo sforzo totale d'attrito a parete, τ_w . Noto questo si è trovata, per ogni configurazione la resistenza del flusso turbolento che scorre in quel canale con quell'altezza di riblet, ed è possibile quindi andare a confrontare i diversi casi studiati. Ciò verrà riportato nella sezione successiva (vedi ??); qui invece, come ultima osservazione, si va ad osservare come, al crescere dell'altezza delle riblet, la curva riportante la somma delle componenti di resistenza, passi da essere quasi una retta, ad un comportamento molto più irregolare; ciò è dovuto allo spostamento della posizione del massimo della componente viscosa di resistenza. Per il caso senza riblet ($H0$), infatti, questo si trova a parete, e va a bilanciare perfettamente ed in modo complementare l'andamento della componente inerziale, la quale è nulla a parete, e cresce, mentre quella viscosa cala. Nel momento in cui vengono introdotte le riblet, la componente viscosa risente molto di più di questo effetto, e si sposta abbastanza verso il centro; tuttavia, invece, la componente inerziale, pur riducendosi leggermente, non cambia la posizione del suo massimo in modo apprezzabile; ciò porta, nei casi di configurazioni di riblet più estremi, ad una quasi sovrapposizione dei massimi delle due componenti, con al contrario una zona vicina alla parete con componenti di resistenza quasi assenti. Ciò è dovuto all'effetto delle riblet, ed è uno dei fenomeni alla base

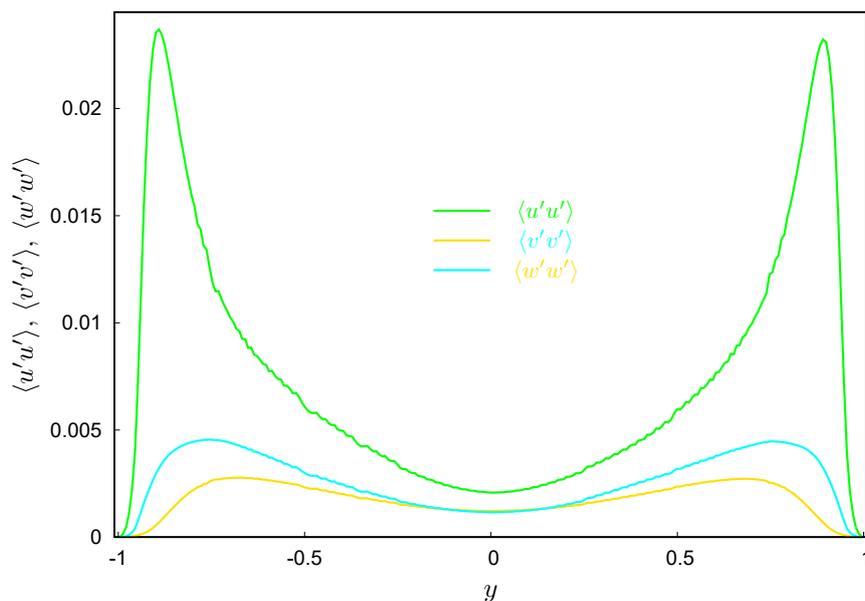


Figura 5.25: Grafico di $\sqrt{\langle u'u' \rangle}$, $\sqrt{\langle v'v' \rangle}$ e $\sqrt{\langle w'w' \rangle}$ in funzione di y , per la configurazione H9.

della riduzione della resistenza che si hanno con questo metodo. Tuttavia si deve tenere anche conto, che lo spostarsi dalla sua *posizione naturale* della curva riportante la resistenza totale, causa anche un errore di interpolazione maggiore, in quanto, risulta più difficile andare ad interpolare la curva in modo lineare, in quanto questa perde, soprattutto nella zona vicina a parete, il suo andamento lineare. Si deve quindi tenere conto anche di questo errore dovuto all'interpolazione, nel valutare i dati ottenuti, oltre a quelli dati dalla discretizzazione numerica e dalla distorsione della griglia che si ha per configurazioni di riblet molto alte.

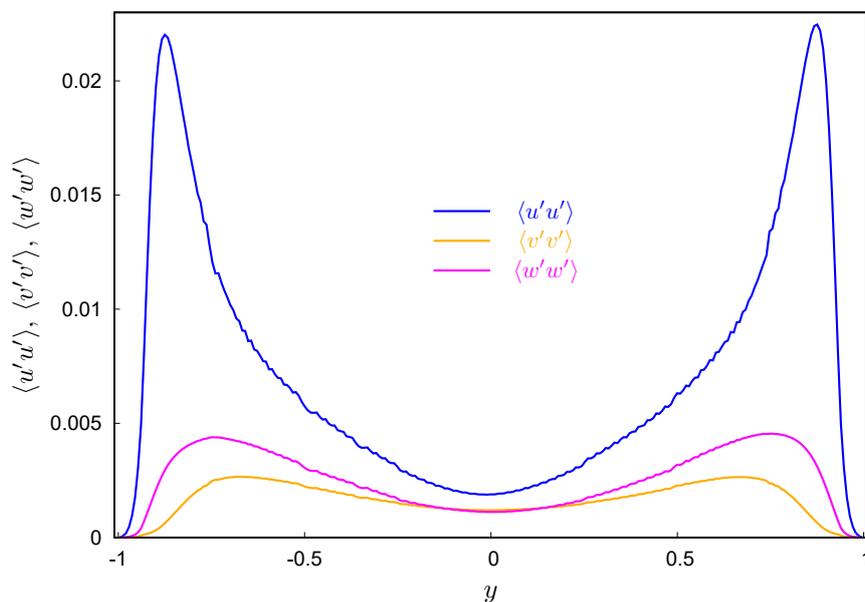


Figura 5.26: Grafico di $\sqrt{\langle u'u' \rangle}$, $\sqrt{\langle v'v' \rangle}$ e $\sqrt{\langle w'w' \rangle}$ in funzione di y , per la configurazione $H12$.

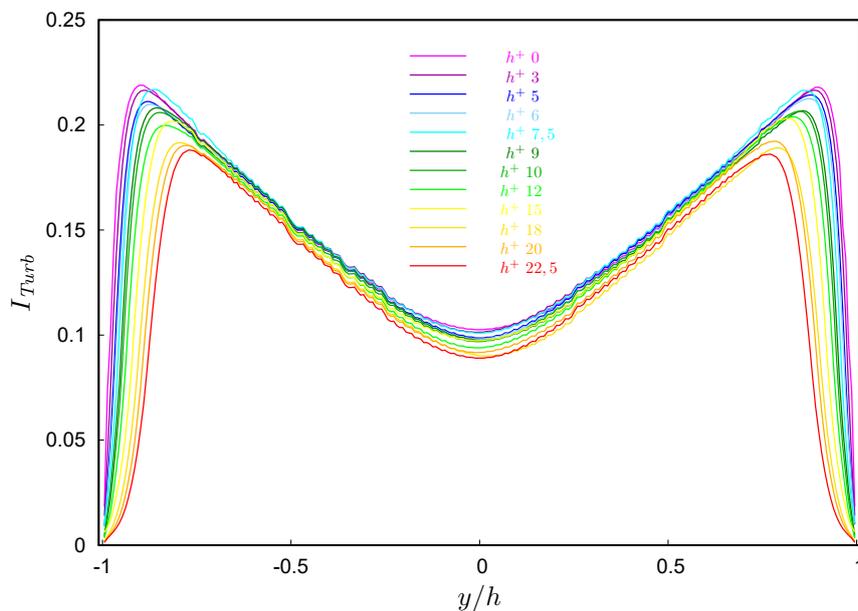


Figura 5.27: Grafico di $\frac{y}{h} - I_{Turb}$, per tutte le configurazioni analizzate.

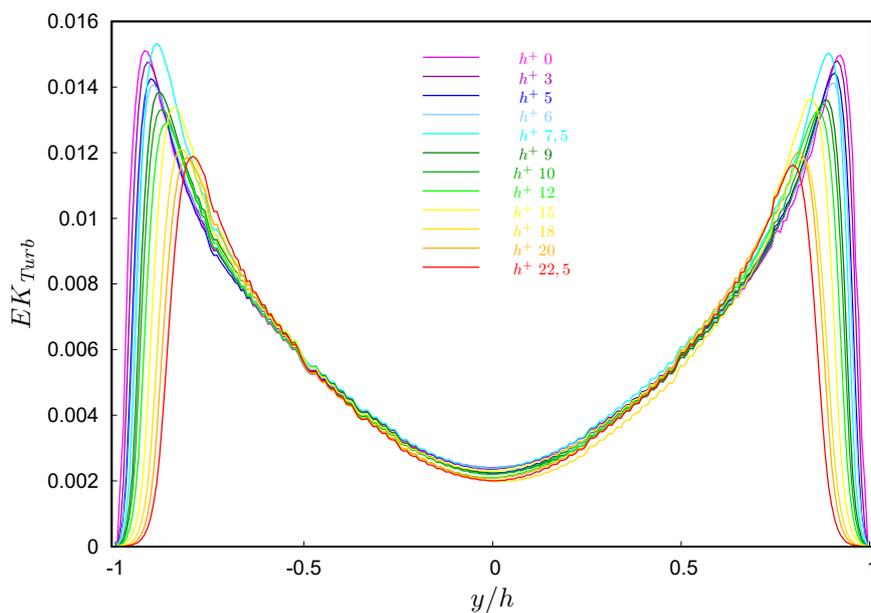


Figura 5.28: Grafico di $\frac{y}{h} - EK_{\text{Turb}}$, per tutte le configurazioni analizzate.

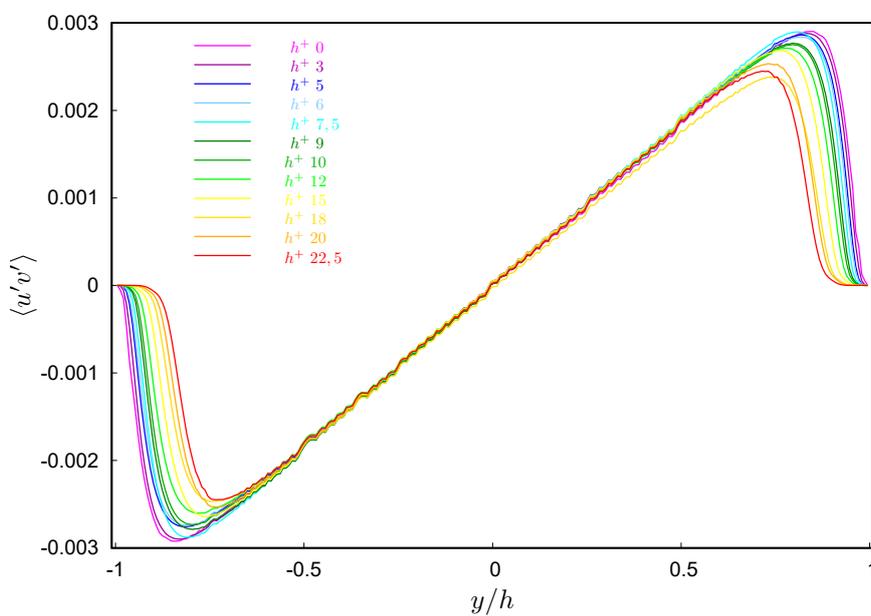


Figura 5.29: Grafico di $\frac{y}{h} - \langle u'v' \rangle$, per tutte le configurazioni analizzate, lungo tutta l'altezza H del canale.

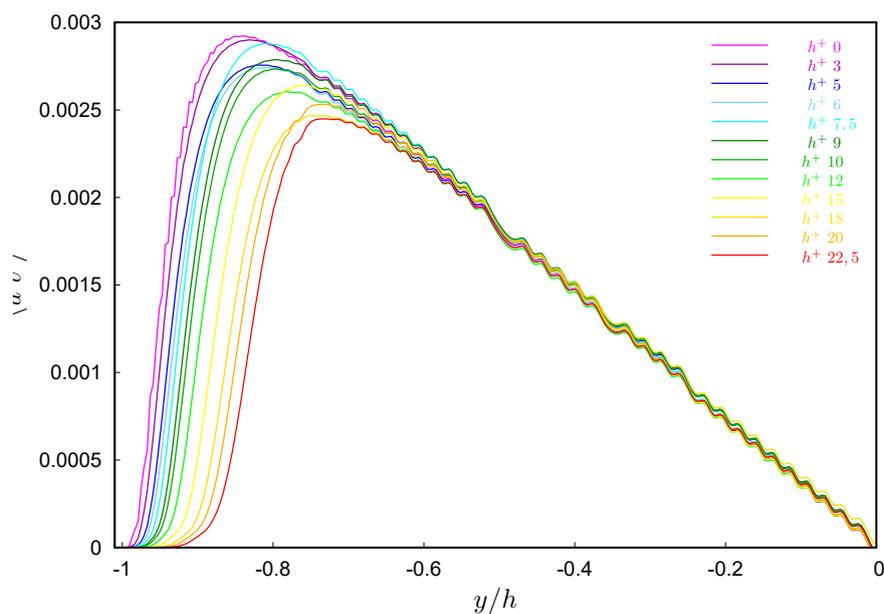


Figura 5.30: Grafico di $\frac{y}{h} - \langle u'v' \rangle$, per tutte le configurazioni analizzate, lungo tutta l'altezza H del canale.

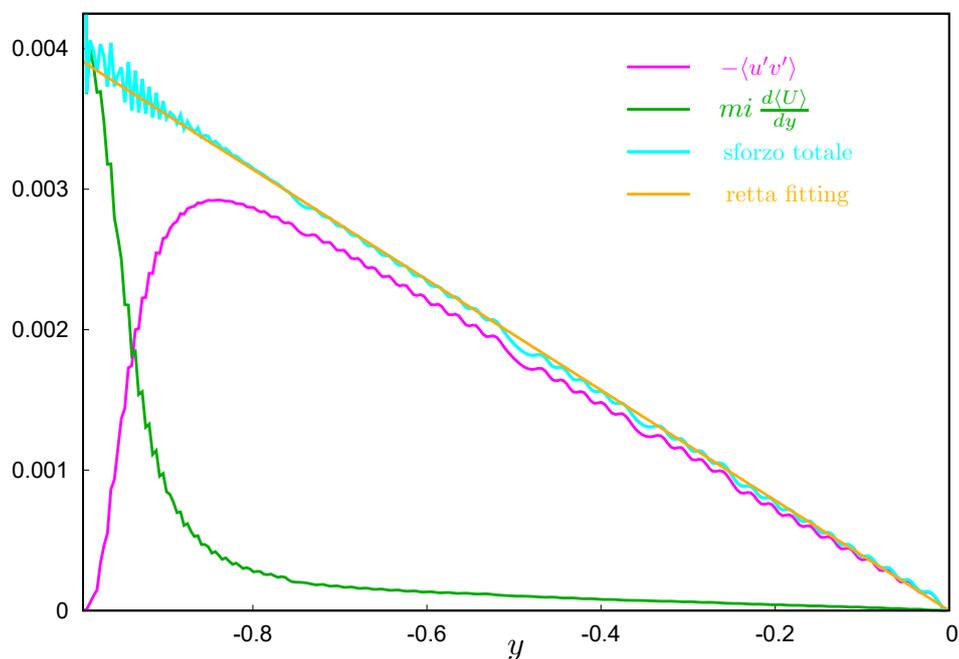


Figura 5.31: Grafico delle componenti della resistenza in un flusso turbolento, in particolare si hanno: la componente inerziale $-\langle u'v' \rangle$, la componente viscosa $mi \frac{d\langle U \rangle}{dy}$, la somma di queste 2 componenti, e la retta che va ad interpolare la somma per calcolare lo sforzo di taglio a parete, riportate in funzione di y . Questa è la configurazione $H0$.

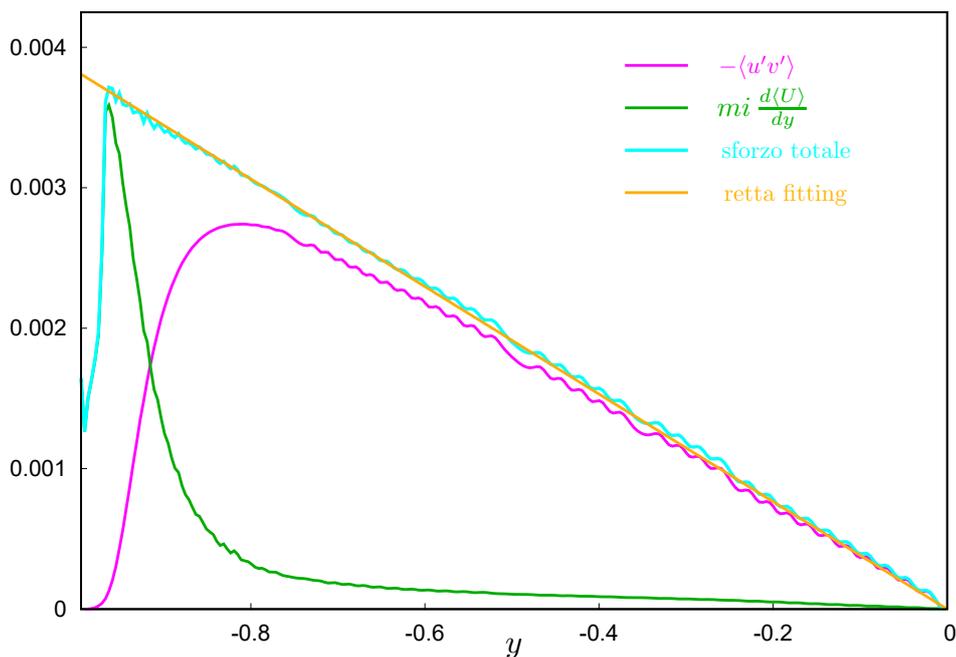


Figura 5.32: Grafico delle componenti della resistenza in un flusso turbolento, in particolare si hanno: la componente inerziale $-\langle u'v' \rangle$, la componente viscosa $mi \frac{d\langle U \rangle}{dy}$, la somma di queste 2 componenti, e la retta che va ad interpolare la somma per calcolare lo sforzo di taglio a parete, riportate in funzione di y . Questa è la configurazione *H6*.

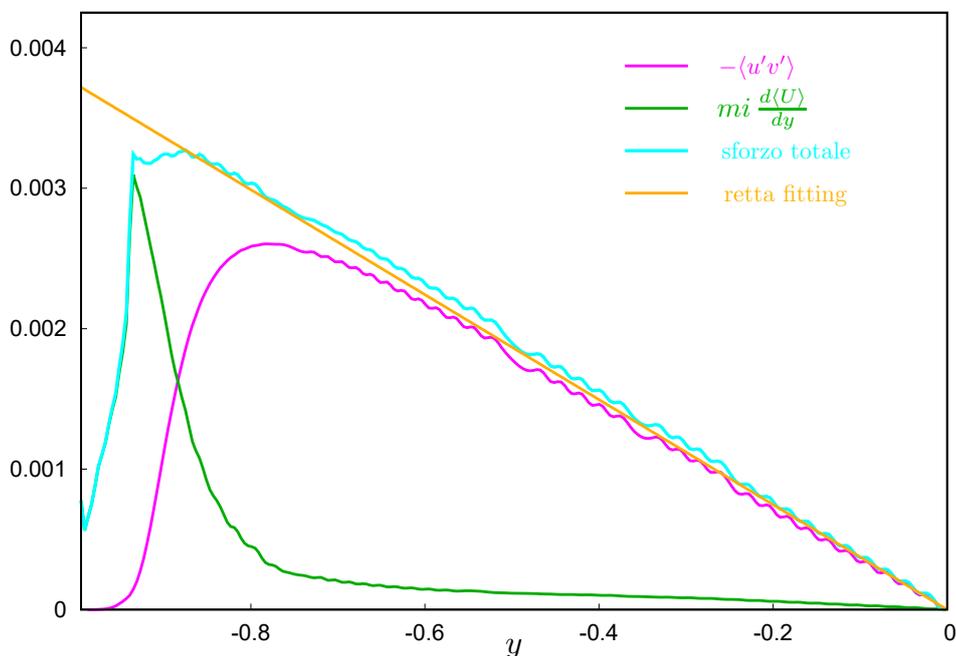


Figura 5.33: Grafico delle componenti della resistenza in un flusso turbolento, in particolare si hanno: la componente inerziale $-\langle u'v' \rangle$, la componente viscosa $mi \frac{d\langle U \rangle}{dy}$, la somma di queste 2 componenti, e la retta che va ad interpolare la somma per calcolare lo sforzo di taglio a parete, riportate in funzione di y . Questa è la configurazione *H12*.

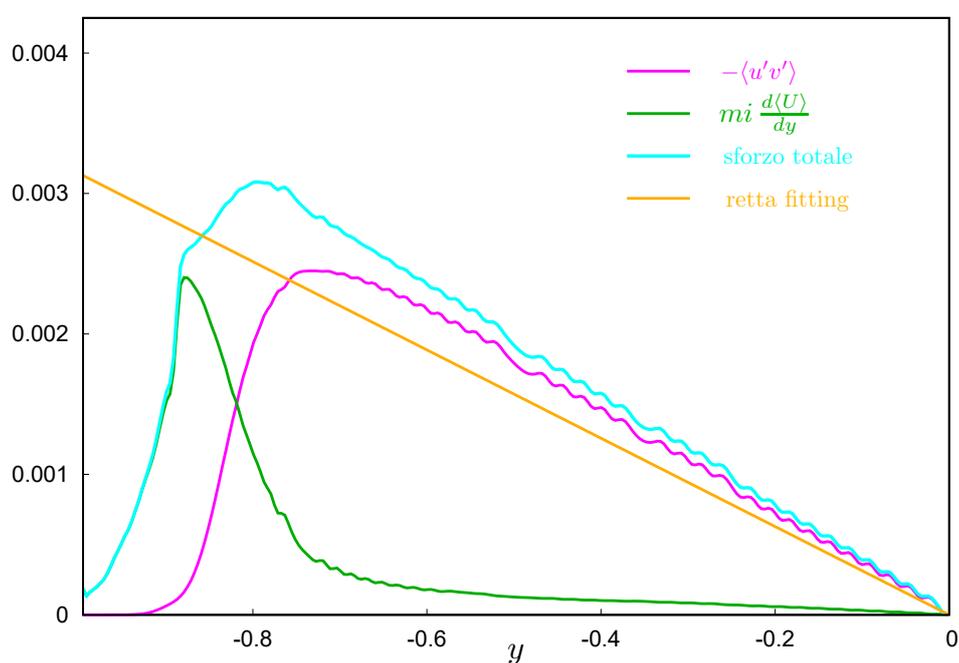


Figura 5.34: Grafico delle componenti della resistenza in un flusso turbolento, in particolare si hanno: la componente inerziale $-\langle u'v' \rangle$, la componente viscosa $mi \frac{d\langle U \rangle}{dy}$, la somma di queste 2 componenti, e la retta che va ad interpolare la somma per calcolare lo sforzo di taglio a parete, riportate in funzione di y . Questa è la configurazione $H22$.

5.6 Resistenza

Tabella 5.1: Valori della resistenza calcolata per tutte le configurazioni analizzate.

Configurazione	h_{rib}	AR	τ_w	C_f	ΔD [%]
H0	0	0	3,92e-03	1,96e-03	0
H3	0,015	0,2	3,95e-03	1,97e-03	-0,77
H5	0,025	0,334	3,90e-03	1,95e-03	0,53
H6	0,03	0,4	3,89e-03	1,94e-03	0,70
H7	0,0375	0,5	3,98e-03	1,99e-03	-1,59
H9	0,045	0,6	3,90e-03	1,95e-03	0,42
H10	0,05	0,667	3,86e-03	1,93e-03	1,41
H12	0,06	0,8	3,78e-03	1,89e-03	3,46
H15	0,075	1	3,67e-03	1,84e-03	6,22
H18	0,09	1,2	3,34e-03	1,67e-03	14,76
H20	0,1	1,334	3,32e-03	1,66e-03	15,17
H22	0,1125	1,5	3,12e-03	1,56e-03	20,42

Dopo averli calcolato, attraverso i grafici riportanti gli andamenti delle componenti della resistenza, vengono qui riportati i valori di *sforzo d'attrito a parete*, che si sono ottenuti per tutte le configurazioni analizzate in questo studio. In particolare, i valori che si sono ricavati dai grafici sono i coefficienti dell'interpolazione lineare della curva riportante la somma delle componenti di resistenza di un flusso turbolento. Come si è visto (4.3.2) questi coefficienti coincidono proprio con il valore di τ_w , lo sforzo d'attrito a parete. Una volta ottenuto questo, è possibile andare a calcolare sia il C_f , il coefficiente d'attrito adimensionale, che fornisce una stima della resistenza della configurazione in forma adimensionale, anche se, come si è visto, dato che il problema è stato già impostato in forma adimensionale, il rapporto tra τ_w e C_f , è solamente un coefficiente 2. Dopo di che, da questo valore, si è potuti andare a calcolare la percentuale di riduzione di resistenza, come $\Delta D = \frac{C_{f0} - C_{f,Hrib}}{C_{f0}}$, che ogni configurazione ha, per stimare in questo modo, quanta resistenza in meno ogni caso offre, anche se nel fare ciò, si deve comunque tener conto di un margine di errore che le simulazioni possono aver dato. Tutti questi valori vengono riportati nella tabella 5.1, in modo tale da essere presentati in modo chiaro e diretto. Inoltre nella tabella, si riporta anche il fattore di forma delle riblet, l'aspect ratio AR , insieme all'altezza adimensionale h^+ , per poter osservare come, il variare della forma (più che dell'altezza) delle riblet, incida sulla resistenza del flusso turbolento. Andando ad osservare i dati riportati in tabella 5.1 si possono fare alcune osservazioni in merito all'andamento della resistenza, anche se, bisogna tenere conto dell'errore che affigge questi dati, che non è visibile nei semplici numeri riportati nella tabella 5.1, mentre è possibile osservare gli intervalli di errore dei dati, nei relativi grafici 5.35 e 5.36. Come riportato, l'errore è funzione della distorsione della griglia a causa delle riblet, dell'interpolazione numerica effettuata, oltre che della discretizzazione numerica della griglia.

Come prima cosa, si osserva immediatamente che il valore più basso di resistenza, e quindi la percentuale più alta di riduzione la si ha per la configurazione $H22$. Ciò ci indica che nel nostro caso, andare ad aumentare l'altezza delle riblet ha prodotto sempre una diminuzione della resistenza, avendo quindi che gli effetti positivi sono

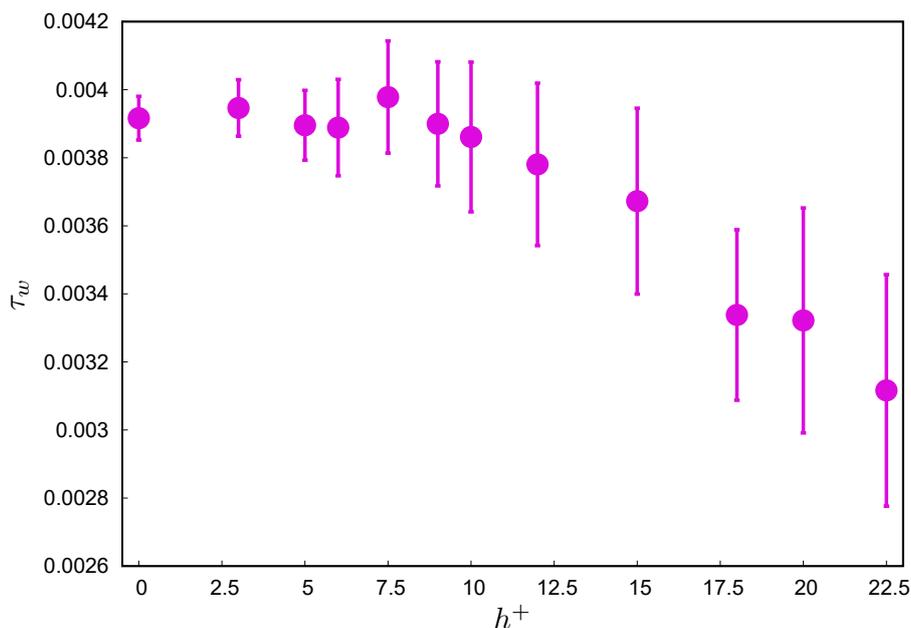


Figura 5.35: Grafico di $h^+ - \tau_w$, riportante i dati ottenuti per tutte le configurazioni analizzate. Inoltre è riportata una banda d'errore verticale, che ci indica l'incertezza del dato, dovuta alla distorsione della griglia ed all'errore di interpolazione. La banda è infatti proporzionale al valore del dato, ed è funzione sia all'altezza della riblet della configurazione e sia dell'errore che si ottiene nell'interpolazione dei dati per il calcolo dello sforzo.

maggiori rispetto a quelli negativi. Si osserva come, per questa configurazione di riblet, si raggiunga inoltre una riduzione di resistenza di circa il 20%, che risulta essere, da un confronto con i dati presenti in letteratura, un ottimo risultato, vicino alle performance massime ottenute per le riblet.

Osservando inoltre gli altri dati riportati, si osserva come, per alcuni casi invece di avere una riduzione della resistenza, si abbia un incremento. Se ciò potrebbe essere valido per la configurazione *H3*, dove, eventualmente l'effetto dell'aumento di superficie bagnata supera quello dell'effetto sulle strutture dello strato limite, avendo un'altezza di riblet abbastanza piccola, è molto più probabile che alcuni dati, come quelli della configurazione *H7*, siano affetti da errore, dovuto alla discretizzazione numerica che il programma ha implicitamente nel suo funzionamento, oltre che alla distorsione della griglia a causa delle riblet, e dall'interpolazione numerica effettuata. Infatti, soprattutto per altezze di riblet ridotte, si osserva come i valori di riduzione della resistenza, siano altalenanti, trovandoci casi con altezza di riblet maggiori, che riducono la resistenza meno rispetto ad altezze minori, seppur l'andamento globale ci mostra come l'aumentare dell'altezza delle riblet riduca la resistenza.

A tal proposito, infatti, anche i dati particolarmente positivi, come quello ottimale, trovato per la configurazione *H22*, potrebbe essere influenzato da errori numerici, dovuti alla discretizzazione delle simulazioni numeriche, all'interpolazione effettuata per il calcolo della resistenza ed alla deformazione eccessiva della griglia, come riportato nella sezione successiva (6). In merito all'errore di interpolazione, se, si osserva la figura 5.34, si nota subito come, in questa, a differenza per esempio della 5.31, la retta in-

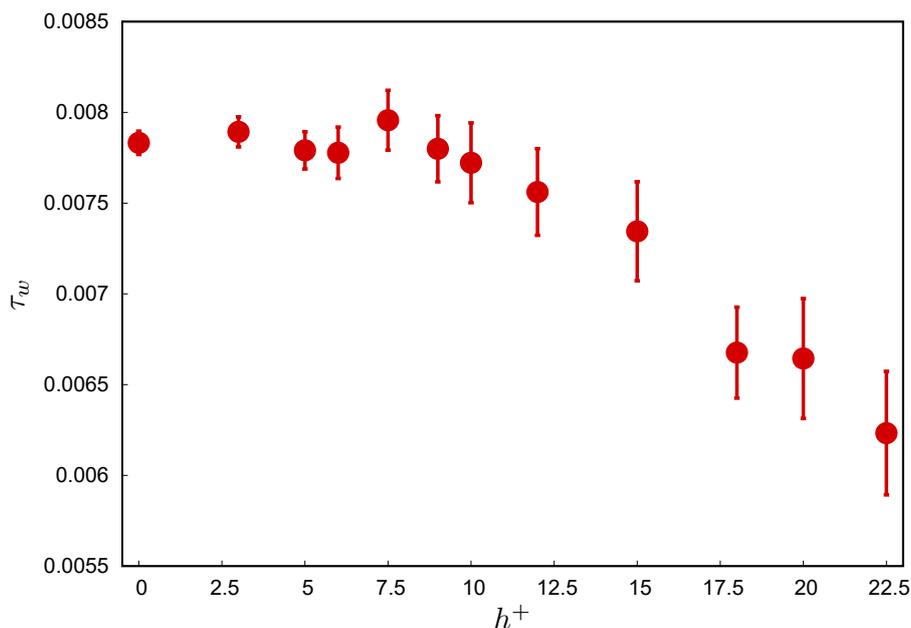


Figura 5.36: Grafico di $h^+ - C_f$, riportante i dati ottenuti per tutte le configurazioni analizzate. Inoltre è riportata una banda d'errore verticale, che ci indica l'incertezza del dato, dovuta alla distorsione della griglia ed all'errore di interpolazione. La banda è infatti proporzionale al valore del dato, ed è funzione sia all'altezza della riblet della configurazione e sia dell'errore che si ottiene nell'interpolazione dei dati per il calcolo dello sforzo.

terpolante ha più difficoltà nel seguire l'andamento della curva di resistenza totale, in quanto questa, da un andamento quasi rettilineo che ha per le configurazioni a riblet assenti o molto basse, passa ad un andamento che presenta una rapida crescita iniziale, un punto di massimo, e poi un decremento, questo sì, con un andamento lineare. Con ciò si vuole intendere come i dati che si sono ottenuti per le configurazioni con riblet più alte, possono essere affetti da errori numerici più elevati, rispetto a quelli con riblet più piccole; ciò lo si è osservato infatti, anche quando si è andati a fare le interpolazioni, le quali, oltre a riportare il coefficiente interpolante, andavano a riportare anche una percentuale di incertezza del risultato, e questa risultava essere crescente al crescere dell'altezza delle riblet; anche quindi questo valore di errore di interpolazione, oltre quello dovuto alla deformazione della griglia, è stato tenuto in conto nel riportare le bande di errore verticale nelle figure 5.35 e 5.36.

I dati riportati nella tabella, in particolare gli andamenti di τ_w e di C_f , sono appunto osservabili anche nei grafici 5.35 e 5.36. In questi, così come nella tabella, è possibile osservare ora la curva di riduzione della resistenza, oltre alla presenza di bande verticali che ci ricordano la possibile presenza di errore nei dati, osservando come, per altezze di riblet ridotte, la resistenza si riduce poco (trascurando i punti in cui si ha addirittura un incremento, causato dall'errore numerico); invece, quando le altezze diventano significative, e si raggiunge il valore di $AR = 1$ (configurazione *H15*), la diminuzione di resistenza diventa significativa, arrivando, come visto, fino al 20% per la configurazione *H22*, tenendo sempre presente il margine di incertezza.

Capitolo 6

Conclusioni finali

Alla fine di questo lavoro di tesi, si vanno a riportare i risultati ottenuti in questo studio, ed a trarre le relative conclusioni. Si è studiati, come tecnica per ridurre la resistenza di un flusso turbolento, la tecnica delle riblet. Le riblet sono una soluzione passiva, ovvero che non richiede energia o un sistema di controllo per funzionare, che va a modificare le strutture turbolente che si hanno in un flusso turbolento, al fine di ridurre la resistenza. In particolare va ad agire sul sottostrato viscoso, riducendo particolarmente la componente di resistenza d'attrito, più che quella inerziale, al quale, però, come si è visto, va comunque a calare leggermente. In questo studio si sono analizzate, attraverso simulazioni numeriche dirette, 12 configurazioni di riblet, dove il parametro che cambia per ognuna è l'altezza delle riblet, espressa in unità di parete. Ciò ci permette di affermare, che i risultati trovati per una delle configurazioni analizzate, dovrebbero essere trasportabili anche a riblet di altezza differente, a patto di avere un numero di Reynolds in unità di parete, Re_τ , tale che da avere la stessa h^+ . Dalle simulazioni effettuati si è trovato il campo di moto che si instaura per un flusso turbolento che scorre in un canale piano sulle cui pareti sono state realizzate riblet. Si è andati a riportare sezioni di questo campo di moto, per avere una prima idea di come il flusso cambia per le diverse configurazioni di riblet analizzate, e dopo di che, dal campo di moto, si sono estratti le medie su piani normali all'asse y , ovvero paralleli alle pareti, delle grandezze di nostro interesse, ovvero la velocità del flusso U , e le fluttuazioni di velocità, u' , v' e w' . Con questi dati, si è visto come le grandezze caratterizzanti la turbolenza cambiano al crescere dell'altezza delle riblet, e si è visti come anche le componenti della resistenza di un flusso turbolento cambiano, e nello specifico si riducono, per la presenza delle riblet. Infine, attraverso la realizzazione di grafici riportati le componenti della resistenza del flusso turbolento, si è andati a calcolare lo sforzo di taglio a parete, da questo il coefficiente d'attrito adimensionale, e si è potuti andare a valutare la riduzione di resistenza che ogni configurazione di riblet offre.

Si è così trovati, che tra le configurazioni analizzate, quella che dà la riduzione più accentuata, è $H22$, ovvero quella con l'altezza di riblet maggiore, tenendo però sempre presente l'incertezza dei dati dovuta agli errori numerici. All'inizio dello studio si era detto che si voleva cercare quale configurazione di riblet offriva la riduzione di resistenza più alta in assoluto, ed avendo noi trovato che è quella con l'altezza massima tra quelle analizzate, non si può essere sicuri che si tratti di quest'altezza. Infatti l'andamento della resistenza in funzione dell'altezza delle riblet ci mostra come, aumentando l'altezza la resistenza cala; non avendo però trovato un valore minimo di resistenza,

non si è trovati una configurazione che da il valore di resistenza più basso in assoluto. Tuttavia, si può comunque affermare che sarebbe stato inutile, se non anche molto problematico, andare ad analizzare configurazioni più estreme, in termini di altezza delle riblet.

Il primo motivo lo si trova nel tipo di mesh analizzato. Infatti, nello studio che si è svolti, il punto di partenza è stato quello, che per le riblet triangolari, la massima riduzione della resistenza, la si ha per $AR = 1$, che nel nostro caso corrisponderebbe alla configurazione $H15$. Tuttavia, si è poi visto, che la configurazione che offre la minima resistenza, dipende dal numero di Reynolds del flusso che si analizza [25]; quindi è possibile che per determinati flusso, si abbia una riduzione maggiore, per $AR > 1$. Ma, per lo studio che si è affrontati, la griglia è stata strutturata per simulare riblet di piccole dimensioni; data la dimensione del blocco di griglia vicino a parete, si ha che questo è alto 0,25, in termini adimensionali. Ciò significa, che si potrebbe arrivare ad avere riblet di altezza, in unità di parete, fino ad $h^+ = 50$, ma questo significherebbe che il blocco di parete, di forma tetragonale, andrebbe a collassare, avendo 2 coppie di punti che si troverebbero a coincidere, e ciò renderebbe impossibile eseguire le simulazioni; infatti già per le configurazioni $H20$ e $H22$, per poterle svolgere, è stato necessario imporre alle simulazioni dei correttori per le direzioni non ortogonali, che hanno permesso lo svolgimento delle simulazioni, ma ne hanno quasi raddoppiato il tempo di calcolo; probabilmente spingersi ancora più su con l'altezza delle riblet, causerebbe problemi ancora maggiori alle simulazioni, che non sarebbero aggirabili.

Una seconda motivazione, riguarda invece l'applicabilità della soluzione. Se è vero, infatti, che con le configurazioni di riblet ad altezza maggiore si ottengono i risultati migliori in termini di riduzione della resistenza, si deve poi andare a valutare se queste configurazioni siano realmente applicabili. Se infatti, dal punto di vista teorico-sperimentale, realizzare superfici con riblet (ovvero scanalature sulle pareti) molto alte può portare a ottimi risultati, dal punto di vista applicativo, bisogna valutare i possibili problemi strutturali (legati alla presenza di scanalature in una sola direzione, e come questo incide sulle matrici strutturali delle superfici su cui sono applicate) ed i problemi legati all'aumento di peso, che si potrebbero avere, se si andasse ad aggiungere superfici scanalate, con scanalature molto alte.

Si deve tenere anche conto, che anche le limitazioni fornite dal programma possono aver influito sulla qualità e la veridicità dei risultati ottenuti, in quanto, se è vero, come si è visto, che i dati per il canale piano rispecchiano molto bene quelli delle DNS di *Kim, Moser & Mansour*, questo vale per il caso di canale piano, senza riblet. Negli altri casi, invece, è anche possibile, che le condizioni al contorno, per come il programma ci obbligava di imporle, abbiano fatto sì di trascurare alcuni aspetti, che invece, in una situazione reale, avrebbero portato determinate configurazioni di riblet, tra quelle analizzate, ad avere prestazioni peggiori, quale per esempio, l'aumento di superficie bagnata, dovuta alle riblet. L'imposizione di una velocità media, applicata come gradiente di pressione al flusso, invece che di un gradiente di pressione diretto, per esempio, può aver in qualche modo influenzato il comportamento del flusso, per configurazioni di riblet ad altezza elevata.

Inoltre, questi risultati sono stati ottenuti solamente per via numerica, e non sono stati verificati per via sperimentale. Quindi è necessario, prima di poter affermare con certezza assoluta quale sia o meno la configurazione migliore, verificare le configurazioni analizzate, anche in via sperimentale. Ovviamente i risultati ottenuti qui ottenuti possono essere una buona base per studi sperimentali successivi, in cui utilizzare i ri-

sultati qui ottenuti, come base di partenza, e sapere in anticipo da quali configurazioni aspettarsi risultati migliori, e capire anche in anticipo, quali configurazioni studiare e quali no, per avere studi sperimentali più mirati, in modo tale da ridurne i costi, non considerando configurazioni dalle prestazioni peggiori, e concentrandosi maggiormente su quelle, che dai dati numerici, risultano essere più promettenti.

Oltre a ciò, si deve sempre tener conto, che trattandosi di simulazioni numeriche, effettuate, soprattutto quelle con configurazioni di riblet molto alte, con griglie di calcolo abbastanza distorte, il che genera un certo errore numerico, a cui poi si somma quello dovuto alla discretizzazione della mesh, e quello dell'interpolazione lineare della resistenza totale.

In conclusione, si può affermare che l'obiettivo che ci si era inizialmente posto, in questo studio, quello di trovare una configurazione di riblet, la quale fosse, in modo assoluto quella che offriva la maggior riduzione di resistenza, non è stato raggiunto, considerando però che, un obiettivo del genere risulta essere quasi impossibile da raggiungere, considerando i numerosi parametri che entrano in gioco nell'andare a definire le performance delle riblet, pur tenendone costanti la forma e la spaziatura tra una riblet ed un'altra. Si è però comunque data una buona indicazione da seguire, per studi futuri, al fine di raggiungere questo difficile obiettivo.

Per quanto riguarda, invece, lo scopo più realistico di questo lavoro di tesi, ovvero andare a studiare il diverso comportamento di un flusso turbolento, su riblet triangolari, di diversa altezza, al fine di capire come queste andassero a ridurre la resistenza del flusso, lo si è raggiunto in pieno. Infatti si è sia compresi quelli che sono i fenomeni analizzati che sono alla base della riduzione della resistenza, e si è anche trovata un'indicazione in merito all'effetto dell'incremento di altezza delle riblet sulla resistenza, ovvero, che questa cala all'aumentare dell'altezza delle riblet, perché queste, aumentando in altezza, vanno ad incidere maggiormente sul sottostrato viscoso di un flusso turbolento, modificando il comportamento delle strutture turbolente di rimescolamento, le quali sono una delle cause della resistenza del flusso turbolento.

Grazie a ciò, quindi, questo studio può essere considerato un'ottima base di partenza per eventuali studi futuri, che dovranno però combinare simulazioni numeriche e prove sperimentali, che avranno lo scopo di perfezionare sempre più la tecnica delle riblet, al fine di trovarne la configurazione migliore.

Appendici

Appendice A

Codici simulazioni numeriche

Nelle appendici, questa e la successiva, si riportano i codici di programmazione che sono stati utilizzati in questo studio di tesi, sia quelli per *OpenFOAM*, che altri scritti in *Fortran*, per altri motivi. Nello specifico, in questa prima sezione dell'appendice, si riportano i codici che sono stati utilizzati per preparare le simulazioni svolte, e quelli utilizzati per la loro esecuzione. Nell'altra appendice (B), invece, si riportano i codici utilizzati per la parte di post processing dei dati, ed anche quelli per il loro riordino ed utilizzo. Si riporta una sola versione del codice, ovvero quello realizzato per una sola simulazione (quella della configurazione *H7*, nello specifico), in quanto tra le differenze tra le varie simulazioni sono riportate nel dettaglio nel capitolo 4.

Vengono di seguito riportati, inizialmente, i codici utilizzati per la realizzazione di una simulazione su *OpenFOAM*, specificando, per ogni codice riportato, dove deve essere collocato all'interno di una simulazione, per funzionare correttamente. In particolare abbiamo, che una simulazione su *OpenFOAM* è rappresentata dalla sua cartella principale, all'interno della quale, devono essere presenti (almeno per le simulazioni da noi svolte) 3 cartelle fondamentali (in alcuni casi 2, ma perché la terza viene creata mediante la mappatura di un'altra simulazione, vedi 4.2.1, A.3.4 e A.3 relativa sezione), che sono: *system*, *constant* ed una cartella che ha per nome il tempo iniziale della simulazione (che è quella che non sempre è presente, in quanto viene generata in seguito mappando le condizioni iniziali e quelle al contorno da un'altra simulazione). Andremo a riportare, nell'ordine, prima i codici che si trovano nella cartella *system*, poi quelli nella cartella *constant*, ed infine quelli presenti nella cartella del primo istante temporale. Inoltre vengono riportati anche altri 2 codici, che sono uno quello che è stato utilizzato per il calcolo dei parametri della mesh per la direzione *y*, *Newton*, scritto in *Fortran*, e l'altro è quello utilizzato per l'esecuzione delle simulazioni sul calcolatore *HACTAR*, *Submit Run*.

Per quanto riguarda i comandi necessari per l'utilizzo di questi codici e quindi l'esecuzione delle simulazioni, fare riferimento alle relative sezioni nel capitolo 4, nello specifico per questa appendice, quelle riportate nella sezione 4.2, e nel file con il codice utilizzato per l'esecuzione delle simulazioni sul calcolatore *HACTAR*, riportato in questa appendice, nella sezione A.6.2. Qui i file sono indicati come *.txt*, in quanto è necessario per farli leggere da *L^AT_EX*. Tuttavia, invece, per essere letti da *OpenFOAM*, è necessario che siano senza estensione.

A.1 *blockMeshDict*

Il primo codice che si trova nella cartella *system*, ed anche uno dei più importanti, è il codice *blockMeshDict*. In questo codice vengono infatti riportate tutte le informazioni relative alla mesh del problema che si vuole analizzare, dalle dimensioni del dominio, la forma che questo ha, il numero di celle con cui discretizzarlo, e le condizioni al contorno. Proprio per quanto riguarda le condizioni al contorno, nel nostro caso, si sono utilizzate 2 versioni di questo codice, una con le condizioni al contorno per il caso iniziale (quello con flusso in ingresso e condizione di uscita) ed un'altra con condizioni periodiche. Qui se ne riporta solamente una versione interamente, in quanto le differenze tra le 2 versioni sono davvero poche, ed alla fine, si riportano quali sono le righe della prima versione che devono essere sostituite, per ottenere la seconda, di cui si riportano appunto solo le righe che sono differenti.

Codici_simulazione/blockMeshDict_iniz.txt

```

1 /*-----* C++ *-----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: 2.2.0 |
5 | \ \ / A n d | Web: http://www.OpenFOAM.org |
6 | \ \ / M a n i p u l a t i o n |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        blockMeshDict;
14 }
15 // * * * * *
16 convertToMeters 1;
17 // Definizione delle grandezze parametriche
18 // Dimensioni, in valore adimensionalizzato
19 h 1.0; // Semi altezza canale, grandezza di adimensionalizzazione
20 L #calc "9.0*$h"; // Lunghezza del canale
21 H #calc "2.0*$h"; // Altezza del canale
22 B #calc "4.5*$h"; // Larghezza del canale
23 Nrib 30.0; // Numero totale di riblet
24
25 y12 #calc "$h*0.25"; // zona griglia vicino pareti, con mesh piu raffinata
    inferiore
26 y23 #calc "$h*0.5"; // zona griglia intermedia, con mesh meno raffinata
    inferiore
27 y45 #calc "$H - $y23"; // zona griglia intermedia, con mesh meno raffinata
    superiore
28 y56 #calc "$H - $y12"; //zona griglia vicino pareti, con mesh piu raffinata
    superiore
29
30 Nx 150; // Numero di celle lungo x per tutti i blocchi
31 Ny16 27; // Numero di celle lungo y, per i blocchi vicino alla parete,
32 Ny25 14; // Numero di celle lungo y ; per i blocchi intermedi
33 Ny34 23; // Numero di celle lungo y ; per i blocchi al centro
34 Nz 240.0; // Numero di celle lungo z per tutti i blocchi
35 Nzrib #calc "$Nz/(2*$Nrib)"; // Numero di celle lungo z per ogni semi riblet
36
37 // dymin = 0.005
38 Ry1 3.0; // Rapporto del mesh grid lungo y, per il blocco inferiore vicino
    alla parete
39 Ry2 1.5; // Rapporto del mesh grid lungo y, per il blocco inferiore
    intermedio
40 Ry5 #calc "1.0/$Ry2"; // Rapporto del mesh grid lungo y, per il blocco superiore
    intermedio
41 Ry6 #calc "1.0/$Ry1"; // Rapporto del mesh grid lungo y, per il blocco superiore
    vicino alla parete
42 Rzdx 1.25; // Rapporto del mesh grid lungo z, per i blocchi destri delle rib.
    quelli discendenti

```

```

43 Rzsx #calc "1.0/$Rzdx"; // Rapporto del mesh grid lungo z, per i blocchi
    sinistri delle rib. quelli ascendenti
44
45 br 0.15; // Base della riblet
46 // rapp hr su br/2 = 1/2
47 hr 0.0375; // Altezza della punta della riblet
48 //hr+ = 7.5
49 zr #calc "$br/2"; // Semi apertura della riblet
50 Hr #calc "$H - $hr"; // Posizione della punta della riblet superiore
51
52 //Definizione dei valori delle coordinate z delle riblet
53 // Cordinate di apertura delle riblet
54 zpr1 #calc "1*$br - $zr"; // Posizione longitudinale punta rib 1.
55 zvr1 #calc "1*$br"; // Posizione longitudinale valle rib 1.
56 zpr2 #calc "2*$br - $zr"; // Posizione longitudinale punta rib 2.
57 zvr2 #calc "2*$br"; // Posizione longitudinale valle rib 2.
58 zpr3 #calc "3*$br - $zr"; // Posizione longitudinale punta rib 3.
59 zvr3 #calc "3*$br"; // Posizione longitudinale valle rib 3.
60 zpr4 #calc "4*$br - $zr"; // Posizione longitudinale punta rib 4.
61 zvr4 #calc "4*$br"; // Posizione longitudinale valle rib 4.
62 zpr5 #calc "5*$br - $zr"; // Posizione longitudinale punta rib 5.
63 zvr5 #calc "5*$br"; // Posizione longitudinale valle rib 5.
64 zpr6 #calc "6*$br - $zr"; // Posizione longitudinale punta rib 6.
65 zvr6 #calc "6*$br"; // Posizione longitudinale valle rib 6.
66 zpr7 #calc "7*$br - $zr"; // Posizione longitudinale punta rib 7.
67 zvr7 #calc "7*$br"; // Posizione longitudinale valle rib 7.
68 zpr8 #calc "8*$br - $zr"; // Posizione longitudinale punta rib 8.
69 zvr8 #calc "8*$br"; // Posizione longitudinale valle rib 8.
70 zpr9 #calc "9*$br - $zr"; // Posizione longitudinale punta rib 9.
71 zvr9 #calc "9*$br"; // Posizione longitudinale valle rib 9.
72 zpr10 #calc "10*$br - $zr"; // Posizione longitudinale punta rib 10.
73 zvr10 #calc "10*$br"; // Posizione longitudinale valle rib 10.
74 zpr11 #calc "11*$br - $zr"; // Posizione longitudinale punta rib 11.
75 zvr11 #calc "11*$br"; // Posizione longitudinale valle rib 11.
76 zpr12 #calc "12*$br - $zr"; // Posizione longitudinale punta rib 12.
77 zvr12 #calc "12*$br"; // Posizione longitudinale valle rib 12.
78 zpr13 #calc "13*$br - $zr"; // Posizione longitudinale punta rib 13.
79 zvr13 #calc "13*$br"; // Posizione longitudinale valle rib 13.
80 zpr14 #calc "14*$br - $zr"; // Posizione longitudinale punta rib 14.
81 zvr14 #calc "14*$br"; // Posizione longitudinale valle rib 14.
82 zpr15 #calc "15*$br - $zr"; // Posizione longitudinale punta rib 15.
83 zvr15 #calc "15*$br"; // Posizione longitudinale valle rib 15.
84 zpr16 #calc "16*$br - $zr"; // Posizione longitudinale punta rib 16.
85 zvr16 #calc "16*$br"; // Posizione longitudinale valle rib 16.
86 zpr17 #calc "17*$br - $zr"; // Posizione longitudinale punta rib 17.
87 zvr17 #calc "17*$br"; // Posizione longitudinale valle rib 17.
88 zpr18 #calc "18*$br - $zr"; // Posizione longitudinale punta rib 18.
89 zvr18 #calc "18*$br"; // Posizione longitudinale valle rib 18.
90 zpr19 #calc "19*$br - $zr"; // Posizione longitudinale punta rib 19.
91 zvr19 #calc "19*$br"; // Posizione longitudinale valle rib 19.
92 zpr20 #calc "20*$br - $zr"; // Posizione longitudinale punta rib 20.
93 zvr20 #calc "20*$br"; // Posizione longitudinale valle rib 20.
94 zpr21 #calc "21*$br - $zr"; // Posizione longitudinale punta rib 21.
95 zvr21 #calc "21*$br"; // Posizione longitudinale valle rib 21.
96 zpr22 #calc "22*$br - $zr"; // Posizione longitudinale punta rib 22.
97 zvr22 #calc "22*$br"; // Posizione longitudinale valle rib 22.
98 zpr23 #calc "23*$br - $zr"; // Posizione longitudinale punta rib 23.
99 zvr23 #calc "23*$br"; // Posizione longitudinale valle rib 23.
100 zpr24 #calc "24*$br - $zr"; // Posizione longitudinale punta rib 24.
101 zvr24 #calc "24*$br"; // Posizione longitudinale valle rib 24.
102 zpr25 #calc "25*$br - $zr"; // Posizione longitudinale punta rib 25.
103 zvr25 #calc "25*$br"; // Posizione longitudinale valle rib 25.
104 zpr26 #calc "26*$br - $zr"; // Posizione longitudinale punta rib 26.
105 zvr26 #calc "26*$br"; // Posizione longitudinale valle rib 26.
106 zpr27 #calc "27*$br - $zr"; // Posizione longitudinale punta rib 27.
107 zvr27 #calc "27*$br"; // Posizione longitudinale valle rib 27.
108 zpr28 #calc "28*$br - $zr"; // Posizione longitudinale punta rib 28.
109 zvr28 #calc "28*$br"; // Posizione longitudinale valle rib 28.
110 zpr29 #calc "29*$br - $zr"; // Posizione longitudinale punta rib 29.

```

```
111 zvr29 #calc "29*$br"; // Posizione longitudinale valle rib 29.
112 zpr30 #calc "30*$br - $zr"; // Posizione longitudinale punta rib 30.
113 zvr30 #calc "30*$br"; // Posizione longitudinale valle rib 30.
114 // il valore della zvr dell ultima riblet coincide con il valore della larghezza
    della base
115
116 // Definizione dei punti della mesh
117 vertices
118 (
119 //faccia iniziale 0 valle, punti da 0 a 13
120 (0 0 0)
121 ($L 0 0)
122 (0 $y12 0)
123 ($L $y12 0)
124 (0 $y23 0)
125 ($L $y23 0)
126 (0 $h 0)
127 ($L $h 0)
128 (0 $y45 0)
129 ($L $y45 0)
130 (0 $y56 0)
131 ($L $y56 0)
132 (0 $H 0)
133 ($L $H 0)
134
135 // riblet 1
136 //faccia punta, punti da 14 a 27
137 (0 $hr $zpr1)
138 ($L $hr $zpr1)
139 (0 $y12 $zpr1)
140 ($L $y12 $zpr1)
141 (0 $y23 $zpr1)
142 ($L $y23 $zpr1)
143 (0 $h $zpr1)
144 ($L $h $zpr1)
145 (0 $y45 $zpr1)
146 ($L $y45 $zpr1)
147 (0 $y56 $zpr1)
148 ($L $y56 $zpr1)
149 (0 $Hr $zpr1)
150 ($L $Hr $zpr1)
151
152 //faccia valle, punti da 28 a 41
153 (0 0 $zvri)
154 ($L 0 $zvri)
155 (0 $y12 $zvri)
156 ($L $y12 $zvri)
157 (0 $y23 $zvri)
158 ($L $y23 $zvri)
159 (0 $h $zvri)
160 ($L $h $zvri)
161 (0 $y45 $zvri)
162 ($L $y45 $zvri)
163 (0 $y56 $zvri)
164 ($L $y56 $zvri)
165 (0 $H $zvri)
166 ($L $H $zvri)
167
168 // riblet 2
169 //faccia punta, punti da 42 a 55
170 (0 $hr $zpr2)
171 ($L $hr $zpr2)
172 (0 $y12 $zpr2)
173 ($L $y12 $zpr2)
174 (0 $y23 $zpr2)
175 ($L $y23 $zpr2)
176 (0 $h $zpr2)
177 ($L $h $zpr2)
178 (0 $y45 $zpr2)
```

```
179 ($L $y45 $zpr2)
180 (0 $y56 $zpr2)
181 ($L $y56 $zpr2)
182 (0 $Hr $zpr2)
183 ($L $Hr $zpr2)
184
185 //faccia valle, punti da 56 a 69
186 (0 0 $zvr2)
187 ($L 0 $zvr2)
188 (0 $y12 $zvr2)
189 ($L $y12 $zvr2)
190 (0 $y23 $zvr2)
191 ($L $y23 $zvr2)
192 (0 $h $zvr2)
193 ($L $h $zvr2)
194 (0 $y45 $zvr2)
195 ($L $y45 $zvr2)
196 (0 $y56 $zvr2)
197 ($L $y56 $zvr2)
198 (0 $H $zvr2)
199 ($L $H $zvr2)
200
201 // riblet 3
202 //faccia punta, punti da 70 a 83
203 (0 $hr $zpr3)
204 ($L $hr $zpr3)
205 (0 $y12 $zpr3)
206 ($L $y12 $zpr3)
207 (0 $y23 $zpr3)
208 ($L $y23 $zpr3)
209 (0 $h $zpr3)
210 ($L $h $zpr3)
211 (0 $y45 $zpr3)
212 ($L $y45 $zpr3)
213 (0 $y56 $zpr3)
214 ($L $y56 $zpr3)
215 (0 $Hr $zpr3)
216 ($L $Hr $zpr3)
217
218 //faccia valle, punti da 84 a 97
219 (0 0 $zvr3)
220 ($L 0 $zvr3)
221 (0 $y12 $zvr3)
222 ($L $y12 $zvr3)
223 (0 $y23 $zvr3)
224 ($L $y23 $zvr3)
225 (0 $h $zvr3)
226 ($L $h $zvr3)
227 (0 $y45 $zvr3)
228 ($L $y45 $zvr3)
229 (0 $y56 $zvr3)
230 ($L $y56 $zvr3)
231 (0 $H $zvr3)
232 ($L $H $zvr3)
233
234 // riblet 4
235 //faccia punta, punti da 98 a 111
236 (0 $hr $zpr4)
237 ($L $hr $zpr4)
238 (0 $y12 $zpr4)
239 ($L $y12 $zpr4)
240 (0 $y23 $zpr4)
241 ($L $y23 $zpr4)
242 (0 $h $zpr4)
243 ($L $h $zpr4)
244 (0 $y45 $zpr4)
245 ($L $y45 $zpr4)
246 (0 $y56 $zpr4)
247 ($L $y56 $zpr4)
```

```
248 (0 $Hr $zpr4)
249 ($L $Hr $zpr4)
250
251 //faccia valle, punti da 112 a 125
252 (0 0 $zvr4)
253 ($L 0 $zvr4)
254 (0 $y12 $zvr4)
255 ($L $y12 $zvr4)
256 (0 $y23 $zvr4)
257 ($L $y23 $zvr4)
258 (0 $h $zvr4)
259 ($L $h $zvr4)
260 (0 $y45 $zvr4)
261 ($L $y45 $zvr4)
262 (0 $y56 $zvr4)
263 ($L $y56 $zvr4)
264 (0 $H $zvr4)
265 ($L $H $zvr4)
266
267 // riblet 5
268 //faccia punta, punti da 126 a 139
269 (0 $hr $zpr5)
270 ($L $hr $zpr5)
271 (0 $y12 $zpr5)
272 ($L $y12 $zpr5)
273 (0 $y23 $zpr5)
274 ($L $y23 $zpr5)
275 (0 $h $zpr5)
276 ($L $h $zpr5)
277 (0 $y45 $zpr5)
278 ($L $y45 $zpr5)
279 (0 $y56 $zpr5)
280 ($L $y56 $zpr5)
281 (0 $Hr $zpr5)
282 ($L $Hr $zpr5)
283
284 //faccia valle, punti da 140 a 153
285 (0 0 $zvr5)
286 ($L 0 $zvr5)
287 (0 $y12 $zvr5)
288 ($L $y12 $zvr5)
289 (0 $y23 $zvr5)
290 ($L $y23 $zvr5)
291 (0 $h $zvr5)
292 ($L $h $zvr5)
293 (0 $y45 $zvr5)
294 ($L $y45 $zvr5)
295 (0 $y56 $zvr5)
296 ($L $y56 $zvr5)
297 (0 $H $zvr5)
298 ($L $H $zvr5)
299
300 // riblet 6
301 //faccia punta, punti da 154 a 167
302 (0 $hr $zpr6)
303 ($L $hr $zpr6)
304 (0 $y12 $zpr6)
305 ($L $y12 $zpr6)
306 (0 $y23 $zpr6)
307 ($L $y23 $zpr6)
308 (0 $h $zpr6)
309 ($L $h $zpr6)
310 (0 $y45 $zpr6)
311 ($L $y45 $zpr6)
312 (0 $y56 $zpr6)
313 ($L $y56 $zpr6)
314 (0 $Hr $zpr6)
315 ($L $Hr $zpr6)
316
```

```
317 //faccia valle, punti da 168 a 181
318 (0 0 $zvr6)
319 ($L 0 $zvr6)
320 (0 $y12 $zvr6)
321 ($L $y12 $zvr6)
322 (0 $y23 $zvr6)
323 ($L $y23 $zvr6)
324 (0 $h $zvr6)
325 ($L $h $zvr6)
326 (0 $y45 $zvr6)
327 ($L $y45 $zvr6)
328 (0 $y56 $zvr6)
329 ($L $y56 $zvr6)
330 (0 $H $zvr6)
331 ($L $H $zvr6)
332
333 // riblet 7
334 //faccia punta, punti da 182 a 195
335 (0 $hr $zpr7)
336 ($L $hr $zpr7)
337 (0 $y12 $zpr7)
338 ($L $y12 $zpr7)
339 (0 $y23 $zpr7)
340 ($L $y23 $zpr7)
341 (0 $h $zpr7)
342 ($L $h $zpr7)
343 (0 $y45 $zpr7)
344 ($L $y45 $zpr7)
345 (0 $y56 $zpr7)
346 ($L $y56 $zpr7)
347 (0 $Hr $zpr7)
348 ($L $Hr $zpr7)
349
350 //faccia valle, punti da 196 a 209
351 (0 0 $zvr7)
352 ($L 0 $zvr7)
353 (0 $y12 $zvr7)
354 ($L $y12 $zvr7)
355 (0 $y23 $zvr7)
356 ($L $y23 $zvr7)
357 (0 $h $zvr7)
358 ($L $h $zvr7)
359 (0 $y45 $zvr7)
360 ($L $y45 $zvr7)
361 (0 $y56 $zvr7)
362 ($L $y56 $zvr7)
363 (0 $H $zvr7)
364 ($L $H $zvr7)
365
366 // riblet 8
367 //faccia punta, punti da 210 a 223
368 (0 $hr $zpr8)
369 ($L $hr $zpr8)
370 (0 $y12 $zpr8)
371 ($L $y12 $zpr8)
372 (0 $y23 $zpr8)
373 ($L $y23 $zpr8)
374 (0 $h $zpr8)
375 ($L $h $zpr8)
376 (0 $y45 $zpr8)
377 ($L $y45 $zpr8)
378 (0 $y56 $zpr8)
379 ($L $y56 $zpr8)
380 (0 $Hr $zpr8)
381 ($L $Hr $zpr8)
382
383 //faccia valle, punti da 224 a 237
384 (0 0 $zvr8)
385 ($L 0 $zvr8)
```

```
386 (0 $y12 $zvr8)
387 ($L $y12 $zvr8)
388 (0 $y23 $zvr8)
389 ($L $y23 $zvr8)
390 (0 $h $zvr8)
391 ($L $h $zvr8)
392 (0 $y45 $zvr8)
393 ($L $y45 $zvr8)
394 (0 $y56 $zvr8)
395 ($L $y56 $zvr8)
396 (0 $H $zvr8)
397 ($L $H $zvr8)
398
399 // riblet 9
400 //faccia punta, punti da 238 a 251
401 (0 $hr $zpr9)
402 ($L $hr $zpr9)
403 (0 $y12 $zpr9)
404 ($L $y12 $zpr9)
405 (0 $y23 $zpr9)
406 ($L $y23 $zpr9)
407 (0 $h $zpr9)
408 ($L $h $zpr9)
409 (0 $y45 $zpr9)
410 ($L $y45 $zpr9)
411 (0 $y56 $zpr9)
412 ($L $y56 $zpr9)
413 (0 $Hr $zpr9)
414 ($L $Hr $zpr9)
415
416 //faccia valle, punti da 252 a 265
417 (0 0 $zvr9)
418 ($L 0 $zvr9)
419 (0 $y12 $zvr9)
420 ($L $y12 $zvr9)
421 (0 $y23 $zvr9)
422 ($L $y23 $zvr9)
423 (0 $h $zvr9)
424 ($L $h $zvr9)
425 (0 $y45 $zvr9)
426 ($L $y45 $zvr9)
427 (0 $y56 $zvr9)
428 ($L $y56 $zvr9)
429 (0 $H $zvr9)
430 ($L $H $zvr9)
431
432 // riblet 10
433 //faccia punta, punti da 266 a 279
434 (0 $hr $zpr10)
435 ($L $hr $zpr10)
436 (0 $y12 $zpr10)
437 ($L $y12 $zpr10)
438 (0 $y23 $zpr10)
439 ($L $y23 $zpr10)
440 (0 $h $zpr10)
441 ($L $h $zpr10)
442 (0 $y45 $zpr10)
443 ($L $y45 $zpr10)
444 (0 $y56 $zpr10)
445 ($L $y56 $zpr10)
446 (0 $Hr $zpr10)
447 ($L $Hr $zpr10)
448
449 //faccia valle, punti da 280 a 293
450 (0 0 $zvr10)
451 ($L 0 $zvr10)
452 (0 $y12 $zvr10)
453 ($L $y12 $zvr10)
454 (0 $y23 $zvr10)
```

```
455 ($L $y23 $zvr10)
456 (0 $h $zvr10)
457 ($L $h $zvr10)
458 (0 $y45 $zvr10)
459 ($L $y45 $zvr10)
460 (0 $y56 $zvr10)
461 ($L $y56 $zvr10)
462 (0 $H $zvr10)
463 ($L $H $zvr10)
464
465 // riblet 11
466 //faccia punta, punti da 294 a 307
467 (0 $hr $zpr11)
468 ($L $hr $zpr11)
469 (0 $y12 $zpr11)
470 ($L $y12 $zpr11)
471 (0 $y23 $zpr11)
472 ($L $y23 $zpr11)
473 (0 $h $zpr11)
474 ($L $h $zpr11)
475 (0 $y45 $zpr11)
476 ($L $y45 $zpr11)
477 (0 $y56 $zpr11)
478 ($L $y56 $zpr11)
479 (0 $Hr $zpr11)
480 ($L $Hr $zpr11)
481
482 //faccia valle, punti da 308 a 321
483 (0 0 $zvr11)
484 ($L 0 $zvr11)
485 (0 $y12 $zvr11)
486 ($L $y12 $zvr11)
487 (0 $y23 $zvr11)
488 ($L $y23 $zvr11)
489 (0 $h $zvr11)
490 ($L $h $zvr11)
491 (0 $y45 $zvr11)
492 ($L $y45 $zvr11)
493 (0 $y56 $zvr11)
494 ($L $y56 $zvr11)
495 (0 $H $zvr11)
496 ($L $H $zvr11)
497
498 // riblet 12
499 //faccia punta, punti da 322 a 335
500 (0 $hr $zpr12)
501 ($L $hr $zpr12)
502 (0 $y12 $zpr12)
503 ($L $y12 $zpr12)
504 (0 $y23 $zpr12)
505 ($L $y23 $zpr12)
506 (0 $h $zpr12)
507 ($L $h $zpr12)
508 (0 $y45 $zpr12)
509 ($L $y45 $zpr12)
510 (0 $y56 $zpr12)
511 ($L $y56 $zpr12)
512 (0 $Hr $zpr12)
513 ($L $Hr $zpr12)
514
515 //faccia valle, punti da 336 a 349
516 (0 0 $zvr12)
517 ($L 0 $zvr12)
518 (0 $y12 $zvr12)
519 ($L $y12 $zvr12)
520 (0 $y23 $zvr12)
521 ($L $y23 $zvr12)
522 (0 $h $zvr12)
523 ($L $h $zvr12)
```

```
524 (0 $y45 $zvr12)
525 ($L $y45 $zvr12)
526 (0 $y56 $zvr12)
527 ($L $y56 $zvr12)
528 (0 $H $zvr12)
529 ($L $H $zvr12)
530
531 // riblet 13
532 //faccia punta, punti da 350 a 363
533 (0 $hr $zpr13)
534 ($L $hr $zpr13)
535 (0 $y12 $zpr13)
536 ($L $y12 $zpr13)
537 (0 $y23 $zpr13)
538 ($L $y23 $zpr13)
539 (0 $h $zpr13)
540 ($L $h $zpr13)
541 (0 $y45 $zpr13)
542 ($L $y45 $zpr13)
543 (0 $y56 $zpr13)
544 ($L $y56 $zpr13)
545 (0 $Hr $zpr13)
546 ($L $Hr $zpr13)
547
548 //faccia valle, punti da 364 a 377
549 (0 0 $zvr13)
550 ($L 0 $zvr13)
551 (0 $y12 $zvr13)
552 ($L $y12 $zvr13)
553 (0 $y23 $zvr13)
554 ($L $y23 $zvr13)
555 (0 $h $zvr13)
556 ($L $h $zvr13)
557 (0 $y45 $zvr13)
558 ($L $y45 $zvr13)
559 (0 $y56 $zvr13)
560 ($L $y56 $zvr13)
561 (0 $H $zvr13)
562 ($L $H $zvr13)
563
564 // riblet 14
565 //faccia punta, punti da 378 a 391
566 (0 $hr $zpr14)
567 ($L $hr $zpr14)
568 (0 $y12 $zpr14)
569 ($L $y12 $zpr14)
570 (0 $y23 $zpr14)
571 ($L $y23 $zpr14)
572 (0 $h $zpr14)
573 ($L $h $zpr14)
574 (0 $y45 $zpr14)
575 ($L $y45 $zpr14)
576 (0 $y56 $zpr14)
577 ($L $y56 $zpr14)
578 (0 $Hr $zpr14)
579 ($L $Hr $zpr14)
580
581 //faccia valle, punti da 392 a 405
582 (0 0 $zvr14)
583 ($L 0 $zvr14)
584 (0 $y12 $zvr14)
585 ($L $y12 $zvr14)
586 (0 $y23 $zvr14)
587 ($L $y23 $zvr14)
588 (0 $h $zvr14)
589 ($L $h $zvr14)
590 (0 $y45 $zvr14)
591 ($L $y45 $zvr14)
592 (0 $y56 $zvr14)
```

```
593 ($L $y56 $zvr14)
594 (0 $H $zvr14)
595 ($L $H $zvr14)
596
597 // riblet 15
598 //faccia punta, punti da 406 a 419
599 (0 $hr $zpr15)
600 ($L $hr $zpr15)
601 (0 $y12 $zpr15)
602 ($L $y12 $zpr15)
603 (0 $y23 $zpr15)
604 ($L $y23 $zpr15)
605 (0 $h $zpr15)
606 ($L $h $zpr15)
607 (0 $y45 $zpr15)
608 ($L $y45 $zpr15)
609 (0 $y56 $zpr15)
610 ($L $y56 $zpr15)
611 (0 $Hr $zpr15)
612 ($L $Hr $zpr15)
613
614 //faccia valle, punti da 420 a 433
615 (0 0 $zvr15)
616 ($L 0 $zvr15)
617 (0 $y12 $zvr15)
618 ($L $y12 $zvr15)
619 (0 $y23 $zvr15)
620 ($L $y23 $zvr15)
621 (0 $h $zvr15)
622 ($L $h $zvr15)
623 (0 $y45 $zvr15)
624 ($L $y45 $zvr15)
625 (0 $y56 $zvr15)
626 ($L $y56 $zvr15)
627 (0 $H $zvr15)
628 ($L $H $zvr15)
629
630 // riblet 16
631 //faccia punta, punti da 434 a 447
632 (0 $hr $zpr16)
633 ($L $hr $zpr16)
634 (0 $y12 $zpr16)
635 ($L $y12 $zpr16)
636 (0 $y23 $zpr16)
637 ($L $y23 $zpr16)
638 (0 $h $zpr16)
639 ($L $h $zpr16)
640 (0 $y45 $zpr16)
641 ($L $y45 $zpr16)
642 (0 $y56 $zpr16)
643 ($L $y56 $zpr16)
644 (0 $Hr $zpr16)
645 ($L $Hr $zpr16)
646
647 //faccia valle, punti da 448 a 461
648 (0 0 $zvr16)
649 ($L 0 $zvr16)
650 (0 $y12 $zvr16)
651 ($L $y12 $zvr16)
652 (0 $y23 $zvr16)
653 ($L $y23 $zvr16)
654 (0 $h $zvr16)
655 ($L $h $zvr16)
656 (0 $y45 $zvr16)
657 ($L $y45 $zvr16)
658 (0 $y56 $zvr16)
659 ($L $y56 $zvr16)
660 (0 $H $zvr16)
661 ($L $H $zvr16)
```

```
662
663 // riblet 17
664 //faccia punta, punti da 462 a 475
665 (0 $hr $zpr17)
666 ($L $hr $zpr17)
667 (0 $y12 $zpr17)
668 ($L $y12 $zpr17)
669 (0 $y23 $zpr17)
670 ($L $y23 $zpr17)
671 (0 $h $zpr17)
672 ($L $h $zpr17)
673 (0 $y45 $zpr17)
674 ($L $y45 $zpr17)
675 (0 $y56 $zpr17)
676 ($L $y56 $zpr17)
677 (0 $Hr $zpr17)
678 ($L $Hr $zpr17)
679
680 //faccia valle, punti da 476 a 489
681 (0 0 $zv17)
682 ($L 0 $zv17)
683 (0 $y12 $zv17)
684 ($L $y12 $zv17)
685 (0 $y23 $zv17)
686 ($L $y23 $zv17)
687 (0 $h $zv17)
688 ($L $h $zv17)
689 (0 $y45 $zv17)
690 ($L $y45 $zv17)
691 (0 $y56 $zv17)
692 ($L $y56 $zv17)
693 (0 $H $zv17)
694 ($L $H $zv17)
695
696 // riblet 18
697 //faccia punta, punti da 490 a 503
698 (0 $hr $zpr18)
699 ($L $hr $zpr18)
700 (0 $y12 $zpr18)
701 ($L $y12 $zpr18)
702 (0 $y23 $zpr18)
703 ($L $y23 $zpr18)
704 (0 $h $zpr18)
705 ($L $h $zpr18)
706 (0 $y45 $zpr18)
707 ($L $y45 $zpr18)
708 (0 $y56 $zpr18)
709 ($L $y56 $zpr18)
710 (0 $Hr $zpr18)
711 ($L $Hr $zpr18)
712
713 //faccia valle, punti da 504 a 517
714 (0 0 $zv18)
715 ($L 0 $zv18)
716 (0 $y12 $zv18)
717 ($L $y12 $zv18)
718 (0 $y23 $zv18)
719 ($L $y23 $zv18)
720 (0 $h $zv18)
721 ($L $h $zv18)
722 (0 $y45 $zv18)
723 ($L $y45 $zv18)
724 (0 $y56 $zv18)
725 ($L $y56 $zv18)
726 (0 $H $zv18)
727 ($L $H $zv18)
728
729 // riblet 19
730 //faccia punta, punti da 518 a 531
```

```
731 (0 $hr $zpr19)
732 ($L $hr $zpr19)
733 (0 $y12 $zpr19)
734 ($L $y12 $zpr19)
735 (0 $y23 $zpr19)
736 ($L $y23 $zpr19)
737 (0 $h $zpr19)
738 ($L $h $zpr19)
739 (0 $y45 $zpr19)
740 ($L $y45 $zpr19)
741 (0 $y56 $zpr19)
742 ($L $y56 $zpr19)
743 (0 $Hr $zpr19)
744 ($L $Hr $zpr19)
745
746 //faccia valle, punti da 532 a 545
747 (0 0 $zv19)
748 ($L 0 $zv19)
749 (0 $y12 $zv19)
750 ($L $y12 $zv19)
751 (0 $y23 $zv19)
752 ($L $y23 $zv19)
753 (0 $h $zv19)
754 ($L $h $zv19)
755 (0 $y45 $zv19)
756 ($L $y45 $zv19)
757 (0 $y56 $zv19)
758 ($L $y56 $zv19)
759 (0 $H $zv19)
760 ($L $H $zv19)
761
762 // riblet 20
763 //faccia punta, punti da 546 a 559
764 (0 $hr $zpr20)
765 ($L $hr $zpr20)
766 (0 $y12 $zpr20)
767 ($L $y12 $zpr20)
768 (0 $y23 $zpr20)
769 ($L $y23 $zpr20)
770 (0 $h $zpr20)
771 ($L $h $zpr20)
772 (0 $y45 $zpr20)
773 ($L $y45 $zpr20)
774 (0 $y56 $zpr20)
775 ($L $y56 $zpr20)
776 (0 $Hr $zpr20)
777 ($L $Hr $zpr20)
778
779 //faccia valle, punti da 560 a 573
780 (0 0 $zv20)
781 ($L 0 $zv20)
782 (0 $y12 $zv20)
783 ($L $y12 $zv20)
784 (0 $y23 $zv20)
785 ($L $y23 $zv20)
786 (0 $h $zv20)
787 ($L $h $zv20)
788 (0 $y45 $zv20)
789 ($L $y45 $zv20)
790 (0 $y56 $zv20)
791 ($L $y56 $zv20)
792 (0 $H $zv20)
793 ($L $H $zv20)
794
795 // riblet 21
796 //faccia punta, punti da 574 a 587
797 (0 $hr $zpr21)
798 ($L $hr $zpr21)
799 (0 $y12 $zpr21)
```

```
800 ($L $y12 $zpr21)
801 (0 $y23 $zpr21)
802 ($L $y23 $zpr21)
803 (0 $h $zpr21)
804 ($L $h $zpr21)
805 (0 $y45 $zpr21)
806 ($L $y45 $zpr21)
807 (0 $y56 $zpr21)
808 ($L $y56 $zpr21)
809 (0 $Hr $zpr21)
810 ($L $Hr $zpr21)
811
812 //faccia valle, punti da 588 a 601
813 (0 0 $zvr21)
814 ($L 0 $zvr21)
815 (0 $y12 $zvr21)
816 ($L $y12 $zvr21)
817 (0 $y23 $zvr21)
818 ($L $y23 $zvr21)
819 (0 $h $zvr21)
820 ($L $h $zvr21)
821 (0 $y45 $zvr21)
822 ($L $y45 $zvr21)
823 (0 $y56 $zvr21)
824 ($L $y56 $zvr21)
825 (0 $H $zvr21)
826 ($L $H $zvr21)
827
828 // riblet 22
829 //faccia punta, punti da 602 a 615
830 (0 $hr $zpr22)
831 ($L $hr $zpr22)
832 (0 $y12 $zpr22)
833 ($L $y12 $zpr22)
834 (0 $y23 $zpr22)
835 ($L $y23 $zpr22)
836 (0 $h $zpr22)
837 ($L $h $zpr22)
838 (0 $y45 $zpr22)
839 ($L $y45 $zpr22)
840 (0 $y56 $zpr22)
841 ($L $y56 $zpr22)
842 (0 $Hr $zpr22)
843 ($L $Hr $zpr22)
844
845 //faccia valle, punti da 616 a 629
846 (0 0 $zvr22)
847 ($L 0 $zvr22)
848 (0 $y12 $zvr22)
849 ($L $y12 $zvr22)
850 (0 $y23 $zvr22)
851 ($L $y23 $zvr22)
852 (0 $h $zvr22)
853 ($L $h $zvr22)
854 (0 $y45 $zvr22)
855 ($L $y45 $zvr22)
856 (0 $y56 $zvr22)
857 ($L $y56 $zvr22)
858 (0 $H $zvr22)
859 ($L $H $zvr22)
860
861 // riblet 23
862 //faccia punta, punti da 630 a 643
863 (0 $hr $zpr23)
864 ($L $hr $zpr23)
865 (0 $y12 $zpr23)
866 ($L $y12 $zpr23)
867 (0 $y23 $zpr23)
868 ($L $y23 $zpr23)
```

```
869 (0 $h $zpr23)
870 ($L $h $zpr23)
871 (0 $y45 $zpr23)
872 ($L $y45 $zpr23)
873 (0 $y56 $zpr23)
874 ($L $y56 $zpr23)
875 (0 $Hr $zpr23)
876 ($L $Hr $zpr23)
877
878 //faccia valle, punti da 644 a 657
879 (0 0 $zvr23)
880 ($L 0 $zvr23)
881 (0 $y12 $zvr23)
882 ($L $y12 $zvr23)
883 (0 $y23 $zvr23)
884 ($L $y23 $zvr23)
885 (0 $h $zvr23)
886 ($L $h $zvr23)
887 (0 $y45 $zvr23)
888 ($L $y45 $zvr23)
889 (0 $y56 $zvr23)
890 ($L $y56 $zvr23)
891 (0 $H $zvr23)
892 ($L $H $zvr23)
893
894 // riblet 24
895 //faccia punta, punti da 658 a 671
896 (0 $hr $zpr24)
897 ($L $hr $zpr24)
898 (0 $y12 $zpr24)
899 ($L $y12 $zpr24)
900 (0 $y23 $zpr24)
901 ($L $y23 $zpr24)
902 (0 $h $zpr24)
903 ($L $h $zpr24)
904 (0 $y45 $zpr24)
905 ($L $y45 $zpr24)
906 (0 $y56 $zpr24)
907 ($L $y56 $zpr24)
908 (0 $Hr $zpr24)
909 ($L $Hr $zpr24)
910
911 //faccia valle, punti da 672 a 685
912 (0 0 $zvr24)
913 ($L 0 $zvr24)
914 (0 $y12 $zvr24)
915 ($L $y12 $zvr24)
916 (0 $y23 $zvr24)
917 ($L $y23 $zvr24)
918 (0 $h $zvr24)
919 ($L $h $zvr24)
920 (0 $y45 $zvr24)
921 ($L $y45 $zvr24)
922 (0 $y56 $zvr24)
923 ($L $y56 $zvr24)
924 (0 $H $zvr24)
925 ($L $H $zvr24)
926
927 // riblet 25
928 //faccia punta, punti da 686 a 699
929 (0 $hr $zpr25)
930 ($L $hr $zpr25)
931 (0 $y12 $zpr25)
932 ($L $y12 $zpr25)
933 (0 $y23 $zpr25)
934 ($L $y23 $zpr25)
935 (0 $h $zpr25)
936 ($L $h $zpr25)
937 (0 $y45 $zpr25)
```

```
938 ($L $y45 $zpr25)
939 (0 $y56 $zpr25)
940 ($L $y56 $zpr25)
941 (0 $Hr $zpr25)
942 ($L $Hr $zpr25)
943
944 //faccia valle, punti da 700 a 713
945 (0 0 $zvr25)
946 ($L 0 $zvr25)
947 (0 $y12 $zvr25)
948 ($L $y12 $zvr25)
949 (0 $y23 $zvr25)
950 ($L $y23 $zvr25)
951 (0 $h $zvr25)
952 ($L $h $zvr25)
953 (0 $y45 $zvr25)
954 ($L $y45 $zvr25)
955 (0 $y56 $zvr25)
956 ($L $y56 $zvr25)
957 (0 $H $zvr25)
958 ($L $H $zvr25)
959
960 // riblet 26
961 //faccia punta, punti da 714 a 727
962 (0 $hr $zpr26)
963 ($L $hr $zpr26)
964 (0 $y12 $zpr26)
965 ($L $y12 $zpr26)
966 (0 $y23 $zpr26)
967 ($L $y23 $zpr26)
968 (0 $h $zpr26)
969 ($L $h $zpr26)
970 (0 $y45 $zpr26)
971 ($L $y45 $zpr26)
972 (0 $y56 $zpr26)
973 ($L $y56 $zpr26)
974 (0 $Hr $zpr26)
975 ($L $Hr $zpr26)
976
977 //faccia valle, punti da 728 a 741
978 (0 0 $zvr26)
979 ($L 0 $zvr26)
980 (0 $y12 $zvr26)
981 ($L $y12 $zvr26)
982 (0 $y23 $zvr26)
983 ($L $y23 $zvr26)
984 (0 $h $zvr26)
985 ($L $h $zvr26)
986 (0 $y45 $zvr26)
987 ($L $y45 $zvr26)
988 (0 $y56 $zvr26)
989 ($L $y56 $zvr26)
990 (0 $H $zvr26)
991 ($L $H $zvr26)
992
993 // riblet 27
994 //faccia punta, punti da 742 a 755
995 (0 $hr $zpr27)
996 ($L $hr $zpr27)
997 (0 $y12 $zpr27)
998 ($L $y12 $zpr27)
999 (0 $y23 $zpr27)
1000 ($L $y23 $zpr27)
1001 (0 $h $zpr27)
1002 ($L $h $zpr27)
1003 (0 $y45 $zpr27)
1004 ($L $y45 $zpr27)
1005 (0 $y56 $zpr27)
1006 ($L $y56 $zpr27)
```

```
1007 (0 $Hr $zpr27)
1008 ($L $Hr $zpr27)
1009
1010 //faccia valle, punti da 756 a 769
1011 (0 0 $zvr27)
1012 ($L 0 $zvr27)
1013 (0 $y12 $zvr27)
1014 ($L $y12 $zvr27)
1015 (0 $y23 $zvr27)
1016 ($L $y23 $zvr27)
1017 (0 $h $zvr27)
1018 ($L $h $zvr27)
1019 (0 $y45 $zvr27)
1020 ($L $y45 $zvr27)
1021 (0 $y56 $zvr27)
1022 ($L $y56 $zvr27)
1023 (0 $H $zvr27)
1024 ($L $H $zvr27)
1025
1026 // riblet 28
1027 //faccia punta, punti da 770 a 783
1028 (0 $hr $zpr28)
1029 ($L $hr $zpr28)
1030 (0 $y12 $zpr28)
1031 ($L $y12 $zpr28)
1032 (0 $y23 $zpr28)
1033 ($L $y23 $zpr28)
1034 (0 $h $zpr28)
1035 ($L $h $zpr28)
1036 (0 $y45 $zpr28)
1037 ($L $y45 $zpr28)
1038 (0 $y56 $zpr28)
1039 ($L $y56 $zpr28)
1040 (0 $Hr $zpr28)
1041 ($L $Hr $zpr28)
1042
1043 //faccia valle, punti da 784 a 797
1044 (0 0 $zvr28)
1045 ($L 0 $zvr28)
1046 (0 $y12 $zvr28)
1047 ($L $y12 $zvr28)
1048 (0 $y23 $zvr28)
1049 ($L $y23 $zvr28)
1050 (0 $h $zvr28)
1051 ($L $h $zvr28)
1052 (0 $y45 $zvr28)
1053 ($L $y45 $zvr28)
1054 (0 $y56 $zvr28)
1055 ($L $y56 $zvr28)
1056 (0 $H $zvr28)
1057 ($L $H $zvr28)
1058
1059 // riblet 29
1060 //faccia punta, punti da 798 a 811
1061 (0 $hr $zpr29)
1062 ($L $hr $zpr29)
1063 (0 $y12 $zpr29)
1064 ($L $y12 $zpr29)
1065 (0 $y23 $zpr29)
1066 ($L $y23 $zpr29)
1067 (0 $h $zpr29)
1068 ($L $h $zpr29)
1069 (0 $y45 $zpr29)
1070 ($L $y45 $zpr29)
1071 (0 $y56 $zpr29)
1072 ($L $y56 $zpr29)
1073 (0 $Hr $zpr29)
1074 ($L $Hr $zpr29)
1075
```

```
1076 //faccia valle, punti da 812 a 825
1077 (0 0 $zvr29)
1078 ($L 0 $zvr29)
1079 (0 $y12 $zvr29)
1080 ($L $y12 $zvr29)
1081 (0 $y23 $zvr29)
1082 ($L $y23 $zvr29)
1083 (0 $h $zvr29)
1084 ($L $h $zvr29)
1085 (0 $y45 $zvr29)
1086 ($L $y45 $zvr29)
1087 (0 $y56 $zvr29)
1088 ($L $y56 $zvr29)
1089 (0 $H $zvr29)
1090 ($L $H $zvr29)
1091
1092 // riblet 30
1093 //faccia punta, punti da 826 a 839
1094 (0 $hr $zpr30)
1095 ($L $hr $zpr30)
1096 (0 $y12 $zpr30)
1097 ($L $y12 $zpr30)
1098 (0 $y23 $zpr30)
1099 ($L $y23 $zpr30)
1100 (0 $h $zpr30)
1101 ($L $h $zpr30)
1102 (0 $y45 $zpr30)
1103 ($L $y45 $zpr30)
1104 (0 $y56 $zpr30)
1105 ($L $y56 $zpr30)
1106 (0 $Hr $zpr30)
1107 ($L $Hr $zpr30)
1108
1109 //faccia valle, punti da 840 a 853
1110 (0 0 $zvr30)
1111 ($L 0 $zvr30)
1112 (0 $y12 $zvr30)
1113 ($L $y12 $zvr30)
1114 (0 $y23 $zvr30)
1115 ($L $y23 $zvr30)
1116 (0 $h $zvr30)
1117 ($L $h $zvr30)
1118 (0 $y45 $zvr30)
1119 ($L $y45 $zvr30)
1120 (0 $y56 $zvr30)
1121 ($L $y56 $zvr30)
1122 (0 $H $zvr30)
1123 ($L $H $zvr30)
1124
1125 ); //Fine vertici
1126
1127 blocks
1128 ( // I blocchi sono ordinati dal basso verso l alto lungo la y, per ogni rib.
1129 //Rib. 1
1130 // Primo lato
1131     hex (0 1 3 2 14 15 17 16) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1 $Rzsx)
1132         // Blocco vicino a parete inferiore
1133     hex (2 3 5 4 16 17 19 18) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2 $Rzsx)
1134         // Blocco intermedio inferiore
1135     hex (4 5 7 6 18 19 21 20) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzsx)
1136         // Blocco centrale inferiore
1137     hex (6 7 9 8 20 21 23 22) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzsx)
1138         // Blocco centrale superiore
1139     hex (8 9 11 10 22 23 25 24) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5 $Rzsx)
1140         // Blocco intermedio superiore
1141     hex (10 11 13 12 24 25 27 26) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6 $Rzsx)
1142         // Blocco vicino a parete superiore
1143 // Secondo lato
```

```

1138   hex (14 15 17 16 28 29 31 30) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1 $Rzdx
    ) // Blocco vicino a parete inferiore
1139   hex (16 17 19 18 30 31 33 32) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2 $Rzdx
    ) // Blocco intermedio inferiore
1140   hex (18 19 21 20 32 33 35 34) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale inferiore
1141   hex (20 21 23 22 34 35 37 36) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale superiore
1142   hex (22 23 25 24 36 37 39 38) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5 $Rzdx
    ) // Blocco intermedio superiore
1143   hex (24 25 27 26 38 39 41 40) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6 $Rzdx
    ) // Blocco vicino a parete superiore
1144
1145 //Rib. 2
1146 // Primo lato
1147   hex (28 29 31 30 42 43 45 44) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1 $Rzdx
    ) // Blocco vicino a parete inferiore
1148   hex (30 31 33 32 44 45 47 46) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2 $Rzdx
    ) // Blocco intermedio inferiore
1149   hex (32 33 35 34 46 47 49 48) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale inferiore
1150   hex (34 35 37 36 48 49 51 50) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale superiore
1151   hex (36 37 39 38 50 51 53 52) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5 $Rzdx
    ) // Blocco intermedio superiore
1152   hex (38 39 41 40 52 53 55 54) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6 $Rzdx
    ) // Blocco vicino a parete superiore
1153 // Secondo lato
1154   hex (42 43 45 44 56 57 59 58) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1 $Rzdx
    ) // Blocco vicino a parete inferiore
1155   hex (44 45 47 46 58 59 61 60) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2 $Rzdx
    ) // Blocco intermedio inferiore
1156   hex (46 47 49 48 60 61 63 62) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale inferiore
1157   hex (48 49 51 50 62 63 65 64) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale superiore
1158   hex (50 51 53 52 64 65 67 66) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5 $Rzdx
    ) // Blocco intermedio superiore
1159   hex (52 53 55 54 66 67 69 68) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6 $Rzdx
    ) // Blocco vicino a parete superiore
1160
1161 //Rib. 3
1162 // Primo lato
1163   hex (56 57 59 58 70 71 73 72) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1 $Rzdx
    ) // Blocco vicino a parete inferiore
1164   hex (58 59 61 60 72 73 75 74) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2 $Rzdx
    ) // Blocco intermedio inferiore
1165   hex (60 61 63 62 74 75 77 76) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale inferiore
1166   hex (62 63 65 64 76 77 79 78) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale superiore
1167   hex (64 65 67 66 78 79 81 80) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5 $Rzdx
    ) // Blocco intermedio superiore
1168   hex (66 67 69 68 80 81 83 82) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6 $Rzdx
    ) // Blocco vicino a parete superiore
1169 // Secondo lato
1170   hex (70 71 73 72 84 85 87 86) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1 $Rzdx
    ) // Blocco vicino a parete inferiore
1171   hex (72 73 75 74 86 87 89 88) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2 $Rzdx
    ) // Blocco intermedio inferiore
1172   hex (74 75 77 76 88 89 91 90) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale inferiore
1173   hex (76 77 79 78 90 91 93 92) ($Nx $Ny34 $Nzrib) simpleGrading (1 1 $Rzdx)
    // Blocco centrale superiore
1174   hex (78 79 81 80 92 93 95 94) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5 $Rzdx
    ) // Blocco intermedio superiore
1175   hex (80 81 83 82 94 95 97 96) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6 $Rzdx
    ) // Blocco vicino a parete superiore
1176

```

```

1177 //Rib. 4
1178 // Primo lato
1179   hex (84 85 87 86 98 99 101 100) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1
    $Rzsx) // Blocco vicino a parete inferiore
1180   hex (86 87 89 88 100 101 103 102) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry2
    $Rzsx) // Blocco intermedio inferiore
1181   hex (88 89 91 90 102 103 105 104) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzsx) // Blocco centrale inferiore
1182   hex (90 91 93 92 104 105 107 106) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzsx) // Blocco centrale superiore
1183   hex (92 93 95 94 106 107 109 108) ($Nx $Ny25 $Nzrib) simpleGrading (1 $Ry5
    $Rzsx) // Blocco intermedio superiore
1184   hex (94 95 97 96 108 109 111 110) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry6
    $Rzsx) // Blocco vicino a parete superiore
1185 // Secondo lato
1186   hex (98 99 101 100 112 113 115 114) ($Nx $Ny16 $Nzrib) simpleGrading (1 $Ry1
    $Rzdx) // Blocco vicino a parete inferiore
1187   hex (100 101 103 102 114 115 117 116) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry2 $Rzdx) // Blocco intermedio inferiore
1188   hex (102 103 105 104 116 117 119 118) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzdx) // Blocco centrale inferiore
1189   hex (104 105 107 106 118 119 121 120) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzdx) // Blocco centrale superiore
1190   hex (106 107 109 108 120 121 123 122) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry5 $Rzdx) // Blocco intermedio superiore
1191   hex (108 109 111 110 122 123 125 124) ($Nx $Ny16 $Nzrib) simpleGrading (1
    $Ry6 $Rzdx) // Blocco vicino a parete superiore
1192
1193 //Rib. 5
1194 // Primo lato
1195   hex (112 113 115 114 126 127 129 128) ($Nx $Ny16 $Nzrib) simpleGrading (1
    $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1196   hex (114 115 117 116 128 129 131 130) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry2 $Rzsx) // Blocco intermedio inferiore
1197   hex (116 117 119 118 130 131 133 132) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzsx) // Blocco centrale inferiore
1198   hex (118 119 121 120 132 133 135 134) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzsx) // Blocco centrale superiore
1199   hex (120 121 123 122 134 135 137 136) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry5 $Rzsx) // Blocco intermedio superiore
1200   hex (122 123 125 124 136 137 139 138) ($Nx $Ny16 $Nzrib) simpleGrading (1
    $Ry6 $Rzsx) // Blocco vicino a parete superiore
1201 // Secondo lato
1202   hex (126 127 129 128 140 141 143 142) ($Nx $Ny16 $Nzrib) simpleGrading (1
    $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1203   hex (128 129 131 130 142 143 145 144) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry2 $Rzdx) // Blocco intermedio inferiore
1204   hex (130 131 133 132 144 145 147 146) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzdx) // Blocco centrale inferiore
1205   hex (132 133 135 134 146 147 149 148) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzdx) // Blocco centrale superiore
1206   hex (134 135 137 136 148 149 151 150) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry5 $Rzdx) // Blocco intermedio superiore
1207   hex (136 137 139 138 150 151 153 152) ($Nx $Ny16 $Nzrib) simpleGrading (1
    $Ry6 $Rzdx) // Blocco vicino a parete superiore
1208
1209 //Rib. 6
1210 // Primo lato
1211   hex (140 141 143 142 154 155 157 156) ($Nx $Ny16 $Nzrib) simpleGrading (1
    $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1212   hex (142 143 145 144 156 157 159 158) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry2 $Rzsx) // Blocco intermedio inferiore
1213   hex (144 145 147 146 158 159 161 160) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzsx) // Blocco centrale inferiore
1214   hex (146 147 149 148 160 161 163 162) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
    $Rzsx) // Blocco centrale superiore
1215   hex (148 149 151 150 162 163 165 164) ($Nx $Ny25 $Nzrib) simpleGrading (1
    $Ry5 $Rzsx) // Blocco intermedio superiore

```

```

1216     hex (150 151 153 152 164 165 167 166) ($Nx $Ny16 $Nzrib) simpleGrading (1
1217 // $Ry6 $Rzsx) // Blocco vicino a parete superiore
1217 // Secondo lato
1218     hex (154 155 157 156 168 169 171 170) ($Nx $Ny16 $Nzrib) simpleGrading (1
1219     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1219     hex (156 157 159 158 170 171 173 172) ($Nx $Ny25 $Nzrib) simpleGrading (1
1220     $Ry2 $Rzdx) // Blocco intermedio inferiore
1220     hex (158 159 161 160 172 173 175 174) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1221     $Rzdx) // Blocco centrale inferiore
1221     hex (160 161 163 162 174 175 177 176) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1222     $Rzdx) // Blocco centrale superiore
1222     hex (162 163 165 164 176 177 179 178) ($Nx $Ny25 $Nzrib) simpleGrading (1
1223     $Ry5 $Rzdx) // Blocco intermedio superiore
1223     hex (164 165 167 166 178 179 181 180) ($Nx $Ny16 $Nzrib) simpleGrading (1
1224     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1224
1225 //Rib. 7
1226 // Primo lato
1227     hex (168 169 171 170 182 183 185 184) ($Nx $Ny16 $Nzrib) simpleGrading (1
1228     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1228     hex (170 171 173 172 184 185 187 186) ($Nx $Ny25 $Nzrib) simpleGrading (1
1229     $Ry2 $Rzsx) // Blocco intermedio inferiore
1229     hex (172 173 175 174 186 187 189 188) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1230     $Rzsx) // Blocco centrale inferiore
1230     hex (174 175 177 176 188 189 191 190) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1231     $Rzsx) // Blocco centrale superiore
1231     hex (176 177 179 178 190 191 193 192) ($Nx $Ny25 $Nzrib) simpleGrading (1
1232     $Ry5 $Rzsx) // Blocco intermedio superiore
1232     hex (178 179 181 180 192 193 195 194) ($Nx $Ny16 $Nzrib) simpleGrading (1
1233     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1233 // Secondo lato
1234     hex (182 183 185 184 196 197 199 198) ($Nx $Ny16 $Nzrib) simpleGrading (1
1235     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1235     hex (184 185 187 186 198 199 201 200) ($Nx $Ny25 $Nzrib) simpleGrading (1
1236     $Ry2 $Rzdx) // Blocco intermedio inferiore
1236     hex (186 187 189 188 200 201 203 202) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1237     $Rzdx) // Blocco centrale inferiore
1237     hex (188 189 191 190 202 203 205 204) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1238     $Rzdx) // Blocco centrale superiore
1238     hex (190 191 193 192 204 205 207 206) ($Nx $Ny25 $Nzrib) simpleGrading (1
1239     $Ry5 $Rzdx) // Blocco intermedio superiore
1239     hex (192 193 195 194 206 207 209 208) ($Nx $Ny16 $Nzrib) simpleGrading (1
1240     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1240
1241 //Rib. 8
1242 // Primo lato
1243     hex (196 197 199 198 210 211 213 212) ($Nx $Ny16 $Nzrib) simpleGrading (1
1244     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1244     hex (198 199 201 200 212 213 215 214) ($Nx $Ny25 $Nzrib) simpleGrading (1
1245     $Ry2 $Rzsx) // Blocco intermedio inferiore
1245     hex (200 201 203 202 214 215 217 216) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1246     $Rzsx) // Blocco centrale inferiore
1246     hex (202 203 205 204 216 217 219 218) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1247     $Rzsx) // Blocco centrale superiore
1247     hex (204 205 207 206 218 219 221 220) ($Nx $Ny25 $Nzrib) simpleGrading (1
1248     $Ry5 $Rzsx) // Blocco intermedio superiore
1248     hex (206 207 209 208 220 221 223 222) ($Nx $Ny16 $Nzrib) simpleGrading (1
1249     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1249 // Secondo lato
1250     hex (210 211 213 212 224 225 227 226) ($Nx $Ny16 $Nzrib) simpleGrading (1
1251     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1251     hex (212 213 215 214 226 227 229 228) ($Nx $Ny25 $Nzrib) simpleGrading (1
1252     $Ry2 $Rzdx) // Blocco intermedio inferiore
1252     hex (214 215 217 216 228 229 231 230) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1253     $Rzdx) // Blocco centrale inferiore
1253     hex (216 217 219 218 230 231 233 232) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1254     $Rzdx) // Blocco centrale superiore
1254     hex (218 219 221 220 232 233 235 234) ($Nx $Ny25 $Nzrib) simpleGrading (1
1255     $Ry5 $Rzdx) // Blocco intermedio superiore

```

```

1255     hex (220 221 223 222 234 235 237 236) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry6 $Rzdx) // Blocco vicino a parete superiore
1256
1257 //Rib. 9
1258 // Primo lato
1259     hex (224 225 227 226 238 239 241 240) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1260     hex (226 227 229 228 240 241 243 242) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry2 $Rzsx) // Blocco intermedio inferiore
1261     hex (228 229 231 230 242 243 245 244) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzsx) // Blocco centrale inferiore
1262     hex (230 231 233 232 244 245 247 246) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzsx) // Blocco centrale superiore
1263     hex (232 233 235 234 246 247 249 248) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry5 $Rzsx) // Blocco intermedio superiore
1264     hex (234 235 237 236 248 249 251 250) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry6 $Rzsx) // Blocco vicino a parete superiore
1265 // Secondo lato
1266     hex (238 239 241 240 252 253 255 254) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1267     hex (240 241 243 242 254 255 257 256) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry2 $Rzdx) // Blocco intermedio inferiore
1268     hex (242 243 245 244 256 257 259 258) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzdx) // Blocco centrale inferiore
1269     hex (244 245 247 246 258 259 261 260) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzdx) // Blocco centrale superiore
1270     hex (246 247 249 248 260 261 263 262) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry5 $Rzdx) // Blocco intermedio superiore
1271     hex (248 249 251 250 262 263 265 264) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry6 $Rzdx) // Blocco vicino a parete superiore
1272
1273 //Rib. 10
1274 // Primo lato
1275     hex (252 253 255 254 266 267 269 268) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1276     hex (254 255 257 256 268 269 271 270) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry2 $Rzsx) // Blocco intermedio inferiore
1277     hex (256 257 259 258 270 271 273 272) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzsx) // Blocco centrale inferiore
1278     hex (258 259 261 260 272 273 275 274) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzsx) // Blocco centrale superiore
1279     hex (260 261 263 262 274 275 277 276) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry5 $Rzsx) // Blocco intermedio superiore
1280     hex (262 263 265 264 276 277 279 278) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry6 $Rzsx) // Blocco vicino a parete superiore
1281 // Secondo lato
1282     hex (266 267 269 268 280 281 283 282) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1283     hex (268 269 271 270 282 283 285 284) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry2 $Rzdx) // Blocco intermedio inferiore
1284     hex (270 271 273 272 284 285 287 286) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzdx) // Blocco centrale inferiore
1285     hex (272 273 275 274 286 287 289 288) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzdx) // Blocco centrale superiore
1286     hex (274 275 277 276 288 289 291 290) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry5 $Rzdx) // Blocco intermedio superiore
1287     hex (276 277 279 278 290 291 293 292) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry6 $Rzdx) // Blocco vicino a parete superiore
1288
1289 //Rib. 11
1290 // Primo lato
1291     hex (280 281 283 282 294 295 297 296) ($Nx $Ny16 $Nzrib) simpleGrading (1
        $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1292     hex (282 283 285 284 296 297 299 298) ($Nx $Ny25 $Nzrib) simpleGrading (1
        $Ry2 $Rzsx) // Blocco intermedio inferiore
1293     hex (284 285 287 286 298 299 301 300) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzsx) // Blocco centrale inferiore
1294     hex (286 287 289 288 300 301 303 302) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
        $Rzsx) // Blocco centrale superiore

```

```

1295     hex (288 289 291 290 302 303 305 304) ($Nx $Ny25 $Nzrib) simpleGrading (1
1296     $Ry5 $Rzsx) // Blocco intermedio superiore
1296     hex (290 291 293 292 304 305 307 306) ($Nx $Ny16 $Nzrib) simpleGrading (1
1297     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1297 // Secondo lato
1298     hex (294 295 297 296 308 309 311 310) ($Nx $Ny16 $Nzrib) simpleGrading (1
1298     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1299     hex (296 297 299 298 310 311 313 312) ($Nx $Ny25 $Nzrib) simpleGrading (1
1299     $Ry2 $Rzdx) // Blocco intermedio inferiore
1300     hex (298 299 301 300 312 313 315 314) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1300     $Rzdx) // Blocco centrale inferiore
1301     hex (300 301 303 302 314 315 317 316) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1301     $Rzdx) // Blocco centrale superiore
1302     hex (302 303 305 304 316 317 319 318) ($Nx $Ny25 $Nzrib) simpleGrading (1
1302     $Ry5 $Rzdx) // Blocco intermedio superiore
1303     hex (304 305 307 306 318 319 321 320) ($Nx $Ny16 $Nzrib) simpleGrading (1
1303     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1304
1305 //Rib. 12
1306 // Primo lato
1307     hex (308 309 311 310 322 323 325 324) ($Nx $Ny16 $Nzrib) simpleGrading (1
1307     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1308     hex (310 311 313 312 324 325 327 326) ($Nx $Ny25 $Nzrib) simpleGrading (1
1308     $Ry2 $Rzsx) // Blocco intermedio inferiore
1309     hex (312 313 315 314 326 327 329 328) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1309     $Rzsx) // Blocco centrale inferiore
1310     hex (314 315 317 316 328 329 331 330) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1310     $Rzsx) // Blocco centrale superiore
1311     hex (316 317 319 318 330 331 333 332) ($Nx $Ny25 $Nzrib) simpleGrading (1
1311     $Ry5 $Rzsx) // Blocco intermedio superiore
1312     hex (318 319 321 320 332 333 335 334) ($Nx $Ny16 $Nzrib) simpleGrading (1
1312     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1313 // Secondo lato
1314     hex (322 323 325 324 336 337 339 338) ($Nx $Ny16 $Nzrib) simpleGrading (1
1314     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1315     hex (324 325 327 326 338 339 341 340) ($Nx $Ny25 $Nzrib) simpleGrading (1
1315     $Ry2 $Rzdx) // Blocco intermedio inferiore
1316     hex (326 327 329 328 340 341 343 342) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1316     $Rzdx) // Blocco centrale inferiore
1317     hex (328 329 331 330 342 343 345 344) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1317     $Rzdx) // Blocco centrale superiore
1318     hex (330 331 333 332 344 345 347 346) ($Nx $Ny25 $Nzrib) simpleGrading (1
1318     $Ry5 $Rzdx) // Blocco intermedio superiore
1319     hex (332 333 335 334 346 347 349 348) ($Nx $Ny16 $Nzrib) simpleGrading (1
1319     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1320
1321 //Rib. 13
1322 // Primo lato
1323     hex (336 337 339 338 350 351 353 352) ($Nx $Ny16 $Nzrib) simpleGrading (1
1323     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1324     hex (338 339 341 340 352 353 355 354) ($Nx $Ny25 $Nzrib) simpleGrading (1
1324     $Ry2 $Rzsx) // Blocco intermedio inferiore
1325     hex (340 341 343 342 354 355 357 356) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1325     $Rzsx) // Blocco centrale inferiore
1326     hex (342 343 345 344 356 357 359 358) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1326     $Rzsx) // Blocco centrale superiore
1327     hex (344 345 347 346 358 359 361 360) ($Nx $Ny25 $Nzrib) simpleGrading (1
1327     $Ry5 $Rzsx) // Blocco intermedio superiore
1328     hex (346 347 349 348 360 361 363 362) ($Nx $Ny16 $Nzrib) simpleGrading (1
1328     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1329 // Secondo lato
1330     hex (350 351 353 352 364 365 367 366) ($Nx $Ny16 $Nzrib) simpleGrading (1
1330     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1331     hex (352 353 355 354 366 367 369 368) ($Nx $Ny25 $Nzrib) simpleGrading (1
1331     $Ry2 $Rzdx) // Blocco intermedio inferiore
1332     hex (354 355 357 356 368 369 371 370) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1332     $Rzdx) // Blocco centrale inferiore
1333     hex (356 357 359 358 370 371 373 372) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1333     $Rzdx) // Blocco centrale superiore

```

```

1334     hex (358 359 361 360 372 373 375 374) ($Nx $Ny25 $Nzrib) simpleGrading (1
1335     $Ry5 $Rzdx) // Blocco intermedio superiore
1336
1337 //Rib. 14
1338 // Primo lato
1339     hex (364 365 367 366 378 379 381 380) ($Nx $Ny16 $Nzrib) simpleGrading (1
1340     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1341     hex (366 367 369 368 380 381 383 382) ($Nx $Ny25 $Nzrib) simpleGrading (1
1342     $Ry2 $Rzsx) // Blocco intermedio inferiore
1343     hex (368 369 371 370 382 383 385 384) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1344     $Rzsx) // Blocco centrale inferiore
1345     hex (370 371 373 372 384 385 387 386) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1346     $Rzsx) // Blocco centrale superiore
1347     hex (372 373 375 374 386 387 389 388) ($Nx $Ny25 $Nzrib) simpleGrading (1
1348     $Ry5 $Rzsx) // Blocco intermedio superiore
1349     hex (374 375 377 376 388 389 391 390) ($Nx $Ny16 $Nzrib) simpleGrading (1
1350     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1351 // Secondo lato
1352     hex (378 379 381 380 392 393 395 394) ($Nx $Ny16 $Nzrib) simpleGrading (1
1353     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1354     hex (380 381 383 382 394 395 397 396) ($Nx $Ny25 $Nzrib) simpleGrading (1
1355     $Ry2 $Rzdx) // Blocco intermedio inferiore
1356     hex (382 383 385 384 396 397 399 398) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1357     $Rzdx) // Blocco centrale inferiore
1358     hex (384 385 387 386 398 399 401 400) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1359     $Rzdx) // Blocco centrale superiore
1360     hex (386 387 389 388 400 401 403 402) ($Nx $Ny25 $Nzrib) simpleGrading (1
1361     $Ry5 $Rzdx) // Blocco intermedio superiore
1362     hex (388 389 391 390 402 403 405 404) ($Nx $Ny16 $Nzrib) simpleGrading (1
1363     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1364
1365 //Rib. 15
1366 // Primo lato
1367     hex (392 393 395 394 406 407 409 408) ($Nx $Ny16 $Nzrib) simpleGrading (1
1368     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1369     hex (394 395 397 396 408 409 411 410) ($Nx $Ny25 $Nzrib) simpleGrading (1
1370     $Ry2 $Rzsx) // Blocco intermedio inferiore
1371     hex (396 397 399 398 410 411 413 412) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1372     $Rzsx) // Blocco centrale inferiore
1373     hex (398 399 401 400 412 413 415 414) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1374     $Rzsx) // Blocco centrale superiore
1375     hex (400 401 403 402 414 415 417 416) ($Nx $Ny25 $Nzrib) simpleGrading (1
1376     $Ry5 $Rzsx) // Blocco intermedio superiore
1377     hex (402 403 405 404 416 417 419 418) ($Nx $Ny16 $Nzrib) simpleGrading (1
1378     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1379 // Secondo lato
1380     hex (406 407 409 408 420 421 423 422) ($Nx $Ny16 $Nzrib) simpleGrading (1
1381     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1382     hex (408 409 411 410 422 423 425 424) ($Nx $Ny25 $Nzrib) simpleGrading (1
1383     $Ry2 $Rzdx) // Blocco intermedio inferiore
1384     hex (410 411 413 412 424 425 427 426) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1385     $Rzdx) // Blocco centrale inferiore
1386     hex (412 413 415 414 426 427 429 428) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1387     $Rzdx) // Blocco centrale superiore
1388     hex (414 415 417 416 428 429 431 430) ($Nx $Ny25 $Nzrib) simpleGrading (1
1389     $Ry5 $Rzdx) // Blocco intermedio superiore
1390     hex (416 417 419 418 430 431 433 432) ($Nx $Ny16 $Nzrib) simpleGrading (1
1391     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1392
1393 //Rib. 16
1394 // Primo lato
1395     hex (420 421 423 422 434 435 437 436) ($Nx $Ny16 $Nzrib) simpleGrading (1
1396     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1397     hex (422 423 425 424 436 437 439 438) ($Nx $Ny25 $Nzrib) simpleGrading (1
1398     $Ry2 $Rzsx) // Blocco intermedio inferiore

```

```

1374   hex (424 425 427 426 438 439 441 440) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzsx) // Blocco centrale inferiore
1375   hex (426 427 429 428 440 441 443 442) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzsx) // Blocco centrale superiore
1376   hex (428 429 431 430 442 443 445 444) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry5 $Rzsx) // Blocco intermedio superiore
1377   hex (430 431 433 432 444 445 447 446) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry6 $Rzsx) // Blocco vicino a parete superiore
1378 // Secondo lato
1379   hex (434 435 437 436 448 449 451 450) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1380   hex (436 437 439 438 450 451 453 452) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry2 $Rzdx) // Blocco intermedio inferiore
1381   hex (438 439 441 440 452 453 455 454) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzdx) // Blocco centrale inferiore
1382   hex (440 441 443 442 454 455 457 456) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzdx) // Blocco centrale superiore
1383   hex (442 443 445 444 456 457 459 458) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry5 $Rzdx) // Blocco intermedio superiore
1384   hex (444 445 447 446 458 459 461 460) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry6 $Rzdx) // Blocco vicino a parete superiore
1385
1386 //Rib. 17
1387 // Primo lato
1388   hex (448 449 451 450 462 463 465 464) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1389   hex (450 451 453 452 464 465 467 466) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry2 $Rzsx) // Blocco intermedio inferiore
1390   hex (452 453 455 454 466 467 469 468) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzsx) // Blocco centrale inferiore
1391   hex (454 455 457 456 468 469 471 470) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzsx) // Blocco centrale superiore
1392   hex (456 457 459 458 470 471 473 472) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry5 $Rzsx) // Blocco intermedio superiore
1393   hex (458 459 461 460 472 473 475 474) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry6 $Rzsx) // Blocco vicino a parete superiore
1394 // Secondo lato
1395   hex (462 463 465 464 476 477 479 478) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1396   hex (464 465 467 466 478 479 481 480) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry2 $Rzdx) // Blocco intermedio inferiore
1397   hex (466 467 469 468 480 481 483 482) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzdx) // Blocco centrale inferiore
1398   hex (468 469 471 470 482 483 485 484) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzdx) // Blocco centrale superiore
1399   hex (470 471 473 472 484 485 487 486) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry5 $Rzdx) // Blocco intermedio superiore
1400   hex (472 473 475 474 486 487 489 488) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry6 $Rzdx) // Blocco vicino a parete superiore
1401
1402 //Rib. 18
1403 // Primo lato
1404   hex (476 477 479 478 490 491 493 492) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1405   hex (478 479 481 480 492 493 495 494) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry2 $Rzsx) // Blocco intermedio inferiore
1406   hex (480 481 483 482 494 495 497 496) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzsx) // Blocco centrale inferiore
1407   hex (482 483 485 484 496 497 499 498) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
      $Rzsx) // Blocco centrale superiore
1408   hex (484 485 487 486 498 499 501 500) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry5 $Rzsx) // Blocco intermedio superiore
1409   hex (486 487 489 488 500 501 503 502) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry6 $Rzsx) // Blocco vicino a parete superiore
1410 // Secondo lato
1411   hex (490 491 493 492 504 505 507 506) ($Nx $Ny16 $Nzrib) simpleGrading (1
      $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1412   hex (492 493 495 494 506 507 509 508) ($Nx $Ny25 $Nzrib) simpleGrading (1
      $Ry2 $Rzdx) // Blocco intermedio inferiore

```

```

1413     hex (494 495 497 496 508 509 511 510) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1414     $Rzdx) // Blocco centrale inferiore
1414     hex (496 497 499 498 510 511 513 512) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1415     $Rzdx) // Blocco centrale superiore
1415     hex (498 499 501 500 512 513 515 514) ($Nx $Ny25 $Nzrib) simpleGrading (1
1416     $Ry5 $Rzdx) // Blocco intermedio superiore
1416     hex (500 501 503 502 514 515 517 516) ($Nx $Ny16 $Nzrib) simpleGrading (1
1417     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1417
1418 //Rib. 19
1419 // Primo lato
1420     hex (504 505 507 506 518 519 521 520) ($Nx $Ny16 $Nzrib) simpleGrading (1
1421     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1421     hex (506 507 509 508 520 521 523 522) ($Nx $Ny25 $Nzrib) simpleGrading (1
1422     $Ry2 $Rzsx) // Blocco intermedio inferiore
1422     hex (508 509 511 510 522 523 525 524) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1423     $Rzsx) // Blocco centrale inferiore
1423     hex (510 511 513 512 524 525 527 526) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1424     $Rzsx) // Blocco centrale superiore
1424     hex (512 513 515 514 526 527 529 528) ($Nx $Ny25 $Nzrib) simpleGrading (1
1425     $Ry5 $Rzsx) // Blocco intermedio superiore
1425     hex (514 515 517 516 528 529 531 530) ($Nx $Ny16 $Nzrib) simpleGrading (1
1426     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1426 // Secondo lato
1427     hex (518 519 521 520 532 533 535 534) ($Nx $Ny16 $Nzrib) simpleGrading (1
1428     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1428     hex (520 521 523 522 534 535 537 536) ($Nx $Ny25 $Nzrib) simpleGrading (1
1429     $Ry2 $Rzdx) // Blocco intermedio inferiore
1429     hex (522 523 525 524 536 537 539 538) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1430     $Rzdx) // Blocco centrale inferiore
1430     hex (524 525 527 526 538 539 541 540) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1431     $Rzdx) // Blocco centrale superiore
1431     hex (526 527 529 528 540 541 543 542) ($Nx $Ny25 $Nzrib) simpleGrading (1
1432     $Ry5 $Rzdx) // Blocco intermedio superiore
1432     hex (528 529 531 530 542 543 545 544) ($Nx $Ny16 $Nzrib) simpleGrading (1
1433     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1433
1434 //Rib. 20
1435 // Primo lato
1436     hex (532 533 535 534 546 547 549 548) ($Nx $Ny16 $Nzrib) simpleGrading (1
1437     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1437     hex (534 535 537 536 548 549 551 550) ($Nx $Ny25 $Nzrib) simpleGrading (1
1438     $Ry2 $Rzsx) // Blocco intermedio inferiore
1438     hex (536 537 539 538 550 551 553 552) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1439     $Rzsx) // Blocco centrale inferiore
1439     hex (538 539 541 540 552 553 555 554) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1440     $Rzsx) // Blocco centrale superiore
1440     hex (540 541 543 542 554 555 557 556) ($Nx $Ny25 $Nzrib) simpleGrading (1
1441     $Ry5 $Rzsx) // Blocco intermedio superiore
1441     hex (542 543 545 544 556 557 559 558) ($Nx $Ny16 $Nzrib) simpleGrading (1
1442     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1442 // Secondo lato
1443     hex (546 547 549 548 560 561 563 562) ($Nx $Ny16 $Nzrib) simpleGrading (1
1444     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1444     hex (548 549 551 550 562 563 565 564) ($Nx $Ny25 $Nzrib) simpleGrading (1
1445     $Ry2 $Rzdx) // Blocco intermedio inferiore
1445     hex (550 551 553 552 564 565 567 566) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1446     $Rzdx) // Blocco centrale inferiore
1446     hex (552 553 555 554 566 567 569 568) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1447     $Rzdx) // Blocco centrale superiore
1447     hex (554 555 557 556 568 569 571 570) ($Nx $Ny25 $Nzrib) simpleGrading (1
1448     $Ry5 $Rzdx) // Blocco intermedio superiore
1448     hex (556 557 559 558 570 571 573 572) ($Nx $Ny16 $Nzrib) simpleGrading (1
1449     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1449
1450 //Rib. 21
1451 // Primo lato
1452     hex (560 561 563 562 574 575 577 576) ($Nx $Ny16 $Nzrib) simpleGrading (1
1453     $Ry1 $Rzsx) // Blocco vicino a parete inferiore

```

```

1453     hex (562 563 565 564 576 577 579 578) ($Nx $Ny25 $Nzrib) simpleGrading (1
1454     $Ry2 $Rzsx) // Blocco intermedio inferiore
1454     hex (564 565 567 566 578 579 581 580) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1455     $Rzsx) // Blocco centrale inferiore
1455     hex (566 567 569 568 580 581 583 582) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1456     $Rzsx) // Blocco centrale superiore
1456     hex (568 569 571 570 582 583 585 584) ($Nx $Ny25 $Nzrib) simpleGrading (1
1457     $Ry5 $Rzsx) // Blocco intermedio superiore
1457     hex (570 571 573 572 584 585 587 586) ($Nx $Ny16 $Nzrib) simpleGrading (1
1458     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1458 // Secondo lato
1459     hex (574 575 577 576 588 589 591 590) ($Nx $Ny16 $Nzrib) simpleGrading (1
1460     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1460     hex (576 577 579 578 590 591 593 592) ($Nx $Ny25 $Nzrib) simpleGrading (1
1461     $Ry2 $Rzdx) // Blocco intermedio inferiore
1461     hex (578 579 581 580 592 593 595 594) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1462     $Rzdx) // Blocco centrale inferiore
1462     hex (580 581 583 582 594 595 597 596) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1463     $Rzdx) // Blocco centrale superiore
1463     hex (582 583 585 584 596 597 599 598) ($Nx $Ny25 $Nzrib) simpleGrading (1
1464     $Ry5 $Rzdx) // Blocco intermedio superiore
1464     hex (584 585 587 586 598 599 601 600) ($Nx $Ny16 $Nzrib) simpleGrading (1
1465     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1465
1466 //Rib. 22
1467 // Primo lato
1468     hex (588 589 591 590 602 603 605 604) ($Nx $Ny16 $Nzrib) simpleGrading (1
1469     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1469     hex (590 591 593 592 604 605 607 606) ($Nx $Ny25 $Nzrib) simpleGrading (1
1470     $Ry2 $Rzsx) // Blocco intermedio inferiore
1470     hex (592 593 595 594 606 607 609 608) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1471     $Rzsx) // Blocco centrale inferiore
1471     hex (594 595 597 596 608 609 611 610) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1472     $Rzsx) // Blocco centrale superiore
1472     hex (596 597 599 598 610 611 613 612) ($Nx $Ny25 $Nzrib) simpleGrading (1
1473     $Ry5 $Rzsx) // Blocco intermedio superiore
1473     hex (598 599 601 600 612 613 615 614) ($Nx $Ny16 $Nzrib) simpleGrading (1
1474     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1474 // Secondo lato
1475     hex (602 603 605 604 616 617 619 618) ($Nx $Ny16 $Nzrib) simpleGrading (1
1476     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1476     hex (604 605 607 606 618 619 621 620) ($Nx $Ny25 $Nzrib) simpleGrading (1
1477     $Ry2 $Rzdx) // Blocco intermedio inferiore
1477     hex (606 607 609 608 620 621 623 622) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1478     $Rzdx) // Blocco centrale inferiore
1478     hex (608 609 611 610 622 623 625 624) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1479     $Rzdx) // Blocco centrale superiore
1479     hex (610 611 613 612 624 625 627 626) ($Nx $Ny25 $Nzrib) simpleGrading (1
1480     $Ry5 $Rzdx) // Blocco intermedio superiore
1480     hex (612 613 615 614 626 627 629 628) ($Nx $Ny16 $Nzrib) simpleGrading (1
1481     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1481
1482 //Rib. 23
1483 // Primo lato
1484     hex (616 617 619 618 630 631 633 632) ($Nx $Ny16 $Nzrib) simpleGrading (1
1485     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1485     hex (618 619 621 620 632 633 635 634) ($Nx $Ny25 $Nzrib) simpleGrading (1
1486     $Ry2 $Rzsx) // Blocco intermedio inferiore
1486     hex (620 621 623 622 634 635 637 636) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1487     $Rzsx) // Blocco centrale inferiore
1487     hex (622 623 625 624 636 637 639 638) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1488     $Rzsx) // Blocco centrale superiore
1488     hex (624 625 627 626 638 639 641 640) ($Nx $Ny25 $Nzrib) simpleGrading (1
1489     $Ry5 $Rzsx) // Blocco intermedio superiore
1489     hex (626 627 629 628 640 641 643 642) ($Nx $Ny16 $Nzrib) simpleGrading (1
1490     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1490 // Secondo lato
1491     hex (630 631 633 632 644 645 647 646) ($Nx $Ny16 $Nzrib) simpleGrading (1
1491     $Ry1 $Rzdx) // Blocco vicino a parete inferiore

```

```

1492     hex (632 633 635 634 646 647 649 648) ($Nx $Ny25 $Nzrib) simpleGrading (1
1493     $Ry2 $Rzdx) // Blocco intermedio inferiore
1493     hex (634 635 637 636 648 649 651 650) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1494     $Rzdx) // Blocco centrale inferiore
1494     hex (636 637 639 638 650 651 653 652) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1495     $Rzdx) // Blocco centrale superiore
1495     hex (638 639 641 640 652 653 655 654) ($Nx $Ny25 $Nzrib) simpleGrading (1
1496     $Ry5 $Rzdx) // Blocco intermedio superiore
1496     hex (640 641 643 642 654 655 657 656) ($Nx $Ny16 $Nzrib) simpleGrading (1
1497     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1497
1498 //Rib. 24
1499 // Primo lato
1500     hex (644 645 647 646 658 659 661 660) ($Nx $Ny16 $Nzrib) simpleGrading (1
1501     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1501     hex (646 647 649 648 660 661 663 662) ($Nx $Ny25 $Nzrib) simpleGrading (1
1502     $Ry2 $Rzsx) // Blocco intermedio inferiore
1502     hex (648 649 651 650 662 663 665 664) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1503     $Rzsx) // Blocco centrale inferiore
1503     hex (650 651 653 652 664 665 667 666) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1504     $Rzsx) // Blocco centrale superiore
1504     hex (652 653 655 654 666 667 669 668) ($Nx $Ny25 $Nzrib) simpleGrading (1
1505     $Ry5 $Rzsx) // Blocco intermedio superiore
1505     hex (654 655 657 656 668 669 671 670) ($Nx $Ny16 $Nzrib) simpleGrading (1
1506     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1506 // Secondo lato
1507     hex (658 659 661 660 672 673 675 674) ($Nx $Ny16 $Nzrib) simpleGrading (1
1508     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1508     hex (660 661 663 662 674 675 677 676) ($Nx $Ny25 $Nzrib) simpleGrading (1
1509     $Ry2 $Rzdx) // Blocco intermedio inferiore
1509     hex (662 663 665 664 676 677 679 678) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1510     $Rzdx) // Blocco centrale inferiore
1510     hex (664 665 667 666 678 679 681 680) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1511     $Rzdx) // Blocco centrale superiore
1511     hex (666 667 669 668 680 681 683 682) ($Nx $Ny25 $Nzrib) simpleGrading (1
1512     $Ry5 $Rzdx) // Blocco intermedio superiore
1512     hex (668 669 671 670 682 683 685 684) ($Nx $Ny16 $Nzrib) simpleGrading (1
1513     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1513
1514 //Rib. 25
1515 // Primo lato
1516     hex (672 673 675 674 686 687 689 688) ($Nx $Ny16 $Nzrib) simpleGrading (1
1517     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1517     hex (674 675 677 676 688 689 691 690) ($Nx $Ny25 $Nzrib) simpleGrading (1
1518     $Ry2 $Rzsx) // Blocco intermedio inferiore
1518     hex (676 677 679 678 690 691 693 692) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1519     $Rzsx) // Blocco centrale inferiore
1519     hex (678 679 681 680 692 693 695 694) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1520     $Rzsx) // Blocco centrale superiore
1520     hex (680 681 683 682 694 695 697 696) ($Nx $Ny25 $Nzrib) simpleGrading (1
1521     $Ry5 $Rzsx) // Blocco intermedio superiore
1521     hex (682 683 685 684 696 697 699 698) ($Nx $Ny16 $Nzrib) simpleGrading (1
1522     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1522 // Secondo lato
1523     hex (686 687 689 688 700 701 703 702) ($Nx $Ny16 $Nzrib) simpleGrading (1
1524     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1524     hex (688 689 691 690 702 703 705 704) ($Nx $Ny25 $Nzrib) simpleGrading (1
1525     $Ry2 $Rzdx) // Blocco intermedio inferiore
1525     hex (690 691 693 692 704 705 707 706) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1526     $Rzdx) // Blocco centrale inferiore
1526     hex (692 693 695 694 706 707 709 708) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1527     $Rzdx) // Blocco centrale superiore
1527     hex (694 695 697 696 708 709 711 710) ($Nx $Ny25 $Nzrib) simpleGrading (1
1528     $Ry5 $Rzdx) // Blocco intermedio superiore
1528     hex (696 697 699 698 710 711 713 712) ($Nx $Ny16 $Nzrib) simpleGrading (1
1529     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1529
1530 //Rib. 26
1531 // Primo lato

```

```

1532     hex (700 701 703 702 714 715 717 716) ($Nx $Ny16 $Nzrib) simpleGrading (1
1533     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1533     hex (702 703 705 704 716 717 719 718) ($Nx $Ny25 $Nzrib) simpleGrading (1
1534     $Ry2 $Rzsx) // Blocco intermedio inferiore
1534     hex (704 705 707 706 718 719 721 720) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1535     $Rzsx) // Blocco centrale inferiore
1535     hex (706 707 709 708 720 721 723 722) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1536     $Rzsx) // Blocco centrale superiore
1536     hex (708 709 711 710 722 723 725 724) ($Nx $Ny25 $Nzrib) simpleGrading (1
1537     $Ry5 $Rzsx) // Blocco intermedio superiore
1537     hex (710 711 713 712 724 725 727 726) ($Nx $Ny16 $Nzrib) simpleGrading (1
1538     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1538 // Secondo lato
1539     hex (714 715 717 716 728 729 731 730) ($Nx $Ny16 $Nzrib) simpleGrading (1
1540     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1540     hex (716 717 719 718 730 731 733 732) ($Nx $Ny25 $Nzrib) simpleGrading (1
1541     $Ry2 $Rzdx) // Blocco intermedio inferiore
1541     hex (718 719 721 720 732 733 735 734) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1542     $Rzdx) // Blocco centrale inferiore
1542     hex (720 721 723 722 734 735 737 736) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1543     $Rzdx) // Blocco centrale superiore
1543     hex (722 723 725 724 736 737 739 738) ($Nx $Ny25 $Nzrib) simpleGrading (1
1544     $Ry5 $Rzdx) // Blocco intermedio superiore
1544     hex (724 725 727 726 738 739 741 740) ($Nx $Ny16 $Nzrib) simpleGrading (1
1545     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1545
1546 //Rib. 27
1547 // Primo lato
1548     hex (728 729 731 730 742 743 745 744) ($Nx $Ny16 $Nzrib) simpleGrading (1
1549     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1549     hex (730 731 733 732 744 745 747 746) ($Nx $Ny25 $Nzrib) simpleGrading (1
1550     $Ry2 $Rzsx) // Blocco intermedio inferiore
1550     hex (732 733 735 734 746 747 749 748) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1551     $Rzsx) // Blocco centrale inferiore
1551     hex (734 735 737 736 748 749 751 750) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1552     $Rzsx) // Blocco centrale superiore
1552     hex (736 737 739 738 750 751 753 752) ($Nx $Ny25 $Nzrib) simpleGrading (1
1553     $Ry5 $Rzsx) // Blocco intermedio superiore
1553     hex (738 739 741 740 752 753 755 754) ($Nx $Ny16 $Nzrib) simpleGrading (1
1554     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1554 // Secondo lato
1555     hex (742 743 745 744 756 757 759 758) ($Nx $Ny16 $Nzrib) simpleGrading (1
1556     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1556     hex (744 745 747 746 758 759 761 760) ($Nx $Ny25 $Nzrib) simpleGrading (1
1557     $Ry2 $Rzdx) // Blocco intermedio inferiore
1557     hex (746 747 749 748 760 761 763 762) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1558     $Rzdx) // Blocco centrale inferiore
1558     hex (748 749 751 750 762 763 765 764) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1559     $Rzdx) // Blocco centrale superiore
1559     hex (750 751 753 752 764 765 767 766) ($Nx $Ny25 $Nzrib) simpleGrading (1
1560     $Ry5 $Rzdx) // Blocco intermedio superiore
1560     hex (752 753 755 754 766 767 769 768) ($Nx $Ny16 $Nzrib) simpleGrading (1
1561     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1561
1562 //Rib. 28
1563 // Primo lato
1564     hex (756 757 759 758 770 771 773 772) ($Nx $Ny16 $Nzrib) simpleGrading (1
1565     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1565     hex (758 759 761 760 772 773 775 774) ($Nx $Ny25 $Nzrib) simpleGrading (1
1566     $Ry2 $Rzsx) // Blocco intermedio inferiore
1566     hex (760 761 763 762 774 775 777 776) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1567     $Rzsx) // Blocco centrale inferiore
1567     hex (762 763 765 764 776 777 779 778) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1568     $Rzsx) // Blocco centrale superiore
1568     hex (764 765 767 766 778 779 781 780) ($Nx $Ny25 $Nzrib) simpleGrading (1
1569     $Ry5 $Rzsx) // Blocco intermedio superiore
1569     hex (766 767 769 768 780 781 783 782) ($Nx $Ny16 $Nzrib) simpleGrading (1
1570     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1570 // Secondo lato

```

```

1571     hex (770 771 773 772 784 785 787 786) ($Nx $Ny16 $Nzrib) simpleGrading (1
1572     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1572     hex (772 773 775 774 786 787 789 788) ($Nx $Ny25 $Nzrib) simpleGrading (1
1573     $Ry2 $Rzdx) // Blocco intermedio inferiore
1573     hex (774 775 777 776 788 789 791 790) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1574     $Rzdx) // Blocco centrale inferiore
1574     hex (776 777 779 778 790 791 793 792) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1575     $Rzdx) // Blocco centrale superiore
1575     hex (778 779 781 780 792 793 795 794) ($Nx $Ny25 $Nzrib) simpleGrading (1
1576     $Ry5 $Rzdx) // Blocco intermedio superiore
1576     hex (780 781 783 782 794 795 797 796) ($Nx $Ny16 $Nzrib) simpleGrading (1
1577     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1577
1578 //Rib. 29
1579 // Primo lato
1580     hex (784 785 787 786 798 799 801 800) ($Nx $Ny16 $Nzrib) simpleGrading (1
1581     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1581     hex (786 787 789 788 800 801 803 802) ($Nx $Ny25 $Nzrib) simpleGrading (1
1582     $Ry2 $Rzsx) // Blocco intermedio inferiore
1582     hex (788 789 791 790 802 803 805 804) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1583     $Rzsx) // Blocco centrale inferiore
1583     hex (790 791 793 792 804 805 807 806) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1584     $Rzsx) // Blocco centrale superiore
1584     hex (792 793 795 794 806 807 809 808) ($Nx $Ny25 $Nzrib) simpleGrading (1
1585     $Ry5 $Rzsx) // Blocco intermedio superiore
1585     hex (794 795 797 796 808 809 811 810) ($Nx $Ny16 $Nzrib) simpleGrading (1
1586     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1586 // Secondo lato
1587     hex (798 799 801 800 812 813 815 814) ($Nx $Ny16 $Nzrib) simpleGrading (1
1588     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1588     hex (800 801 803 802 814 815 817 816) ($Nx $Ny25 $Nzrib) simpleGrading (1
1589     $Ry2 $Rzdx) // Blocco intermedio inferiore
1589     hex (802 803 805 804 816 817 819 818) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1590     $Rzdx) // Blocco centrale inferiore
1590     hex (804 805 807 806 818 819 821 820) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1591     $Rzdx) // Blocco centrale superiore
1591     hex (806 807 809 808 820 821 823 822) ($Nx $Ny25 $Nzrib) simpleGrading (1
1592     $Ry5 $Rzdx) // Blocco intermedio superiore
1592     hex (808 809 811 810 822 823 825 824) ($Nx $Ny16 $Nzrib) simpleGrading (1
1593     $Ry6 $Rzdx) // Blocco vicino a parete superiore
1593
1594 //Rib. 30
1595 // Primo lato
1596     hex (812 813 815 814 826 827 829 828) ($Nx $Ny16 $Nzrib) simpleGrading (1
1597     $Ry1 $Rzsx) // Blocco vicino a parete inferiore
1597     hex (814 815 817 816 828 829 831 830) ($Nx $Ny25 $Nzrib) simpleGrading (1
1598     $Ry2 $Rzsx) // Blocco intermedio inferiore
1598     hex (816 817 819 818 830 831 833 832) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1599     $Rzsx) // Blocco centrale inferiore
1599     hex (818 819 821 820 832 833 835 834) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1600     $Rzsx) // Blocco centrale superiore
1600     hex (820 821 823 822 834 835 837 836) ($Nx $Ny25 $Nzrib) simpleGrading (1
1601     $Ry5 $Rzsx) // Blocco intermedio superiore
1601     hex (822 823 825 824 836 837 839 838) ($Nx $Ny16 $Nzrib) simpleGrading (1
1602     $Ry6 $Rzsx) // Blocco vicino a parete superiore
1602 // Secondo lato
1603     hex (826 827 829 828 840 841 843 842) ($Nx $Ny16 $Nzrib) simpleGrading (1
1604     $Ry1 $Rzdx) // Blocco vicino a parete inferiore
1604     hex (828 829 831 830 842 843 845 844) ($Nx $Ny25 $Nzrib) simpleGrading (1
1605     $Ry2 $Rzdx) // Blocco intermedio inferiore
1605     hex (830 831 833 832 844 845 847 846) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1606     $Rzdx) // Blocco centrale inferiore
1606     hex (832 833 835 834 846 847 849 848) ($Nx $Ny34 $Nzrib) simpleGrading (1 1
1607     $Rzdx) // Blocco centrale superiore
1607     hex (834 835 837 836 848 849 851 850) ($Nx $Ny25 $Nzrib) simpleGrading (1
1608     $Ry5 $Rzdx) // Blocco intermedio superiore
1608     hex (836 837 839 838 850 851 853 852) ($Nx $Ny16 $Nzrib) simpleGrading (1
1609     $Ry6 $Rzdx) // Blocco vicino a parete superiore

```

```

1610 ); // Fine blocchi
1611
1612 boundary
1613 ( inlet
1614     { type patch;
1615       faces (
1616 (2 0 14 16) (4 2 16 18) (6 4 18 20) (8 6 20 22) (10 8 22 24) (12 10 24 26)
//sx rib. 1
1617 (16 14 28 30) (18 16 30 32) (20 18 32 34) (22 20 34 36) (24 22 36 38) (26 24 38
40) //dx rib. 1
1618 (30 28 42 44) (32 30 44 46) (34 32 46 48) (36 34 48 50) (38 36 50 52) (40 38 52
54) //sx rib. 2
1619 (44 42 56 58) (46 44 58 60) (48 46 60 62) (50 48 62 64) (52 50 64 66) (54 52 66
68) //dx rib. 2
1620 (58 56 70 72) (60 58 72 74) (62 60 74 76) (64 62 76 78) (66 64 78 80) (68 66 80
82) //sx rib. 3
1621 (72 70 84 86) (74 72 86 88) (76 74 88 90) (78 76 90 92) (80 78 92 94) (82 80 94
96) //dx rib. 3
1622 (86 84 98 100) (88 86 100 102) (90 88 102 104) (92 90 104 106) (94 92 106 108)
(96 94 108 110) //sx rib. 4
1623 (100 98 112 114) (102 100 114 116) (104 102 116 118) (106 104 118 120) (108 106
120 122) (110 108 122 124) //dx rib. 4
1624 (114 112 126 128) (116 114 128 130) (118 116 130 132) (120 118 132 134) (122 120
134 136) (124 122 136 138) //sx rib. 5
1625 (128 126 140 142) (130 128 142 144) (132 130 144 146) (134 132 146 148) (136 134
148 150) (138 136 150 152) //dx rib. 5
1626 (142 140 154 156) (144 142 158 156) (146 144 160 158) (148 146 160 162) (150 148
162 164) (152 150 164 166) //sx rib. 6
1627 (156 154 168 170) (158 156 170 172) (160 158 172 174) (162 160 174 176) (164 162
176 178) (166 164 178 180) //dx rib. 6
1628 (170 168 182 184) (172 170 184 186) (174 172 186 188) (176 174 188 190) (178 176
190 192) (180 178 192 194) //sx rib. 7
1629 (184 182 196 198) (186 184 198 200) (188 186 200 202) (190 188 202 204) (192 190
204 206) (194 192 206 208) //dx rib. 7
1630 (198 196 210 212) (200 198 212 214) (202 200 214 216) (204 202 216 218) (206 204
218 220) (208 206 220 222) //sx rib. 8
1631 (212 210 224 226) (214 212 226 228) (216 214 228 230) (218 216 230 232) (220 218
232 234) (222 220 234 236) //dx rib. 8
1632 (226 224 238 240) (228 226 240 242) (230 228 242 244) (232 230 244 246) (234 232
246 248) (236 234 248 250) //sx rib. 9
1633 (240 238 252 254) (242 240 254 256) (244 242 256 258) (246 244 258 260) (248 246
260 262) (250 248 262 264) //dx rib. 9
1634 (254 252 266 268) (256 254 268 270) (258 256 270 272) (260 258 272 274) (262 260
274 276) (264 262 276 278) //sx rib. 10
1635 (268 266 280 282) (270 268 282 284) (272 270 284 286) (274 272 286 288) (276 274
288 290) (278 276 290 292) //dx rib. 10
1636 (282 280 294 296) (284 282 296 298) (286 284 298 300) (288 286 300 302) (290 288
302 304) (292 290 304 306) //sx rib. 11
1637 (296 294 308 310) (298 296 310 312) (300 298 312 314) (302 300 314 316) (304 302
316 318) (306 304 318 320) //dx rib. 11
1638 (310 308 322 324) (312 310 324 326) (314 312 326 328) (316 314 328 330) (318 316
330 332) (320 318 332 334) //sx rib. 12
1639 (324 322 336 338) (326 324 338 340) (328 326 340 342) (330 328 342 344) (332 330
344 346) (334 332 346 348) //dx rib. 12
1640 (338 336 350 352) (340 338 352 354) (342 340 354 356) (344 342 356 358) (346 344
358 360) (348 346 360 362) //sx rib. 13
1641 (352 350 364 366) (354 352 366 368) (356 354 368 370) (358 356 370 372) (360 358
372 374) (362 360 374 376) //dx rib. 13
1642 (366 364 378 380) (368 366 380 382) (370 368 382 384) (372 370 384 386) (374 372
386 388) (376 374 388 390) //sx rib. 14
1643 (380 378 392 394) (382 380 394 396) (384 382 396 398) (386 384 398 400) (388 386
400 402) (390 388 402 404) //dx rib. 14
1644 (394 392 406 408) (396 394 408 410) (398 396 410 412) (400 398 412 414) (402 400
414 416) (404 402 416 418) //sx rib. 15
1645 (408 406 420 422) (410 408 422 424) (412 410 424 426) (414 412 426 428) (416 414
428 430) (418 416 430 432) //dx rib. 15
1646 (422 420 434 436) (424 422 436 438) (426 424 438 440) (428 426 440 442) (430 428
442 444) (432 430 444 446) //sx rib. 16

```

```

1647 (436 434 448 450) (438 436 450 452) (440 438 452 454) (442 440 454 456) (444 442
    456 458) (446 444 458 460) //dx rib. 16
1648 (450 448 462 464) (452 450 464 466) (454 452 466 468) (456 454 468 470) (458 456
    470 472) (460 458 472 474) //sx rib. 17
1649 (464 462 476 478) (466 464 478 480) (468 466 480 482) (470 468 482 484) (472 470
    484 486) (474 472 486 488) //dx rib. 17
1650 (478 476 490 492) (480 478 492 494) (482 480 494 496) (484 482 496 498) (486 484
    498 500) (488 486 500 502) //sx rib. 18
1651 (492 490 504 506) (494 492 506 508) (496 494 508 510) (498 496 510 512) (500 498
    512 514) (502 500 514 516) //dx rib. 18
1652 (506 504 518 520) (508 506 520 522) (510 508 522 524) (512 510 524 526) (514 512
    526 528) (516 514 528 530) //sx rib. 19
1653 (520 518 532 534) (522 520 534 536) (524 522 536 538) (526 524 538 540) (528 526
    540 542) (530 528 542 544) //dx rib. 19
1654 (534 532 546 548) (536 534 548 550) (538 536 550 552) (540 538 552 554) (542 540
    554 556) (544 542 556 558) //sx rib. 20
1655 (548 546 560 562) (550 548 562 564) (552 550 564 566) (554 552 566 568) (556 554
    568 570) (558 556 570 572) //dx rib. 20
1656 (562 560 574 576) (564 562 576 578) (566 564 578 580) (568 566 580 582) (570 568
    582 584) (572 570 584 586) //sx rib. 21
1657 (576 574 588 590) (578 576 590 592) (580 578 592 594) (582 580 594 596) (584 582
    596 598) (586 584 598 600) //dx rib. 21
1658 (590 588 602 604) (592 590 604 606) (594 592 606 608) (596 594 608 610) (598 596
    610 612) (600 598 612 614) //sx rib. 22
1659 (604 602 616 618) (606 604 618 620) (608 606 620 622) (610 608 622 624) (612 610
    624 626) (614 612 626 628) //dx rib. 22
1660 (618 616 630 632) (620 618 632 634) (622 620 634 636) (624 622 636 638) (626 624
    638 640) (628 626 640 642) //sx rib. 23
1661 (632 630 644 646) (634 632 646 648) (636 634 648 650) (638 636 650 652) (640 638
    652 654) (642 640 654 656) //dx rib. 23
1662 (646 644 658 660) (648 646 660 662) (650 648 662 664) (652 650 664 666) (654 652
    666 668) (656 654 668 670) //sx rib. 24
1663 (660 658 672 674) (662 660 674 676) (664 662 676 678) (666 664 678 680) (668 666
    680 682) (670 668 682 684) //dx rib. 24
1664 (674 672 686 688) (676 674 688 690) (678 676 690 692) (680 678 692 694) (682 680
    694 696) (684 682 696 698) //sx rib. 25
1665 (688 686 700 702) (690 688 702 704) (692 690 704 706) (694 692 706 708) (696 694
    708 710) (698 696 710 712) //dx rib. 25
1666 (702 700 714 716) (704 702 716 718) (706 704 718 720) (708 706 720 722) (710 708
    722 724) (712 710 724 726) //sx rib. 26
1667 (716 714 728 730) (718 716 730 732) (720 718 732 734) (722 720 734 736) (724 722
    736 738) (726 724 738 740) //dx rib. 26
1668 (730 728 742 744) (732 730 744 746) (734 732 746 748) (736 734 748 750) (738 736
    750 752) (740 738 752 754) //sx rib. 27
1669 (744 742 756 758) (746 744 758 760) (748 746 760 762) (750 748 762 764) (752 750
    764 766) (754 752 766 768) //dx rib. 27
1670 (758 756 770 772) (760 758 772 774) (762 760 774 776) (764 762 776 778) (766 764
    778 780) (768 766 780 782) //sx rib. 28
1671 (772 770 784 786) (774 772 786 788) (776 774 788 790) (778 776 790 792) (780 778
    792 794) (782 780 794 796) //dx rib. 28
1672 (786 784 798 800) (788 786 800 802) (790 788 802 804) (792 790 804 806) (794 792
    806 808) (796 794 808 810) //sx rib. 29
1673 (800 798 812 814) (802 800 814 816) (804 802 816 818) (806 804 818 820) (808 806
    820 822) (810 808 822 824) //dx rib. 29
1674 (814 812 826 828) (816 814 828 830) (818 816 830 832) (820 818 832 834) (822 820
    834 836) (824 822 836 838) //sx rib. 30
1675 (828 826 840 842) (830 828 842 844) (832 830 844 846) (834 832 846 848) (836 834
    848 850) (838 836 850 852) //dx rib. 30
1676 );
1677 } // Fine inlet
1678
1679 outlet
1680 { type patch;
1681 faces (
1682 (1 3 17 15) (3 5 19 17) (5 7 21 19) (7 9 23 21) (9 11 25 23) (11 13 27 25)
    //sx rib. 1
1683 (15 17 31 29) (17 19 33 31) (19 21 35 33) (21 23 37 35) (23 25 39 37) (25 27 41
    39) //dx rib. 1

```

```

1684 (29 31 45 43) (31 33 47 45) (33 35 49 47) (35 37 51 49) (37 39 53 51) (39 41 55
      53) //sx rib. 2
1685 (43 45 59 57) (45 47 61 59) (47 49 63 61) (49 51 65 63) (51 53 67 65) (53 55 69
      67) //dx rib. 2
1686 (57 59 73 71) (59 61 75 73) (61 63 77 75) (63 65 79 77) (65 67 81 79) (67 69 83
      81) //sx rib. 3
1687 (71 73 87 85) (73 75 89 87) (75 77 91 89) (77 79 93 91) (79 81 95 93) (81 83 97
      95) //dx rib. 3
1688 (85 87 101 99) (87 89 103 101) (89 91 105 103) (91 93 107 105) (93 95 109 107)
      (95 97 111 109) //sx rib. 4
1689 (99 101 115 113) (101 103 117 115) (103 105 119 117) (105 107 121 119) (107 109
      123 121) (109 111 125 123) //dx rib. 4
1690 (113 115 129 127) (115 117 131 129) (117 119 133 131) (119 121 135 133) (121 123
      137 135) (123 125 139 137) //sx rib. 5
1691 (127 129 143 141) (129 131 145 143) (131 133 147 145) (133 135 149 147) (135 137
      151 149) (137 139 153 151) //dx rib. 5
1692 (141 143 157 155) (143 145 159 157) (145 147 161 159) (147 149 163 161) (149 151
      165 163) (151 153 167 165) //sx rib. 6
1693 (155 157 171 169) (157 159 173 171) (159 161 175 173) (161 163 177 175) (163 165
      179 177) (165 167 181 179) //dx rib. 6
1694 (169 171 185 183) (171 173 187 185) (173 175 189 187) (175 177 191 189) (177 179
      193 191) (179 181 195 193) //sx rib. 7
1695 (183 185 199 197) (185 187 201 199) (187 189 203 201) (189 191 205 203) (191 193
      207 205) (193 195 209 207) //dx rib. 7
1696 (197 199 213 211) (199 201 215 213) (201 203 217 215) (203 205 219 217) (205 207
      221 219) (207 209 223 221) //sx rib. 8
1697 (211 213 227 225) (213 215 229 227) (215 217 231 229) (217 219 233 231) (219 221
      235 233) (221 223 237 235) //dx rib. 8
1698 (225 227 241 239) (227 229 243 241) (229 231 245 243) (231 233 247 245) (233 235
      249 247) (235 237 251 249) //sx rib. 9
1699 (239 241 255 253) (241 243 257 255) (243 245 259 257) (245 247 261 259) (247 249
      263 261) (249 251 265 263) //dx rib. 9
1700 (253 255 269 267) (255 257 271 269) (257 259 273 271) (259 261 275 273) (261 263
      277 275) (263 265 279 277) //sx rib. 10
1701 (267 269 283 281) (269 271 285 283) (271 273 287 285) (273 275 289 287) (275 277
      291 289) (277 279 293 291) //dx rib. 10
1702 (281 283 297 295) (283 285 299 297) (285 287 301 299) (287 289 303 301) (289 291
      305 303) (291 293 307 305) //sx rib. 11
1703 (295 297 311 309) (297 299 313 311) (299 301 315 313) (301 303 317 315) (303 305
      319 317) (305 307 321 319) //dx rib. 11
1704 (309 311 325 323) (311 313 327 325) (313 315 329 327) (315 317 331 329) (317 319
      333 331) (319 321 335 333) //sx rib. 12
1705 (323 325 339 337) (325 327 341 339) (327 329 343 341) (329 331 345 343) (331 333
      347 345) (333 335 349 347) //dx rib. 12
1706 (337 339 353 351) (339 341 355 353) (341 343 357 355) (343 345 359 357) (345 347
      361 359) (347 349 363 361) //sx rib. 13
1707 (351 353 367 365) (353 355 369 367) (355 357 371 369) (357 359 373 371) (359 361
      375 373) (361 363 377 375) //dx rib. 13
1708 (365 367 381 379) (367 369 383 381) (369 371 385 383) (371 373 387 385) (373 375
      389 387) (375 377 391 389) //sx rib. 14
1709 (379 381 395 393) (381 383 397 395) (383 385 399 397) (385 387 401 399) (387 389
      403 401) (389 391 405 403) //dx rib. 14
1710 (393 395 409 407) (395 397 411 409) (397 399 413 411) (399 401 415 413) (401 403
      417 415) (403 405 419 417) //sx rib. 15
1711 (407 409 423 421) (409 411 425 423) (411 413 427 425) (413 415 429 427) (415 417
      431 429) (417 419 433 431) //dx rib. 15
1712 (421 423 437 435) (423 425 439 437) (425 427 441 439) (427 429 443 441) (429 431
      445 443) (431 433 447 445) //sx rib. 16
1713 (435 437 451 449) (437 439 453 451) (439 441 455 453) (441 443 457 455) (443 445
      459 457) (445 447 461 459) //dx rib. 16
1714 (449 451 465 463) (451 453 467 465) (453 455 469 467) (455 457 471 469) (457 459
      473 471) (459 461 475 473) //sx rib. 17
1715 (463 465 479 477) (465 467 481 479) (467 469 483 481) (469 471 485 483) (471 473
      487 485) (473 475 489 487) //dx rib. 17
1716 (477 479 493 491) (479 481 495 493) (481 483 497 495) (483 485 499 497) (485 487
      501 499) (487 489 503 501) //sx rib. 18
1717 (491 493 507 505) (493 495 509 507) (495 497 511 509) (497 499 513 511) (499 501
      515 513) (501 503 517 515) //dx rib. 18

```

```

1718 (505 507 521 519) (507 509 523 521) (509 511 525 523) (511 513 527 525) (513 515
      529 527) (515 517 531 529) //sx rib. 19
1719 (519 521 535 533) (521 523 537 535) (523 525 539 537) (525 527 541 539) (527 529
      543 541) (529 531 545 543) //dx rib. 19
1720 (533 535 549 547) (535 537 551 549) (537 539 553 551) (539 541 555 553) (541 543
      557 555) (543 545 559 557) //sx rib. 20
1721 (547 549 563 561) (549 551 565 563) (551 553 567 565) (553 555 569 567) (555 557
      571 569) (557 559 573 571) //dx rib. 20
1722 (561 563 577 575) (563 565 579 577) (565 567 581 579) (567 569 583 581) (569 571
      585 583) (571 573 587 585) //sx rib. 21
1723 (575 577 591 589) (577 579 593 591) (579 581 595 593) (581 583 597 595) (583 585
      599 597) (585 587 601 599) //dx rib. 21
1724 (589 591 605 603) (591 593 607 605) (593 595 609 607) (595 597 611 609) (597 599
      613 611) (599 601 615 613) //sx rib. 22
1725 (603 605 619 617) (605 607 621 619) (607 609 623 621) (609 611 625 623) (611 613
      627 625) (613 615 629 627) //dx rib. 22
1726 (617 619 633 631) (619 621 635 633) (621 623 637 635) (623 625 639 637) (625 627
      641 639) (627 629 643 641) //sx rib. 23
1727 (631 633 647 645) (633 635 649 647) (635 637 651 649) (637 639 653 651) (639 641
      655 653) (641 643 657 655) //dx rib. 23
1728 (645 647 661 659) (647 649 663 661) (649 651 665 663) (651 653 667 665) (653 655
      669 667) (655 657 671 669) //sx rib. 24
1729 (659 661 675 673) (661 663 677 675) (663 665 679 677) (665 667 681 679) (667 669
      683 681) (669 671 685 683) //dx rib. 24
1730 (673 675 689 687) (675 677 691 689) (677 679 693 691) (679 681 695 693) (681 683
      697 695) (683 685 699 697) //sx rib. 25
1731 (687 689 703 701) (689 691 705 703) (691 693 707 705) (693 695 709 707) (695 697
      711 709) (697 699 713 711) //dx rib. 25
1732 (701 703 717 715) (703 705 719 717) (705 707 721 719) (707 709 723 721) (709 711
      725 723) (711 713 727 725) //sx rib. 26
1733 (715 717 731 729) (717 719 733 731) (719 721 735 733) (721 723 737 735) (723 725
      739 737) (725 727 741 739) //dx rib. 26
1734 (729 731 745 743) (731 733 747 745) (733 735 749 747) (735 737 751 749) (737 739
      753 751) (739 741 755 753) //sx rib. 27
1735 (743 745 759 757) (745 747 761 759) (747 749 763 761) (749 751 765 763) (751 753
      767 765) (753 755 769 767) //dx rib. 27
1736 (757 759 773 771) (759 761 775 773) (761 763 777 775) (763 765 779 777) (765 767
      781 779) (767 769 783 781) //sx rib. 28
1737 (771 773 787 785) (773 775 789 787) (775 777 791 789) (777 779 793 791) (779 781
      795 793) (781 783 797 795) //dx rib. 28
1738 (785 787 801 799) (787 789 803 801) (789 791 805 803) (791 793 807 805) (793 795
      809 807) (795 797 811 809) //sx rib. 29
1739 (799 801 815 813) (801 803 817 815) (803 805 819 817) (805 807 821 819) (807 809
      823 821) (809 811 825 823) //dx rib. 29
1740 (813 815 829 827) (815 817 831 829) (817 819 833 831) (819 821 835 833) (821 823
      837 835) (823 825 839 837) //sx rib. 30
1741 (827 829 843 841) (829 831 845 843) (831 833 847 845) (833 835 849 847) (835 837
      851 849) (837 839 853 851) //dx rib. 30
1742 );
1743 } // Fine outlet
1744
1745 bottomWall
1746 { type wall;
1747   faces (
1748 (0 1 15 14) (14 15 29 28) // rib. 1
1749 (28 29 43 42) (42 43 57 56) // rib. 2
1750 (56 57 71 70) (70 71 85 84) // rib. 3
1751 (84 85 99 98) (98 99 113 112) // rib. 4
1752 (112 113 127 126) (126 127 141 140) // rib. 5
1753 (140 141 155 154) (154 155 169 168) // rib. 6
1754 (168 169 183 182) (182 183 197 196) // rib. 7
1755 (196 197 211 210) (210 211 225 224) // rib. 8
1756 (224 225 239 238) (238 239 253 252) // rib. 9
1757 (252 253 267 266) (266 267 281 280) // rib. 10
1758 (280 281 295 294) (294 295 309 308) // rib. 11
1759 (308 309 323 322) (322 323 337 336) // rib. 12
1760 (336 337 351 350) (350 351 365 364) // rib. 13
1761 (364 365 379 378) (378 379 393 392) // rib. 14
1762 (392 393 407 406) (406 407 421 420) // rib. 15

```

```

1763 (420 421 435 434) (434 435 449 448) // rib. 16
1764 (448 449 463 462) (462 463 477 476) // rib. 17
1765 (476 477 491 490) (490 491 505 504) // rib. 18
1766 (504 505 519 518) (518 519 533 532) // rib. 19
1767 (532 533 547 546) (546 547 561 560) // rib. 20
1768 (560 561 575 574) (574 575 589 588) // rib. 21
1769 (588 589 603 602) (602 603 617 616) // rib. 22
1770 (616 617 631 630) (630 631 645 644) // rib. 23
1771 (644 645 659 658) (658 659 673 672) // rib. 24
1772 (672 673 687 686) (686 687 701 700) // rib. 25
1773 (700 701 715 714) (714 715 729 728) // rib. 26
1774 (728 729 743 742) (742 743 757 756) // rib. 27
1775 (756 757 771 770) (770 771 785 784) // rib. 28
1776 (784 785 799 798) (798 799 813 812) // rib. 29
1777 (812 813 827 826) (826 827 841 840) // rib. 30
1778 );
1779 } // Fine bottomWall
1780
1781 topWall
1782 { type wall;
1783 faces (
1784 (13 12 26 27) (27 26 40 41) // rib. 1
1785 (41 40 54 55) (55 54 68 69) // rib. 2
1786 (69 68 82 83) (83 82 96 97) // rib. 3
1787 (97 96 110 111) (111 110 124 125) // rib. 4
1788 (125 124 138 139) (139 138 152 153) // rib. 5
1789 (153 152 166 167) (167 166 180 181) // rib. 6
1790 (181 180 194 195) (195 194 208 209) // rib. 7
1791 (209 208 222 223) (223 222 236 237) // rib. 8
1792 (237 236 250 251) (251 250 264 265) // rib. 9
1793 (265 264 278 279) (279 278 292 293) // rib. 10
1794 (293 292 306 307) (307 306 320 321) // rib. 11
1795 (321 320 334 335) (335 334 348 349) // rib. 12
1796 (349 348 362 363) (363 362 376 377) // rib. 13
1797 (377 376 390 391) (391 390 404 405) // rib. 14
1798 (405 404 418 419) (419 418 432 433) // rib. 15
1799 (433 432 446 447) (447 446 460 461) // rib. 16
1800 (461 460 474 475) (475 474 488 489) // rib. 17
1801 (489 488 502 503) (503 502 516 517) // rib. 18
1802 (517 516 530 531) (531 530 544 545) // rib. 19
1803 (545 544 558 559) (559 558 572 573) // rib. 20
1804 (573 572 586 587) (587 586 600 601) // rib. 21
1805 (601 600 614 615) (615 614 628 629) // rib. 22
1806 (629 628 642 643) (643 642 656 657) // rib. 23
1807 (657 656 670 671) (671 670 684 685) // rib. 24
1808 (685 684 698 699) (699 698 712 713) // rib. 25
1809 (713 712 726 727) (727 726 740 741) // rib. 26
1810 (741 740 754 755) (755 754 768 769) // rib. 27
1811 (769 768 782 783) (783 782 796 797) // rib. 28
1812 (797 796 810 811) (811 810 824 825) // rib. 29
1813 (825 824 838 839) (839 838 852 853) // rib. 30
1814 );
1815 } // Fine topWall
1816
1817 left //faccia dietro
1818 { type cyclic;
1819 neighbourPatch right;
1820 faces (
1821 // queste faccie sono sempre le stesse, anche se cambia il numero di riblet
1822 (1 0 2 3) (3 2 4 5) (5 4 6 7) (7 6 8 9) (9 8 10 11) (11 10 12 13)
1823 );
1824 }
1825
1826 right // faccia avanti
1827 { type cyclic;
1828 neighbourPatch left;
1829 faces (
1830 // queste faccie devono essere prese ogni volta sull ultima riblet, vedendo i
punti dell ultimo lato

```

```

1831 (840 841 843 842) (842 843 845 844) (844 845 847 846) (846 847 849 848) (848
      849 851 850) (850 851 853 852)
1832 );
1833 }
1834 ); // Fine definizioni condizioni al contorno
1835
1836 mergePatchPairs
1837 (
1838 );
1839 // ***** //

```

Modifiche al blockMeshDict

Per andare a simulare le altre configurazioni riportate in questo studio, è necessario andare a cambiare solo il valore riportato nella riga 47 (o eventualmente anche i commenti nelle righe 46 e 48, come promemoria), dove si specifica l'altezza delle riblet. Si riporta l'esempio per la configurazione *H15*.

```

1 // rapp hr su br/2 = 1
2 hr 0.075; // Altezza della punta della riblet
3 //hr+ = 15

```

Se invece si vogliono andare a cambiare le condizioni al contorno, per passare dal nostro caso iniziale (quello riportato) al nostro caso periodico, si devono andare a cambiare i comandi riportati nelle righe 1613-1614 e quelli nelle righe 1679-1680, rispettivamente con:

righe 1613-1614

```

1 ( inlet
2   { type cyclic;
3     neighbourPatch outlet;

```

righe 1679-1680

```

1 outlet
2 { type cyclic;
3   neighbourPatch inlet;

```

A.2 controlDictRun

Il secondo codice che si trova nella cartella *system*, un altro dei più importanti, è il codice *controlDict*, che qui viene chiamato *controlDictRun*, per distinguerlo da quello utilizzato per il post processing (vedi B.1). Tuttavia, quando la simulazione deve essere eseguita, è necessario che questo codice, si trovi nella cartella *system*, e si chiami *controlDict*. In questo codice vengono riportate tutte le informazioni relative al tempo per la simulazione da eseguire, oltre al tipo di simulazione da utilizzare, ed informazioni in merito alla precisione numerica. Si possono anche inserire funzioni per avere altri strumenti di calcolo, come nel caso del postprocessing (vedi B.1), ma nel caso di questo codice, non si aggiunge nulla.

Codici_simulazione/controlDictRun.txt

```

1 /*-----* C++ *-----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: 2.2.0 |
5 | \ \ / A n d | Web: www.OpenFOAM.org |
6 | \ \ / M a n i p u l a t i o n | |
7 /*-----*/
8 FoamFile
9 {
10   version 2.0;

```



```

5 |  \ \ /   A nd           | Web:      www.OpenFOAM.org           |
6 |  \ \ /   M anipulation |                               |
7 \*-----*
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        decomposeParDict;
15 }
16 // * * * * *
17 numberOfSubdomains 24;
18 method              simple;
19 simpleCoeffs
20 {
21     n                (6 2 2);
22     delta            0.001;
23 }
24 hierarchicalCoeffs
25 {
26     n                (1 1 1);
27     delta            0.001;
28     order            xyz;
29 }
30 manualCoeffs
31 {
32     dataFile         "";
33 }
34 distributed        no;
35 roots              ( );
36 // * * * * *

```

A.3.2 fvSchemes

Nel codice *fvSchemes* si vanno ad indicare i solutori che si vogliono utilizzare nella simulazione, per discretizzare le derivate temporali, i gradienti, le divergenze, i laplaciani, le interpolazioni, ed altre discretizzazioni che sono necessarie per le simulazioni che si vogliono svolgere.

Codici_simulazione/fvSchemes.txt

```

1 /*-----* C++ *-----*\
2 | ===== |
3 |  \ \ /   F ield           | OpenFOAM: The Open Source CFD Toolbox |
4 |  \ \ /   O peration      | Version:  2.2.0           |
5 |  \ \ /   A nd           | Web:      www.OpenFOAM.org           |
6 |  \ \ /   M anipulation |                               |
7 \*-----*
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        fvSchemes;
15 }
16 // * * * * *
17 ddtSchemes
18 {  default          backward;
19 }
20
21 gradSchemes
22 {  default          Gauss linear;
23     grad(p)         Gauss linear;
24 }
25

```

```

26 divSchemes
27 {   default          Gauss linear;
28     div(phi,U)       Gauss linear;
29 }
30
31 laplacianSchemes
32 {   default          Gauss linear corrected;
33 }
34
35 interpolationSchemes
36 {   default          linear;
37 }
38
39 snGradSchemes
40 {   default          corrected;
41 }
42
43 fluxRequired
44 {   default          no;
45     p                ;
46 }
47 // ***** //

```

A.3.3 fvSolution

Nel codice *fvSolutions* si vanno ad indicare i solutori che si vogliono utilizzare nella simulazione, per calcolare le grandezze fisiche di interesse, presenti nella simulazione, quali pressione, velocità, ed altre che possono eventualmente servire. Si indica anche, per ognuna di queste grandezze, l'ordine di accuratezza con cui si vuole che le cifre siano riportate. Infine si vanno ad indicare eventuali strumenti correttori, da applicare al tipo di simulazione scelta nel *controlDictRun*, vedi A.2.

Codici_simulazione/fvSolution.txt

```

1  /*-----* C++ *-----*/
2  | ===== |
3  | \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
4  | \\      / O peration  | Version: 2.2.0 |
5  | \\      / A nd        | Web: www.OpenFOAM.org |
6  | \\      / M anipulation | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       fvSolution;
15 }
16 // ***** //
17 solvers
18 {
19     p
20     {
21         solver          PCG;
22         preconditioner  DIC;
23         tolerance       1e-06;
24         relTol          0.05;
25     }
26
27     pFinal
28     {
29         solver          PCG;
30         preconditioner  DIC;
31         tolerance       1e-06;
32         relTol          0;
33     }
34
35     U
36     {
37         solver          PBiCG;

```

```

34     preconditioner  DILU;
35     tolerance      1e-05;
36     relTol         0.1;
37 }
38
39 UFinal
40 {
41     $U;
42     tolerance      1e-05;
43     relTol         0;
44 }
45 }
46
47 PISO
48 {  nCorrectors      2;
49     nNonOrthogonalCorrectors 0;
50     pRefCell        0;
51     pRefValue       0;
52 }
53 // *****

```

A.3.4 mapFieldsDict

Nel codice *mapFieldsDict* si vanno ad inserire le eventuali opzioni che possono servire per fare la mappatura da una simulazione ad un'altra. Questo file deve essere presente, solamente nel caso si vada a fare la mappatura non *consistent*, ovvero quella in cui nel comando `mapFields` non è presente l'opzione `-consistent`, e che si utilizza per mappare un campo di moto su di un'altra simulazione, ma con condizioni al contorno differenti.

Codici_simulazione/mapFieldsDict.txt

```

1  /-----* C++ *-----\
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: 5 |
5  | \ \ / A n d | Web: www.OpenFOAM.org |
6  | \ \ / M a n i p u l a t i o n | |
7  \-----*-----\
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       mapFieldsDict;
15 }
16 // *****
17 patchMap      ( );
18 cuttingPatches ( );
19 // *****

```

A.4 Codici della cartella constant

Vengono riportati ora i codici, che devono trovarsi, se servono, nella cartella *constant*, per poter eseguire le simulazioni. I codici che andiamo a riportare sono:

- transportProperties
- turbulenceProperties
- fvOptions

A.4.1 transportProperties

Nel codice *transportProperties* si vanno ad inserire le proprietà fisiche del fluido che si vuole simulare. Nel nostro caso si è inserirà solamente la viscosità cinematica ν (avendo considerato la densità $\rho = 1$), e non essendoci altre proprietà che fossero utili (come per esempio la conducibilità termica, che a noi non serve).

Codici_simulazione/transportProperties.txt

```

1 /*----- C++ -----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: 2.2.0 |
5 | \ \ / A n d | Web: www.OpenFOAM.org |
6 | \ \ / M a n i p u l a t i o n | |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "constant";
14     object       transportProperties;
15 }
16 // * * * * * //
17 transportModel  Newtonian;
18 nu              nu [ 0 2 -1 0 0 0 ] 3.18e-4;
19 // viscosità cinematica adimensionalizzata, per Re corrispondente a Re tau 200
20 // * * * * * //

```

A.4.2 turbulenceProperties

Nel codice *turbulenceProperties* (se presente) si vanno ad inserire le eventuali proprietà o caratteristiche del flusso turbolento che si va a simulare, ed anche il modello di turbolenza da utilizzare, se richiesto dal solutore. Nel caso del nostro solutore, *pisoFoam*, questo richiede un modello di turbolenza, ma volendo noi eseguire DNS, si inserisce come modello *laminar*, ovvero nessun modello di turbolenza (e quindi il programma esegue DNS).

Codici_simulazione/turbulenceProperties.txt

```

1 /*----- C++ -----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: 5 |
5 | \ \ / A n d | Web: www.OpenFOAM.org |
6 | \ \ / M a n i p u l a t i o n | |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "constant";
14     object       turbulenceProperties;
15 }
16 // * * * * * //
17 simulationType  laminar;
18 // * * * * * //

```

A.4.3 fvOptions

Nel codice *fvOptions* (se presente) si vanno ad inserire eventuali altre opzioni di personalizzazione che si vogliono fornire alla simulazione. Nel nostro caso si è utilizzati

questo codice, solamente nelle simulazioni con condizioni al contorno periodiche, per imporre al flusso il gradiente di pressione che sostenesse il moto.

Codici_simulazione/fvOptions.txt

```

1 /*-----*-- C++ *-----*\
2 | ===== |
3 | \\ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4 | \\ / O peration | Version: 5 |
5 | \\ / A nd | Web: www.OpenFOAM.org |
6 | \\ / M anipulation |
7 \*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "constant";
14     object       fvOptions;
15 }
16 // ***** //
17 momentumSource
18 {
19     type          meanVelocityForce;
20     selectionMode all;
21     fields        (U);
22     Ubar          (1 0 0);
23 }
24 // ***** //

```

A.5 Codici della cartella I istante temporale

Vengono riportati ora i codici, che devono trovarsi, nella cartella chiamata come il primo istante temporale della simulazione (se questa è presente), per poter eseguire le simulazioni. I codici presenti in questa cartella sono essenzialmente le condizioni al contorno e/o quelle iniziali per tutte le grandezze che si vogliono simulare. Nel nostro caso sono:

- U
- p

A.5.1 U e P

Nei codici U e p (che sono gli unici utilizzati nel nostro caso) si vanno ad inserire le condizioni al contorno e quelle iniziali che si hanno per queste grandezze nella simulazione da eseguire. Questi devono trovarsi nella cartella chiamata come il primo istante temporale, in quanto rappresentano le proprietà del flusso in quell'istante temporale. Laddove la cartella non è presente, vuol dire che questi due codici saranno estratti, tramite mappatura da un'altra simulazione (nel caso **consistent**). Invece, nel caso si voglia mappare un'altra simulazione, ma cambiando le condizioni al contorno, questi file (e la cartella che li contiene) devono essere presenti, ed al loro interno devono essere riportate solamente le nuove condizioni al contorno, senza invece riportare nessuna condizione iniziale (che viene invece mappata). Si riportano ora sia la versione di questi file per le simulazioni con condizioni iniziali, e sia la versione per le condizioni periodiche. I codici che vengono ora riportati, sono chiamati, per poterli distinguere, U_{in} e p_{in} quelli della condizione iniziale, e U_{per} e p_{per} quelli della condizione periodica. Nella simulazione, però, affinché vengano letti, è necessario che si chiamino solamente U e p .

Condizioni iniziali

Codici_simulazione/U_in.txt

```

1 /*-----* C++ *-----*\
2 | ===== |
3 | \\ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4 | \\ / O peration | Version: 2.3.1 |
5 | \\ / A nd | Web: www.OpenFOAM.org |
6 | \\/ M anipulation |
7 /*-----*
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        volVectorField;
13     location     "0";
14     object       U;
15 }
16 // ***** //
17 dimensions      [0 1 -1 0 0 0];
18 internalField   uniform (0 0 0);
19 boundaryField
20 {
21     inlet
22     {
23         type      fixedValue;
24         value     uniform (1.0 0 0);
25     }
26     outlet
27     {
28         type      zeroGradient;
29     }
30     bottomWall
31     {
32         type      fixedValue;
33         value     uniform (0 0 0);
34     }
35     topWall
36     {
37         type      fixedValue;
38         value     uniform (0 0 0);
39     }
40     left
41     {
42         type      cyclic;
43     }
44     right
45     {
46         type      cyclic;
47     }
48 }
49 // ***** //

```

Codici_simulazione/p_in.txt

```

1 /*-----* C++ *-----*\
2 | ===== |
3 | \\ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4 | \\ / O peration | Version: 2.3.1 |
5 | \\ / A nd | Web: www.OpenFOAM.org |
6 | \\/ M anipulation |
7 /*-----*
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        volScalarField;
13     location     "0";
14     object       p;
15 }
16 // ***** //
17 dimensions      [0 2 -2 0 0 0];

```

```

18 internalField    uniform 0;
19 boundaryField
20 {
21     inlet
22     {
23         type          zeroGradient;
24     }
25
26     outlet
27     {
28         type          fixedValue;
29         value         uniform 0;
30     }
31
32     bottomWall
33     {
34         type          zeroGradient;
35     }
36
37     topWall
38     {
39         type          zeroGradient;
40     }
41
42     left
43     {
44         type          cyclic;
45     }
46
47     right
48     {
49         type          cyclic;
50     }
51 }
52 // *****

```

Condizioni periodiche

Codici_simulazione/U_per.txt

```

1 /*-----* C++ *-----*\
2 | ===== |
3 | \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O peration | Version: 2.3.1 |
5 | \ \ / A nd | Web: www.OpenFOAM.org |
6 | \ \ / M anipulation | |
7 \*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        volVectorField;
13     location     "0";
14     object       U;
15 }
16 // * * * * *
17 dimensions      [0 1 -1 0 0 0];
18 internalField   uniform (0 0 0);
19 boundaryField
20 {
21     inlet
22     {
23         type          cyclic;
24     }
25
26     outlet
27     {
28         type          cyclic;
29     }
30
31     bottomWall
32     {
33         type          fixedValue;
34         value         uniform (0 0 0);
35     }
36
37     topWall
38     {
39         type          fixedValue;
40         value         uniform (0 0 0);
41     }
42 }

```

```

36     }
37
38     left
39     {   type           cyclic;
40     }
41
42     right
43     {   type           cyclic;
44     }
45 }
46 // *****

```

Codici_simulazione/p_per.txt

```

1  /*-----*-- C++ -----*\
2  | ===== |
3  | \\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
4  | \\      /  O peration  | Version: 2.3.1 |
5  | \\      /  A nd        | Web: www.OpenFOAM.org |
6  |  \\\    /  M anipulation | |
7  \*-----*--\
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        volScalarField;
13     location     "0";
14     object       p;
15 }
16 // * * * * *
17 dimensions      [0 2 -2 0 0 0 0];
18 internalField   uniform 0;
19 boundaryField
20 {
21     inlet
22     {
23         type           cyclic;
24     }
25
26     outlet
27     {
28         type           cyclic;
29     }
30
31     bottomWall
32     {
33         type           zeroGradient;
34     }
35
36     topWall
37     {
38         type           zeroGradient;
39     }
40
41     left
42     {
43         type           cyclic;
44     }
45
46     right
47     {
48         type           cyclic;
49     }
50 }
51 // *****

```

A.6 Altri codici per l'esecuzione della simulazione

Vengono ora riportati, altri codici, che pur non servendo direttamente per il funzionamento di *OpenFOAM*, sono stati utilizzati per la realizzazione e/o l'esecuzione della simulazione. Nello specifico il primo codice riportato è un codice *Fortran* realizzato per trovare i valori degli input da dare ad *OpenFOAM* in merito alla spaziatura in direzione

y , per ottenere il grading delle celle da noi voluto. Questo codice è chiamato *Newton*, in quanto, nel suo funzionamento, sfrutta il metodo di Newton per la risoluzione dell'equazione differenziale che si ha al suo interno. Il secondo codice riportato, invece, è quello che viene utilizzato per poter eseguire le simulazioni sul sistema di calcolo utilizzato, *HACTAR*, per mettere in coda la simulazione da eseguire.

A.6.1 Newton

Questo codice, scritto nel linguaggio *Fortran*, serve per definire i parametri da fornire nel blockMeshDict (vedi A.1) per avere la spaziatura delle celle con grading voluta. In particolare richiede come input la dimensione del blocco da discretizzare, il rapporto di grading che si vuole avere, e la dimensione della prima cella. Questo fornirà in uscita, il numero di punti da imporre al programma per avere quel grading, oltre a riportare la dimensione di tutte le celle che si otterranno per quel blocco. Come tutti i codici *Fortran*, prima di essere utilizzato, deve essere compilato, e poi si lancia l'eseguibile che si ottiene dalla compilazione.

Codici_simulazione/Newton.f90

```

1 program newton
2 real(kind=8) :: eps
3
4 real(kind=8) :: t1,t2,soglia ,rapporto_dx ,Dx_min ,xN ,elle ,F,DF ,alpha ,Ddx_min
5 character(len=20) :: nome
6
7 ! write(*,*)'epsilon/d=?'
8 ! read(*,*)eps
9
10 !Definisco l'intervallo di Reynolds
11
12 write(*,*)'Rapporto Dx_fine/Dx_inizio=?'
13 read(*,*)rapporto_dx
14
15 write(*,*)'Lunghezza blocco=?'
16 read(*,*)elle
17
18 write(*,*)'Dx_minimo=?'
19 read(*,*)Dx_min
20 Ddx_min=Dx_min
21 Dx_min=Dx_min/elle
22
23 !...metodo di Newton:
24 !...inizializzo
25 call inizio(rapporto_dx,Dx_min,xN)
26 write(*,*)'valore iniziale =' ,xN
27
28 iter=0
29 soglia=5.d-3
30 t2=100*soglia
31 t1=xN
32
33 !...risolvo Newton:
34 do while(dabs(t2).gt.soglia)
35
36 call effe(t1,rapporto_dx ,Dx_min ,F,DF)
37 t2 = - F/Df
38 t1 = t1+t2
39 iter=iter+1
40 write(*,*)'iter=',iter , 'agg.',t1 , 'corr=',t2 , 'residuo=',F
41 ! pause 'sono in pausa'
42
43 end do
44
45 xN=t1
46

```

```

47 !...scrittura a schermo:
48 write(*,*) '-----'
49 write(*,'(A,1e13.6,A,1I6,A)') 'Numero punti=',xN,' [in iterazioni=',iter,']'
50 !write(*,*) 'Numero punti suggerito=',NINT(xN)
51 !write(*,*) '-----'
52
53 alpha=rapporto_dx**(1.0d0/(xN-1))
54
55 write(*,*) 'Deltax_1=',elle*(alpha-1)/(alpha**xN-1)
56 write(*,*) 'Deltax_N=',elle*(alpha-1)/(alpha**xN-1)*rapporto_dx
57
58 write(*,*) '-----'
59 write(*,*) 'Numero punti suggerito=',NINT(xN)
60 xN=NINT(xN)
61 alpha=rapporto_dx**(1.0d0/(xN-1))
62 write(*,*) 'Deltax_1=',elle*(alpha-1)/(alpha**xN-1)
63 write(*,*) 'Deltax_N=',elle*(alpha-1)/(alpha**xN-1)*rapporto_dx
64 write(*,*) 'alpha=',alpha
65 write(*,*) '-----'
66
67 write(*,*) 'Ymin=?'
68 read(*,*) Ymin
69
70 Deltax = elle*(alpha-1)/(alpha**xN-1)
71
72 write(*,'(1I6,10e15.8)') j, Ymin
73 do j=1,NINT(xN)-1
74 Y=Y+Deltax
75 write(*,'(1I6,10e15.8)') j, Ymin+(1-alpha**(j+1))/(1-alpha)*Deltax
76 end do
77
78 write(*,*) 'Conto da=?'
79 read(*,*) Conto
80
81 Deltax = elle*(alpha-1)/(alpha**xN-1)
82
83 write(*,'(1I6,10e15.8)') j, Ymin
84 do j=1,NINT(xN)-1
85 Y=Y+Deltax
86 write(*,'(10e15.8,10e15.8)') Conto+NINT(xN)-j, 2-(Ymin+(1-alpha**(j+1))/(1-alpha)*
      Deltax)
87 end do
88
89 stop
90 end program newton
91
92 !=====
93 subroutine effe(xN, rapporto_dx, Dx_min, F,DF)
94
95 real(kind=8) :: xN,Dx_min,rapporto_dx
96 real(kind=8),intent(OUT) :: F,DF
97 real(kind=8) :: A1,A2,tmp1,tmp2
98
99 tmp1=rapporto_dx**(xN/(xN-1))
100 tmp2=rapporto_dx**(1.0d0/(xN-1))
101
102 F = Dx_min*(tmp1-1) - (tmp2-1)
103
104 ! DF = Dx_min*tmp1*(-1.0d0/(xN-1)**2) + tmp2/(xN-1)**2
105 DF = (tmp2/(xN-1)**2) *(1-Dx_min*rapporto_dx)
106
107 ! write(*,*) '*', xN, rapporto_dx, Dx_min, tmp1, tmp2, F, DF
108
109 return
110 end
111
112 subroutine inizio(rapporto_dx,Dx_min,xN)
113 real(kind=8) :: xN,Dx_min,rapporto_dx
114

```

```

115 xN = log(rapporto_dx)/log(1+(rapporto_dx-1)*Dx_min)
116
117 return
118 end

```

A.6.2 Submit Run Simulazione

Questo codice serve per poter mettere in coda di esecuzione una simulazione, per farla poi eseguire sul sistema di calcolo utilizzato. In questo caso si riporta il codice per eseguire la simulazione della configurazione *H7*, nella versione periodica. Tra le diverse versioni cambiano solamente i nomi delle cartelle in cui ci si trova e/o da cui si deve interpolare, ed i tempi necessari a svolgere le simulazioni, che rendono necessario chiedere più o meno tempo di calcolo. Inoltre è possibile personalizzare altri aspetti del codice per ognuna delle configurazioni analizzate, per rendere più facilmente riconoscibili le simulazioni quando queste sono in coda e/o in esecuzione. Il nome del file può essere qualsiasi, mentre la sua estensione deve essere `.q`. Inoltre si invia al sistema di gestione della coda della macchina con `sbatch nome.q`.

Questo codice è molto interessante, non solo perché ci mostra quali sono i parametri da fornire per l'utilizzo del sistema di calcolo *HACTAR*, ma anche perché contiene al suo interno tutti i comandi necessari per l'esecuzione completa di una simulazione su *OpenFOAM*.

Codici_simulazione/Run_C_per_h7_5.txt

```

1 #!/bin/bash
2 #SBATCH --job-name=C_per_h7_5
3 # le due righe servono per mandare email, modificare guardando la guida
4 #SBATCH --mail-type=ALL
5 #SBATCH --mail-user=s230951@studenti.polito.it
6 #SBATCH --partition=global
7 #SBATCH --time=60:55:00
8 #SBATCH --nodes=1
9 #SBATCH --ntasks-per-node=18
10 #SBATCH --output=C_per_h7_5_%j.log
11 #SBATCH --mem-per-cpu=1536M
12
13 module load openfoam/5.0
14 source /share/apps/openfoam/5.0/OpenFOAM-5.0/etc/bashrc
15 cp system/controlDictRun system/controlDict
16 blockMesh
17 mapFields ../C_per_h7_5_transitorio -consistent
18 decomposePar
19 mpirun pisoFoam -parallel &> log
20 reconstructPar

```

Appendice B

Codici post processing

In questa seconda parte delle appendici, vengono riportati i codici che sono stati utilizzati per effettuare il post processing dei dati, ovvero la parte di elaborazione dei dati ottenuti dalla simulazioni numeriche per estrarre le velocità medie e le fluttuazioni di velocità, per ordinarli in modo tale da poterli graficare ed utilizzare, ed anche il codice utilizzato per l'esecuzione del post processing sul calcolatore *HACTAR*. In questo caso, ogni codice è riportato singolarmente, con eventuali aggiunte di possibili modifiche che possono essere fatte al codice per ottenerne altre versioni, e/o codici ausiliari che servono per il loro funzionamento. Per il codice del post processing, si riporta solamente quello utilizzato per la configurazione *H7*, in quanto ci sono poche differenze tra una configurazione ed un'altra, vedi 4.3.

Per quanto riguarda i comandi necessari per l'utilizzo di questi codici e quindi l'esecuzione del post processing, fare riferimento alle relative sezioni nel capitolo 4, nello specifico per questa appendice, quelle riportate nella sezione 4.3, e nel file con il codice utilizzato per l'esecuzione delle simulazioni sul calcolatore *HACTAR*, riportato in questa appendice, nella sezione B.3.

Qui i file relativi ad *OpenFOAM* sono indicati come `.txt`, in quanto è necessario per farli leggere da \LaTeX . Tuttavia, invece, per essere letti da *OpenFOAM*, è necessario che siano senza estensione. Gli altri codici, invece, sono riportati con la loro estensione, o viene specificato nel codice, se per l'uso è necessario cambiare estensione.

B.1 controlDictReynolds

Questo codice, è una variante del *controlDict* utilizzato per l'esecuzione della simulazione, è viene appunto chiamato *controlDictReynolds*, per distinguerlo dall'altro (vedi B.1). Tuttavia, anche per questo codice, quando il post processing deve essere eseguito, è necessario che questo codice, si trovi nella cartella *system*, e si chiami *controlDict*. In questo codice vengono riportate, oltre a tutte le informazioni già presenti nel *controlDictRun*, anche quelle per effettuare il post processing dei dati. In particolare, nel nostro caso, troviamo una funzione, la quale serve per estrarre le grandezze specificate al suo interno, facendone la media per piani, sui piani per le quali si indica di utilizzare questa funzione. Nel nostro caso, si è andati ad estrarre la velocità media e le componenti di fluttuazione della velocità, su piani normali all'asse *y*.

Codici_simulazione/controlDictReynolds.txt

```

1 /-----* C++ *-----\
2 | ===== | |
3 | \ \ / F i e l d | O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x |
4 | \ \ / O p e r a t i o n | V e r s i o n : 2 . 2 . 0 |

```



```

73         operation          areaAverage;
74 sampledSurfaceDict
75     { type                  plane;
76 planeType pointAndNormal;
77 triangulate false;
78 interpolate false;
79 pointAndNormalDict
80     { basePoint             (0.0501 0.50358107E-02 0.005);
81       normalVector         (0 1 0);
82     }
83     fields
84     ( Ux UPrime2Mean
85     ); }
86
87 piano_2
88     { type                  surfaceFieldValue;
89       libs                  ("libfieldFunctionObjects.so");
90       log                   true;
91       writeControl          writeTime;
92       writeFields           true;
93       surfaceFormat         raw;
94       regionType            sampledSurface;
95       name                  yNormal1;
96       operation             areaAverage;
97 sampledSurfaceDict
98     { type                  plane;
99 planeType pointAndNormal;
100 triangulate false;
101 interpolate false;
102 pointAndNormalDict
103     { basePoint             (0.0501 0.76316051E-02 0.005);
104       normalVector         (0 1 0);
105     }
106     fields
107     ( Ux UPrime2Mean
108     ); }
109
110 piano_3
111     { type                  surfaceFieldValue;
112       libs                  ("libfieldFunctionObjects.so");
113       log                   true;
114       writeControl          writeTime;
115       writeFields           true;
116       surfaceFormat         raw;
117       regionType            sampledSurface;
118       name                  yNormal2;
119       operation             areaAverage;
120 sampledSurfaceDict
121     { type                  plane;
122 planeType pointAndNormal;
123 triangulate false;
124 interpolate false;
125 pointAndNormalDict
126     { basePoint             (0.0501 0.10280751E-01 0.005);
127       normalVector         (0 1 0);
128     }
129     fields
130     ( Ux UPrime2Mean
131     ); }
132
133 piano_4
134     { type                  surfaceFieldValue;
135       libs                  ("libfieldFunctionObjects.so");
136       log                   true;
137       writeControl          writeTime;
138       writeFields           true;
139       surfaceFormat         raw;
140       regionType            sampledSurface;
141       name                  yNormal3;

```

```

142         operation          areaAverage;
143 sampledSurfaceDict
144     { type                  plane;
145 planeType pointAndNormal;
146 triangulate false;
147 interpolate false;
148 pointAndNormalDict
149     { basePoint             (0.0501 0.12984345E-01 0.005);
150       normalVector         (0 1 0);
151     }
152     fields
153     ( Ux UPrime2Mean
154     ); }
155
156 piano_5
157     { type                  surfaceFieldValue;
158       libs                  ("libfieldFunctionObjects.so");
159       log                   true;
160       writeControl          writeTime;
161       writeFields          true;
162       surfaceFormat        raw;
163       regionType           sampledSurface;
164       name                  yNormal4;
165       operation            areaAverage;
166 sampledSurfaceDict
167     { type                  plane;
168 planeType pointAndNormal;
169 triangulate false;
170 interpolate false;
171 pointAndNormalDict
172     { basePoint             (0.0501 0.15743506E-01 0.005);
173       normalVector         (0 1 0);
174     }
175     fields
176     ( Ux UPrime2Mean
177     ); }
178
179 piano_6
180     { type                  surfaceFieldValue;
181       libs                  ("libfieldFunctionObjects.so");
182       log                   true;
183       writeControl          writeTime;
184       writeFields          true;
185       surfaceFormat        raw;
186       regionType           sampledSurface;
187       name                  yNormal5;
188       operation            areaAverage;
189 sampledSurfaceDict
190     { type                  plane;
191 planeType pointAndNormal;
192 triangulate false;
193 interpolate false;
194 pointAndNormalDict
195     { basePoint             (0.0501 0.18559376E-01 0.005);
196       normalVector         (0 1 0);
197     }
198     fields
199     ( Ux UPrime2Mean
200     ); }
201
202 piano_7
203     { type                  surfaceFieldValue;
204       libs                  ("libfieldFunctionObjects.so");
205       log                   true;
206       writeControl          writeTime;
207       writeFields          true;
208       surfaceFormat        raw;
209       regionType           sampledSurface;
210       name                  yNormal10;

```

```

211         operation          areaAverage;
212 sampledSurfaceDict
213     { type                  plane;
214 planeType pointAndNormal;
215 triangulate false;
216 interpolate false;
217 pointAndNormalDict
218     { basePoint             (0.0501 0.21433121E-01 0.005);
219       normalVector         (0 1 0);
220 }
221     fields
222     ( Ux UPrime2Mean
223     ); }
224
225 piano_8
226     { type                  surfaceFieldValue;
227       libs                  ("libfieldFunctionObjects.so");
228       log                   true;
229       writeControl          writeTime;
230       writeFields           true;
231       surfaceFormat         raw;
232       regionType            sampledSurface;
233       name                   yNormal1;
234       operation             areaAverage;
235 sampledSurfaceDict
236     { type                  plane;
237 planeType pointAndNormal;
238 triangulate false;
239 interpolate false;
240 pointAndNormalDict
241     { basePoint             (0.0501 0.24365930E-01 0.005);
242       normalVector         (0 1 0);
243 }
244     fields
245     ( Ux UPrime2Mean
246     ); }
247
248 piano_9
249     { type                  surfaceFieldValue;
250       libs                  ("libfieldFunctionObjects.so");
251       log                   true;
252       writeControl          writeTime;
253       writeFields           true;
254       surfaceFormat         raw;
255       regionType            sampledSurface;
256       name                   yNormal2;
257       operation             areaAverage;
258 sampledSurfaceDict
259     { type                  plane;
260 planeType pointAndNormal;
261 triangulate false;
262 interpolate false;
263 pointAndNormalDict
264     { basePoint             (0.0501 0.27359017E-01 0.005);
265       normalVector         (0 1 0);
266 }
267     fields
268     ( Ux UPrime2Mean
269     ); }
270
271 piano_10
272     { type                  surfaceFieldValue;
273       libs                  ("libfieldFunctionObjects.so");
274       log                   true;
275       writeControl          writeTime;
276       writeFields           true;
277       surfaceFormat         raw;
278       regionType            sampledSurface;
279       name                   yNormal3;

```

```

280         operation          areaAverage;
281 sampledSurfaceDict
282     { type                  plane;
283 planeType pointAndNormal;
284 triangulate false;
285 interpolate false;
286 pointAndNormalDict
287     { basePoint            (0.0501 0.30413621E-01 0.005);
288       normalVector        (0 1 0);
289     }
290     fields
291     ( Ux UPrime2Mean
292     ); }
293
294 piano_11
295     { type                  surfaceFieldValue;
296       libs                  ("libfieldFunctionObjects.so");
297       log                    true;
298       writeControl          writeTime;
299       writeFields           true;
300       surfaceFormat        raw;
301       regionType           sampledSurface;
302       name                  yNormal4;
303       operation            areaAverage;
304 sampledSurfaceDict
305     { type                  plane;
306 planeType pointAndNormal;
307 triangulate false;
308 interpolate false;
309 pointAndNormalDict
310     { basePoint            (0.0501 0.33531006E-01 0.005);
311       normalVector        (0 1 0);
312     }
313     fields
314     ( Ux UPrime2Mean
315     ); }
316
317 piano_12
318     { type                  surfaceFieldValue;
319       libs                  ("libfieldFunctionObjects.so");
320       log                    true;
321       writeControl          writeTime;
322       writeFields           true;
323       surfaceFormat        raw;
324       regionType           sampledSurface;
325       name                  yNormal5;
326       operation            areaAverage;
327 sampledSurfaceDict
328     { type                  plane;
329 planeType pointAndNormal;
330 triangulate false;
331 interpolate false;
332 pointAndNormalDict
333     { basePoint            (0.0501 0.36712464E-01 0.005);
334       normalVector        (0 1 0);
335     }
336     fields
337     ( Ux UPrime2Mean
338     ); }
339
340 piano_13
341     { type                  surfaceFieldValue;
342       libs                  ("libfieldFunctionObjects.so");
343       log                    true;
344       writeControl          writeTime;
345       writeFields           true;
346       surfaceFormat        raw;
347       regionType           sampledSurface;
348       name                  yNormal10;

```

```

349         operation          areaAverage;
350 sampledSurfaceDict
351     { type                  plane;
352 planeType pointAndNormal;
353 triangulate false;
354 interpolate false;
355     pointAndNormalDict
356         { basePoint          (0.0501 0.39959309E-01 0.005);
357           normalVector       (0 1 0);
358     }
359     fields
360     ( Ux UPrime2Mean
361     ); }
362
363 piano_14
364     { type                  surfaceFieldValue;
365       libs                  ("libfieldFunctionObjects.so");
366       log                   true;
367       writeControl          writeTime;
368       writeFields           true;
369       surfaceFormat         raw;
370       regionType            sampledSurface;
371       name                  yNormal1;
372       operation             areaAverage;
373 sampledSurfaceDict
374     { type                  plane;
375 planeType pointAndNormal;
376 triangulate false;
377 interpolate false;
378     pointAndNormalDict
379         { basePoint          (0.0501 0.43272888E-01 0.005);
380           normalVector       (0 1 0);
381     }
382     fields
383     ( Ux UPrime2Mean
384     ); }
385
386 piano_15
387     { type                  surfaceFieldValue;
388       libs                  ("libfieldFunctionObjects.so");
389       log                   true;
390       writeControl          writeTime;
391       writeFields           true;
392       surfaceFormat         raw;
393       regionType            sampledSurface;
394       name                  yNormal2;
395       operation             areaAverage;
396 sampledSurfaceDict
397     { type                  plane;
398 planeType pointAndNormal;
399 triangulate false;
400 interpolate false;
401     pointAndNormalDict
402         { basePoint          (0.0501 0.46654570E-01 0.005);
403           normalVector       (0 1 0);
404     }
405     fields
406     ( Ux UPrime2Mean
407     ); }
408
409 piano_16
410     { type                  surfaceFieldValue;
411       libs                  ("libfieldFunctionObjects.so");
412       log                   true;
413       writeControl          writeTime;
414       writeFields           true;
415       surfaceFormat         raw;
416       regionType            sampledSurface;
417       name                  yNormal3;

```

```

418         operation          areaAverage;
419 sampledSurfaceDict
420     { type                  plane;
421 planeType pointAndNormal;
422 triangulate false;
423 interpolate false;
424 pointAndNormalDict
425     { basePoint            (0.0501 0.50105756E-01 0.005);
426       normalVector        (0 1 0);
427     }
428     fields
429     ( Ux UPrime2Mean
430     ); }
431
432 piano_17
433     { type                  surfaceFieldValue;
434       libs                  ("libfieldFunctionObjects.so");
435       log                    true;
436       writeControl          writeTime;
437       writeFields           true;
438       surfaceFormat         raw;
439       regionType            sampledSurface;
440       name                   yNormal4;
441       operation             areaAverage;
442 sampledSurfaceDict
443     { type                  plane;
444 planeType pointAndNormal;
445 triangulate false;
446 interpolate false;
447 pointAndNormalDict
448     { basePoint            (0.0501 0.53627875E-01 0.005);
449       normalVector        (0 1 0);
450     }
451     fields
452     ( Ux UPrime2Mean
453     ); }
454
455 piano_18
456     { type                  surfaceFieldValue;
457       libs                  ("libfieldFunctionObjects.so");
458       log                    true;
459       writeControl          writeTime;
460       writeFields           true;
461       surfaceFormat         raw;
462       regionType            sampledSurface;
463       name                   yNormal5;
464       operation             areaAverage;
465 sampledSurfaceDict
466     { type                  plane;
467 planeType pointAndNormal;
468 triangulate false;
469 interpolate false;
470 pointAndNormalDict
471     { basePoint            (0.0501 0.57222384E-01 0.005);
472       normalVector        (0 1 0);
473     }
474     fields
475     ( Ux UPrime2Mean
476     ); }
477
478 piano_19
479     { type                  surfaceFieldValue;
480       libs                  ("libfieldFunctionObjects.so");
481       log                    true;
482       writeControl          writeTime;
483       writeFields           true;
484       surfaceFormat         raw;
485       regionType            sampledSurface;
486       name                   yNormal10;

```

```

487         operation          areaAverage;
488 sampledSurfaceDict
489     { type                  plane;
490 planeType pointAndNormal;
491 triangulate false;
492 interpolate false;
493 pointAndNormalDict
494     { basePoint            (0.0501 0.60890771E-01 0.005);
495       normalVector        (0 1 0);
496     }
497     fields
498     ( Ux UPrime2Mean
499     ); }
500
501 piano_20
502     { type                  surfaceFieldValue;
503       libs                  ("libfieldFunctionObjects.so");
504       log                    true;
505       writeControl          writeTime;
506       writeFields           true;
507       surfaceFormat         raw;
508       regionType            sampledSurface;
509       name                   yNormal1;
510       operation             areaAverage;
511 sampledSurfaceDict
512     { type                  plane;
513 planeType pointAndNormal;
514 triangulate false;
515 interpolate false;
516 pointAndNormalDict
517     { basePoint            (0.0501 0.64634554E-01 0.005);
518       normalVector        (0 1 0);
519     }
520     fields
521     ( Ux UPrime2Mean
522     ); }
523
524 piano_21
525     { type                  surfaceFieldValue;
526       libs                  ("libfieldFunctionObjects.so");
527       log                    true;
528       writeControl          writeTime;
529       writeFields           true;
530       surfaceFormat         raw;
531       regionType            sampledSurface;
532       name                   yNormal2;
533       operation             areaAverage;
534 sampledSurfaceDict
535     { type                  plane;
536 planeType pointAndNormal;
537 triangulate false;
538 interpolate false;
539 pointAndNormalDict
540     { basePoint            (0.0501 0.68455284E-01 0.005);
541       normalVector        (0 1 0);
542     }
543     fields
544     ( Ux UPrime2Mean
545     ); }
546
547 piano_22
548     { type                  surfaceFieldValue;
549       libs                  ("libfieldFunctionObjects.so");
550       log                    true;
551       writeControl          writeTime;
552       writeFields           true;
553       surfaceFormat         raw;
554       regionType            sampledSurface;
555       name                   yNormal3;

```

```

556         operation          areaAverage;
557 sampledSurfaceDict
558     { type                  plane;
559 planeType pointAndNormal;
560 triangulate false;
561 interpolate false;
562 pointAndNormalDict
563     { basePoint             (0.0501 0.72354541E-01 0.005);
564       normalVector         (0 1 0);
565     }
566     fields
567     ( Ux UPrime2Mean
568       );
569
570 piano_23
571     { type                  surfaceFieldValue;
572       libs                  ("libfieldFunctionObjects.so");
573       log                   true;
574       writeControl          writeTime;
575       writeFields           true;
576       surfaceFormat         raw;
577       regionType            sampledSurface;
578       name                  yNormal4;
579       operation             areaAverage;
580 sampledSurfaceDict
581     { type                  plane;
582 planeType pointAndNormal;
583 triangulate false;
584 interpolate false;
585 pointAndNormalDict
586     { basePoint             (0.0501 0.76333940E-01 0.005);
587       normalVector         (0 1 0);
588     }
589     fields
590     ( Ux UPrime2Mean
591       );
592
593 piano_24
594     { type                  surfaceFieldValue;
595       libs                  ("libfieldFunctionObjects.so");
596       log                   true;
597       writeControl          writeTime;
598       writeFields           true;
599       surfaceFormat         raw;
600       regionType            sampledSurface;
601       name                  yNormal5;
602       operation             areaAverage;
603 sampledSurfaceDict
604     {
605         type                  plane;
606 planeType pointAndNormal;
607 triangulate false;
608 interpolate false;
609 pointAndNormalDict
610     { basePoint             (0.0501 0.80395127E-01 0.005);
611       normalVector         (0 1 0);
612     }
613     fields
614     ( Ux UPrime2Mean
615       );
616
617 piano_25
618     { type                  surfaceFieldValue;
619       libs                  ("libfieldFunctionObjects.so");
620       log                   true;
621       writeControl          writeTime;
622       writeFields           true;
623       surfaceFormat         raw;
624       regionType            sampledSurface;

```

```

625     name          yNormal0;
626     operation      areaAverage;
627 sampledSurfaceDict
628     { type          plane;
629 planeType pointAndNormal;
630 triangulate false;
631 interpolate false;
632 pointAndNormalDict
633     { basePoint      (0.0501 0.84539784E-01 0.005);
634       normalVector   (0 1 0);
635     }
636     fields
637     ( Ux UPrime2Mean
638     ); }
639
640 piano_26
641     { type          surfaceFieldValue;
642       libs          ("libfieldFunctionObjects.so");
643       log           true;
644       writeControl  writeTime;
645       writeFields  true;
646       surfaceFormat raw;
647       regionType   sampledSurface;
648       name         yNormal1;
649       operation    areaAverage;
650 sampledSurfaceDict
651     { type          plane;
652 planeType pointAndNormal;
653 triangulate false;
654 interpolate false;
655 pointAndNormalDict
656     { basePoint      (0.0501 0.88769627E-01 0.005);
657       normalVector   (0 1 0);
658     }
659     fields
660     ( Ux UPrime2Mean
661     ); }
662
663 piano_27
664     { type          surfaceFieldValue;
665       libs          ("libfieldFunctionObjects.so");
666       log           true;
667       writeControl  writeTime;
668       writeFields  true;
669       surfaceFormat raw;
670       regionType   sampledSurface;
671       name         yNormal2;
672       operation    areaAverage;
673 sampledSurfaceDict
674     { type          plane;
675 planeType pointAndNormal;
676 triangulate false;
677 interpolate false;
678 pointAndNormalDict
679     { basePoint      (0.0501 0.93086405E-01 0.005);
680       normalVector   (0 1 0);
681     }
682     fields
683     ( Ux UPrime2Mean
684     ); }
685
686 piano_28
687     { type          surfaceFieldValue;
688       libs          ("libfieldFunctionObjects.so");
689       log           true;
690       writeControl  writeTime;
691       writeFields  true;
692       surfaceFormat raw;
693       regionType   sampledSurface;

```

```

694     name          yNormal3;
695     operation      areaAverage;
696 sampledSurfaceDict
697     { type          plane;
698 planeType pointAndNormal;
699 triangulate false;
700 interpolate false;
701     pointAndNormalDict
702     { basePoint      (0.0501 0.97491907E-01 0.005);
703       normalVector   (0 1 0);
704     }
705     fields
706     ( Ux UPrime2Mean
707     ); }
708
709 piano_29
710     { type          surfaceFieldValue;
711       libs          ("libfieldFunctionObjects.so");
712       log           true;
713       writeControl  writeTime;
714       writeFields  true;
715       surfaceFormat raw;
716       regionType   sampledSurface;
717       name          yNormal4;
718       operation      areaAverage;
719 sampledSurfaceDict
720     { type          plane;
721 planeType pointAndNormal;
722 triangulate false;
723 interpolate false;
724     pointAndNormalDict
725     { basePoint      (0.0501 0.10198795 0.005);
726       normalVector   (0 1 0);
727     }
728     fields
729     ( Ux UPrime2Mean
730     ); }
731
732 piano_30
733     { type          surfaceFieldValue;
734       libs          ("libfieldFunctionObjects.so");
735       log           true;
736       writeControl  writeTime;
737       writeFields  true;
738       surfaceFormat raw;
739       regionType   sampledSurface;
740       name          yNormal5;
741       operation      areaAverage;
742 sampledSurfaceDict
743     { type          plane;
744 planeType pointAndNormal;
745 triangulate false;
746 interpolate false;
747     pointAndNormalDict
748     { basePoint      (0.0501 0.10657641 0.005);
749       normalVector   (0 1 0);
750     }
751     fields
752     ( Ux UPrime2Mean
753     ); }
754
755 piano_31
756     { type          surfaceFieldValue;
757       libs          ("libfieldFunctionObjects.so");
758       log           true;
759       writeControl  writeTime;
760       writeFields  true;
761       surfaceFormat raw;
762       regionType   sampledSurface;

```

```

763     name          yNormal0;
764     operation      areaAverage;
765 sampledSurfaceDict
766     { type          plane;
767 planeType pointAndNormal;
768 triangulate false;
769 interpolate false;
770     pointAndNormalDict
771     { basePoint      (0.0501 0.11125917 0.005);
772       normalVector   (0 1 0);
773     }
774     fields
775     ( Ux UPrime2Mean
776     ); }
777
778 piano_32
779     { type          surfaceFieldValue;
780       libs          ("libfieldFunctionObjects.so");
781       log           true;
782       writeControl  writeTime;
783       writeFields   true;
784       surfaceFormat raw;
785       regionType    sampledSurface;
786       name          yNormal1;
787       operation      areaAverage;
788 sampledSurfaceDict
789     { type          plane;
790 planeType pointAndNormal;
791 triangulate false;
792 interpolate false;
793     pointAndNormalDict
794     { basePoint      (0.0501 0.11603818 0.005);
795       normalVector   (0 1 0);
796     }
797     fields
798     ( Ux UPrime2Mean
799     ); }
800
801 piano_33
802     { type          surfaceFieldValue;
803       libs          ("libfieldFunctionObjects.so");
804       log           true;
805       writeControl  writeTime;
806       writeFields   true;
807       surfaceFormat raw;
808       regionType    sampledSurface;
809       name          yNormal2;
810       operation      areaAverage;
811 sampledSurfaceDict
812     { type          plane;
813 planeType pointAndNormal;
814 triangulate false;
815 interpolate false;
816     pointAndNormalDict
817     { basePoint      (0.0501 0.12091541 0.005);
818       normalVector   (0 1 0);
819     }
820     fields
821     ( Ux UPrime2Mean
822     ); }
823
824 piano_34
825     { type          surfaceFieldValue;
826       libs          ("libfieldFunctionObjects.so");
827       log           true;
828       writeControl  writeTime;
829       writeFields   true;
830       surfaceFormat raw;
831       regionType    sampledSurface;

```

```

832     name          yNormal3;
833     operation      areaAverage;
834 sampledSurfaceDict
835     { type          plane;
836 planeType pointAndNormal;
837 triangulate false;
838 interpolate false;
839     pointAndNormalDict
840     { basePoint      (0.0501 0.12589288 0.005);
841       normalVector   (0 1 0);
842     }
843     fields
844     ( Ux UPrime2Mean
845     ); }
846
847 piano_35
848     { type          surfaceFieldValue;
849       libs          ("libfieldFunctionObjects.so");
850       log           true;
851       writeControl  writeTime;
852       writeFields   true;
853       surfaceFormat raw;
854       regionType    sampledSurface;
855       name          yNormal4;
856       operation      areaAverage;
857 sampledSurfaceDict
858     { type          plane;
859 planeType pointAndNormal;
860 triangulate false;
861 interpolate false;
862     pointAndNormalDict
863     { basePoint      (0.0501 0.13097265 0.005);
864       normalVector   (0 1 0);
865     }
866     fields
867     ( Ux UPrime2Mean
868     ); }
869
870 piano_36
871     { type          surfaceFieldValue;
872       libs          ("libfieldFunctionObjects.so");
873       log           true;
874       writeControl  writeTime;
875       writeFields   true;
876       surfaceFormat raw;
877       regionType    sampledSurface;
878       name          yNormal5;
879       operation      areaAverage;
880 sampledSurfaceDict
881     { type          plane;
882 planeType pointAndNormal;
883 triangulate false;
884 interpolate false;
885     pointAndNormalDict
886     { basePoint      (0.0501 0.13615683 0.005);
887       normalVector   (0 1 0);
888     }
889     fields
890     ( Ux UPrime2Mean
891     ); }
892
893 piano_37
894     { type          surfaceFieldValue;
895       libs          ("libfieldFunctionObjects.so");
896       log           true;
897       writeControl  writeTime;
898       writeFields   true;
899       surfaceFormat raw;
900       regionType    sampledSurface;

```

```

901     name          yNormal0;
902     operation      areaAverage;
903 sampledSurfaceDict
904     { type          plane;
905 planeType pointAndNormal;
906 triangulate false;
907 interpolate false;
908     pointAndNormalDict
909     { basePoint      (0.0501 0.14144756 0.005);
910       normalVector   (0 1 0);
911     }
912     fields
913     ( Ux UPrime2Mean
914     ); }
915
916 piano_38
917     { type          surfaceFieldValue;
918       libs          ("libfieldFunctionObjects.so");
919       log           true;
920       writeControl  writeTime;
921       writeFields  true;
922       surfaceFormat raw;
923       regionType   sampledSurface;
924       name         yNormal1;
925       operation    areaAverage;
926 sampledSurfaceDict
927     { type          plane;
928 planeType pointAndNormal;
929 triangulate false;
930 interpolate false;
931     pointAndNormalDict
932     { basePoint      (0.0501 0.14684703 0.005);
933       normalVector   (0 1 0);
934     }
935     fields
936     ( Ux UPrime2Mean
937     ); }
938
939 piano_39
940     { type          surfaceFieldValue;
941       libs          ("libfieldFunctionObjects.so");
942       log           true;
943       writeControl  writeTime;
944       writeFields  true;
945       surfaceFormat raw;
946       regionType   sampledSurface;
947       name         yNormal2;
948       operation    areaAverage;
949 sampledSurfaceDict
950     { type          plane;
951 planeType pointAndNormal;
952 triangulate false;
953 interpolate false;
954     pointAndNormalDict
955     { basePoint      (0.0501 0.15235748 0.005);
956       normalVector   (0 1 0);
957     }
958     fields
959     ( Ux UPrime2Mean
960     ); }
961
962 piano_40
963     { type          surfaceFieldValue;
964       libs          ("libfieldFunctionObjects.so");
965       log           true;
966       writeControl  writeTime;
967       writeFields  true;
968       surfaceFormat raw;
969       regionType   sampledSurface;

```

```

970     name          yNormal3;
971     operation      areaAverage;
972 sampledSurfaceDict
973     { type          plane;
974 planeType pointAndNormal;
975 triangulate false;
976 interpolate false;
977     pointAndNormalDict
978     { basePoint      (0.0501 0.15798118 0.005);
979       normalVector   (0 1 0);
980     }
981     fields
982     ( Ux UPrime2Mean
983     ); }
984
985 piano_41
986     { type          surfaceFieldValue;
987       libs          ("libfieldFunctionObjects.so");
988       log           true;
989       writeControl  writeTime;
990       writeFields  true;
991       surfaceFormat raw;
992       regionType   sampledSurface;
993       name         yNormal4;
994       operation    areaAverage;
995 sampledSurfaceDict
996     { type          plane;
997 planeType pointAndNormal;
998 triangulate false;
999 interpolate false;
1000    pointAndNormalDict
1001    { basePoint      (0.0501 0.16372047 0.005);
1002      normalVector   (0 1 0);
1003    }
1004    fields
1005    ( Ux UPrime2Mean
1006    ); }
1007
1008 piano_42
1009     { type          surfaceFieldValue;
1010       libs          ("libfieldFunctionObjects.so");
1011       log           true;
1012       writeControl  writeTime;
1013       writeFields  true;
1014       surfaceFormat raw;
1015       regionType   sampledSurface;
1016       name         yNormal5;
1017       operation    areaAverage;
1018 sampledSurfaceDict
1019     { type          plane;
1020 planeType pointAndNormal;
1021 triangulate false;
1022 interpolate false;
1023    pointAndNormalDict
1024    { basePoint      (0.0501 0.16957771 0.005);
1025      normalVector   (0 1 0);
1026    }
1027    fields
1028    ( Ux UPrime2Mean
1029    ); }
1030
1031 piano_43
1032     { type          surfaceFieldValue;
1033       libs          ("libfieldFunctionObjects.so");
1034       log           true;
1035       writeControl  writeTime;
1036       writeFields  true;
1037       surfaceFormat raw;
1038       regionType   sampledSurface;

```

```

1039     name          yNormal0;
1040     operation      areaAverage;
1041 sampledSurfaceDict
1042     { type          plane;
1043 planeType pointAndNormal;
1044 triangulate false;
1045 interpolate false;
1046     pointAndNormalDict
1047     { basePoint      (0.0501 0.17555534 0.005);
1048       normalVector   (0 1 0);
1049     }
1050     fields
1051     ( Ux UPrime2Mean
1052     ); }
1053
1054 piano_44
1055     { type          surfaceFieldValue;
1056       libs          ("libfieldFunctionObjects.so");
1057       log           true;
1058       writeControl  writeTime;
1059       writeFields   true;
1060       surfaceFormat raw;
1061       regionType    sampledSurface;
1062       name          yNormal1;
1063       operation      areaAverage;
1064 sampledSurfaceDict
1065     { type          plane;
1066 planeType pointAndNormal;
1067 triangulate false;
1068 interpolate false;
1069     pointAndNormalDict
1070     { basePoint      (0.0501 0.18165583 0.005);
1071       normalVector   (0 1 0);
1072     }
1073     fields
1074     ( Ux UPrime2Mean
1075     ); }
1076
1077 piano_45
1078     { type          surfaceFieldValue;
1079       libs          ("libfieldFunctionObjects.so");
1080       log           true;
1081       writeControl  writeTime;
1082       writeFields   true;
1083       surfaceFormat raw;
1084       regionType    sampledSurface;
1085       name          yNormal2;
1086       operation      areaAverage;
1087 sampledSurfaceDict
1088     { type          plane;
1089 planeType pointAndNormal;
1090 triangulate false;
1091 interpolate false;
1092     pointAndNormalDict
1093     { basePoint      (0.0501 0.1878817 0.005);
1094       normalVector   (0 1 0);
1095     }
1096     fields
1097     ( Ux UPrime2Mean
1098     ); }
1099
1100 piano_46
1101     { type          surfaceFieldValue;
1102       libs          ("libfieldFunctionObjects.so");
1103       log           true;
1104       writeControl  writeTime;
1105       writeFields   true;
1106       surfaceFormat raw;
1107       regionType    sampledSurface;

```

```

1108     name          yNormal3;
1109     operation      areaAverage;
1110 sampledSurfaceDict
1111     { type          plane;
1112 planeType pointAndNormal;
1113 triangulate false;
1114 interpolate false;
1115     pointAndNormalDict
1116     { basePoint    (0.0501 0.19423554 0.005);
1117       normalVector (0 1 0);
1118     }
1119     fields
1120     ( Ux UPrime2Mean
1121     ); }
1122
1123 piano_47
1124     { type          surfaceFieldValue;
1125       libs          ("libfieldFunctionObjects.so");
1126       log           true;
1127       writeControl  writeTime;
1128       writeFields   true;
1129       surfaceFormat raw;
1130       regionType    sampledSurface;
1131       name          yNormal4;
1132       operation      areaAverage;
1133 sampledSurfaceDict
1134     { type          plane;
1135 planeType pointAndNormal;
1136 triangulate false;
1137 interpolate false;
1138     pointAndNormalDict
1139     { basePoint    (0.0501 0.20071996 0.005);
1140       normalVector (0 1 0);
1141     }
1142     fields
1143     ( Ux UPrime2Mean
1144     ); }
1145
1146 piano_48
1147     { type          surfaceFieldValue;
1148       libs          ("libfieldFunctionObjects.so");
1149       log           true;
1150       writeControl  writeTime;
1151       writeFields   true;
1152       surfaceFormat raw;
1153       regionType    sampledSurface;
1154       name          yNormal5;
1155       operation      areaAverage;
1156 sampledSurfaceDict
1157     { type          plane;
1158 planeType pointAndNormal;
1159 triangulate false;
1160 interpolate false;
1161     pointAndNormalDict
1162     { basePoint    (0.0501 0.20733766 0.005);
1163       normalVector (0 1 0);
1164     }
1165     fields
1166     ( Ux UPrime2Mean
1167     ); }
1168
1169 piano_49
1170     { type          surfaceFieldValue;
1171       libs          ("libfieldFunctionObjects.so");
1172       log           true;
1173       writeControl  writeTime;
1174       writeFields   true;
1175       surfaceFormat raw;
1176       regionType    sampledSurface;

```

```

1177     name          yNormal0;
1178     operation      areaAverage;
1179 sampledSurfaceDict
1180     { type          plane;
1181 planeType pointAndNormal;
1182 triangulate false;
1183 interpolate false;
1184 pointAndNormalDict
1185     { basePoint      (0.0501 0.21409137 0.005);
1186       normalVector   (0 1 0);
1187     }
1188     fields
1189     ( Ux UPrime2Mean
1190     ); }
1191
1192 piano_50
1193     { type          surfaceFieldValue;
1194       libs          ("libfieldFunctionObjects.so");
1195       log           true;
1196       writeControl  writeTime;
1197       writeFields   true;
1198       surfaceFormat raw;
1199       regionType    sampledSurface;
1200       name          yNormal1;
1201       operation      areaAverage;
1202 sampledSurfaceDict
1203     { type          plane;
1204 planeType pointAndNormal;
1205 triangulate false;
1206 interpolate false;
1207 pointAndNormalDict
1208     { basePoint      (0.0501 0.22098389 0.005);
1209       normalVector   (0 1 0);
1210     }
1211     fields
1212     ( Ux UPrime2Mean
1213     ); }
1214
1215 piano_51
1216     { type          surfaceFieldValue;
1217       libs          ("libfieldFunctionObjects.so");
1218       log           true;
1219       writeControl  writeTime;
1220       writeFields   true;
1221       surfaceFormat raw;
1222       regionType    sampledSurface;
1223       name          yNormal2;
1224       operation      areaAverage;
1225 sampledSurfaceDict
1226     { type          plane;
1227 planeType pointAndNormal;
1228 triangulate false;
1229 interpolate false;
1230 pointAndNormalDict
1231     { basePoint      (0.0501 0.22801807 0.005);
1232       normalVector   (0 1 0);
1233     }
1234     fields
1235     ( Ux UPrime2Mean
1236     ); }
1237
1238 piano_52
1239     { type          surfaceFieldValue;
1240       libs          ("libfieldFunctionObjects.so");
1241       log           true;
1242       writeControl  writeTime;
1243       writeFields   true;
1244       surfaceFormat raw;
1245       regionType    sampledSurface;

```

```

1246     name                yNormal3;
1247     operation            areaAverage;
1248 sampledSurfaceDict
1249     { type                plane;
1250 planeType pointAndNormal;
1251 triangulate false;
1252 interpolate false;
1253     pointAndNormalDict
1254     { basePoint          (0.0501 0.23519683 0.005);
1255       normalVector      (0 1 0);
1256     }
1257     fields
1258     ( Ux UPrime2Mean
1259     ); }
1260
1261 piano_53
1262     { type                surfaceFieldValue;
1263       libs                ("libfieldFunctionObjects.so");
1264       log                  true;
1265       writeControl        writeTime;
1266       writeFields        true;
1267       surfaceFormat      raw;
1268       regionType         sampledSurface;
1269       name                yNormal4;
1270       operation            areaAverage;
1271 sampledSurfaceDict
1272     { type                plane;
1273 planeType pointAndNormal;
1274 triangulate false;
1275 interpolate false;
1276     pointAndNormalDict
1277     { basePoint          (0.0501 0.24252313 0.005);
1278       normalVector      (0 1 0);
1279     }
1280     fields
1281     ( Ux UPrime2Mean
1282     ); }
1283
1284 piano_54
1285     { type                surfaceFieldValue;
1286       libs                ("libfieldFunctionObjects.so");
1287       log                  true;
1288       writeControl        writeTime;
1289       writeFields        true;
1290       surfaceFormat      raw;
1291       regionType         sampledSurface;
1292       name                yNormal5;
1293       operation            areaAverage;
1294 sampledSurfaceDict
1295     { type                plane;
1296 planeType pointAndNormal;
1297 triangulate false;
1298 interpolate false;
1299     pointAndNormalDict
1300     { basePoint          (0.0501 0.2500001 0.005);
1301       normalVector      (0 1 0);
1302     }
1303     fields
1304     ( Ux UPrime2Mean
1305     ); }
1306
1307 // Piani del II blocco in cui è diviso il canale, parte intermedia inferiore,
1308 // con grading 1.5, circa 26 piani
1309 // Il numero dei piani continua dal blocco precedente con la numerazione, per
1310 // non creare problemi di sovrapposizione dei numeri; il blocco precedente
1311 // aveva 54 punti, quindi l 1 di questo blocco è il 55 totale
1312 piano_55
1313     { type                surfaceFieldValue;
1314       libs                ("libfieldFunctionObjects.so");

```

```

1312     log                true;
1313     writeControl       writeTime;
1314     writeFields        true;
1315     surfaceFormat      raw;
1316     regionType         sampledSurface;
1317     name                yNormal0;
1318     operation          areaAverage;
1319 sampledSurfaceDict
1320     { type                plane;
1321 planeType pointAndNormal;
1322 triangulate false;
1323 interpolate false;
1324     pointAndNormalDict
1325     { basePoint          (0.0501 0.26512729 0.005);
1326       normalVector      (0 1 0);
1327     }
1328     fields
1329     ( Ux UPrime2Mean
1330     ); }
1331
1332 piano_56
1333     { type                surfaceFieldValue;
1334       libs                ("libfieldFunctionObjects.so");
1335       log                  true;
1336       writeControl        writeTime;
1337       writeFields         true;
1338       surfaceFormat       raw;
1339       regionType          sampledSurface;
1340       name                 yNormal1;
1341       operation           areaAverage;
1342 sampledSurfaceDict
1343     { type                plane;
1344 planeType pointAndNormal;
1345 triangulate false;
1346 interpolate false;
1347     pointAndNormalDict
1348     { basePoint          (0.0501 0.27286971 0.005);
1349       normalVector      (0 1 0);
1350     }
1351     fields
1352     ( Ux UPrime2Mean
1353     ); }
1354
1355 piano_57
1356     { type                surfaceFieldValue;
1357       libs                ("libfieldFunctionObjects.so");
1358       log                  true;
1359       writeControl        writeTime;
1360       writeFields         true;
1361       surfaceFormat       raw;
1362       regionType          sampledSurface;
1363       name                 yNormal2;
1364       operation           areaAverage;
1365 sampledSurfaceDict
1366     { type                plane;
1367 planeType pointAndNormal;
1368 triangulate false;
1369 interpolate false;
1370     pointAndNormalDict
1371     { basePoint          (0.0501 0.28073383 0.005);
1372       normalVector      (0 1 0);
1373     }
1374     fields
1375     ( Ux UPrime2Mean
1376     ); }
1377
1378 piano_58
1379     { type                surfaceFieldValue;
1380       libs                ("libfieldFunctionObjects.so");

```

```

1381     log                true;
1382     writeControl       writeTime;
1383     writeFields        true;
1384     surfaceFormat      raw;
1385     regionType         sampledSurface;
1386     name                yNormal3;
1387     operation          areaAverage;
1388     sampledSurfaceDict
1389     { type              plane;
1390     planeType pointAndNormal;
1391     triangulate false;
1392     interpolate false;
1393     pointAndNormalDict
1394     { basePoint         (0.0501 0.28872154 0.005);
1395       normalVector     (0 1 0);
1396     }
1397     fields
1398     ( Ux UPrime2Mean
1399     ); }
1400
1401     piano_59
1402     { type              surfaceFieldValue;
1403       libs              ("libfieldFunctionObjects.so");
1404       log                true;
1405       writeControl       writeTime;
1406       writeFields        true;
1407       surfaceFormat      raw;
1408       regionType         sampledSurface;
1409       name                yNormal4;
1410       operation          areaAverage;
1411     sampledSurfaceDict
1412     { type              plane;
1413     planeType pointAndNormal;
1414     triangulate false;
1415     interpolate false;
1416     pointAndNormalDict
1417     { basePoint         (0.0501 0.29683480 0.005);
1418       normalVector     (0 1 0);
1419     }
1420     fields
1421     ( Ux UPrime2Mean
1422     ); }
1423
1424     piano_60
1425     { type              surfaceFieldValue;
1426       libs              ("libfieldFunctionObjects.so");
1427       log                true;
1428       writeControl       writeTime;
1429       writeFields        true;
1430       surfaceFormat      raw;
1431       regionType         sampledSurface;
1432       name                yNormal5;
1433       operation          areaAverage;
1434     sampledSurfaceDict
1435     { type              plane;
1436     planeType pointAndNormal;
1437     triangulate false;
1438     interpolate false;
1439     pointAndNormalDict
1440     { basePoint         (0.0501 0.30507557 0.005);
1441       normalVector     (0 1 0);
1442     }
1443     fields
1444     ( Ux UPrime2Mean
1445     ); }
1446
1447     piano_61
1448     { type              surfaceFieldValue;
1449       libs              ("libfieldFunctionObjects.so");

```

```

1450     log                true;
1451     writeControl       writeTime;
1452     writeFields        true;
1453     surfaceFormat      raw;
1454     regionType         sampledSurface;
1455     name                yNormal0;
1456     operation          areaAverage;
1457 sampledSurfaceDict
1458     { type              plane;
1459 planeType pointAndNormal;
1460 triangulate false;
1461 interpolate false;
1462     pointAndNormalDict
1463     { basePoint         (0.0501 0.31344587 0.005);
1464       normalVector      (0 1 0);
1465     }
1466     fields
1467     ( Ux UPrime2Mean
1468     ); }
1469
1470 piano_62
1471     { type              surfaceFieldValue;
1472       libs              ("libfieldFunctionObjects.so");
1473       log                true;
1474       writeControl       writeTime;
1475       writeFields        true;
1476       surfaceFormat      raw;
1477       regionType         sampledSurface;
1478       name                yNormal1;
1479       operation          areaAverage;
1480 sampledSurfaceDict
1481     { type              plane;
1482 planeType pointAndNormal;
1483 triangulate false;
1484 interpolate false;
1485     pointAndNormalDict
1486     { basePoint         (0.0501 0.32194772 0.005);
1487       normalVector      (0 1 0);
1488     }
1489     fields
1490     ( Ux UPrime2Mean
1491     ); }
1492
1493 piano_63
1494     { type              surfaceFieldValue;
1495       libs              ("libfieldFunctionObjects.so");
1496       log                true;
1497       writeControl       writeTime;
1498       writeFields        true;
1499       surfaceFormat      raw;
1500       regionType         sampledSurface;
1501       name                yNormal2;
1502       operation          areaAverage;
1503 sampledSurfaceDict
1504     { type              plane;
1505 planeType pointAndNormal;
1506 triangulate false;
1507 interpolate false;
1508     pointAndNormalDict
1509     { basePoint         (0.0501 0.33058319 0.005);
1510       normalVector      (0 1 0);
1511     }
1512     fields
1513     ( Ux UPrime2Mean
1514     ); }
1515
1516 piano_64
1517     { type              surfaceFieldValue;
1518       libs              ("libfieldFunctionObjects.so");

```

```

1519     log                true;
1520     writeControl       writeTime;
1521     writeFields        true;
1522     surfaceFormat      raw;
1523     regionType         sampledSurface;
1524     name                yNormal3;
1525     operation          areaAverage;
1526 sampledSurfaceDict
1527     { type              plane;
1528 planeType pointAndNormal;
1529 triangulate false;
1530 interpolate false;
1531 pointAndNormalDict
1532     { basePoint        (0.0501 0.33935439 0.005);
1533       normalVector     (0 1 0);
1534     }
1535     fields
1536     ( Ux UPrime2Mean
1537     ); }
1538
1539 piano_65
1540     { type              surfaceFieldValue;
1541       libs              ("libfieldFunctionObjects.so");
1542       log                true;
1543       writeControl       writeTime;
1544       writeFields        true;
1545       surfaceFormat      raw;
1546       regionType         sampledSurface;
1547       name                yNormal4;
1548       operation          areaAverage;
1549 sampledSurfaceDict
1550     { type              plane;
1551 planeType pointAndNormal;
1552 triangulate false;
1553 interpolate false;
1554 pointAndNormalDict
1555     { basePoint        (0.0501 0.34826345 0.005);
1556       normalVector     (0 1 0);
1557     }
1558     fields
1559     ( Ux UPrime2Mean
1560     ); }
1561
1562 piano_66
1563     { type              surfaceFieldValue;
1564       libs              ("libfieldFunctionObjects.so");
1565       log                true;
1566       writeControl       writeTime;
1567       writeFields        true;
1568       surfaceFormat      raw;
1569       regionType         sampledSurface;
1570       name                yNormal5;
1571       operation          areaAverage;
1572 sampledSurfaceDict
1573     { type              plane;
1574 planeType pointAndNormal;
1575 triangulate false;
1576 interpolate false;
1577 pointAndNormalDict
1578     { basePoint        (0.0501 0.35731253 0.005);
1579       normalVector     (0 1 0);
1580     }
1581     fields
1582     ( Ux UPrime2Mean
1583     ); }
1584
1585 piano_67
1586     { type              surfaceFieldValue;
1587       libs              ("libfieldFunctionObjects.so");

```

```

1588     log                true;
1589     writeControl       writeTime;
1590     writeFields       true;
1591     surfaceFormat     raw;
1592     regionType       sampledSurface;
1593     name              yNormal0;
1594     operation        areaAverage;
1595     sampledSurfaceDict
1596     { type              plane;
1597     planeType pointAndNormal;
1598     triangulate false;
1599     interpolate  false;
1600     pointAndNormalDict
1601     { basePoint      (0.0501 0.36650383 0.005);
1602       normalVector   (0 1 0);
1603     }
1604     fields
1605     ( Ux UPrime2Mean
1606     ); }
1607
1608     piano_68
1609     { type              surfaceFieldValue;
1610       libs              ("libfieldFunctionObjects.so");
1611       log                true;
1612       writeControl     writeTime;
1613       writeFields     true;
1614       surfaceFormat    raw;
1615       regionType      sampledSurface;
1616       name              yNormal1;
1617       operation        areaAverage;
1618     sampledSurfaceDict
1619     { type              plane;
1620     planeType pointAndNormal;
1621     triangulate false;
1622     interpolate  false;
1623     pointAndNormalDict
1624     { basePoint      (0.0501 0.37583960 0.005);
1625       normalVector   (0 1 0);
1626     }
1627     fields
1628     ( Ux UPrime2Mean
1629     ); }
1630
1631     piano_69
1632     { type              surfaceFieldValue;
1633       libs              ("libfieldFunctionObjects.so");
1634       log                true;
1635       writeControl     writeTime;
1636       writeFields     true;
1637       surfaceFormat    raw;
1638       regionType      sampledSurface;
1639       name              yNormal2;
1640       operation        areaAverage;
1641     sampledSurfaceDict
1642     { type              plane;
1643     planeType pointAndNormal;
1644     triangulate false;
1645     interpolate  false;
1646     pointAndNormalDict
1647     { basePoint      (0.0501 0.38532209 0.005);
1648       normalVector   (0 1 0);
1649     }
1650     fields
1651     ( Ux UPrime2Mean
1652     ); }
1653
1654     piano_70
1655     { type              surfaceFieldValue;
1656       libs              ("libfieldFunctionObjects.so");

```

```

1657     log                true;
1658     writeControl       writeTime;
1659     writeFields        true;
1660     surfaceFormat      raw;
1661     regionType         sampledSurface;
1662     name                yNormal3;
1663     operation          areaAverage;
1664 sampledSurfaceDict
1665     { type                plane;
1666 planeType pointAndNormal;
1667 triangulate false;
1668 interpolate false;
1669     pointAndNormalDict
1670     { basePoint          (0.0501 0.39495363 0.005);
1671       normalVector      (0 1 0);
1672     }
1673     fields
1674     ( Ux UPrime2Mean
1675     ); }
1676
1677 piano_71
1678     { type                surfaceFieldValue;
1679       libs                ("libfieldFunctionObjects.so");
1680       log                 true;
1681       writeControl       writeTime;
1682       writeFields        true;
1683       surfaceFormat      raw;
1684       regionType         sampledSurface;
1685       name                yNormal4;
1686       operation          areaAverage;
1687 sampledSurfaceDict
1688     { type                plane;
1689 planeType pointAndNormal;
1690 triangulate false;
1691 interpolate false;
1692     pointAndNormalDict
1693     { basePoint          (0.0501 0.40473654 0.005);
1694       normalVector      (0 1 0);
1695     }
1696     fields
1697     ( Ux UPrime2Mean
1698     ); }
1699
1700 piano_72
1701     { type                surfaceFieldValue;
1702       libs                ("libfieldFunctionObjects.so");
1703       log                 true;
1704       writeControl       writeTime;
1705       writeFields        true;
1706       surfaceFormat      raw;
1707       regionType         sampledSurface;
1708       name                yNormal5;
1709       operation          areaAverage;
1710 sampledSurfaceDict
1711     { type                plane;
1712 planeType pointAndNormal;
1713 triangulate false;
1714 interpolate false;
1715     pointAndNormalDict
1716     { basePoint          (0.0501 0.41467321 0.005);
1717       normalVector      (0 1 0);
1718     }
1719     fields
1720     ( Ux UPrime2Mean
1721     ); }
1722
1723 piano_73
1724     { type                surfaceFieldValue;
1725       libs                ("libfieldFunctionObjects.so");

```

```

1726     log                true;
1727     writeControl       writeTime;
1728     writeFields       true;
1729     surfaceFormat     raw;
1730     regionType       sampledSurface;
1731     name              yNormal0;
1732     operation        areaAverage;
1733 sampledSurfaceDict
1734     { type              plane;
1735 planeType pointAndNormal;
1736 triangulate false;
1737 interpolate  false;
1738 pointAndNormalDict
1739     { basePoint       (0.0501 0.42476605 0.005);
1740       normalVector   (0 1 0);
1741     }
1742     fields
1743     (   Ux UPrime2Mean
1744     ); }
1745
1746 piano_74
1747     { type              surfaceFieldValue;
1748       libs              ("libfieldFunctionObjects.so");
1749       log                true;
1750       writeControl     writeTime;
1751       writeFields     true;
1752       surfaceFormat   raw;
1753       regionType     sampledSurface;
1754       name            yNormal1;
1755       operation      areaAverage;
1756 sampledSurfaceDict
1757     { type              plane;
1758 planeType pointAndNormal;
1759 triangulate false;
1760 interpolate  false;
1761 pointAndNormalDict
1762     { basePoint       (0.0501 0.43501753 0.005);
1763       normalVector   (0 1 0);
1764     }
1765     fields
1766     (   Ux UPrime2Mean
1767     ); }
1768
1769 piano_75
1770     { type              surfaceFieldValue;
1771       libs              ("libfieldFunctionObjects.so");
1772       log                true;
1773       writeControl     writeTime;
1774       writeFields     true;
1775       surfaceFormat   raw;
1776       regionType     sampledSurface;
1777       name            yNormal2;
1778       operation      areaAverage;
1779 sampledSurfaceDict
1780     { type              plane;
1781 planeType pointAndNormal;
1782 triangulate false;
1783 interpolate  false;
1784 pointAndNormalDict
1785     { basePoint       (0.0501 0.44543013 0.005);
1786       normalVector   (0 1 0);
1787     }
1788     fields
1789     (   Ux UPrime2Mean
1790     ); }
1791
1792 piano_76
1793     { type              surfaceFieldValue;
1794       libs              ("libfieldFunctionObjects.so");

```

```

1795     log                true;
1796     writeControl       writeTime;
1797     writeFields        true;
1798     surfaceFormat      raw;
1799     regionType         sampledSurface;
1800     name                yNormal3;
1801     operation          areaAverage;
1802 sampledSurfaceDict
1803     { type                plane;
1804 planeType pointAndNormal;
1805 triangulate false;
1806 interpolate false;
1807     pointAndNormalDict
1808         { basePoint      (0.0501 0.45600638 0.005);
1809           normalVector   (0 1 0);
1810         }
1811     fields
1812         ( Ux UPrime2Mean
1813         ); }
1814
1815 piano_77
1816     { type                surfaceFieldValue;
1817       libs                ("libfieldFunctionObjects.so");
1818       log                 true;
1819       writeControl       writeTime;
1820       writeFields        true;
1821       surfaceFormat      raw;
1822       regionType         sampledSurface;
1823       name                yNormal4;
1824       operation          areaAverage;
1825 sampledSurfaceDict
1826     { type                plane;
1827 planeType pointAndNormal;
1828 triangulate false;
1829 interpolate false;
1830     pointAndNormalDict
1831         { basePoint      (0.0501 0.46674886 0.005);
1832           normalVector   (0 1 0);
1833         }
1834     fields
1835         ( Ux UPrime2Mean
1836         ); }
1837
1838 piano_78
1839     { type                surfaceFieldValue;
1840       libs                ("libfieldFunctionObjects.so");
1841       log                 true;
1842       writeControl       writeTime;
1843       writeFields        true;
1844       surfaceFormat      raw;
1845       regionType         sampledSurface;
1846       name                yNormal5;
1847       operation          areaAverage;
1848 sampledSurfaceDict
1849     { type                plane;
1850 planeType pointAndNormal;
1851 triangulate false;
1852 interpolate false;
1853     pointAndNormalDict
1854         { basePoint      (0.0501 0.47766018 0.005);
1855           normalVector   (0 1 0);
1856         }
1857     fields
1858         ( Ux UPrime2Mean
1859         ); }
1860
1861 piano_79
1862     { type                surfaceFieldValue;
1863       libs                ("libfieldFunctionObjects.so");

```

```

1864     log                true;
1865     writeControl       writeTime;
1866     writeFields        true;
1867     surfaceFormat      raw;
1868     regionType         sampledSurface;
1869     name                yNormal0;
1870     operation          areaAverage;
1871 sampledSurfaceDict
1872     { type                plane;
1873 planeType pointAndNormal;
1874 triangulate false;
1875 interpolate false;
1876 pointAndNormalDict
1877     { basePoint          (0.0501 0.48874299 0.005);
1878       normalVector      (0 1 0);
1879     }
1880     fields
1881     (   Ux UPrime2Mean
1882     ); }
1883
1884 piano_80
1885     { type                surfaceFieldValue;
1886       libs                ("libfieldFunctionObjects.so");
1887       log                  true;
1888       writeControl        writeTime;
1889       writeFields         true;
1890       surfaceFormat        raw;
1891       regionType           sampledSurface;
1892       name                 yNormal1;
1893       operation            areaAverage;
1894 sampledSurfaceDict
1895     { type                plane;
1896 planeType pointAndNormal;
1897 triangulate false;
1898 interpolate false;
1899 pointAndNormalDict
1900     { basePoint          (0.0501 0.500001 0.005);
1901       normalVector      (0 1 0);
1902     }
1903     fields
1904     (   Ux UPrime2Mean
1905     ); }
1906 // Piani del III blocco in cui è diviso il canale, parte centrale inferiore del
1907 // Il numero dei piani continua dai blocchi precedenti con la numerazione, per
1908 // non creare problemi di sovrapposizione dei numeri; i blocchi precedente
1909 // avevano 54 e 26 punti, quindi 1 1 di questo blocco è 1 81 totale
1908 piano_81
1909     { type                surfaceFieldValue;
1910       libs                ("libfieldFunctionObjects.so");
1911       log                  true;
1912       writeControl        writeTime;
1913       writeFields         true;
1914       surfaceFormat        raw;
1915       regionType           sampledSurface;
1916       name                 yNormal2;
1917       operation            areaAverage;
1918 sampledSurfaceDict
1919     { type                plane;
1920 planeType pointAndNormal;
1921 triangulate false;
1922 interpolate false;
1923 pointAndNormalDict
1924     { basePoint          (0.0501 0.52272727 0.005);
1925       normalVector      (0 1 0);
1926     }
1927     fields
1928     (   Ux UPrime2Mean
1929     ); }

```

```

1930
1931 piano_82
1932     { type          surfaceFieldValue;
1933       libs          ("libfieldFunctionObjects.so");
1934       log           true;
1935       writeControl  writeTime;
1936       writeFields  true;
1937       surfaceFormat raw;
1938       regionType   sampledSurface;
1939       name         yNormal3;
1940       operation    areaAverage;
1941 sampledSurfaceDict
1942     { type          plane;
1943 planeType pointAndNormal;
1944 triangulate false;
1945 interpolate  false;
1946 pointAndNormalDict
1947     { basePoint    (0.0501 0.53409091 0.005);
1948       normalVector (0 1 0);
1949     }
1950     fields
1951     (   Ux UPrime2Mean
1952       );
1953
1954 piano_83
1955     { type          surfaceFieldValue;
1956       libs          ("libfieldFunctionObjects.so");
1957       log           true;
1958       writeControl  writeTime;
1959       writeFields  true;
1960       surfaceFormat raw;
1961       regionType   sampledSurface;
1962       name         yNormal4;
1963       operation    areaAverage;
1964 sampledSurfaceDict
1965     { type          plane;
1966 planeType pointAndNormal;
1967 triangulate false;
1968 interpolate  false;
1969 pointAndNormalDict
1970     { basePoint    (0.0501 0.54545455 0.005);
1971       normalVector (0 1 0);
1972     }
1973     fields
1974     (   Ux UPrime2Mean
1975       );
1976
1977 piano_84
1978     { type          surfaceFieldValue;
1979       libs          ("libfieldFunctionObjects.so");
1980       log           true;
1981       writeControl  writeTime;
1982       writeFields  true;
1983       surfaceFormat raw;
1984       regionType   sampledSurface;
1985       name         yNormal5;
1986       operation    areaAverage;
1987 sampledSurfaceDict
1988     { type          plane;
1989 planeType pointAndNormal;
1990 triangulate false;
1991 interpolate  false;
1992 pointAndNormalDict
1993     { basePoint    (0.0501 0.55681818 0.005);
1994       normalVector (0 1 0);
1995     }
1996     fields
1997     (   Ux UPrime2Mean
1998       );

```

```

1999
2000 piano_85
2001     { type          surfaceFieldValue;
2002       libs          ("libfieldFunctionObjects.so");
2003       log           true;
2004       writeControl  writeTime;
2005       writeFields  true;
2006       surfaceFormat raw;
2007       regionType   sampledSurface;
2008       name         yNormal0;
2009       operation    areaAverage;
2010 sampledSurfaceDict
2011     { type          plane;
2012 planeType pointAndNormal;
2013 triangulate false;
2014 interpolate  false;
2015 pointAndNormalDict
2016     { basePoint    (0.0501 0.56818182 0.005);
2017       normalVector (0 1 0);
2018 }
2019     fields
2020     (   Ux UPrime2Mean
2021       );
2022
2023 piano_86
2024     { type          surfaceFieldValue;
2025       libs          ("libfieldFunctionObjects.so");
2026       log           true;
2027       writeControl  writeTime;
2028       writeFields  true;
2029       surfaceFormat raw;
2030       regionType   sampledSurface;
2031       name         yNormal1;
2032       operation    areaAverage;
2033 sampledSurfaceDict
2034     { type          plane;
2035 planeType pointAndNormal;
2036 triangulate false;
2037 interpolate  false;
2038 pointAndNormalDict
2039     { basePoint    (0.0501 0.57954546 0.005);
2040       normalVector (0 1 0);
2041 }
2042     fields
2043     (   Ux UPrime2Mean
2044       );
2045
2046 piano_87
2047     { type          surfaceFieldValue;
2048       libs          ("libfieldFunctionObjects.so");
2049       log           true;
2050       writeControl  writeTime;
2051       writeFields  true;
2052       surfaceFormat raw;
2053       regionType   sampledSurface;
2054       name         yNormal2;
2055       operation    areaAverage;
2056 sampledSurfaceDict
2057     { type          plane;
2058 planeType pointAndNormal;
2059 triangulate false;
2060 interpolate  false;
2061 pointAndNormalDict
2062     { basePoint    (0.0501 0.59090909 0.005);
2063       normalVector (0 1 0);
2064 }
2065     fields
2066     (   Ux UPrime2Mean
2067       );

```

```

2068
2069 piano_88
2070     { type          surfaceFieldValue;
2071       libs          ("libfieldFunctionObjects.so");
2072       log           true;
2073       writeControl  writeTime;
2074       writeFields  true;
2075       surfaceFormat raw;
2076       regionType   sampledSurface;
2077       name         yNormal3;
2078       operation    areaAverage;
2079 sampledSurfaceDict
2080     { type          plane;
2081 planeType pointAndNormal;
2082 triangulate false;
2083 interpolate  false;
2084 pointAndNormalDict
2085     { basePoint    (0.0501 0.60227273 0.005);
2086       normalVector (0 1 0);
2087     }
2088     fields
2089     (   Ux UPrime2Mean
2090       );
2091
2092 piano_89
2093     { type          surfaceFieldValue;
2094       libs          ("libfieldFunctionObjects.so");
2095       log           true;
2096       writeControl  writeTime;
2097       writeFields  true;
2098       surfaceFormat raw;
2099       regionType   sampledSurface;
2100       name         yNormal4;
2101       operation    areaAverage;
2102 sampledSurfaceDict
2103     { type          plane;
2104 planeType pointAndNormal;
2105 triangulate false;
2106 interpolate  false;
2107 pointAndNormalDict
2108     { basePoint    (0.0501 0.61363637 0.005);
2109       normalVector (0 1 0);
2110     }
2111     fields
2112     (   Ux UPrime2Mean
2113       );
2114
2115 piano_90
2116     { type          surfaceFieldValue;
2117       libs          ("libfieldFunctionObjects.so");
2118       log           true;
2119       writeControl  writeTime;
2120       writeFields  true;
2121       surfaceFormat raw;
2122       regionType   sampledSurface;
2123       name         yNormal5;
2124       operation    areaAverage;
2125 sampledSurfaceDict
2126     { type          plane;
2127 planeType pointAndNormal;
2128 triangulate false;
2129 interpolate  false;
2130 pointAndNormalDict
2131     { basePoint    (0.0501 0.625001 0.005);
2132       normalVector (0 1 0);
2133     }
2134     fields
2135     (   Ux UPrime2Mean
2136       );

```

```

2137
2138 piano_91
2139     { type          surfaceFieldValue;
2140       libs          ("libfieldFunctionObjects.so");
2141       log           true;
2142       writeControl  writeTime;
2143       writeFields  true;
2144       surfaceFormat raw;
2145       regionType   sampledSurface;
2146       name         yNormal0;
2147       operation    areaAverage;
2148 sampledSurfaceDict
2149     { type          plane;
2150 planeType pointAndNormal;
2151 triangulate false;
2152 interpolate  false;
2153 pointAndNormalDict
2154     { basePoint    (0.0501 0.63636364 0.005);
2155       normalVector (0 1 0);
2156 }
2157     fields
2158     (    Ux UPrime2Mean
2159     ); }
2160
2161 piano_92
2162     { type          surfaceFieldValue;
2163       libs          ("libfieldFunctionObjects.so");
2164       log           true;
2165       writeControl  writeTime;
2166       writeFields  true;
2167       surfaceFormat raw;
2168       regionType   sampledSurface;
2169       name         yNormal1;
2170       operation    areaAverage;
2171 sampledSurfaceDict
2172     { type          plane;
2173 planeType pointAndNormal;
2174 triangulate false;
2175 interpolate  false;
2176 pointAndNormalDict
2177     { basePoint    (0.0501 0.65909091 0.005);
2178       normalVector (0 1 0);
2179 }
2180     fields
2181     (    Ux UPrime2Mean
2182     ); }
2183
2184 piano_93
2185     { type          surfaceFieldValue;
2186       libs          ("libfieldFunctionObjects.so");
2187       log           true;
2188       writeControl  writeTime;
2189       writeFields  true;
2190       surfaceFormat raw;
2191       regionType   sampledSurface;
2192       name         yNormal2;
2193       operation    areaAverage;
2194 sampledSurfaceDict
2195     { type          plane;
2196 planeType pointAndNormal;
2197 triangulate false;
2198 interpolate  false;
2199 pointAndNormalDict
2200     { basePoint    (0.0501 0.65909091 0.005);
2201       normalVector (0 1 0);
2202 }
2203     fields
2204     (    Ux UPrime2Mean
2205     ); }

```

```

2206
2207 piano_94
2208     { type          surfaceFieldValue;
2209       libs          ("libfieldFunctionObjects.so");
2210       log           true;
2211       writeControl  writeTime;
2212       writeFields  true;
2213       surfaceFormat raw;
2214       regionType   sampledSurface;
2215       name         yNormal3;
2216       operation    areaAverage;
2217 sampledSurfaceDict
2218     { type          plane;
2219 planeType pointAndNormal;
2220 triangulate false;
2221 interpolate  false;
2222 pointAndNormalDict
2223     { basePoint    (0.0501 0.67045455 0.005);
2224       normalVector (0 1 0);
2225     }
2226     fields
2227     (   Ux UPrime2Mean
2228       );
2229
2230 piano_95
2231     { type          surfaceFieldValue;
2232       libs          ("libfieldFunctionObjects.so");
2233       log           true;
2234       writeControl  writeTime;
2235       writeFields  true;
2236       surfaceFormat raw;
2237       regionType   sampledSurface;
2238       name         yNormal4;
2239       operation    areaAverage;
2240 sampledSurfaceDict
2241     { type          plane;
2242 planeType pointAndNormal;
2243 triangulate false;
2244 interpolate  false;
2245 pointAndNormalDict
2246     { basePoint    (0.0501 0.68181819 0.005);
2247       normalVector (0 1 0);
2248     }
2249     fields
2250     (   Ux UPrime2Mean
2251       );
2252
2253 piano_96
2254     { type          surfaceFieldValue;
2255       libs          ("libfieldFunctionObjects.so");
2256       log           true;
2257       writeControl  writeTime;
2258       writeFields  true;
2259       surfaceFormat raw;
2260       regionType   sampledSurface;
2261       name         yNormal5;
2262       operation    areaAverage;
2263 sampledSurfaceDict
2264     { type          plane;
2265 planeType pointAndNormal;
2266 triangulate false;
2267 interpolate  false;
2268 pointAndNormalDict
2269     { basePoint    (0.0501 0.69318182 0.005);
2270       normalVector (0 1 0);
2271     }
2272     fields
2273     (   Ux UPrime2Mean
2274       );

```

```

2275
2276 piano_97
2277     { type          surfaceFieldValue;
2278       libs          ("libfieldFunctionObjects.so");
2279       log            true;
2280       writeControl  writeTime;
2281       writeFields   true;
2282       surfaceFormat raw;
2283       regionType    sampledSurface;
2284       name          yNormal0;
2285       operation     areaAverage;
2286 sampledSurfaceDict
2287     { type          plane;
2288 planeType pointAndNormal;
2289 triangulate false;
2290 interpolate  false;
2291 pointAndNormalDict
2292     { basePoint    (0.0501 0.70454546 0.005);
2293       normalVector (0 1 0);
2294     }
2295     fields
2296     (   Ux UPrime2Mean
2297     ); }
2298
2299 piano_98
2300     { type          surfaceFieldValue;
2301       libs          ("libfieldFunctionObjects.so");
2302       log            true;
2303       writeControl  writeTime;
2304       writeFields   true;
2305       surfaceFormat raw;
2306       regionType    sampledSurface;
2307       name          yNormal1;
2308       operation     areaAverage;
2309 sampledSurfaceDict
2310     { type          plane;
2311 planeType pointAndNormal;
2312 triangulate false;
2313 interpolate  false;
2314 pointAndNormalDict
2315     { basePoint    (0.0501 0.71590910 0.005);
2316       normalVector (0 1 0);
2317     }
2318     fields
2319     (   Ux UPrime2Mean
2320     ); }
2321
2322 piano_99
2323     { type          surfaceFieldValue;
2324       libs          ("libfieldFunctionObjects.so");
2325       log            true;
2326       writeControl  writeTime;
2327       writeFields   true;
2328       surfaceFormat raw;
2329       regionType    sampledSurface;
2330       name          yNormal2;
2331       operation     areaAverage;
2332 sampledSurfaceDict
2333     { type          plane;
2334 planeType pointAndNormal;
2335 triangulate false;
2336 interpolate  false;
2337 pointAndNormalDict
2338     { basePoint    (0.0501 0.72727273 0.005);
2339       normalVector (0 1 0);
2340     }
2341     fields
2342     (   Ux UPrime2Mean
2343     ); }

```

```

2344
2345 piano_100
2346     { type          surfaceFieldValue;
2347       libs          ("libfieldFunctionObjects.so");
2348       log           true;
2349       writeControl  writeTime;
2350       writeFields  true;
2351       surfaceFormat raw;
2352       regionType   sampledSurface;
2353       name         yNormal3;
2354       operation    areaAverage;
2355 sampledSurfaceDict
2356     { type          plane;
2357 planeType pointAndNormal;
2358 triangulate false;
2359 interpolate  false;
2360 pointAndNormalDict
2361     { basePoint    (0.0501 0.73863637 0.005);
2362       normalVector (0 1 0);
2363     }
2364     fields
2365     (   Ux UPrime2Mean
2366       );
2367
2368 piano_101
2369     { type          surfaceFieldValue;
2370       libs          ("libfieldFunctionObjects.so");
2371       log           true;
2372       writeControl  writeTime;
2373       writeFields  true;
2374       surfaceFormat raw;
2375       regionType   sampledSurface;
2376       name         yNormal4;
2377       operation    areaAverage;
2378 sampledSurfaceDict
2379     { type          plane;
2380 planeType pointAndNormal;
2381 triangulate false;
2382 interpolate  false;
2383 pointAndNormalDict
2384     { basePoint    (0.0501 0.75000001 0.005);
2385       normalVector (0 1 0);
2386     }
2387     fields
2388     (   Ux UPrime2Mean
2389       );
2390
2391 piano_102
2392     { type          surfaceFieldValue;
2393       libs          ("libfieldFunctionObjects.so");
2394       log           true;
2395       writeControl  writeTime;
2396       writeFields  true;
2397       surfaceFormat raw;
2398       regionType   sampledSurface;
2399       name         yNormal5;
2400       operation    areaAverage;
2401 sampledSurfaceDict
2402     { type          plane;
2403 planeType pointAndNormal;
2404 triangulate false;
2405 interpolate  false;
2406 pointAndNormalDict
2407     { basePoint    (0.0501 0.76136364 0.005);
2408       normalVector (0 1 0);
2409     }
2410     fields
2411     (   Ux UPrime2Mean
2412       );

```

```

2413
2414 piano_103
2415     { type          surfaceFieldValue;
2416       libs          ("libfieldFunctionObjects.so");
2417       log            true;
2418       writeControl  writeTime;
2419       writeFields   true;
2420       surfaceFormat raw;
2421       regionType    sampledSurface;
2422       name          yNormal0;
2423       operation     areaAverage;
2424 sampledSurfaceDict
2425     { type          plane;
2426 planeType pointAndNormal;
2427 triangulate false;
2428 interpolate  false;
2429 pointAndNormalDict
2430     { basePoint    (0.0501 0.77272728 0.005);
2431       normalVector (0 1 0);
2432 }
2433     fields
2434     (   Ux UPrime2Mean
2435       );
2436
2437 piano_104
2438     { type          surfaceFieldValue;
2439       libs          ("libfieldFunctionObjects.so");
2440       log            true;
2441       writeControl  writeTime;
2442       writeFields   true;
2443       surfaceFormat raw;
2444       regionType    sampledSurface;
2445       name          yNormal1;
2446       operation     areaAverage;
2447 sampledSurfaceDict
2448     { type          plane;
2449 planeType pointAndNormal;
2450 triangulate false;
2451 interpolate  false;
2452 pointAndNormalDict
2453     { basePoint    (0.0501 0.78409092 0.005);
2454       normalVector (0 1 0);
2455 }
2456     fields
2457     (   Ux UPrime2Mean
2458       );
2459
2460 piano_105
2461     { type          surfaceFieldValue;
2462       libs          ("libfieldFunctionObjects.so");
2463       log            true;
2464       writeControl  writeTime;
2465       writeFields   true;
2466       surfaceFormat raw;
2467       regionType    sampledSurface;
2468       name          yNormal2;
2469       operation     areaAverage;
2470 sampledSurfaceDict
2471     { type          plane;
2472 planeType pointAndNormal;
2473 triangulate false;
2474 interpolate  false;
2475 pointAndNormalDict
2476     { basePoint    (0.0501 0.79545455 0.005);
2477       normalVector (0 1 0);
2478 }
2479     fields
2480     (   Ux UPrime2Mean
2481       );

```

```

2482
2483 piano_106
2484     { type          surfaceFieldValue;
2485       libs          ("libfieldFunctionObjects.so");
2486       log           true;
2487       writeControl  writeTime;
2488       writeFields  true;
2489       surfaceFormat raw;
2490       regionType   sampledSurface;
2491       name         yNormal3;
2492       operation    areaAverage;
2493 sampledSurfaceDict
2494     { type          plane;
2495 planeType pointAndNormal;
2496 triangulate false;
2497 interpolate false;
2498 pointAndNormalDict
2499     { basePoint    (0.0501 0.80681819 0.005);
2500       normalVector (0 1 0);
2501     }
2502     fields
2503     (   Ux UPrime2Mean
2504       );
2505
2506 piano_107
2507     { type          surfaceFieldValue;
2508       libs          ("libfieldFunctionObjects.so");
2509       log           true;
2510       writeControl  writeTime;
2511       writeFields  true;
2512       surfaceFormat raw;
2513       regionType   sampledSurface;
2514       name         yNormal4;
2515       operation    areaAverage;
2516 sampledSurfaceDict
2517     { type          plane;
2518 planeType pointAndNormal;
2519 triangulate false;
2520 interpolate false;
2521 pointAndNormalDict
2522     { basePoint    (0.0501 0.81818183 0.005);
2523       normalVector (0 1 0);
2524     }
2525     fields
2526     (   Ux UPrime2Mean
2527       );
2528
2529 piano_108
2530     { type          surfaceFieldValue;
2531       libs          ("libfieldFunctionObjects.so");
2532       log           true;
2533       writeControl  writeTime;
2534       writeFields  true;
2535       surfaceFormat raw;
2536       regionType   sampledSurface;
2537       name         yNormal5;
2538       operation    areaAverage;
2539 sampledSurfaceDict
2540     { type          plane;
2541 planeType pointAndNormal;
2542 triangulate false;
2543 interpolate false;
2544 pointAndNormalDict
2545     { basePoint    (0.0501 0.82954546 0.005);
2546       normalVector (0 1 0);
2547     }
2548     fields
2549     (   Ux UPrime2Mean
2550       );

```

```

2551
2552 piano_109
2553     { type          surfaceFieldValue;
2554       libs          ("libfieldFunctionObjects.so");
2555       log           true;
2556       writeControl  writeTime;
2557       writeFields   true;
2558       surfaceFormat raw;
2559       regionType    sampledSurface;
2560       name          yNormal3;
2561       operation     areaAverage;
2562 sampledSurfaceDict
2563     { type          plane;
2564 planeType pointAndNormal;
2565 triangulate false;
2566 interpolate false;
2567 pointAndNormalDict
2568     { basePoint    (0.0501 0.84090910 0.005);
2569       normalVector (0 1 0);
2570 }
2571     fields
2572     ( Ux UPrime2Mean
2573     );
2574
2575 piano_110
2576     { type          surfaceFieldValue;
2577       libs          ("libfieldFunctionObjects.so");
2578       log           true;
2579       writeControl  writeTime;
2580       writeFields   true;
2581       surfaceFormat raw;
2582       regionType    sampledSurface;
2583       name          yNormal4;
2584       operation     areaAverage;
2585 sampledSurfaceDict
2586     { type          plane;
2587 planeType pointAndNormal;
2588 triangulate false;
2589 interpolate false;
2590 pointAndNormalDict
2591     { basePoint    (0.0501 0.85227274 0.005);
2592       normalVector (0 1 0);
2593 }
2594     fields
2595     ( Ux UPrime2Mean
2596     );
2597
2598 piano_111
2599     { type          surfaceFieldValue;
2600       libs          ("libfieldFunctionObjects.so");
2601       log           true;
2602       writeControl  writeTime;
2603       writeFields   true;
2604       surfaceFormat raw;
2605       regionType    sampledSurface;
2606       name          yNormal5;
2607       operation     areaAverage;
2608 sampledSurfaceDict
2609     { type          plane;
2610 planeType pointAndNormal;
2611 triangulate false;
2612 interpolate false;
2613 pointAndNormalDict
2614     { basePoint    (0.0501 0.86363637 0.005);
2615       normalVector (0 1 0);
2616 }
2617     fields
2618     ( Ux UPrime2Mean
2619     );

```

```

2620
2621 piano_112
2622     { type          surfaceFieldValue;
2623       libs          ("libfieldFunctionObjects.so");
2624       log           true;
2625       writeControl  writeTime;
2626       writeFields   true;
2627       surfaceFormat raw;
2628       regionType    sampledSurface;
2629       name          yNormal0;
2630       operation     areaAverage;
2631 sampledSurfaceDict
2632     { type          plane;
2633 planeType pointAndNormal;
2634 triangulate false;
2635 interpolate  false;
2636 pointAndNormalDict
2637     { basePoint    (0.0501 0.87500001 0.005);
2638       normalVector (0 1 0);
2639     }
2640     fields
2641     (   Ux UPrime2Mean
2642       );
2643
2644 piano_113
2645     { type          surfaceFieldValue;
2646       libs          ("libfieldFunctionObjects.so");
2647       log           true;
2648       writeControl  writeTime;
2649       writeFields   true;
2650       surfaceFormat raw;
2651       regionType    sampledSurface;
2652       name          yNormal1;
2653       operation     areaAverage;
2654 sampledSurfaceDict
2655     { type          plane;
2656 planeType pointAndNormal;
2657 triangulate false;
2658 interpolate  false;
2659 pointAndNormalDict
2660     { basePoint    (0.0501 0.88636365 0.005);
2661       normalVector (0 1 0);
2662     }
2663     fields
2664     (   Ux UPrime2Mean
2665       );
2666
2667 piano_114
2668     { type          surfaceFieldValue;
2669       libs          ("libfieldFunctionObjects.so");
2670       log           true;
2671       writeControl  writeTime;
2672       writeFields   true;
2673       surfaceFormat raw;
2674       regionType    sampledSurface;
2675       name          yNormal2;
2676       operation     areaAverage;
2677 sampledSurfaceDict
2678     { type          plane;
2679 planeType pointAndNormal;
2680 triangulate false;
2681 interpolate  false;
2682 pointAndNormalDict
2683     { basePoint    (0.0501 0.89772728 0.005);
2684       normalVector (0 1 0);
2685     }
2686     fields
2687     (   Ux UPrime2Mean
2688       );

```

```

2689
2690 piano_115
2691     { type          surfaceFieldValue;
2692       libs          ("libfieldFunctionObjects.so");
2693       log           true;
2694       writeControl  writeTime;
2695       writeFields  true;
2696       surfaceFormat raw;
2697       regionType   sampledSurface;
2698       name         yNormal3;
2699       operation    areaAverage;
2700 sampledSurfaceDict
2701     { type          plane;
2702 planeType pointAndNormal;
2703 triangulate false;
2704 interpolate  false;
2705 pointAndNormalDict
2706     { basePoint    (0.0501 0.90909092 0.005);
2707       normalVector (0 1 0);
2708 }
2709     fields
2710     ( Ux UPrime2Mean
2711     );
2712
2713 piano_116
2714     { type          surfaceFieldValue;
2715       libs          ("libfieldFunctionObjects.so");
2716       log           true;
2717       writeControl  writeTime;
2718       writeFields  true;
2719       surfaceFormat raw;
2720       regionType   sampledSurface;
2721       name         yNormal4;
2722       operation    areaAverage;
2723 sampledSurfaceDict
2724     { type          plane;
2725 planeType pointAndNormal;
2726 triangulate false;
2727 interpolate  false;
2728 pointAndNormalDict
2729     { basePoint    (0.0501 0.92045456 0.005);
2730       normalVector (0 1 0);
2731 }
2732     fields
2733     ( Ux UPrime2Mean
2734     );
2735
2736 piano_117
2737     { type          surfaceFieldValue;
2738       libs          ("libfieldFunctionObjects.so");
2739       log           true;
2740       writeControl  writeTime;
2741       writeFields  true;
2742       surfaceFormat raw;
2743       regionType   sampledSurface;
2744       name         yNormal5;
2745       operation    areaAverage;
2746 sampledSurfaceDict
2747     { type          plane;
2748 planeType pointAndNormal;
2749 triangulate false;
2750 interpolate  false;
2751 pointAndNormalDict
2752     { basePoint    (0.0501 0.93181819 0.005);
2753       normalVector (0 1 0);
2754 }
2755     fields
2756     ( Ux UPrime2Mean
2757     );

```

```

2758
2759 piano_118
2760     { type          surfaceFieldValue;
2761       libs          ("libfieldFunctionObjects.so");
2762       log           true;
2763       writeControl  writeTime;
2764       writeFields  true;
2765       surfaceFormat raw;
2766       regionType   sampledSurface;
2767       name         yNormal3;
2768       operation    areaAverage;
2769 sampledSurfaceDict
2770     { type          plane;
2771 planeType pointAndNormal;
2772 triangulate false;
2773 interpolate  false;
2774 pointAndNormalDict
2775     { basePoint    (0.0501 0.94318183 0.005);
2776       normalVector (0 1 0);
2777 }
2778     fields
2779     (   Ux UPrime2Mean
2780       );
2781
2782 piano_119
2783     { type          surfaceFieldValue;
2784       libs          ("libfieldFunctionObjects.so");
2785       log           true;
2786       writeControl  writeTime;
2787       writeFields  true;
2788       surfaceFormat raw;
2789       regionType   sampledSurface;
2790       name         yNormal4;
2791       operation    areaAverage;
2792 sampledSurfaceDict
2793     { type          plane;
2794 planeType pointAndNormal;
2795 triangulate false;
2796 interpolate  false;
2797 pointAndNormalDict
2798     { basePoint    (0.0501 0.95454547 0.005);
2799       normalVector (0 1 0);
2800 }
2801     fields
2802     (   Ux UPrime2Mean
2803       );
2804
2805 piano_120
2806     { type          surfaceFieldValue;
2807       libs          ("libfieldFunctionObjects.so");
2808       log           true;
2809       writeControl  writeTime;
2810       writeFields  true;
2811       surfaceFormat raw;
2812       regionType   sampledSurface;
2813       name         yNormal5;
2814       operation    areaAverage;
2815 sampledSurfaceDict
2816     { type          plane;
2817 planeType pointAndNormal;
2818 triangulate false;
2819 interpolate  false;
2820 pointAndNormalDict
2821     { basePoint    (0.0501 0.96590910 0.005);
2822       normalVector (0 1 0);
2823 }
2824     fields
2825     (   Ux UPrime2Mean
2826       );

```

```

2827
2828 piano_121
2829     {   type           surfaceFieldValue;
2830         libs           ("libfieldFunctionObjects.so");
2831         log            true;
2832         writeControl   writeTime;
2833         writeFields    true;
2834         surfaceFormat  raw;
2835         regionType     sampledSurface;
2836         name           yNormal0;
2837         operation      areaAverage;
2838     sampledSurfaceDict
2839     {   type           plane;
2840     planeType pointAndNormal;
2841     triangulate false;
2842     interpolate  false;
2843     pointAndNormalDict
2844     {   basePoint     (0.0501 0.97727274 0.005);
2845         normalVector  (0 1 0);
2846     }
2847     fields
2848     (   Ux UPrime2Mean
2849     ); }
2850
2851 piano_122
2852     {   type           surfaceFieldValue;
2853         libs           ("libfieldFunctionObjects.so");
2854         log            true;
2855         writeControl   writeTime;
2856         writeFields    true;
2857         surfaceFormat  raw;
2858         regionType     sampledSurface;
2859         name           yNormal1;
2860         operation      areaAverage;
2861     sampledSurfaceDict
2862     {   type           plane;
2863     planeType pointAndNormal;
2864     triangulate false;
2865     interpolate  false;
2866     pointAndNormalDict
2867     {   basePoint     (0.0501 0.98863638 0.005);
2868         normalVector  (0 1 0);
2869     }
2870     fields
2871     (   Ux UPrime2Mean
2872     ); }
2873
2874 // Piano centrale del canale
2875 piano_123
2876     {   type           surfaceFieldValue;
2877         libs           ("libfieldFunctionObjects.so");
2878         log            true;
2879         writeControl   writeTime;
2880         writeFields    true;
2881         surfaceFormat  raw;
2882         regionType     sampledSurface;
2883         name           yNormal2;
2884         operation      areaAverage;
2885     sampledSurfaceDict
2886     {   type           plane;
2887     planeType pointAndNormal;
2888     triangulate false;
2889     interpolate  false;
2890     pointAndNormalDict
2891     {   basePoint     (0.0501 1.0000001 0.005);
2892         normalVector  (0 1 0);
2893     }
2894     fields
2895     (   Ux UPrime2Mean

```

```

2896     );    }
2897 //Piani del IV blocco in cui è diviso il canale, parte centrale superiore del
      canale, senza grading , circa 43 piani
2898 //Il numero dei piani continua dali blocchi precedenti con la numerazione, per
      non creare problemi di sovrapposizione dei numeri; i blocchi precedenti
      avevano 54, 26 e 43 punti, piu quello centrale, quindi 1 1 di questo blocco
      è il 124 totale
2899 piano_124
2900     { type          surfaceFieldValue;
2901       libs          ("libfieldFunctionObjects.so");
2902       log            true;
2903       writeControl  writeTime;
2904       writeFields   true;
2905       surfaceFormat raw;
2906       regionType    sampledSurface;
2907       name          yNormal0;
2908       operation     areaAverage;
2909 sampledSurfaceDict
2910     { type          plane;
2911 planeType pointAndNormal;
2912 triangulate false;
2913 interpolate false;
2914 pointAndNormalDict
2915     { basePoint     (0.0501 0.10113636E+01 0.005);
2916       normalVector  (0 1 0);
2917     }
2918     fields
2919     (   Ux UPrime2Mean
2920     );   }
2921
2922 piano_125
2923     { type          surfaceFieldValue;
2924       libs          ("libfieldFunctionObjects.so");
2925       log            true;
2926       writeControl  writeTime;
2927       writeFields   true;
2928       surfaceFormat raw;
2929       regionType    sampledSurface;
2930       name          yNormal1;
2931       operation     areaAverage;
2932 sampledSurfaceDict
2933     { type          plane;
2934 planeType pointAndNormal;
2935 triangulate false;
2936 interpolate false;
2937 pointAndNormalDict
2938     { basePoint     (0.0501 0.10227273E+01 0.005);
2939       normalVector  (0 1 0);
2940     }
2941     fields
2942     (   Ux UPrime2Mean
2943     );   }
2944
2945 piano_126
2946     { type          surfaceFieldValue;
2947       libs          ("libfieldFunctionObjects.so");
2948       log            true;
2949       writeControl  writeTime;
2950       writeFields   true;
2951       surfaceFormat raw;
2952       regionType    sampledSurface;
2953       name          yNormal2;
2954       operation     areaAverage;
2955 sampledSurfaceDict
2956     { type          plane;
2957 planeType pointAndNormal;
2958 triangulate false;
2959 interpolate false;
2960 pointAndNormalDict

```

```

2961     { basePoint      (0.0501 0.10340909E+01 0.005);
2962       normalVector   (0 1 0);
2963     }
2964     fields
2965     (   Ux UPrime2Mean
2966     );   }
2967
2968 piano_127
2969     {   type           surfaceFieldValue;
2970       libs            ("libfieldFunctionObjects.so");
2971       log             true;
2972       writeControl    writeTime;
2973       writeFields     true;
2974       surfaceFormat   raw;
2975       regionType      sampledSurface;
2976       name            yNormal3;
2977       operation       areaAverage;
2978     sampledSurfaceDict
2979     {   type           plane;
2980     planeType pointAndNormal;
2981     triangulate false;
2982     interpolate  false;
2983     pointAndNormalDict
2984     {   basePoint      (0.0501 0.10454546E+01 0.005);
2985       normalVector   (0 1 0);
2986     }
2987     fields
2988     (   Ux UPrime2Mean
2989     );   }
2990
2991 piano_128
2992     {   type           surfaceFieldValue;
2993       libs            ("libfieldFunctionObjects.so");
2994       log             true;
2995       writeControl    writeTime;
2996       writeFields     true;
2997       surfaceFormat   raw;
2998       regionType      sampledSurface;
2999       name            yNormal4;
3000       operation       areaAverage;
3001     sampledSurfaceDict
3002     {   type           plane;
3003     planeType pointAndNormal;
3004     triangulate false;
3005     interpolate  false;
3006     pointAndNormalDict
3007     {   basePoint      (0.0501 0.10568182E+01 0.005);
3008       normalVector   (0 1 0);
3009     }
3010     fields
3011     (   Ux UPrime2Mean
3012     );   }
3013
3014 piano_129
3015     {   type           surfaceFieldValue;
3016       libs            ("libfieldFunctionObjects.so");
3017       log             true;
3018       writeControl    writeTime;
3019       writeFields     true;
3020       surfaceFormat   raw;
3021       regionType      sampledSurface;
3022       name            yNormal5;
3023       operation       areaAverage;
3024     sampledSurfaceDict
3025     {   type           plane;
3026     planeType pointAndNormal;
3027     triangulate false;
3028     interpolate  false;
3029     pointAndNormalDict

```

```

3030     { basePoint      (0.0501 0.10681818E+01 0.005);
3031       normalVector   (0 1 0);
3032   }
3033   fields
3034   (   Ux UPrime2Mean
3035   );   }
3036
3037 piano_130
3038   {   type           surfaceFieldValue;
3039       libs           ("libfieldFunctionObjects.so");
3040       log            true;
3041       writeControl   writeTime;
3042       writeFields    true;
3043       surfaceFormat  raw;
3044       regionType     sampledSurface;
3045       name           yNormal0;
3046       operation      areaAverage;
3047   sampledSurfaceDict
3048   {   type           plane;
3049   planeType pointAndNormal;
3050   triangulate false;
3051   interpolate  false;
3052   pointAndNormalDict
3053   {   basePoint      (0.0501 0.10795454E+01 0.005);
3054       normalVector   (0 1 0);
3055   }
3056   fields
3057   (   Ux UPrime2Mean
3058   );   }
3059
3060 piano_131
3061   {   type           surfaceFieldValue;
3062       libs           ("libfieldFunctionObjects.so");
3063       log            true;
3064       writeControl   writeTime;
3065       writeFields    true;
3066       surfaceFormat  raw;
3067       regionType     sampledSurface;
3068       name           yNormal1;
3069       operation      areaAverage;
3070   sampledSurfaceDict
3071   {   type           plane;
3072   planeType pointAndNormal;
3073   triangulate false;
3074   interpolate  false;
3075   pointAndNormalDict
3076   {   basePoint      (0.0501 0.10909091E+01 0.005);
3077       normalVector   (0 1 0);
3078   }
3079   fields
3080   (   Ux UPrime2Mean
3081   );   }
3082
3083 piano_132
3084   {   type           surfaceFieldValue;
3085       libs           ("libfieldFunctionObjects.so");
3086       log            true;
3087       writeControl   writeTime;
3088       writeFields    true;
3089       surfaceFormat  raw;
3090       regionType     sampledSurface;
3091       name           yNormal2;
3092       operation      areaAverage;
3093   sampledSurfaceDict
3094   {   type           plane;
3095   planeType pointAndNormal;
3096   triangulate false;
3097   interpolate  false;
3098   pointAndNormalDict

```

```

3099     { basePoint      (0.0501 0.11022727E+01 0.005);
3100       normalVector   (0 1 0);
3101   }
3102   fields
3103   (   Ux UPrime2Mean
3104     );   }
3105
3106 piano_133
3107   {   type           surfaceFieldValue;
3108       libs           ("libfieldFunctionObjects.so");
3109       log            true;
3110       writeControl   writeTime;
3111       writeFields    true;
3112       surfaceFormat  raw;
3113       regionType     sampledSurface;
3114       name           yNormal3;
3115       operation      areaAverage;
3116   sampledSurfaceDict
3117     {   type           plane;
3118   planeType pointAndNormal;
3119   triangulate false;
3120   interpolate  false;
3121   pointAndNormalDict
3122     {   basePoint      (0.0501 0.11136364E+01 0.005);
3123       normalVector   (0 1 0);
3124     }
3125     fields
3126     (   Ux UPrime2Mean
3127       );   }
3128
3129 piano_134
3130   {   type           surfaceFieldValue;
3131       libs           ("libfieldFunctionObjects.so");
3132       log            true;
3133       writeControl   writeTime;
3134       writeFields    true;
3135       surfaceFormat  raw;
3136       regionType     sampledSurface;
3137       name           yNormal4;
3138       operation      areaAverage;
3139   sampledSurfaceDict
3140     {   type           plane;
3141   planeType pointAndNormal;
3142   triangulate false;
3143   interpolate  false;
3144   pointAndNormalDict
3145     {   basePoint      (0.0501 0.11250000E+01 0.005);
3146       normalVector   (0 1 0);
3147     }
3148     fields
3149     (   Ux UPrime2Mean
3150       );   }
3151
3152 piano_135
3153   {   type           surfaceFieldValue;
3154       libs           ("libfieldFunctionObjects.so");
3155       log            true;
3156       writeControl   writeTime;
3157       writeFields    true;
3158       surfaceFormat  raw;
3159       regionType     sampledSurface;
3160       name           yNormal5;
3161       operation      areaAverage;
3162   sampledSurfaceDict
3163     {   type           plane;
3164   planeType pointAndNormal;
3165   triangulate false;
3166   interpolate  false;
3167   pointAndNormalDict

```

```

3168     { basePoint      (0.0501 0.11363636E+01 0.005);
3169       normalVector   (0 1 0);
3170 }
3171     fields
3172     (   Ux UPrime2Mean
3173     ); }
3174
3175 piano_136
3176 {   type           surfaceFieldValue;
3177     libs           ("libfieldFunctionObjects.so");
3178     log            true;
3179     writeControl   writeTime;
3180     writeFields   true;
3181     surfaceFormat  raw;
3182     regionType    sampledSurface;
3183     name          yNormal0;
3184     operation     areaAverage;
3185 sampledSurfaceDict
3186 {   type           plane;
3187 planeType pointAndNormal;
3188 triangulate false;
3189 interpolate  false;
3190 pointAndNormalDict
3191 {   basePoint      (0.0501 0.11477273E+01 0.005);
3192     normalVector   (0 1 0);
3193 }
3194     fields
3195     (   Ux UPrime2Mean
3196     ); }
3197
3198 piano_137
3199 {   type           surfaceFieldValue;
3200     libs           ("libfieldFunctionObjects.so");
3201     log            true;
3202     writeControl   writeTime;
3203     writeFields   true;
3204     surfaceFormat  raw;
3205     regionType    sampledSurface;
3206     name          yNormal1;
3207     operation     areaAverage;
3208 sampledSurfaceDict
3209 {   type           plane;
3210 planeType pointAndNormal;
3211 triangulate false;
3212 interpolate  false;
3213 pointAndNormalDict
3214 {   basePoint      (0.0501 0.11590909E+01 0.005);
3215     normalVector   (0 1 0);
3216 }
3217     fields
3218     (   Ux UPrime2Mean
3219     ); }
3220
3221 piano_138
3222 {   type           surfaceFieldValue;
3223     libs           ("libfieldFunctionObjects.so");
3224     log            true;
3225     writeControl   writeTime;
3226     writeFields   true;
3227     surfaceFormat  raw;
3228     regionType    sampledSurface;
3229     name          yNormal2;
3230     operation     areaAverage;
3231 sampledSurfaceDict
3232 {   type           plane;
3233 planeType pointAndNormal;
3234 triangulate false;
3235 interpolate  false;
3236 pointAndNormalDict

```

```

3237     { basePoint      (0.0501 0.11704545E+01 0.005);
3238       normalVector   (0 1 0);
3239   }
3240   fields
3241   (   Ux UPrime2Mean
3242   );   }
3243
3244   piano_139
3245   {   type           surfaceFieldValue;
3246       libs           ("libfieldFunctionObjects.so");
3247       log            true;
3248       writeControl   writeTime;
3249       writeFields    true;
3250       surfaceFormat  raw;
3251       regionType     sampledSurface;
3252       name           yNormal3;
3253       operation      areaAverage;
3254   sampledSurfaceDict
3255   {   type           plane;
3256   planeType pointAndNormal;
3257   triangulate false;
3258   interpolate  false;
3259   pointAndNormalDict
3260   {   basePoint      (0.0501 0.11818182E+01 0.005);
3261       normalVector   (0 1 0);
3262   }
3263   fields
3264   (   Ux UPrime2Mean
3265   );   }
3266
3267   piano_140
3268   {   type           surfaceFieldValue;
3269       libs           ("libfieldFunctionObjects.so");
3270       log            true;
3271       writeControl   writeTime;
3272       writeFields    true;
3273       surfaceFormat  raw;
3274       regionType     sampledSurface;
3275       name           yNormal4;
3276       operation      areaAverage;
3277   sampledSurfaceDict
3278   {   type           plane;
3279   planeType pointAndNormal;
3280   triangulate false;
3281   interpolate  false;
3282   pointAndNormalDict
3283   {   basePoint      (0.0501 0.11931818E+01 0.005);
3284       normalVector   (0 1 0);
3285   }
3286   fields
3287   (   Ux UPrime2Mean
3288   );   }
3289
3290   piano_141
3291   {   type           surfaceFieldValue;
3292       libs           ("libfieldFunctionObjects.so");
3293       log            true;
3294       writeControl   writeTime;
3295       writeFields    true;
3296       surfaceFormat  raw;
3297       regionType     sampledSurface;
3298       name           yNormal5;
3299       operation      areaAverage;
3300   sampledSurfaceDict
3301   {   type           plane;
3302   planeType pointAndNormal;
3303   triangulate false;
3304   interpolate  false;
3305   pointAndNormalDict

```

```

3306     { basePoint      (0.0501 0.12045454E+01 0.005);
3307       normalVector   (0 1 0);
3308   }
3309   fields
3310   (   Ux UPrime2Mean
3311     );   }
3312
3313 piano_142
3314   {   type           surfaceFieldValue;
3315     libs             ("libfieldFunctionObjects.so");
3316     log              true;
3317     writeControl     writeTime;
3318     writeFields      true;
3319     surfaceFormat    raw;
3320     regionType       sampledSurface;
3321     name             yNormal0;
3322     operation        areaAverage;
3323   sampledSurfaceDict
3324     {   type           plane;
3325     planeType pointAndNormal;
3326     triangulate false;
3327     interpolate false;
3328     pointAndNormalDict
3329       { basePoint      (0.0501 0.12159091E+01 0.005);
3330         normalVector   (0 1 0);
3331       }
3332     fields
3333     (   Ux UPrime2Mean
3334       );   }
3335
3336 piano_143
3337   {   type           surfaceFieldValue;
3338     libs             ("libfieldFunctionObjects.so");
3339     log              true;
3340     writeControl     writeTime;
3341     writeFields      true;
3342     surfaceFormat    raw;
3343     regionType       sampledSurface;
3344     name             yNormal1;
3345     operation        areaAverage;
3346   sampledSurfaceDict
3347     {   type           plane;
3348     planeType pointAndNormal;
3349     triangulate false;
3350     interpolate false;
3351     pointAndNormalDict
3352       { basePoint      (0.0501 0.12272727E+01 0.005);
3353         normalVector   (0 1 0);
3354       }
3355     fields
3356     (   Ux UPrime2Mean
3357       );   }
3358
3359 piano_144
3360   {   type           surfaceFieldValue;
3361     libs             ("libfieldFunctionObjects.so");
3362     log              true;
3363     writeControl     writeTime;
3364     writeFields      true;
3365     surfaceFormat    raw;
3366     regionType       sampledSurface;
3367     name             yNormal2;
3368     operation        areaAverage;
3369   sampledSurfaceDict
3370     {   type           plane;
3371     planeType pointAndNormal;
3372     triangulate false;
3373     interpolate false;
3374     pointAndNormalDict

```

```

3375     { basePoint      (0.0501 0.12386364E+01 0.005);
3376       normalVector  (0 1 0);
3377   }
3378   fields
3379   (   Ux UPrime2Mean
3380     );   }
3381
3382 piano_145
3383   {   type           surfaceFieldValue;
3384       libs           ("libfieldFunctionObjects.so");
3385       log            true;
3386       writeControl   writeTime;
3387       writeFields    true;
3388       surfaceFormat  raw;
3389       regionType     sampledSurface;
3390       name           yNormal3;
3391       operation      areaAverage;
3392   sampledSurfaceDict
3393     {   type           plane;
3394   planeType pointAndNormal;
3395   triangulate false;
3396   interpolate  false;
3397   pointAndNormalDict
3398     {   basePoint      (0.0501 0.12500000E+01 0.005);
3399       normalVector  (0 1 0);
3400   }
3401   fields
3402   (   Ux UPrime2Mean
3403     );   }
3404
3405 piano_146
3406   {   type           surfaceFieldValue;
3407       libs           ("libfieldFunctionObjects.so");
3408       log            true;
3409       writeControl   writeTime;
3410       writeFields    true;
3411       surfaceFormat  raw;
3412       regionType     sampledSurface;
3413       name           yNormal4;
3414       operation      areaAverage;
3415   sampledSurfaceDict
3416     {   type           plane;
3417   planeType pointAndNormal;
3418   triangulate false;
3419   interpolate  false;
3420   pointAndNormalDict
3421     {   basePoint      (0.0501 0.12613636E+01 0.005);
3422       normalVector  (0 1 0);
3423   }
3424   fields
3425   (   Ux UPrime2Mean
3426     );   }
3427
3428 piano_147
3429   {   type           surfaceFieldValue;
3430       libs           ("libfieldFunctionObjects.so");
3431       log            true;
3432       writeControl   writeTime;
3433       writeFields    true;
3434       surfaceFormat  raw;
3435       regionType     sampledSurface;
3436       name           yNormal5;
3437       operation      areaAverage;
3438   sampledSurfaceDict
3439     {   type           plane;
3440   planeType pointAndNormal;
3441   triangulate false;
3442   interpolate  false;
3443   pointAndNormalDict

```

```

3444     { basePoint      (0.0501 0.12727273E+01 0.005);
3445       normalVector   (0 1 0);
3446     }
3447     fields
3448     (   Ux UPrime2Mean
3449     );   }
3450
3451 piano_148
3452     {   type           surfaceFieldValue;
3453       libs             ("libfieldFunctionObjects.so");
3454       log              true;
3455       writeControl     writeTime;
3456       writeFields     true;
3457       surfaceFormat    raw;
3458       regionType      sampledSurface;
3459       name             yNormal0;
3460       operation        areaAverage;
3461 sampledSurfaceDict
3462     {   type           plane;
3463 planeType pointAndNormal;
3464 triangulate false;
3465 interpolate false;
3466   pointAndNormalDict
3467     {   basePoint      (0.0501 0.12840909E+01 0.005);
3468       normalVector   (0 1 0);
3469     }
3470     fields
3471     (   Ux UPrime2Mean
3472     );   }
3473
3474 piano_149
3475     {   type           surfaceFieldValue;
3476       libs             ("libfieldFunctionObjects.so");
3477       log              true;
3478       writeControl     writeTime;
3479       writeFields     true;
3480       surfaceFormat    raw;
3481       regionType      sampledSurface;
3482       name             yNormal1;
3483       operation        areaAverage;
3484 sampledSurfaceDict
3485     {   type           plane;
3486 planeType pointAndNormal;
3487 triangulate false;
3488 interpolate false;
3489   pointAndNormalDict
3490     {   basePoint      (0.0501 0.12954545E+01 0.005);
3491       normalVector   (0 1 0);
3492     }
3493     fields
3494     (   Ux UPrime2Mean
3495     );   }
3496
3497 piano_150
3498     {   type           surfaceFieldValue;
3499       libs             ("libfieldFunctionObjects.so");
3500       log              true;
3501       writeControl     writeTime;
3502       writeFields     true;
3503       surfaceFormat    raw;
3504       regionType      sampledSurface;
3505       name             yNormal2;
3506       operation        areaAverage;
3507 sampledSurfaceDict
3508     {   type           plane;
3509 planeType pointAndNormal;
3510 triangulate false;
3511 interpolate false;
3512   pointAndNormalDict

```

```

3513     { basePoint      (0.0501 0.13068182E+01 0.005);
3514       normalVector   (0 1 0);
3515   }
3516   fields
3517   (   Ux UPrime2Mean
3518     );   }
3519
3520 piano_151
3521   {   type           surfaceFieldValue;
3522     libs             ("libfieldFunctionObjects.so");
3523     log              true;
3524     writeControl     writeTime;
3525     writeFields      true;
3526     surfaceFormat    raw;
3527     regionType       sampledSurface;
3528     name             yNormal3;
3529     operation        areaAverage;
3530 sampledSurfaceDict
3531   {   type           plane;
3532     planeType        pointAndNormal;
3533     triangulate      false;
3534     interpolate      false;
3535     pointAndNormalDict
3536   {   basePoint      (0.0501 0.13181818E+01 0.005);
3537     normalVector     (0 1 0);
3538   }
3539   fields
3540   (   Ux UPrime2Mean
3541     );   }
3542
3543 piano_152
3544   {   type           surfaceFieldValue;
3545     libs             ("libfieldFunctionObjects.so");
3546     log              true;
3547     writeControl     writeTime;
3548     writeFields      true;
3549     surfaceFormat    raw;
3550     regionType       sampledSurface;
3551     name             yNormal4;
3552     operation        areaAverage;
3553 sampledSurfaceDict
3554   {   type           plane;
3555     planeType        pointAndNormal;
3556     triangulate      false;
3557     interpolate      false;
3558     pointAndNormalDict
3559   {   basePoint      (0.0501 0.13295454E+01 0.005);
3560     normalVector     (0 1 0);
3561   }
3562   fields
3563   (   Ux UPrime2Mean
3564     );   }
3565
3566 piano_153
3567   {   type           surfaceFieldValue;
3568     libs             ("libfieldFunctionObjects.so");
3569     log              true;
3570     writeControl     writeTime;
3571     writeFields      true;
3572     surfaceFormat    raw;
3573     regionType       sampledSurface;
3574     name             yNormal5;
3575     operation        areaAverage;
3576 sampledSurfaceDict
3577   {   type           plane;
3578     planeType        pointAndNormal;
3579     triangulate      false;
3580     interpolate      false;
3581     pointAndNormalDict

```

```

3582     { basePoint      (0.0501 0.13409091E+01 0.005);
3583       normalVector   (0 1 0);
3584   }
3585   fields
3586   (   Ux UPrime2Mean
3587   );   }
3588
3589 piano_154
3590   {   type           surfaceFieldValue;
3591       libs           ("libfieldFunctionObjects.so");
3592       log            true;
3593       writeControl   writeTime;
3594       writeFields    true;
3595       surfaceFormat  raw;
3596       regionType    sampledSurface;
3597       name           yNormal0;
3598       operation      areaAverage;
3599   sampledSurfaceDict
3600   {   type           plane;
3601   planeType pointAndNormal;
3602   triangulate false;
3603   interpolate  false;
3604   pointAndNormalDict
3605   {   basePoint      (0.0501 0.13522727E+01 0.005);
3606       normalVector   (0 1 0);
3607   }
3608   fields
3609   (   Ux UPrime2Mean
3610   );   }
3611
3612 piano_155
3613   {   type           surfaceFieldValue;
3614       libs           ("libfieldFunctionObjects.so");
3615       log            true;
3616       writeControl   writeTime;
3617       writeFields    true;
3618       surfaceFormat  raw;
3619       regionType    sampledSurface;
3620       name           yNormal1;
3621       operation      areaAverage;
3622   sampledSurfaceDict
3623   {   type           plane;
3624   planeType pointAndNormal;
3625   triangulate false;
3626   interpolate  false;
3627   pointAndNormalDict
3628   {   basePoint      (0.0501 0.13636364E+01 0.005);
3629       normalVector   (0 1 0);
3630   }
3631   fields
3632   (   Ux UPrime2Mean
3633   );   }
3634
3635 piano_156
3636   {   type           surfaceFieldValue;
3637       libs           ("libfieldFunctionObjects.so");
3638       log            true;
3639       writeControl   writeTime;
3640       writeFields    true;
3641       surfaceFormat  raw;
3642       regionType    sampledSurface;
3643       name           yNormal2;
3644       operation      areaAverage;
3645   sampledSurfaceDict
3646   {   type           plane;
3647   planeType pointAndNormal;
3648   triangulate false;
3649   interpolate  false;
3650   pointAndNormalDict

```

```

3651     { basePoint      (0.0501 0.13750000E+01 0.005);
3652       normalVector   (0 1 0);
3653   }
3654   fields
3655   (   Ux UPrime2Mean
3656     );   }
3657
3658 piano_157
3659   {   type           surfaceFieldValue;
3660     libs             ("libfieldFunctionObjects.so");
3661     log              true;
3662     writeControl     writeTime;
3663     writeFields      true;
3664     surfaceFormat    raw;
3665     regionType       sampledSurface;
3666     name             yNormal3;
3667     operation        areaAverage;
3668   sampledSurfaceDict
3669     {   type           plane;
3670   planeType pointAndNormal;
3671   triangulate false;
3672   interpolate  false;
3673   pointAndNormalDict
3674     {   basePoint      (0.0501 0.13863636E+01 0.005);
3675       normalVector   (0 1 0);
3676     }
3677     fields
3678     (   Ux UPrime2Mean
3679       );   }
3680
3681 piano_158
3682   {   type           surfaceFieldValue;
3683     libs             ("libfieldFunctionObjects.so");
3684     log              true;
3685     writeControl     writeTime;
3686     writeFields      true;
3687     surfaceFormat    raw;
3688     regionType       sampledSurface;
3689     name             yNormal4;
3690     operation        areaAverage;
3691   sampledSurfaceDict
3692     {   type           plane;
3693   planeType pointAndNormal;
3694   triangulate false;
3695   interpolate  false;
3696   pointAndNormalDict
3697     {   basePoint      (0.0501 0.13977273E+01 0.005);
3698       normalVector   (0 1 0);
3699     }
3700     fields
3701     (   Ux UPrime2Mean
3702       );   }
3703
3704 piano_159
3705   {   type           surfaceFieldValue;
3706     libs             ("libfieldFunctionObjects.so");
3707     log              true;
3708     writeControl     writeTime;
3709     writeFields      true;
3710     surfaceFormat    raw;
3711     regionType       sampledSurface;
3712     name             yNormal5;
3713     operation        areaAverage;
3714   sampledSurfaceDict
3715     {   type           plane;
3716   planeType pointAndNormal;
3717   triangulate false;
3718   interpolate  false;
3719   pointAndNormalDict

```

```

3720     { basePoint      (0.0501 0.14090909E+01 0.005);
3721       normalVector   (0 1 0);
3722   }
3723   fields
3724   (   Ux UPrime2Mean
3725     );   }
3726
3727   piano_160
3728   {   type           surfaceFieldValue;
3729     libs             ("libfieldFunctionObjects.so");
3730     log              true;
3731     writeControl     writeTime;
3732     writeFields      true;
3733     surfaceFormat    raw;
3734     regionType       sampledSurface;
3735     name             yNormal0;
3736     operation        areaAverage;
3737   sampledSurfaceDict
3738   {   type           plane;
3739     planeType        pointAndNormal;
3740     triangulate      false;
3741     interpolate       false;
3742     pointAndNormalDict
3743     {   basePoint      (0.0501 0.14204545E+01 0.005);
3744       normalVector   (0 1 0);
3745     }
3746     fields
3747     (   Ux UPrime2Mean
3748       );   }
3749
3750   piano_161
3751   {   type           surfaceFieldValue;
3752     libs             ("libfieldFunctionObjects.so");
3753     log              true;
3754     writeControl     writeTime;
3755     writeFields      true;
3756     surfaceFormat    raw;
3757     regionType       sampledSurface;
3758     name             yNormal1;
3759     operation        areaAverage;
3760   sampledSurfaceDict
3761   {   type           plane;
3762     planeType        pointAndNormal;
3763     triangulate      false;
3764     interpolate       false;
3765     pointAndNormalDict
3766     {   basePoint      (0.0501 0.14318182E+01 0.005);
3767       normalVector   (0 1 0);
3768     }
3769     fields
3770     (   Ux UPrime2Mean
3771       );   }
3772
3773   piano_162
3774   {   type           surfaceFieldValue;
3775     libs             ("libfieldFunctionObjects.so");
3776     log              true;
3777     writeControl     writeTime;
3778     writeFields      true;
3779     surfaceFormat    raw;
3780     regionType       sampledSurface;
3781     name             yNormal2;
3782     operation        areaAverage;
3783   sampledSurfaceDict
3784   {   type           plane;
3785     planeType        pointAndNormal;
3786     triangulate      false;
3787     interpolate       false;
3788     pointAndNormalDict

```

```

3789     { basePoint      (0.0501 0.14431818E+01 0.005);
3790       normalVector   (0 1 0);
3791   }
3792   fields
3793   (   Ux UPrime2Mean
3794     );   }
3795
3796 piano_163
3797   {   type           surfaceFieldValue;
3798       libs           ("libfieldFunctionObjects.so");
3799       log            true;
3800       writeControl   writeTime;
3801       writeFields    true;
3802       surfaceFormat  raw;
3803       regionType     sampledSurface;
3804       name           yNormal3;
3805       operation      areaAverage;
3806   sampledSurfaceDict
3807     {   type           plane;
3808   planeType pointAndNormal;
3809   triangulate false;
3810   interpolate  false;
3811   pointAndNormalDict
3812     {   basePoint      (0.0501 0.14545455E+01 0.005);
3813       normalVector   (0 1 0);
3814     }
3815     fields
3816     (   Ux UPrime2Mean
3817       );   }
3818
3819 piano_164
3820   {   type           surfaceFieldValue;
3821       libs           ("libfieldFunctionObjects.so");
3822       log            true;
3823       writeControl   writeTime;
3824       writeFields    true;
3825       surfaceFormat  raw;
3826       regionType     sampledSurface;
3827       name           yNormal4;
3828       operation      areaAverage;
3829   sampledSurfaceDict
3830     {   type           plane;
3831   planeType pointAndNormal;
3832   triangulate false;
3833   interpolate  false;
3834   pointAndNormalDict
3835     {   basePoint      (0.0501 0.14659091E+01 0.005);
3836       normalVector   (0 1 0);
3837     }
3838     fields
3839     (   Ux UPrime2Mean
3840       );   }
3841
3842 piano_165
3843   {   type           surfaceFieldValue;
3844       libs           ("libfieldFunctionObjects.so");
3845       log            true;
3846       writeControl   writeTime;
3847       writeFields    true;
3848       surfaceFormat  raw;
3849       regionType     sampledSurface;
3850       name           yNormal5;
3851       operation      areaAverage;
3852   sampledSurfaceDict
3853     {   type           plane;
3854   planeType pointAndNormal;
3855   triangulate false;
3856   interpolate  false;
3857   pointAndNormalDict

```

```

3858     { basePoint      (0.0501 0.14772727E+01 0.005);
3859       normalVector  (0 1 0);
3860 }
3861     fields
3862     (   Ux UPrime2Mean
3863     ); }
3864
3865 piano_166
3866     { type            surfaceFieldValue;
3867       libs           ("libfieldFunctionObjects.so");
3868       log            true;
3869       writeControl   writeTime;
3870       writeFields    true;
3871       surfaceFormat  raw;
3872       regionType     sampledSurface;
3873       name           yNormal0;
3874       operation      areaAverage;
3875 sampledSurfaceDict
3876     { type            plane;
3877 planeType pointAndNormal;
3878 triangulate false;
3879 interpolate false;
3880 pointAndNormalDict
3881     { basePoint      (0.0501 0.15000000E+01 0.005);
3882       normalVector  (0 1 0);
3883 }
3884     fields
3885     (   Ux UPrime2Mean
3886     ); }
3887 //Piani del V blocco in cui è diviso il canale, parte intermedia superiore, con
3888 //Il numero dei piani continua dai blocchi precedenti con la numerazione, per
3889 //non creare problemi di sovrapposizione dei numeri; i blocchi precedenti
3890 //avevano 54, 26, 43 e 43 punti, piu quello centrale, quindi 1 1 di questo
3891 //blocco è il 167 totale
3892 piano_167
3893     { type            surfaceFieldValue;
3894       libs           ("libfieldFunctionObjects.so");
3895       log            true;
3896       writeControl   writeTime;
3897       writeFields    true;
3898       surfaceFormat  raw;
3899       regionType     sampledSurface;
3900       name           yNormal1;
3901       operation      areaAverage;
3902 sampledSurfaceDict
3903     { type            plane;
3904 planeType pointAndNormal;
3905 triangulate false;
3906 interpolate false;
3907 pointAndNormalDict
3908     { basePoint      (0.0501 0.15112570E+01 0.005);
3909       normalVector  (0 1 0);
3910 }
3911     fields
3912     (   Ux UPrime2Mean
3913     ); }
3914
3915 piano_168
3916     { type            surfaceFieldValue;
3917       libs           ("libfieldFunctionObjects.so");
3918       log            true;
3919       writeControl   writeTime;
3920       writeFields    true;
3921       surfaceFormat  raw;
3922       regionType     sampledSurface;
3923       name           yNormal2;
3924       operation      areaAverage;
3925 sampledSurfaceDict

```

```

3923     {   type           plane;
3924 planeType pointAndNormal;
3925 triangulate false;
3926 interpolate  false;
3927   pointAndNormalDict
3928     {   basePoint      (0.0501 0.15223398E+01 0.005);
3929         normalVector   (0 1 0);
3930   }
3931     fields
3932     (   Ux UPrime2Mean
3933       );   }
3934
3935 piano_169
3936   {   type           surfaceFieldValue;
3937       libs           ("libfieldFunctionObjects.so");
3938       log            true;
3939       writeControl   writeTime;
3940       writeFields    true;
3941       surfaceFormat  raw;
3942       regionType     sampledSurface;
3943       name           yNormal3;
3944       operation      areaAverage;
3945   sampledSurfaceDict
3946     {   type           plane;
3947 planeType pointAndNormal;
3948 triangulate false;
3949 interpolate  false;
3950   pointAndNormalDict
3951     {   basePoint      (0.0501 0.15332511E+01 0.005);
3952         normalVector   (0 1 0);
3953   }
3954     fields
3955     (   Ux UPrime2Mean
3956       );   }
3957
3958 piano_170
3959   {   type           surfaceFieldValue;
3960       libs           ("libfieldFunctionObjects.so");
3961       log            true;
3962       writeControl   writeTime;
3963       writeFields    true;
3964       surfaceFormat  raw;
3965       regionType     sampledSurface;
3966       name           yNormal4;
3967       operation      areaAverage;
3968   sampledSurfaceDict
3969     {   type           plane;
3970 planeType pointAndNormal;
3971 triangulate false;
3972 interpolate  false;
3973   pointAndNormalDict
3974     {   basePoint      (0.0501 0.15439936E+01 0.005);
3975         normalVector   (0 1 0);
3976   }
3977     fields
3978     (   Ux UPrime2Mean
3979       );   }
3980
3981 piano_171
3982   {   type           surfaceFieldValue;
3983       libs           ("libfieldFunctionObjects.so");
3984       log            true;
3985       writeControl   writeTime;
3986       writeFields    true;
3987       surfaceFormat  raw;
3988       regionType     sampledSurface;
3989       name           yNormal5;
3990       operation      areaAverage;
3991   sampledSurfaceDict

```

```

3992     {   type           plane;
3993 planeType pointAndNormal;
3994 triangulate false;
3995 interpolate  false;
3996   pointAndNormalDict
3997     {   basePoint      (0.0501 0.15545699E+01 0.005);
3998       normalVector    (0 1 0);
3999   }
4000   fields
4001   (   Ux UPrime2Mean
4002     );   }
4003
4004 piano_172
4005   {   type           surfaceFieldValue;
4006     libs             ("libfieldFunctionObjects.so");
4007     log              true;
4008     writeControl     writeTime;
4009     writeFields      true;
4010     surfaceFormat    raw;
4011     regionType       sampledSurface;
4012     name             yNormal0;
4013     operation        areaAverage;
4014   sampledSurfaceDict
4015     {   type           plane;
4016   planeType pointAndNormal;
4017   triangulate false;
4018   interpolate  false;
4019   pointAndNormalDict
4020     {   basePoint      (0.0501 0.15649825E+01 0.005);
4021       normalVector    (0 1 0);
4022   }
4023   fields
4024   (   Ux UPrime2Mean
4025     );   }
4026
4027 piano_173
4028   {   type           surfaceFieldValue;
4029     libs             ("libfieldFunctionObjects.so");
4030     log              true;
4031     writeControl     writeTime;
4032     writeFields      true;
4033     surfaceFormat    raw;
4034     regionType       sampledSurface;
4035     name             yNormal1;
4036     operation        areaAverage;
4037   sampledSurfaceDict
4038     {   type           plane;
4039   planeType pointAndNormal;
4040   triangulate false;
4041   interpolate  false;
4042   pointAndNormalDict
4043     {   basePoint      (0.0501 0.15752339E+01 0.005);
4044       normalVector    (0 1 0);
4045   }
4046   fields
4047   (   Ux UPrime2Mean
4048     );   }
4049
4050 piano_174
4051   {   type           surfaceFieldValue;
4052     libs             ("libfieldFunctionObjects.so");
4053     log              true;
4054     writeControl     writeTime;
4055     writeFields      true;
4056     surfaceFormat    raw;
4057     regionType       sampledSurface;
4058     name             yNormal2;
4059     operation        areaAverage;
4060   sampledSurfaceDict

```

```

4061     {   type           plane;
4062 planeType pointAndNormal;
4063 triangulate false;
4064 interpolate  false;
4065   pointAndNormalDict
4066     {   basePoint      (0.0501 0.15853268E+01 0.005);
4067       normalVector    (0 1 0);
4068   }
4069   fields
4070   (   Ux UPrime2Mean
4071     );   }
4072
4073 piano_175
4074   {   type           surfaceFieldValue;
4075     libs             ("libfieldFunctionObjects.so");
4076     log              true;
4077     writeControl     writeTime;
4078     writeFields      true;
4079     surfaceFormat    raw;
4080     regionType       sampledSurface;
4081     name             yNormal3;
4082     operation        areaAverage;
4083   sampledSurfaceDict
4084     {   type           plane;
4085 planeType pointAndNormal;
4086 triangulate false;
4087 interpolate  false;
4088   pointAndNormalDict
4089     {   basePoint      (0.0501 0.15952635E+01 0.005);
4090       normalVector    (0 1 0);
4091   }
4092   fields
4093   (   Ux UPrime2Mean
4094     );   }
4095
4096 piano_176
4097   {   type           surfaceFieldValue;
4098     libs             ("libfieldFunctionObjects.so");
4099     log              true;
4100     writeControl     writeTime;
4101     writeFields      true;
4102     surfaceFormat    raw;
4103     regionType       sampledSurface;
4104     name             yNormal4;
4105     operation        areaAverage;
4106   sampledSurfaceDict
4107     {   type           plane;
4108 planeType pointAndNormal;
4109 triangulate false;
4110 interpolate  false;
4111   pointAndNormalDict
4112     {   basePoint      (0.0501 0.16050464E+01 0.005);
4113       normalVector    (0 1 0);
4114   }
4115   fields
4116   (   Ux UPrime2Mean
4117     );   }
4118
4119 piano_177
4120   {   type           surfaceFieldValue;
4121     libs             ("libfieldFunctionObjects.so");
4122     log              true;
4123     writeControl     writeTime;
4124     writeFields      true;
4125     surfaceFormat    raw;
4126     regionType       sampledSurface;
4127     name             yNormal5;
4128     operation        areaAverage;
4129   sampledSurfaceDict

```

```

4130     {   type           plane;
4131 planeType pointAndNormal;
4132 triangulate false;
4133 interpolate  false;
4134   pointAndNormalDict
4135     {   basePoint      (0.0501 0.16146779E+01 0.005);
4136         normalVector   (0 1 0);
4137   }
4138     fields
4139     (   Ux UPrime2Mean
4140     );   }
4141
4142 piano_178
4143   {   type           surfaceFieldValue;
4144       libs           ("libfieldFunctionObjects.so");
4145       log             true;
4146       writeControl   writeTime;
4147       writeFields    true;
4148       surfaceFormat  raw;
4149       regionType     sampledSurface;
4150       name           yNormal0;
4151       operation      areaAverage;
4152 sampledSurfaceDict
4153   {   type           plane;
4154 planeType pointAndNormal;
4155 triangulate false;
4156 interpolate  false;
4157   pointAndNormalDict
4158     {   basePoint      (0.0501 0.16241604E+01 0.005);
4159         normalVector   (0 1 0);
4160   }
4161     fields
4162     (   Ux UPrime2Mean
4163     );   }
4164
4165 piano_179
4166   {   type           surfaceFieldValue;
4167       libs           ("libfieldFunctionObjects.so");
4168       log             true;
4169       writeControl   writeTime;
4170       writeFields    true;
4171       surfaceFormat  raw;
4172       regionType     sampledSurface;
4173       name           yNormal1;
4174       operation      areaAverage;
4175 sampledSurfaceDict
4176   {   type           plane;
4177 planeType pointAndNormal;
4178 triangulate false;
4179 interpolate  false;
4180   pointAndNormalDict
4181     {   basePoint      (0.0501 0.16334962E+01 0.005);
4182         normalVector   (0 1 0);
4183   }
4184     fields
4185     (   Ux UPrime2Mean
4186     );   }
4187
4188 piano_180
4189   {   type           surfaceFieldValue;
4190       libs           ("libfieldFunctionObjects.so");
4191       log             true;
4192       writeControl   writeTime;
4193       writeFields    true;
4194       surfaceFormat  raw;
4195       regionType     sampledSurface;
4196       name           yNormal2;
4197       operation      areaAverage;
4198 sampledSurfaceDict

```

```

4199     {   type           plane;
4200 planeType pointAndNormal;
4201 triangulate false;
4202 interpolate  false;
4203   pointAndNormalDict
4204     {   basePoint      (0.0501 0.16426875E+01 0.005);
4205         normalVector   (0 1 0);
4206   }
4207     fields
4208     (   Ux UPrime2Mean
4209     );   }
4210
4211 piano_181
4212   {   type           surfaceFieldValue;
4213       libs           ("libfieldFunctionObjects.so");
4214       log             true;
4215       writeControl   writeTime;
4216       writeFields    true;
4217       surfaceFormat  raw;
4218       regionType     sampledSurface;
4219       name           yNormal3;
4220       operation      areaAverage;
4221 sampledSurfaceDict
4222   {   type           plane;
4223 planeType pointAndNormal;
4224 triangulate false;
4225 interpolate  false;
4226   pointAndNormalDict
4227     {   basePoint      (0.0501 0.16517366E+01 0.005);
4228         normalVector   (0 1 0);
4229   }
4230     fields
4231     (   Ux UPrime2Mean
4232     );   }
4233
4234 piano_182
4235   {   type           surfaceFieldValue;
4236       libs           ("libfieldFunctionObjects.so");
4237       log             true;
4238       writeControl   writeTime;
4239       writeFields    true;
4240       surfaceFormat  raw;
4241       regionType     sampledSurface;
4242       name           yNormal4;
4243       operation      areaAverage;
4244 sampledSurfaceDict
4245   {   type           plane;
4246 planeType pointAndNormal;
4247 triangulate false;
4248 interpolate  false;
4249   pointAndNormalDict
4250     {   basePoint      (0.0501 0.16606456E+01 0.005);
4251         normalVector   (0 1 0);
4252   }
4253     fields
4254     (   Ux UPrime2Mean
4255     );   }
4256
4257 piano_183
4258   {   type           surfaceFieldValue;
4259       libs           ("libfieldFunctionObjects.so");
4260       log             true;
4261       writeControl   writeTime;
4262       writeFields    true;
4263       surfaceFormat  raw;
4264       regionType     sampledSurface;
4265       name           yNormal5;
4266       operation      areaAverage;
4267 sampledSurfaceDict

```

```

4268     {   type           plane;
4269 planeType pointAndNormal;
4270 triangulate false;
4271 interpolate  false;
4272   pointAndNormalDict
4273     {   basePoint      (0.0501 0.16694168E+01 0.005);
4274         normalVector   (0 1 0);
4275   }
4276     fields
4277     (   Ux UPrime2Mean
4278     );   }
4279
4280 piano_184
4281   {   type           surfaceFieldValue;
4282       libs           ("libfieldFunctionObjects.so");
4283       log             true;
4284       writeControl   writeTime;
4285       writeFields    true;
4286       surfaceFormat  raw;
4287       regionType     sampledSurface;
4288       name           yNormal0;
4289       operation      areaAverage;
4290 sampledSurfaceDict
4291   {   type           plane;
4292 planeType pointAndNormal;
4293 triangulate false;
4294 interpolate  false;
4295   pointAndNormalDict
4296     {   basePoint      (0.0501 0.16780523E+01 0.005);
4297         normalVector   (0 1 0);
4298   }
4299     fields
4300     (   Ux UPrime2Mean
4301     );   }
4302
4303 piano_185
4304   {   type           surfaceFieldValue;
4305       libs           ("libfieldFunctionObjects.so");
4306       log             true;
4307       writeControl   writeTime;
4308       writeFields    true;
4309       surfaceFormat  raw;
4310       regionType     sampledSurface;
4311       name           yNormal1;
4312       operation      areaAverage;
4313 sampledSurfaceDict
4314   {   type           plane;
4315 planeType pointAndNormal;
4316 triangulate false;
4317 interpolate  false;
4318   pointAndNormalDict
4319     {   basePoint      (0.0501 0.16865541E+01 0.005);
4320         normalVector   (0 1 0);
4321   }
4322     fields
4323     (   Ux UPrime2Mean
4324     );   }
4325
4326 piano_186
4327   {   type           surfaceFieldValue;
4328       libs           ("libfieldFunctionObjects.so");
4329       log             true;
4330       writeControl   writeTime;
4331       writeFields    true;
4332       surfaceFormat  raw;
4333       regionType     sampledSurface;
4334       name           yNormal2;
4335       operation      areaAverage;
4336 sampledSurfaceDict

```

```

4337     {   type           plane;
4338 planeType pointAndNormal;
4339 triangulate false;
4340 interpolate  false;
4341   pointAndNormalDict
4342     {   basePoint      (0.0501 0.16949244E+01 0.005);
4343       normalVector    (0 1 0);
4344   }
4345   fields
4346   (   Ux UPrime2Mean
4347   );   }
4348
4349 piano_187
4350   {   type           surfaceFieldValue;
4351     libs             ("libfieldFunctionObjects.so");
4352     log              true;
4353     writeControl     writeTime;
4354     writeFields      true;
4355     surfaceFormat    raw;
4356     regionType       sampledSurface;
4357     name             yNormal3;
4358     operation        areaAverage;
4359 sampledSurfaceDict
4360   {   type           plane;
4361 planeType pointAndNormal;
4362 triangulate false;
4363 interpolate  false;
4364   pointAndNormalDict
4365     {   basePoint      (0.0501 0.17031652E+01 0.005);
4366       normalVector    (0 1 0);
4367   }
4368   fields
4369   (   Ux UPrime2Mean
4370   );   }
4371
4372 piano_188
4373   {   type           surfaceFieldValue;
4374     libs             ("libfieldFunctionObjects.so");
4375     log              true;
4376     writeControl     writeTime;
4377     writeFields      true;
4378     surfaceFormat    raw;
4379     regionType       sampledSurface;
4380     name             yNormal4;
4381     operation        areaAverage;
4382 sampledSurfaceDict
4383   {   type           plane;
4384 planeType pointAndNormal;
4385 triangulate false;
4386 interpolate  false;
4387   pointAndNormalDict
4388     {   basePoint      (0.0501 0.17112785E+01 0.005);
4389       normalVector    (0 1 0);
4390   }
4391   fields
4392   (   Ux UPrime2Mean
4393   );   }
4394
4395 piano_189
4396   {   type           surfaceFieldValue;
4397     libs             ("libfieldFunctionObjects.so");
4398     log              true;
4399     writeControl     writeTime;
4400     writeFields      true;
4401     surfaceFormat    raw;
4402     regionType       sampledSurface;
4403     name             yNormal5;
4404     operation        areaAverage;
4405 sampledSurfaceDict

```

```

4406     {   type           plane;
4407 planeType pointAndNormal;
4408 triangulate false;
4409 interpolate  false;
4410   pointAndNormalDict
4411     {   basePoint      (0.0501 0.17192662E+01 0.005);
4412       normalVector    (0 1 0);
4413   }
4414   fields
4415   (   Ux UPrime2Mean
4416     );   }
4417
4418 piano_190
4419   {   type           surfaceFieldValue;
4420     libs             ("libfieldFunctionObjects.so");
4421     log              true;
4422     writeControl     writeTime;
4423     writeFields      true;
4424     surfaceFormat    raw;
4425     regionType       sampledSurface;
4426     name             yNormal0;
4427     operation        areaAverage;
4428   sampledSurfaceDict
4429     {   type           plane;
4430 planeType pointAndNormal;
4431 triangulate false;
4432 interpolate  false;
4433   pointAndNormalDict
4434     {   basePoint      (0.0501 0.17271303E+01 0.005);
4435       normalVector    (0 1 0);
4436   }
4437   fields
4438   (   Ux UPrime2Mean
4439     );   }
4440
4441 piano_191
4442   {   type           surfaceFieldValue;
4443     libs             ("libfieldFunctionObjects.so");
4444     log              true;
4445     writeControl     writeTime;
4446     writeFields      true;
4447     surfaceFormat    raw;
4448     regionType       sampledSurface;
4449     name             yNormal1;
4450     operation        areaAverage;
4451   sampledSurfaceDict
4452     {   type           plane;
4453 planeType pointAndNormal;
4454 triangulate false;
4455 interpolate  false;
4456   pointAndNormalDict
4457     {   basePoint      (0.0501 0.17348727E+01 0.005);
4458       normalVector    (0 1 0);
4459   }
4460   fields
4461   (   Ux UPrime2Mean
4462     );   }
4463
4464 piano_192
4465   {   type           surfaceFieldValue;
4466     libs             ("libfieldFunctionObjects.so");
4467     log              true;
4468     writeControl     writeTime;
4469     writeFields      true;
4470     surfaceFormat    raw;
4471     regionType       sampledSurface;
4472     name             yNormal2;
4473     operation        areaAverage;
4474   sampledSurfaceDict

```

```

4475     {   type           plane;
4476 planeType pointAndNormal;
4477 triangulate false;
4478 interpolate  false;
4479   pointAndNormalDict
4480     {   basePoint      (0.0501 0.17500000E+01 0.005);
4481       normalVector    (0 1 0);
4482   }
4483   fields
4484   (   Ux UPrime2Mean
4485     );   }
4486 //Piani del VI blocco in cui è diviso il canale, parte piu fitta superiore del
         canale, senza grading, con grading 3.0, circa 54 piani
4487 //Il numero dei piani continua dal blocco precedente con la numerazione, per non
         creare problemi di sovrapposizione dei numeri; i blocchi precedente avevano
         54, 26, 43, 43 e 26 punti, piu quello centrale, quindi 1 1 di questo
         blocco è il 193 totale
4488 piano_193
4489   {   type           surfaceFieldValue;
4490     libs             ("libfieldFunctionObjects.so");
4491     log              true;
4492     writeControl     writeTime;
4493     writeFields      true;
4494     surfaceFormat    raw;
4495     regionType       sampledSurface;
4496     name             yNormal3;
4497     operation        areaAverage;
4498 sampledSurfaceDict
4499   {   type           plane;
4500 planeType pointAndNormal;
4501 triangulate false;
4502 interpolate  false;
4503   pointAndNormalDict
4504     {   basePoint      (0.0501 0.17574769E+01 0.005);
4505       normalVector    (0 1 0);
4506   }
4507   fields
4508   (   Ux UPrime2Mean
4509     );   }
4510
4511 piano_194
4512   {   type           surfaceFieldValue;
4513     libs             ("libfieldFunctionObjects.so");
4514     log              true;
4515     writeControl     writeTime;
4516     writeFields      true;
4517     surfaceFormat    raw;
4518     regionType       sampledSurface;
4519     name             yNormal4;
4520     operation        areaAverage;
4521 sampledSurfaceDict
4522   {   type           plane;
4523 planeType pointAndNormal;
4524 triangulate false;
4525 interpolate  false;
4526   pointAndNormalDict
4527     {   basePoint      (0.0501 0.17648032E+01 0.005);
4528       normalVector    (0 1 0);
4529   }
4530   fields
4531   (   Ux UPrime2Mean
4532     );   }
4533
4534 piano_195
4535   {   type           surfaceFieldValue;
4536     libs             ("libfieldFunctionObjects.so");
4537     log              true;
4538     writeControl     writeTime;
4539     writeFields      true;

```

```

4540     surfaceFormat    raw;
4541     regionType       sampledSurface;
4542     name              yNormal5;
4543     operation         areaAverage;
4544 sampledSurfaceDict
4545     { type            plane;
4546 planeType pointAndNormal;
4547 triangulate false;
4548 interpolate false;
4549     pointAndNormalDict
4550     { basePoint      (0.0501 0.17719819E+01 0.005);
4551       normalVector   (0 1 0);
4552     }
4553     fields
4554     ( Ux UPrime2Mean
4555       ); }
4556
4557 piano_196
4558     { type            surfaceFieldValue;
4559       libs            ("libfieldFunctionObjects.so");
4560       log              true;
4561       writeControl     writeTime;
4562       writeFields      true;
4563       surfaceFormat    raw;
4564       regionType       sampledSurface;
4565       name              yNormal0;
4566       operation         areaAverage;
4567 sampledSurfaceDict
4568     { type            plane;
4569 planeType pointAndNormal;
4570 triangulate false;
4571 interpolate false;
4572     pointAndNormalDict
4573     { basePoint      (0.0501 0.17790161E+01 0.005);
4574       normalVector   (0 1 0);
4575     }
4576     fields
4577     ( Ux UPrime2Mean
4578       ); }
4579
4580 piano_197
4581     { type            surfaceFieldValue;
4582       libs            ("libfieldFunctionObjects.so");
4583       log              true;
4584       writeControl     writeTime;
4585       writeFields      true;
4586       surfaceFormat    raw;
4587       regionType       sampledSurface;
4588       name              yNormal1;
4589       operation         areaAverage;
4590 sampledSurfaceDict
4591     { type            plane;
4592 planeType pointAndNormal;
4593 triangulate false;
4594 interpolate false;
4595     pointAndNormalDict
4596     { basePoint      (0.0501 0.17859086E+01 0.005);
4597       normalVector   (0 1 0);
4598     }
4599     fields
4600     ( Ux UPrime2Mean
4601       ); }
4602
4603 piano_198
4604     { type            surfaceFieldValue;
4605       libs            ("libfieldFunctionObjects.so");
4606       log              true;
4607       writeControl     writeTime;
4608       writeFields      true;

```

```

4609     surfaceFormat    raw;
4610     regionType       sampledSurface;
4611     name              yNormal2;
4612     operation        areaAverage;
4613 sampledSurfaceDict
4614     { type            plane;
4615 planeType pointAndNormal;
4616 triangulate false;
4617 interpolate false;
4618 pointAndNormalDict
4619     { basePoint      (0.0501 0.17926623E+01 0.005);
4620     normalVector    (0 1 0);
4621     }
4622     fields
4623     ( Ux UPrime2Mean
4624     ); }
4625
4626 piano_199
4627     { type            surfaceFieldValue;
4628     libs              ("libfieldFunctionObjects.so");
4629     log               true;
4630     writeControl      writeTime;
4631     writeFields       true;
4632     surfaceFormat    raw;
4633     regionType       sampledSurface;
4634     name              yNormal3;
4635     operation        areaAverage;
4636 sampledSurfaceDict
4637     { type            plane;
4638 planeType pointAndNormal;
4639 triangulate false;
4640 interpolate false;
4641 pointAndNormalDict
4642     { basePoint      (0.0501 0.17992800E+01 0.005);
4643     normalVector    (0 1 0);
4644     }
4645     fields
4646     ( Ux UPrime2Mean
4647     ); }
4648
4649 piano_200
4650     { type            surfaceFieldValue;
4651     libs              ("libfieldFunctionObjects.so");
4652     log               true;
4653     writeControl      writeTime;
4654     writeFields       true;
4655     surfaceFormat    raw;
4656     regionType       sampledSurface;
4657     name              yNormal4;
4658     operation        areaAverage;
4659 sampledSurfaceDict
4660     { type            plane;
4661 planeType pointAndNormal;
4662 triangulate false;
4663 interpolate false;
4664 pointAndNormalDict
4665     { basePoint      (0.0501 0.18057645E+01 0.005);
4666     normalVector    (0 1 0);
4667     }
4668     fields
4669     ( Ux UPrime2Mean
4670     ); }
4671
4672 piano_201
4673     { type            surfaceFieldValue;
4674     libs              ("libfieldFunctionObjects.so");
4675     log               true;
4676     writeControl      writeTime;
4677     writeFields       true;

```

```

4678     surfaceFormat    raw;
4679     regionType       sampledSurface;
4680     name              yNormal5;
4681     operation         areaAverage;
4682 sampledSurfaceDict
4683     { type            plane;
4684 planeType pointAndNormal;
4685 triangulate false;
4686 interpolate  false;
4687     pointAndNormalDict
4688     { basePoint      (0.0501 0.18121183E+01 0.005);
4689       normalVector   (0 1 0);
4690     }
4691     fields
4692     ( Ux UPrime2Mean
4693     ); }
4694
4695 piano_202
4696     { type            surfaceFieldValue;
4697       libs            ("libfieldFunctionObjects.so");
4698       log              true;
4699       writeControl     writeTime;
4700       writeFields      true;
4701       surfaceFormat    raw;
4702       regionType       sampledSurface;
4703       name              yNormal0;
4704       operation         areaAverage;
4705 sampledSurfaceDict
4706     { type            plane;
4707 planeType pointAndNormal;
4708 triangulate false;
4709 interpolate  false;
4710     pointAndNormalDict
4711     { basePoint      (0.0501 0.18183442E+01 0.005);
4712       normalVector   (0 1 0);
4713     }
4714     fields
4715     ( Ux UPrime2Mean
4716     ); }
4717
4718 piano_203
4719     { type            surfaceFieldValue;
4720       libs            ("libfieldFunctionObjects.so");
4721       log              true;
4722       writeControl     writeTime;
4723       writeFields      true;
4724       surfaceFormat    raw;
4725       regionType       sampledSurface;
4726       name              yNormal1;
4727       operation         areaAverage;
4728 sampledSurfaceDict
4729     { type            plane;
4730 planeType pointAndNormal;
4731 triangulate false;
4732 interpolate  false;
4733     pointAndNormalDict
4734     { basePoint      (0.0501 0.18244447E+01 0.005);
4735       normalVector   (0 1 0);
4736     }
4737     fields
4738     ( Ux UPrime2Mean
4739     ); }
4740
4741 piano_204
4742     { type            surfaceFieldValue;
4743       libs            ("libfieldFunctionObjects.so");
4744       log              true;
4745       writeControl     writeTime;
4746       writeFields      true;

```

```

4747     surfaceFormat    raw;
4748     regionType      sampledSurface;
4749     name             yNormal2;
4750     operation        areaAverage;
4751 sampledSurfaceDict
4752 {   type            plane;
4753 planeType pointAndNormal;
4754 triangulate false;
4755 interpolate  false;
4756   pointAndNormalDict
4757   {   basePoint      (0.0501 0.18304223E+01 0.005);
4758       normalVector   (0 1 0);
4759   }
4760   fields
4761   (   Ux UPrime2Mean
4762   ); }
4763
4764 piano_205
4765 {   type            surfaceFieldValue;
4766     libs             ("libfieldFunctionObjects.so");
4767     log              true;
4768     writeControl     writeTime;
4769     writeFields      true;
4770     surfaceFormat    raw;
4771     regionType      sampledSurface;
4772     name             yNormal3;
4773     operation        areaAverage;
4774 sampledSurfaceDict
4775 {   type            plane;
4776 planeType pointAndNormal;
4777 triangulate false;
4778 interpolate  false;
4779   pointAndNormalDict
4780   {   basePoint      (0.0501 0.18362795E+01 0.005);
4781       normalVector   (0 1 0);
4782   }
4783   fields
4784   (   Ux UPrime2Mean
4785   ); }
4786
4787 piano_206
4788 {   type            surfaceFieldValue;
4789     libs             ("libfieldFunctionObjects.so");
4790     log              true;
4791     writeControl     writeTime;
4792     writeFields      true;
4793     surfaceFormat    raw;
4794     regionType      sampledSurface;
4795     name             yNormal4;
4796     operation        areaAverage;
4797 sampledSurfaceDict
4798 {   type            plane;
4799 planeType pointAndNormal;
4800 triangulate false;
4801 interpolate  false;
4802   pointAndNormalDict
4803   {   basePoint      (0.0501 0.18420188E+01 0.005);
4804       normalVector   (0 1 0);
4805   }
4806   fields
4807   (   Ux UPrime2Mean
4808   ); }
4809
4810 piano_207
4811 {   type            surfaceFieldValue;
4812     libs             ("libfieldFunctionObjects.so");
4813     log              true;
4814     writeControl     writeTime;
4815     writeFields      true;

```

```

4816     surfaceFormat    raw;
4817     regionType       sampledSurface;
4818     name              yNormal5;
4819     operation         areaAverage;
4820 sampledSurfaceDict
4821     { type            plane;
4822 planeType pointAndNormal;
4823 triangulate false;
4824 interpolate false;
4825     pointAndNormalDict
4826     { basePoint      (0.0501 0.18476425E+01 0.005);
4827       normalVector   (0 1 0);
4828     }
4829     fields
4830     ( Ux UPrime2Mean
4831       ); }
4832
4833 piano_208
4834     { type            surfaceFieldValue;
4835       libs             ("libfieldFunctionObjects.so");
4836       log              true;
4837       writeControl     writeTime;
4838       writeFields      true;
4839       surfaceFormat    raw;
4840       regionType       sampledSurface;
4841       name              yNormal0;
4842       operation         areaAverage;
4843 sampledSurfaceDict
4844     { type            plane;
4845 planeType pointAndNormal;
4846 triangulate false;
4847 interpolate false;
4848     pointAndNormalDict
4849     { basePoint      (0.0501 0.18531530E+01 0.005);
4850       normalVector   (0 1 0);
4851     }
4852     fields
4853     ( Ux UPrime2Mean
4854       ); }
4855
4856 piano_209
4857     { type            surfaceFieldValue;
4858       libs             ("libfieldFunctionObjects.so");
4859       log              true;
4860       writeControl     writeTime;
4861       writeFields      true;
4862       surfaceFormat    raw;
4863       regionType       sampledSurface;
4864       name              yNormal1;
4865       operation         areaAverage;
4866 sampledSurfaceDict
4867     { type            plane;
4868 planeType pointAndNormal;
4869 triangulate false;
4870 interpolate false;
4871     pointAndNormalDict
4872     { basePoint      (0.0501 0.18585524E+01 0.005);
4873       normalVector   (0 1 0);
4874     }
4875     fields
4876     ( Ux UPrime2Mean
4877       ); }
4878
4879 piano_210
4880     { type            surfaceFieldValue;
4881       libs             ("libfieldFunctionObjects.so");
4882       log              true;
4883       writeControl     writeTime;
4884       writeFields      true;

```

```

4885     surfaceFormat    raw;
4886     regionType       sampledSurface;
4887     name              yNormal2;
4888     operation         areaAverage;
4889 sampledSurfaceDict
4890 {   type              plane;
4891 planeType pointAndNormal;
4892 triangulate false;
4893 interpolate  false;
4894   pointAndNormalDict
4895     {   basePoint      (0.0501 0.18638432E+01 0.005);
4896         normalVector   (0 1 0);
4897     }
4898     fields
4899     (   Ux UPrime2Mean
4900     ); }
4901
4902 piano_211
4903 {   type              surfaceFieldValue;
4904     libs               ("libfieldFunctionObjects.so");
4905     log                 true;
4906     writeControl        writeTime;
4907     writeFields         true;
4908     surfaceFormat       raw;
4909     regionType          sampledSurface;
4910     name                yNormal3;
4911     operation           areaAverage;
4912 sampledSurfaceDict
4913 {   type              plane;
4914 planeType pointAndNormal;
4915 triangulate false;
4916 interpolate  false;
4917   pointAndNormalDict
4918     {   basePoint      (0.0501 0.18690273E+01 0.005);
4919         normalVector   (0 1 0);
4920     }
4921     fields
4922     (   Ux UPrime2Mean
4923     ); }
4924
4925 piano_212
4926 {   type              surfaceFieldValue;
4927     libs               ("libfieldFunctionObjects.so");
4928     log                 true;
4929     writeControl        writeTime;
4930     writeFields         true;
4931     surfaceFormat       raw;
4932     regionType          sampledSurface;
4933     name                yNormal4;
4934     operation           areaAverage;
4935 sampledSurfaceDict
4936 {   type              plane;
4937 planeType pointAndNormal;
4938 triangulate false;
4939 interpolate  false;
4940   pointAndNormalDict
4941     {   basePoint      (0.0501 0.18741071E+01 0.005);
4942         normalVector   (0 1 0);
4943     }
4944     fields
4945     (   Ux UPrime2Mean
4946     ); }
4947
4948 piano_213
4949 {   type              surfaceFieldValue;
4950     libs               ("libfieldFunctionObjects.so");
4951     log                 true;
4952     writeControl        writeTime;
4953     writeFields         true;

```

```

4954     surfaceFormat    raw;
4955     regionType       sampledSurface;
4956     name              yNormal5;
4957     operation         areaAverage;
4958 sampledSurfaceDict
4959     { type            plane;
4960 planeType pointAndNormal;
4961 triangulate false;
4962 interpolate false;
4963     pointAndNormalDict
4964     { basePoint      (0.0501 0.18790846E+01 0.005);
4965       normalVector   (0 1 0);
4966     }
4967     fields
4968     ( Ux UPrime2Mean
4969     ); }
4970
4971 piano_214
4972     { type            surfaceFieldValue;
4973       libs            ("libfieldFunctionObjects.so");
4974       log              true;
4975       writeControl     writeTime;
4976       writeFields      true;
4977       surfaceFormat    raw;
4978       regionType       sampledSurface;
4979       name              yNormal0;
4980       operation         areaAverage;
4981 sampledSurfaceDict
4982     { type            plane;
4983 planeType pointAndNormal;
4984 triangulate false;
4985 interpolate false;
4986     pointAndNormalDict
4987     { basePoint      (0.0501 0.18839618E+01 0.005);
4988       normalVector   (0 1 0);
4989     }
4990     fields
4991     ( Ux UPrime2Mean
4992     ); }
4993
4994 piano_215
4995     { type            surfaceFieldValue;
4996       libs            ("libfieldFunctionObjects.so");
4997       log              true;
4998       writeControl     writeTime;
4999       writeFields      true;
5000       surfaceFormat    raw;
5001       regionType       sampledSurface;
5002       name              yNormal1;
5003       operation         areaAverage;
5004 sampledSurfaceDict
5005     { type            plane;
5006 planeType pointAndNormal;
5007 triangulate false;
5008 interpolate false;
5009     pointAndNormalDict
5010     { basePoint      (0.0501 0.18887408E+01 0.005);
5011       normalVector   (0 1 0);
5012     }
5013     fields
5014     ( Ux UPrime2Mean
5015     ); }
5016
5017 piano_216
5018     { type            surfaceFieldValue;
5019       libs            ("libfieldFunctionObjects.so");
5020       log              true;
5021       writeControl     writeTime;
5022       writeFields      true;

```

```

5023     surfaceFormat    raw;
5024     regionType       sampledSurface;
5025     name              yNormal2;
5026     operation        areaAverage;
5027 sampledSurfaceDict
5028     { type            plane;
5029 planeType pointAndNormal;
5030 triangulate false;
5031 interpolate false;
5032     pointAndNormalDict
5033     { basePoint      (0.0501 0.18934236E+01 0.005);
5034       normalVector   (0 1 0);
5035     }
5036     fields
5037     (   Ux UPrime2Mean
5038       );   }
5039
5040 piano_217
5041     { type            surfaceFieldValue;
5042       libs            ("libfieldFunctionObjects.so");
5043       log              true;
5044       writeControl     writeTime;
5045       writeFields      true;
5046       surfaceFormat    raw;
5047       regionType       sampledSurface;
5048       name              yNormal3;
5049       operation        areaAverage;
5050 sampledSurfaceDict
5051     { type            plane;
5052 planeType pointAndNormal;
5053 triangulate false;
5054 interpolate false;
5055     pointAndNormalDict
5056     { basePoint      (0.0501 0.18980120E+01 0.005);
5057       normalVector   (0 1 0);
5058     }
5059     fields
5060     (   Ux UPrime2Mean
5061       );   }
5062
5063 piano_218
5064     { type            surfaceFieldValue;
5065       libs            ("libfieldFunctionObjects.so");
5066       log              true;
5067       writeControl     writeTime;
5068       writeFields      true;
5069       surfaceFormat    raw;
5070       regionType       sampledSurface;
5071       name              yNormal4;
5072       operation        areaAverage;
5073 sampledSurfaceDict
5074     { type            plane;
5075 planeType pointAndNormal;
5076 triangulate false;
5077 interpolate false;
5078     pointAndNormalDict
5079     { basePoint      (0.0501 0.19025081E+01 0.005);
5080       normalVector   (0 1 0);
5081     }
5082     fields
5083     (   Ux UPrime2Mean
5084       );   }
5085
5086 piano_219
5087     { type            surfaceFieldValue;
5088       libs            ("libfieldFunctionObjects.so");
5089       log              true;
5090       writeControl     writeTime;
5091       writeFields      true;

```

```

5092     surfaceFormat    raw;
5093     regionType       sampledSurface;
5094     name              yNormal5;
5095     operation         areaAverage;
5096 sampledSurfaceDict
5097     { type            plane;
5098 planeType pointAndNormal;
5099 triangulate false;
5100 interpolate  false;
5101   pointAndNormalDict
5102     { basePoint      (0.0501 0.19069136E+01 0.005);
5103       normalVector   (0 1 0);
5104   }
5105     fields
5106     (   Ux UPrime2Mean
5107       );   }
5108
5109 piano_220
5110   { type            surfaceFieldValue;
5111     libs            ("libfieldFunctionObjects.so");
5112     log              true;
5113     writeControl    writeTime;
5114     writeFields     true;
5115     surfaceFormat   raw;
5116     regionType      sampledSurface;
5117     name            yNormal0;
5118     operation       areaAverage;
5119 sampledSurfaceDict
5120   { type            plane;
5121 planeType pointAndNormal;
5122 triangulate false;
5123 interpolate  false;
5124   pointAndNormalDict
5125     { basePoint      (0.0501 0.19112304E+01 0.005);
5126       normalVector   (0 1 0);
5127   }
5128     fields
5129     (   Ux UPrime2Mean
5130       );   }
5131
5132 piano_221
5133   { type            surfaceFieldValue;
5134     libs            ("libfieldFunctionObjects.so");
5135     log              true;
5136     writeControl    writeTime;
5137     writeFields     true;
5138     surfaceFormat   raw;
5139     regionType      sampledSurface;
5140     name            yNormal1;
5141     operation       areaAverage;
5142 sampledSurfaceDict
5143   { type            plane;
5144 planeType pointAndNormal;
5145 triangulate false;
5146 interpolate  false;
5147   pointAndNormalDict
5148     { basePoint      (0.0501 0.19154602E+01 0.005);
5149       normalVector   (0 1 0);
5150   }
5151     fields
5152     (   Ux UPrime2Mean
5153       );   }
5154
5155 piano_222
5156   { type            surfaceFieldValue;
5157     libs            ("libfieldFunctionObjects.so");
5158     log              true;
5159     writeControl    writeTime;
5160     writeFields     true;

```

```

5161     surfaceFormat    raw;
5162     regionType       sampledSurface;
5163     name              yNormal2;
5164     operation         areaAverage;
5165 sampledSurfaceDict
5166     { type            plane;
5167 planeType pointAndNormal;
5168 triangulate false;
5169 interpolate false;
5170     pointAndNormalDict
5171     { basePoint      (0.0501 0.19196049E+01 0.005);
5172       normalVector   (0 1 0);
5173     }
5174     fields
5175     ( Ux UPrime2Mean
5176     ); }
5177
5178 piano_223
5179     { type            surfaceFieldValue;
5180     libs              ("libfieldFunctionObjects.so");
5181     log               true;
5182     writeControl      writeTime;
5183     writeFields       true;
5184     surfaceFormat     raw;
5185     regionType       sampledSurface;
5186     name              yNormal3;
5187     operation         areaAverage;
5188 sampledSurfaceDict
5189     { type            plane;
5190 planeType pointAndNormal;
5191 triangulate false;
5192 interpolate false;
5193     pointAndNormalDict
5194     { basePoint      (0.0501 0.19236661E+01 0.005);
5195       normalVector   (0 1 0);
5196     }
5197     fields
5198     ( Ux UPrime2Mean
5199     ); }
5200
5201 piano_224
5202     { type            surfaceFieldValue;
5203     libs              ("libfieldFunctionObjects.so");
5204     log               true;
5205     writeControl      writeTime;
5206     writeFields       true;
5207     surfaceFormat     raw;
5208     regionType       sampledSurface;
5209     name              yNormal4;
5210     operation         areaAverage;
5211 sampledSurfaceDict
5212     { type            plane;
5213 planeType pointAndNormal;
5214 triangulate false;
5215 interpolate false;
5216     pointAndNormalDict
5217     { basePoint      (0.0501 0.19276455E+01 0.005);
5218       normalVector   (0 1 0);
5219     }
5220     fields
5221     ( Ux UPrime2Mean
5222     ); }
5223
5224 piano_225
5225     { type            surfaceFieldValue;
5226     libs              ("libfieldFunctionObjects.so");
5227     log               true;
5228     writeControl      writeTime;
5229     writeFields       true;

```

```

5230     surfaceFormat    raw;
5231     regionType       sampledSurface;
5232     name              yNormal5;
5233     operation         areaAverage;
5234 sampledSurfaceDict
5235     { type            plane;
5236 planeType pointAndNormal;
5237 triangulate false;
5238 interpolate false;
5239     pointAndNormalDict
5240     { basePoint      (0.0501 0.19315447E+01 0.005);
5241       normalVector   (0 1 0);
5242     }
5243     fields
5244     ( Ux UPrime2Mean
5245     ); }
5246
5247 piano_226
5248     { type            surfaceFieldValue;
5249       libs             ("libfieldFunctionObjects.so");
5250       log               true;
5251       writeControl      writeTime;
5252       writeFields       true;
5253       surfaceFormat     raw;
5254       regionType       sampledSurface;
5255       name              yNormal0;
5256       operation         areaAverage;
5257 sampledSurfaceDict
5258     { type            plane;
5259 planeType pointAndNormal;
5260 triangulate false;
5261 interpolate false;
5262     pointAndNormalDict
5263     { basePoint      (0.0501 0.19353654E+01 0.005);
5264       normalVector   (0 1 0);
5265     }
5266     fields
5267     ( Ux UPrime2Mean
5268     ); }
5269
5270 piano_227
5271     { type            surfaceFieldValue;
5272       libs             ("libfieldFunctionObjects.so");
5273       log               true;
5274       writeControl      writeTime;
5275       writeFields       true;
5276       surfaceFormat     raw;
5277       regionType       sampledSurface;
5278       name              yNormal1;
5279       operation         areaAverage;
5280 sampledSurfaceDict
5281     { type            plane;
5282 planeType pointAndNormal;
5283 triangulate false;
5284 interpolate false;
5285     pointAndNormalDict
5286     { basePoint      (0.0501 0.19391092E+01 0.005);
5287       normalVector   (0 1 0);
5288     }
5289     fields
5290     ( Ux UPrime2Mean
5291     ); }
5292
5293 piano_228
5294     { type            surfaceFieldValue;
5295       libs             ("libfieldFunctionObjects.so");
5296       log               true;
5297       writeControl      writeTime;
5298       writeFields       true;

```

```

5299     surfaceFormat    raw;
5300     regionType        sampledSurface;
5301     name              yNormal2;
5302     operation         areaAverage;
5303 sampledSurfaceDict
5304     { type            plane;
5305 planeType pointAndNormal;
5306 triangulate false;
5307 interpolate false;
5308   pointAndNormalDict
5309     { basePoint      (0.0501 0.19427776E+01 0.005);
5310       normalVector   (0 1 0);
5311   }
5312   fields
5313   (   Ux UPrime2Mean
5314     ); }
5315
5316 piano_229
5317   { type              surfaceFieldValue;
5318     libs              ("libfieldFunctionObjects.so");
5319     log               true;
5320     writeControl      writeTime;
5321     writeFields       true;
5322     surfaceFormat     raw;
5323     regionType        sampledSurface;
5324     name              yNormal3;
5325     operation         areaAverage;
5326 sampledSurfaceDict
5327     { type            plane;
5328 planeType pointAndNormal;
5329 triangulate false;
5330 interpolate false;
5331   pointAndNormalDict
5332     { basePoint      (0.0501 0.19463721E+01 0.005);
5333       normalVector   (0 1 0);
5334   }
5335   fields
5336   (   Ux UPrime2Mean
5337     ); }
5338
5339 piano_230
5340   { type              surfaceFieldValue;
5341     libs              ("libfieldFunctionObjects.so");
5342     log               true;
5343     writeControl      writeTime;
5344     writeFields       true;
5345     surfaceFormat     raw;
5346     regionType        sampledSurface;
5347     name              yNormal4;
5348     operation         areaAverage;
5349 sampledSurfaceDict
5350     { type            plane;
5351 planeType pointAndNormal;
5352 triangulate false;
5353 interpolate false;
5354   pointAndNormalDict
5355     { basePoint      (0.0501 0.19498942E+01 0.005);
5356       normalVector   (0 1 0);
5357   }
5358   fields
5359   (   Ux UPrime2Mean
5360     ); }
5361
5362 piano_231
5363   { type              surfaceFieldValue;
5364     libs              ("libfieldFunctionObjects.so");
5365     log               true;
5366     writeControl      writeTime;
5367     writeFields       true;

```

```

5368     surfaceFormat    raw;
5369     regionType       sampledSurface;
5370     name              yNormal5;
5371     operation         areaAverage;
5372 sampledSurfaceDict
5373     { type            plane;
5374 planeType pointAndNormal;
5375 triangulate false;
5376 interpolate false;
5377     pointAndNormalDict
5378     { basePoint      (0.0501 0.19533454E+01 0.005);
5379       normalVector   (0 1 0);
5380     }
5381     fields
5382     ( Ux UPrime2Mean
5383     ); }
5384
5385 piano_232
5386     { type            surfaceFieldValue;
5387       libs            ("libfieldFunctionObjects.so");
5388       log              true;
5389       writeControl    writeTime;
5390       writeFields     true;
5391       surfaceFormat   raw;
5392       regionType      sampledSurface;
5393       name             yNormal3;
5394       operation        areaAverage;
5395 sampledSurfaceDict
5396     { type            plane;
5397 planeType pointAndNormal;
5398 triangulate false;
5399 interpolate false;
5400     pointAndNormalDict
5401     { basePoint      (0.0501 0.19567271E+01 0.005);
5402       normalVector   (0 1 0);
5403     }
5404     fields
5405     ( Ux UPrime2Mean
5406     ); }
5407
5408 piano_233
5409     { type            surfaceFieldValue;
5410       libs            ("libfieldFunctionObjects.so");
5411       log              true;
5412       writeControl    writeTime;
5413       writeFields     true;
5414       surfaceFormat   raw;
5415       regionType      sampledSurface;
5416       name             yNormal4;
5417       operation        areaAverage;
5418 sampledSurfaceDict
5419     { type            plane;
5420 planeType pointAndNormal;
5421 triangulate false;
5422 interpolate false;
5423     pointAndNormalDict
5424     { basePoint      (0.0501 0.19600407E+01 0.005);
5425       normalVector   (0 1 0);
5426     }
5427     fields
5428     ( Ux UPrime2Mean
5429     ); }
5430
5431 piano_234
5432     { type            surfaceFieldValue;
5433       libs            ("libfieldFunctionObjects.so");
5434       log              true;
5435       writeControl    writeTime;
5436       writeFields     true;

```

```

5437     surfaceFormat    raw;
5438     regionType       sampledSurface;
5439     name              yNormal5;
5440     operation         areaAverage;
5441 sampledSurfaceDict
5442     { type            plane;
5443 planeType pointAndNormal;
5444 triangulate false;
5445 interpolate false;
5446     pointAndNormalDict
5447     { basePoint      (0.0501 0.19664690E+01 0.005);
5448       normalVector   (0 1 0);
5449     }
5450     fields
5451     ( Ux UPrime2Mean
5452     ); }
5453
5454 piano_235
5455     { type            surfaceFieldValue;
5456       libs            ("libfieldFunctionObjects.so");
5457       log              true;
5458       writeControl     writeTime;
5459       writeFields      true;
5460       surfaceFormat    raw;
5461       regionType       sampledSurface;
5462       name              yNormal0;
5463       operation         areaAverage;
5464 sampledSurfaceDict
5465     { type            plane;
5466 planeType pointAndNormal;
5467 triangulate false;
5468 interpolate false;
5469     pointAndNormalDict
5470     { basePoint      (0.0501 0.19664690E+01 0.005);
5471       normalVector   (0 1 0);
5472     }
5473     fields
5474     ( Ux UPrime2Mean
5475     ); }
5476
5477 piano_236
5478     { type            surfaceFieldValue;
5479       libs            ("libfieldFunctionObjects.so");
5480       log              true;
5481       writeControl     writeTime;
5482       writeFields      true;
5483       surfaceFormat    raw;
5484       regionType       sampledSurface;
5485       name              yNormal1;
5486       operation         areaAverage;
5487 sampledSurfaceDict
5488     { type            plane;
5489 planeType pointAndNormal;
5490 triangulate false;
5491 interpolate false;
5492     pointAndNormalDict
5493     { basePoint      (0.0501 0.19695864E+01 0.005);
5494       normalVector   (0 1 0);
5495     }
5496     fields
5497     ( Ux UPrime2Mean
5498     ); }
5499
5500 piano_237
5501     { type            surfaceFieldValue;
5502       libs            ("libfieldFunctionObjects.so");
5503       log              true;
5504       writeControl     writeTime;
5505       writeFields      true;

```

```

5506     surfaceFormat    raw;
5507     regionType       sampledSurface;
5508     name              yNormal2;
5509     operation         areaAverage;
5510 sampledSurfaceDict
5511     { type            plane;
5512 planeType pointAndNormal;
5513 triangulate false;
5514 interpolate false;
5515     pointAndNormalDict
5516     { basePoint      (0.0501 0.19726410E+01 0.005);
5517       normalVector   (0 1 0);
5518     }
5519     fields
5520     ( Ux UPrime2Mean
5521     ); }
5522
5523 piano_238
5524     { type            surfaceFieldValue;
5525     libs              ("libfieldFunctionObjects.so");
5526     log               true;
5527     writeControl      writeTime;
5528     writeFields       true;
5529     surfaceFormat     raw;
5530     regionType       sampledSurface;
5531     name              yNormal3;
5532     operation         areaAverage;
5533 sampledSurfaceDict
5534     { type            plane;
5535 planeType pointAndNormal;
5536 triangulate false;
5537 interpolate false;
5538     pointAndNormalDict
5539     { basePoint      (0.0501 0.19756341E+01 0.005);
5540       normalVector   (0 1 0);
5541     }
5542     fields
5543     ( Ux UPrime2Mean
5544     ); }
5545
5546 piano_239
5547     { type            surfaceFieldValue;
5548     libs              ("libfieldFunctionObjects.so");
5549     log               true;
5550     writeControl      writeTime;
5551     writeFields       true;
5552     surfaceFormat     raw;
5553     regionType       sampledSurface;
5554     name              yNormal4;
5555     operation         areaAverage;
5556 sampledSurfaceDict
5557     { type            plane;
5558 planeType pointAndNormal;
5559 triangulate false;
5560 interpolate false;
5561     pointAndNormalDict
5562     { basePoint      (0.0501 0.19785669E+01 0.005);
5563       normalVector   (0 1 0);
5564     }
5565     fields
5566     ( Ux UPrime2Mean
5567     ); }
5568
5569 piano_240
5570     { type            surfaceFieldValue;
5571     libs              ("libfieldFunctionObjects.so");
5572     log               true;
5573     writeControl      writeTime;
5574     writeFields       true;

```

```

5575     surfaceFormat    raw;
5576     regionType       sampledSurface;
5577     name              yNormal5;
5578     operation         areaAverage;
5579 sampledSurfaceDict
5580     { type            plane;
5581 planeType pointAndNormal;
5582 triangulate false;
5583 interpolate false;
5584 pointAndNormalDict
5585     { basePoint      (0.0501 0.19785669E+01 0.005);
5586     normalVector    (0 1 0);
5587 }
5588     fields
5589     ( Ux UPrime2Mean
5590     ); }
5591
5592 piano_241
5593     { type            surfaceFieldValue;
5594     libs              ("libfieldFunctionObjects.so");
5595     log               true;
5596     writeControl      writeTime;
5597     writeFields       true;
5598     surfaceFormat     raw;
5599     regionType        sampledSurface;
5600     name              yNormal3;
5601     operation         areaAverage;
5602 sampledSurfaceDict
5603     { type            plane;
5604 planeType pointAndNormal;
5605 triangulate false;
5606 interpolate false;
5607 pointAndNormalDict
5608     { basePoint      (0.0501 0.19842565E+01 0.005);
5609     normalVector    (0 1 0);
5610 }
5611     fields
5612     ( Ux UPrime2Mean
5613     ); }
5614
5615 piano_242
5616     { type            surfaceFieldValue;
5617     libs              ("libfieldFunctionObjects.so");
5618     log               true;
5619     writeControl      writeTime;
5620     writeFields       true;
5621     surfaceFormat     raw;
5622     regionType        sampledSurface;
5623     name              yNormal4;
5624     operation         areaAverage;
5625 sampledSurfaceDict
5626     { type            plane;
5627 planeType pointAndNormal;
5628 triangulate false;
5629 interpolate false;
5630 pointAndNormalDict
5631     { basePoint      (0.0501 0.19870157E+01 0.005);
5632     normalVector    (0 1 0);
5633 }
5634     fields
5635     ( Ux UPrime2Mean
5636     ); }
5637
5638 piano_243
5639     { type            surfaceFieldValue;
5640     libs              ("libfieldFunctionObjects.so");
5641     log               true;
5642     writeControl      writeTime;
5643     writeFields       true;

```

```

5644     surfaceFormat    raw;
5645     regionType       sampledSurface;
5646     name              yNormal5;
5647     operation        areaAverage;
5648 sampledSurfaceDict
5649     { type            plane;
5650 planeType pointAndNormal;
5651 triangulate false;
5652 interpolate false;
5653     pointAndNormalDict
5654         { basePoint    (0.0501 0.19897192E+01 0.005);
5655           normalVector (0 1 0);
5656         }
5657     fields
5658     ( Ux UPrime2Mean
5659     ); }
5660
5661 piano_244
5662     { type            surfaceFieldValue;
5663     libs              ("libfieldFunctionObjects.so");
5664     log               true;
5665     writeControl      writeTime;
5666     writeFields       true;
5667     surfaceFormat    raw;
5668     regionType       sampledSurface;
5669     name              yNormal0;
5670     operation        areaAverage;
5671 sampledSurfaceDict
5672     { type            plane;
5673 planeType pointAndNormal;
5674 triangulate false;
5675 interpolate false;
5676     pointAndNormalDict
5677         { basePoint    (0.0501 0.19923684E+01 0.005);
5678           normalVector (0 1 0);
5679         }
5680     fields
5681     ( Ux UPrime2Mean
5682     ); }
5683
5684 piano_245
5685     { type            surfaceFieldValue;
5686     libs              ("libfieldFunctionObjects.so");
5687     log               true;
5688     writeControl      writeTime;
5689     writeFields       true;
5690     surfaceFormat    raw;
5691     regionType       sampledSurface;
5692     name              yNormal1;
5693     operation        areaAverage;
5694 sampledSurfaceDict
5695     { type            plane;
5696 planeType pointAndNormal;
5697 triangulate false;
5698 interpolate false;
5699     pointAndNormalDict
5700         { basePoint    (0.0501 0.19949642E+01 0.005);
5701           normalVector (0 1 0);
5702         }
5703     fields
5704     ( Ux UPrime2Mean
5705     ); }
5706 } // Chiusura della functions

```

Modifiche al controlDictReynolds

Nel nostro caso, si è utilizzato solamente questa versione del codice, per tutte le simulazioni. Comunque, se si vuole andare a cambiare le grandezze da estrarre con questo codice, lo si deve andare a specificare, sia nella righe 41, 53 e 60 della funzione, e sia in tutte le applicazioni sui vari piani indicati, come per esempio nelle righe 84 o 107.

Se invece si vogliono cambiare i piani da cui vengono estratti i dati, si devono cambiare le coordinate del punto di origine e/o la direzione normale al piano da prendere, ovvero, per esempio, nelle righe 80-81, o in quelle 103-104.

Infine se si vogliono aggiungere nuovi piani, si deve copiare la relativa parte di codice, per esempio le righe da 64 a 85, specificando poi il nuovo piano. Se invece si vuole eliminare un piano, basta eliminare o commentare la relativa parte di codice.

B.2 Codice riordino grandezze Reynolds

Questo codice, scritto nel linguaggio *Fortran*, serve per riordinare i dati che si ottengono dal lancio del codice di post processing, come descritto in 4.3.1. Inoltre dei dati estratti, ne fa la media temporale, calcola la derivata della velocità, e genera diversi file, ognuno con una grandezza riordinata, più quello con le medie temporali. Come tutti i codici *Fortran*, prima di essere utilizzato, deve essere compilato, e poi si lancia l'eseguibile che si ottiene dalla compilazione. Inoltre per il suo utilizzo richiede un file di input, ed un file con le coordinate lungo y , che vengono riportati di seguito, dopo questo codice.

Codici_simulazione/ricostruisci_umedia_v2bis_ureynolds_RE_all.f90

```

1 program ricostruisci
2
3 implicit none
4
5 character(LEN=60) :: nome, nomedir
6 integer :: N1,N2,Ntempi,Itmin,Itmax,Nrighe,Nmedia
7 integer :: i,j,it
8
9 real(kind=8) :: it_real
10 real(kind=8), allocatable, dimension(:) :: ymedia,Umm, dumdy
11 real(kind=8), allocatable, dimension(:,) :: umedia
12 real(kind=8), allocatable, dimension(:,) :: uu,uv,uw,vv,vw,ww
13 real(kind=8), allocatable, dimension(:) :: uum,uvm,uwm,vvm,vwm,wwm, tempi
14 real(kind=8) :: Tmin,Tmax,Deltat,Tmedia
15 character(len=150) :: riga
16 integer :: pos1,pos2
17 integer :: iflag
18
19 !esecuzione
20
21 write(*,*)'Indici primo e ultimo piano (=Numero di piani se numerati da 1) '
22 read(*,*) N1, N2
23 write(*,*)'Istante inizio/istante fine & Deltat '
24 read(*,*) Tmin,Tmax, Deltat
25 write(*,*)'Istante inizio media '
26 read(*,*) Tmedia
27 Nrighe=NINT((Tmax-Tmin)/Deltat)+1
28 Nmedia=NINT((Tmedia-Tmin)/Deltat)+1
29
30 !allocazione file:
31
32 allocate(umedia(N1:N2,Nmedia:Nrighe),ymedia(N1:N2),Umm(N1:N2),dumdy(N1:N2))
33 !...sforzi di Reynolds: valori medi istantanei e medie globali
34 allocate(uu(N1:N2,Nmedia:Nrighe),uum(N1:N2))
35 allocate(uv(N1:N2,Nmedia:Nrighe),uvm(N1:N2))

```

```

36 allocate(uw(N1:N2, Nmedia:Nrighe), uwm(N1:N2))
37 allocate(vv(N1:N2, Nmedia:Nrighe), vvm(N1:N2))
38 allocate(vw(N1:N2, Nmedia:Nrighe), vwm(N1:N2))
39 allocate(ww(N1:N2, Nmedia:Nrighe), wwm(N1:N2))
40
41 !...variabile tempo
42 allocate(tempi(Nmedia:Nrighe))
43 !nota: umedia: primo indice spazio, seconmdo indice tempo
44
45 !..Leggo i piani:
46
47 open(10, file='coordinate_y.txt', status='old', action='read', iostat=iflag)
48 if(iflag.ne.0) then
49   write(*,*) 'Errore nel file coordinate_y.txt. Esiste?'
50   stop
51 end if
52 do j=N1, N2
53   read(10,*) ymedia(j)
54 end do
55 close(10)
56
57 ! Ciclo principale. Indice j indica il piano (qui da uno ad N)
58 !~~~~~
59 do j=N1, N2 !<-- i piani sono numerati da 1 ad N
60 !~~~~~
61
62 write(*,*) 'Piano numero=', j
63 write(nome, '(./postProcessing_RE_all/piano_', iIO, '/', iIO, '/surfaceFieldValue.
   dat)")' j, NINT(Tmin)
64 write(*,*) 'File piano = ', nome
65
66 open(1, file=nome, status='old', action='read')
67
68 ! Lettura righe vuote:
69 read(1,*)
70 read(1,*)
71 read(1,*)
72 read(1,*)
73
74 do i=1, Nmedia-1
75   read(1,*)
76 end do
77
78 do i=Nmedia, Nrighe
79   read(1, '(A)') riga
80   !..leggo tempo e velocità:
81   read(riga,*) tempi(i), umedia(j, i)
82   !..leggo gli sforzi di reynolds, scritti tra le parentesi, eliminando le
   parentesi:
83   pos1=scan(riga, "(")
84   pos2=scan(riga, ")", back=.true.)
85   read(riga(pos1+1:pos2-1), *) uu(j, i), uv(j, i), uw(j, i), vv(j, i), vw(j, i), ww(j, i)
86 end do
87 close(1)
88
89 write(*,*) 'Fine iterazione numero.....', j
90 !~~~~~
91 end do
92 !~~~~~
93 write(*,*) 'tempo=', tempi(Nmedia), tempi(Nmedia+1), tempi(Nrighe)
94
95 nome='umedia_tutti_v2.txt'
96 open(1, file=nome)
97 !write(1, '(A)') '# y, umedia(t1), umedia(t2) ecc.'
98 write(1, '(A, I14, 600I15)') '#colonna n:', (i, i=1, (Nrighe-Nmedia+2))
99 write(1, '(A, 600F15.3)') '# y_piano, t= ', tempi(Nmedia:Nrighe)
100 do j=N1, N2
101 ! write(1, '(300e15.7)') dfloat(j), umedia(j, Itmin:Itmax)
102 write(1, '(600e15.7)') ymedia(j), umedia(j, Nmedia:Nrighe)

```

```

103 end do
104 close(1)
105
106 nome='uu_tutti_v2.txt'
107 open(1,file=nome)
108 ! write(1,'(A)')'# y, umedia(t1), umedia(t2) ecc.'
109 write(1,'(A,1I4,600I15)')'#colonna n:',(i,i=1,(Nrighe-Nmedia+2))
110 write(1,'(A,600F15.3)')'# y_piano, t= ',tempi(Nmedia:Nrighe)
111 do j=N1,N2
112 ! write(1,'(300e15.7)')dfloat(j),umedia(j,Itmin:Itmax)
113 write(1,'(600e15.7)')ymedia(j),uu(j,Nmedia:Nrighe)
114 end do
115 close(1)
116
117 nome='uv_tutti_v2.txt'
118 open(1,file=nome)
119 ! write(1,'(A)')'# y, umedia(t1), umedia(t2) ecc.'
120 write(1,'(A,1I4,600I15)')'#colonna n:',(i,i=1,(Nrighe-Nmedia+2))
121 write(1,'(A,600F15.3)')'# y_piano, t= ',tempi(Nmedia:Nrighe)
122 do j=N1,N2
123 ! write(1,'(300e15.7)')dfloat(j),umedia(j,Itmin:Itmax)
124 write(1,'(600e15.7)')ymedia(j),uv(j,Nmedia:Nrighe)
125 end do
126 close(1)
127
128 nome='vv_tutti_v2.txt'
129 open(1,file=nome)
130 ! write(1,'(A)')'# y, umedia(t1), umedia(t2) ecc.'
131 write(1,'(A,1I4,600I15)')'#colonna n:',(i,i=1,(Nrighe-Nmedia+2))
132 write(1,'(A,600F15.3)')'# y_piano, t= ',tempi(Nmedia:Nrighe)
133 do j=N1,N2
134 ! write(1,'(300e15.7)')dfloat(j),umedia(j,Itmin:Itmax)
135 write(1,'(600e15.7)')ymedia(j),vv(j,Nmedia:Nrighe)
136 end do
137 close(1)
138
139 nome='ww_tutti_v2.txt'
140 open(1,file=nome)
141 ! write(1,'(A)')'# y, umedia(t1), umedia(t2) ecc.'
142 write(1,'(A,1I4,600I15)')'#colonna n:',(i,i=1,(Nrighe-Nmedia+2))
143 write(1,'(A,600F15.3)')'# y_piano, t= ',tempi(Nmedia:Nrighe)
144 do j=N1,N2
145 ! write(1,'(300e15.7)')dfloat(j),umedia(j,Itmin:Itmax)
146 write(1,'(600e15.7)')ymedia(j),ww(j,Nmedia:Nrighe)
147 end do
148 close(1)
149
150 write(*,*)'Fine scrittura U,reynolds istatanei'
151
152 Umm = SUM(umedia,dim=2)/ (Nrighe-Nmedia+1)
153 UUM = SUM(UU,dim=2)/ (Nrighe-Nmedia+1)
154 UVM = SUM(UV,dim=2)/ (Nrighe-Nmedia+1)
155 UWM = SUM(UW,dim=2)/ (Nrighe-Nmedia+1)
156 VVM = SUM(VV,dim=2)/ (Nrighe-Nmedia+1)
157 VWM = SUM(VW,dim=2)/ (Nrighe-Nmedia+1)
158 WWM = SUM(WW,dim=2)/ (Nrighe-Nmedia+1)
159
160 !..calcolo dU/dy:
161
162 if(N1.eq.N2) then
163 dumdy(:) = 0
164 else
165 dumdy(N1) = (umm(N1+1)-umm(N1))/(ymedia(N1+1)-ymedia(N1))
166 do j=N1+1,N2-1
167 dumdy(j) = (Umm(j+1)-Umm(j))/(ymedia(j+1)-ymedia(j))*(ymedia(j)-ymedia(j-1))
168 + (Umm(j)-Umm(j-1))/(ymedia(j)-ymedia(j-1))*(ymedia(j+1)-ymedia(j))/(
169 ymedia(j+1)-ymedia(j-1))
169 end do

```

```

170 dumdy(N2) = (umm(N2)-umm(N2-1))/(ymedia(N2)-ymedia(N2-1))
171 end if
172
173 nome='umedia_fin_v2_ureynolds.txt'
174
175 open(1,file=nome)
176 write(1,'(A,1I1,1I11,600I15)')'#colonna n:',(i,i=1,9)
177 write(1,'(12A)')'# y_piano ',&
178 '          umedia ',&
179 '          d<u>/dy ',&
180 '          <uu> ',&
181 '          <uv> ',&
182 '          <uw> ',&
183 '          <vv> ',&
184 '          <vw> ',&
185 '          <ww> '
186
187 do j=N1,N2
188   write(1,'(300e15.7)')ymedia(j),Umm(j),dumdy(j),UUM(J),UVM(J),UWM(J),VVM(J),VWM
(J),WWM(J)
189 end do
190
191 close(1)
192
193 deallocate(uu,uv,uw,vv,vw,ww)
194 deallocate(uum,uvm,uwm,vvm,vwm,wmm,tempi)
195
196 stop
197 end

```

B.2.1 Input codice Riordina

Questo è il file contenente gli input che devono essere dati al codice di riordino, affinché questo funzioni. Viene specificato, per ogni riga di input, cosa questi siano. Il comando per lanciare il codice con già imposto il file di input è riportato nella sezione 4.3.1.

Codici_simulazione/input_media.riordina_rey

```

1 1 245      ### Indice piano 1 ed indice ultimo piano
2 700 1150 2    ### Tempo iniziale, tempo finale, passo temporale
3 1050        ### Tempo di inizio media,

```

B.2.2 Coordinate Y

Questo è il file contenente i valori delle coordinate lungo y su cui si sono presi i dati del post processing, che deve trovarsi nella stessa cartella da cui viene lanciato il codice di riordino, affinché venga letto ed i suoi valori vengano inseriti nei file di input che il codice di riordino fornisce. Il file viene qui riportato diviso in 3 colonne, ma il file di testo che si deve fornire al codice deve essere strutturato con tutti i dati su di un'unica colonna.

	coordinate_y.txt				
		12	0.36712464E-01	24	0.80395127E-01
		13	0.39959309E-01	25	0.84539784E-01
1	0.50358107E-02	14	0.43272888E-01	26	0.88769627E-01
2	0.76316051E-02	15	0.46654570E-01	27	0.93086405E-01
3	0.10280751E-01	16	0.50105756E-01	28	0.97491907E-01
4	0.12984345E-01	17	0.53627875E-01	29	0.10198795
5	0.15743506E-01	18	0.57222384E-01	30	0.10657641
6	0.18559376E-01	19	0.60890771E-01	31	0.11125917
7	0.21433121E-01	20	0.64634554E-01	32	0.11603818
8	0.24365930E-01	21	0.68455284E-01	33	0.12091541
9	0.27359017E-01	22	0.72354541E-01	34	0.12589288
10	0.30413621E-01	23	0.76333940E-01	35	0.13097265
11	0.33531006E-01				

36	0.13615683	105	0.79545455	174	0.15853268E+01
37	0.14144756	106	0.80681819	175	0.15952635E+01
38	0.14684703	107	0.81818183	176	0.16050464E+01
39	0.15235748	108	0.82954546	177	0.16146779E+01
40	0.15798118	109	0.84090910	178	0.16241604E+01
41	0.16372047	110	0.85227274	179	0.16334962E+01
42	0.16957771	111	0.86363637	180	0.16426875E+01
43	0.17555534	112	0.87500001	181	0.16517366E+01
44	0.18165583	113	0.88636365	182	0.16606456E+01
45	0.1878817	114	0.89772728	183	0.16694168E+01
46	0.19423554	115	0.90909092	184	0.16780523E+01
47	0.20071996	116	0.92045456	185	0.16865541E+01
48	0.20733766	117	0.93181819	186	0.16949244E+01
49	0.21409137	118	0.94318183	187	0.17031652E+01
50	0.22098389	119	0.95454547	188	0.17112785E+01
51	0.22801807	120	0.96590910	189	0.17192662E+01
52	0.23519683	121	0.97727274	190	0.17271303E+01
53	0.24252313	122	0.98863638	191	0.17348727E+01
54	0.25000001	123	1.00000001	192	0.17500000E+01
55	0.26512729	124	0.10113636E+01	193	0.17574769E+01
56	0.27286971	125	0.10227273E+01	194	0.17648032E+01
57	0.28073383	126	0.10340909E+01	195	0.17719819E+01
58	0.28872154	127	0.10454546E+01	196	0.17790161E+01
59	0.29683480	128	0.10568182E+01	197	0.17859086E+01
60	0.30507557	129	0.10681818E+01	198	0.17926623E+01
61	0.31344587	130	0.10795454E+01	199	0.17992800E+01
62	0.32194772	131	0.10909091E+01	200	0.18057645E+01
63	0.33058319	132	0.11022727E+01	201	0.18121183E+01
64	0.33935439	133	0.11136364E+01	202	0.18183442E+01
65	0.34826345	134	0.11250000E+01	203	0.18244447E+01
66	0.35731253	135	0.11363636E+01	204	0.18304223E+01
67	0.36650383	136	0.11477273E+01	205	0.18362795E+01
68	0.37583960	137	0.11590909E+01	206	0.18420188E+01
69	0.38532209	138	0.11704545E+01	207	0.18476425E+01
70	0.39495363	139	0.11818182E+01	208	0.18531530E+01
71	0.40473654	140	0.11931818E+01	209	0.18585524E+01
72	0.41467321	141	0.12045454E+01	210	0.18638432E+01
73	0.42476605	142	0.12159091E+01	211	0.18690273E+01
74	0.43501753	143	0.12272727E+01	212	0.18741071E+01
75	0.44543013	144	0.12386364E+01	213	0.18790846E+01
76	0.45600638	145	0.12500000E+01	214	0.18839618E+01
77	0.46674886	146	0.12613636E+01	215	0.18887408E+01
78	0.47766018	147	0.12727273E+01	216	0.18934236E+01
79	0.48874299	148	0.12840909E+01	217	0.18980120E+01
80	0.50000001	149	0.12954545E+01	218	0.19025081E+01
81	0.52272727	150	0.13068182E+01	219	0.19069136E+01
82	0.53409091	151	0.13181818E+01	220	0.19112304E+01
83	0.54545455	152	0.13295454E+01	221	0.19154602E+01
84	0.55681818	153	0.13409091E+01	222	0.19196049E+01
85	0.56818182	154	0.13522727E+01	223	0.19236661E+01
86	0.57954546	155	0.13636364E+01	224	0.19276455E+01
87	0.59090909	156	0.13750000E+01	225	0.19315447E+01
88	0.60227273	157	0.13863636E+01	226	0.19353654E+01
89	0.61363637	158	0.13977273E+01	227	0.19391092E+01
90	0.62500001	159	0.14090909E+01	228	0.19427776E+01
91	0.63636364	160	0.14204545E+01	229	0.19463721E+01
92	0.65909091	161	0.14318182E+01	230	0.19498942E+01
93	0.65909091	162	0.14431818E+01	231	0.19533454E+01
94	0.67045455	163	0.14545455E+01	232	0.19567271E+01
95	0.68181819	164	0.14659091E+01	233	0.19600407E+01
96	0.69318182	165	0.14772727E+01	234	0.19664690E+01
97	0.70454546	166	0.15000000E+01	235	0.19664690E+01
98	0.71590910	167	0.15112570E+01	236	0.19695864E+01
99	0.72727273	168	0.15223398E+01	237	0.19726410E+01
100	0.73863637	169	0.15332511E+01	238	0.19756341E+01
101	0.75000000	170	0.15439936E+01	239	0.19785669E+01
102	0.76136364	171	0.15545699E+01	240	0.19785669E+01
103	0.77272728	172	0.15649825E+01	241	0.19842565E+01
104	0.78409092	173	0.15752339E+01	242	0.19870157E+01

243 0.19897192E+01

244 0.19923684E+01

245 0.19949642E+01

B.2.3 Interpolazione Gnuplot

Questo è un estratto di uno dei codici utilizzato per realizzare i grafici relativi alle componenti della resistenza di un flusso turbolento. Questa parte è quella di maggiore interesse, in quanto è quella che riporta il fitting delle curve della resistenza totale, ovvero il procedimento utilizzato per calcolare lo sforzo d'attrito a parete.

Codici_simulazione/sforzi_tot.gnu

```

1 #####
2
3 #INIZIO PARTE DEL FITTING
4
5 # Definisco le rette (sforzo totale in funzione della distanza da centro canale)
   per i vari dati
6
7 retta0(y) = -a0*y
8 retta3(y) = -a3*y
9 retta5(y) = -a5*y
10 retta6(y) = -a6*y
11 retta7(y) = -a7*y
12 retta9(y) = -a9*y
13 retta10(y) = -a10*y
14 retta12(y) = -a12*y
15 retta15(y) = -a15*y
16 retta18(y) = -a18*y
17 retta20(y) = -a20*y
18 retta22(y) = -a22*y
19
20 #viscosità:
21 mi = 3.18e-4
22
23 # numero piani
24 Npiani = 245
25
26 #Piani esclusi a parete:
27 Ntogli = 20
28
29 #####Inizio sequenza di fitting
30
31 #fitting H0
32 fit retta0(x) './Dati/H0/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi) every
   ::Ntogli::(Npiani-Ntogli) via a0
33
34 # disegno in unità di parete, definizione di utau per H0
35 utauH0 = a0**0.5
36 retauH0= utauH0/mi
37
38 #fitting H3
39 fit retta3(x) './Dati/H3/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi) every
   ::Ntogli::(Npiani-Ntogli) via a3
40
41 # disegno in unità di parete, definizione di utau per H3
42 utauH3 = a3**0.5
43 retauH3= utauH3/mi
44
45 #fitting H5
46 fit retta5(x) './Dati/H5/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi) every
   ::Ntogli::(Npiani-Ntogli) via a5
47
48 # disegno in unità di parete, definizione di utau per H5
49 utauH5 = a5**0.5
50 retauH5= utauH5/mi
51

```

```

52 #fitting H6
53 fit retta6(x) './Dati/H6/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi) every
    ::Ntogli::(Npiani-Ntogli) via a6
54
55 # disegno in unità di parete, definizione di utau per H6
56 utauH6 = a6**0.5
57 retauH6= utauH6/mi
58
59 #fitting H7,5
60 fit retta7(x) './Dati/H7_5/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a7
61
62 # disegno in unità di parete, definizione di utau per H7,5
63 utauH7 = a7**0.5
64 retauH7= utauH7/mi
65
66 #fitting H9
67 fit retta9(x) './Dati/H9/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi) every
    ::Ntogli::(Npiani-Ntogli) via a9
68
69 # disegno in unità di parete, definizione di utau per H9
70 utauH9 = a9**0.5
71 retauH9= utauH9/mi
72
73 #fitting H10
74 fit retta10(x) './Dati/H10/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a10
75
76 # disegno in unità di parete, definizione di utau per H10
77 utauH10 = a10**0.5
78 retauH10= utauH10/mi
79
80 #fitting H12
81 fit retta12(x) './Dati/H12/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a12
82
83 # disegno in unità di parete, definizione di utau per H12
84 utauH12 = a12**0.5
85 retauH12= utauH12/mi
86
87 #fitting H15
88 fit retta15(x) './Dati/H15/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a15
89
90 # disegno in unità di parete, definizione di utau per H15
91 utauH15 = a15**0.5
92 retauH15= utauH15/mi
93
94 #fitting H18
95 fit retta18(x) './Dati/H18/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a18
96
97 # disegno in unità di parete, definizione di utau per H18
98 utauH18 = a18**0.5
99 retauH18= utauH18/mi
100
101 #fitting H20
102 fit retta20(x) './Dati/H20/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a20
103
104 # disegno in unità di parete, definizione di utau per H0
105 utauH20 = a20**0.5
106 retauH20= utauH20/mi
107
108 #fitting H22,5
109 fit retta22(x) './Dati/H22_5/umedia_fin_v2_ureynolds.txt' u ($1-1):(-$5+$3*mi)
    every::Ntogli::(Npiani-Ntogli) via a22
110
111 # disegno in unità di parete, definizione di utau per H22,5

```

```
112 utauH22 = a22**0.5
113 retauH22= utauH22/mi
114
115 #####Fine sequenza di fitting
```

B.3 Submit Post processing delle simulazioni

Questo codice serve per poter mettere in coda di esecuzione il postprocessing di una simulazione, per farlo poi eseguire sul sistema di calcolo utilizzato. In questo caso si riporta il codice per eseguire il postprocessing della configurazione *H7*. A differenza del codice per l'esecuzione delle simulazioni (vedi A.6.2) qui non ci sono differenze vere nei codici, se non nelle opzioni di personalizzazione. Il nome del file può essere qualsiasi, mentre la sua estensione deve essere *.q*. Inoltre si invia al sistema di gestione della coda della macchina con `sbatch nome.q`.

Questo codice è molto interessante, non solo perché ci mostra quali sono i parametri da fornire per l'utilizzo del sistema di calcolo *HACTAR*, ma anche perché contiene al suo interno tutti i comandi necessari per l'esecuzione del postprocessing di una simulazione su *OpenFOAM*.

Codici_simulazione/PoProc_UReyMed_h7_5.txt

```
1 #!/bin/bash
2 #SBATCH --job-name=PPR_h7_5
3 # le due righe servono per mandare email, modificare guardando la guida
4 #SBATCH --mail-type=ALL
5 #SBATCH --mail-user=s230951@studenti.polito.it
6 #SBATCH --partition=global
7 #SBATCH --time=30:00:00
8 #SBATCH --nodes=1
9 #SBATCH --ntasks-per-node=2
10 #SBATCH --output=PPR_h7_5_%j.log
11 #SBATCH --mem-per-cpu=4096M
12
13 module load openfoam/5.0
14 source /share/apps/openfoam/5.0/OpenFOAM-5.0/etc/bashrc
15 cp system/controlDictReynolds system/controlDict
16 pisoFoam -postProcess
```

Bibliografia

- [1] Y.F. Fu, C.Q. Yuan e X.Q. Baiab. «Marine drag reduction of shark skin inspired riblet surfaces. Part I: Biosurface and Biotribology». In: *ScienceDirect* 3 (1 2017), pp. 11–24. URL: <https://doi.org/10.1016/j.bsbt.2017.02.001>.
- [2] Renzo Arina. *Fondamenti di Aerodinamica*. Vol. 2. Teoria del flusso viscoso incompressibile. Torino: Levrotto&Bella, 2015.
- [3] Gianfranco Chiocchia e Massimo Germano. *Termofluidodinamica*. Torino: Levrotto&Bella, 2012.
- [4] D.J. Tritton. *Physical Fluid Dynamics*. Second Edition. Oxford: Oxford Science Publications, 1988.
- [5] Frank M. White. *Viscous Fluid Flow*. Third Edition. New York: McGraw-Hill International Edition, 2006.
- [6] Stephen B. Pope. *Turbulent Flows*. Cambridge: Cambridge University Press, 2000.
- [7] H. Tennekes e J.L. Lumley. *A First Course in Turbulence*. Cambridge, Massachusetts e London, England: The MIT Press, 1972.
- [8] J. Kim, R.D. Moser e N.N. Mansour. «Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$ ». In: *Physic of fluids* 11.4 (1999), pp. 943–945.
- [9] T.C. Corke e F.O. Thomas. «Active and Passive Turbulent Boundary-Layer Drag Reduction». In: *AIAA Journal* 56.10 (2018), pp. 3835–3847. URL: <https://DOI:10.2514/1.J056949>.
- [10] J. Kim. «Physics and control of wall turbulence for drag reduction». In: *Philosophical transactions of the royal society* 369 (2011), pp. 1396–1411. URL: <https://doi:10.1098/rsta.2010.0360>.
- [11] W. Li. *Experimental Investigation of Passive and Active Friction Drag Reduction*. Dissertation.
- [12] P. Ricco e M. Quadrio. *Wall-oscillation conditions for drag reduction in turbulent channel flow*. Paper. URL: <https://arXiv:0802.0114v1>.
- [13] C. Viotti, M. Quadrio e P. Luchini. «Streamwise oscillation of spanwise velocity at the wall of a channel for turbulent drag reduction». In: *Physic of fluids* 21 (2009). URL: <https://doi:10.1063/1.3266945>.
- [14] S. C. Schlanderer, N. Hutchins e R. D. Sandberg. «The Effect of Wall Normal Actuation on a Turbulent Boundary Layer». In: *Flow Turbulence Combust* 99 (2017). A cura di Springer Science+Business Media, pp. 807–821. URL: <https://doi.org/10.1007/s10494-017-9868-0>.

- [15] I. Arenas et al. «Comparison between super-hydrophobic, liquid infused and rough surfaces: a direct numerical simulation study». In: *J. Fluid Mech.* 869 (2019). A cura di Cambridge University Press, pp. 500–525. URL: <https://doi:10.1017/jfm.2019.222>.
- [16] J. Seo, R. Garcia-Mayoral e A. Mani. «Turbulent flows over superhydrophobic surfaces: flow-induced capillary waves, and robustness of air-water interfaces». In: *J. Fluid Mech.* 835 (2018). A cura di Cambridge University Press, pp. 45–85. URL: <https://doi:10.1017/jfm.2017.733>.
- [17] W. Abu Rowin, J. Hou e S. Ghaemia. «Inner and outer layer turbulence over a superhydrophobic surface with low roughness level at low Reynolds number». In: *Physics of Fluids* 9.29 (2017).
- [18] L. Sirovich e S. Karlsson. «Turbulent drag reduction by passive mechanisms». In: *Nature* 388 (1997). A cura di Macmillan Publishers, pp. 753–755. URL: <https://DOI:10.1038/41966>.
- [19] A. Silvestri et al. «Mechanism of sweep event attenuation using micro-cavities in a turbulent boundary layer». In: *American Institute of Physics* 30 (5 2018). URL: <https://DOI:10.1063/1.5026130>.
- [20] M. E. Rosti, L. Brandt e A. Pinelli. «Turbulent channel flow over an anisotropic porous wall - Drag increase and reduction». In: *Journal of Fluid Mechanics* 842 (2018), pp. 381–394. URL: <https://DOI:10.1017/jfm.2018.152>.
- [21] D.W. Bechert e W. Hage. «Drag reduction with riblets in nature and engineering». In: *Transactions on State of the Art in Science and Engineering* 4 (2006). A cura di WIT Press, pp. 457–469. URL: <https://doi:10.2495/1-84564-095-0/5g>.
- [22] R. Garcia-Mayoral e J. Jimenez. «Drag reduction by riblets». In: *Philosophical Transactions of the Royal Society A* 369 (2011), 1412–1427. URL: <https://doi:10.1098/rsta.2010.0359>.
- [23] A. Baron, M. Quadrio e L. Vigevano. «On the boundary layer/riblets interaction mechanisms and the prediction of turbulent drag reduction». In: *J. Heat and Fluid Flow* 14.4 (1993). A cura di Butterworth-Heinemann, pp. 324–332.
- [24] S. Di Giorgio et al. *On the relationship between drag modification and vertical velocity fluctuations in flow over riblets*. Symposium. Southampton, UK. URL: <https://arXiv:0802.0114v1>.
- [25] S. Raayai-Ardakani e G. H. McKinley. «Geometric Optimization of Riblet-Textured Surfaces for Drag Reduction in Laminar Boundary Layer Flows». In: *Physics of Fluids* (2019). URL: <https://DOI:10.1063/1.5090881>.
- [26] J. H. Ferziger e M. Peric. *Computational Methods for Fluid Dynamics*. Third, rev. Edition. Berlin: Springer, 2002.

Alla fine di questo lavoro di tesi, vorrei ringraziare tutti coloro che lo hanno reso possibile e mi hanno sostenuto in questo lungo percorso, dal suo inizio, fino a questo traguardo così importante.

Un ringraziamento doveroso va:

Al prof.re Michele Iovieno

Il quale non solo come professore, ma anche come incredibile esempio di quello che deve essere il ruolo dell'ingegnere, mi ha trasmesso una passione ancora più grande per lo studio della fluidodinamica, in tutti i suoi aspetti, seguendomi e consigliandomi in ogni momento, nel compimento di questo lavoro di tesi, mettendo a mia disposizione la sua sconfinata conoscenza, non solo della materia, ma di tutto ciò di cui avessi bisogno.

All'ing. Irene Gned

Per avermi fornito il suo aiuto e le sue competenze, per risolvere i numerosi problemi emersi durante questo percorso, in merito al funzionamento del programma di simulazione, e seppur non direttamente coinvolta nella tesi, avermi spronato a dare il meglio e concludere nel migliore dei modi questo percorso.

hpc@polito

Ringrazio anche, per le "Risorse di calcolo fornite hpc@polito, progetto di Academic Computing del Dipartimento di Automatica e Informatica presso il Politecnico di Torino (<http://www.hpc.polito.it>)".

I miei ringraziamenti personali,

Sono poi rivolti a coloro che mi hanno supportato nello studio, ed a cui dedico questo lavoro di tesi.

Ai miei genitori

Che mi hanno sempre sostenuto nello studio, hanno sempre creduto in me, anche nei momenti in cui non credevo fosse possibile raggiungere il traguardo che mi ero prefissato, e mi hanno sempre incentivato a seguire il mio percorso, aiutandomi il più possibile a dare sempre il meglio. E sono sempre stati pronti a sostenermi nei momenti di difficoltà e a gioire con me in quelli felici, non facendomi mai mancare la loro presenza al mio fianco in questo lungo e difficile viaggio, che anche grazie a loro sono riuscito finalmente a terminare.

A mio fratello Donato

Che, sia a distanza che di persona, mi è sempre stato vicino, è sempre stato pronto a sostenermi a sua volta, per permettermi di andare avanti anche nei momenti di difficoltà, e mi ha fatto sentire fiero ed orgoglioso del percorso intrapreso e del traguardo raggiunto.

Alla mia fidanzata Mafalda

Che mi è sempre stata accanto, sia fisicamente che a distanza, mi ha aiutato nei momenti difficili, sostenuto in questo lungo studio, e reso un pò più facile e sereno questa esperienza, anche grazie ai suoi consigli. Soprattutto nelle parti più dure di questo percorso, quando la paura mi assaliva, sei sempre stata pronta ad aiutarmi a tirarmi su e proseguire. Ti amo.

Ai miei nonni, Donato e Salvatore

Che sono stati i miei modelli di ispirazione in questo lungo percorso, che sono stati sempre accanto a me, seppur non fisicamente, in questa lunga e difficile sfida, dandomi coraggio ogni volta che ne avessi avuto bisogno; per avermi protetto in ogni momento della mia vita, consigliatomi, dall'alto, in ogni mia scelta, ed essere sempre la luce da seguire sulla mia strada.

Alle mie nonne, Giovanna e Tecla

Le quali sono sempre state felici dei miei successi, fornendomi sempre il loro supporto, gioendo con me dei traguardi raggiunti, e per avermi dato la forza di andare avanti, sempre con affetto, nei momenti di solitudine, e soprattutto per avermi aiutato, nella mia crescita, in tutta la mia vita ed in ogni momento.

A tutti i miei familiari

Che attraverso i loro pensieri, mi hanno sempre incentivato a perseguire il mio sogno e la mia strada, hanno gioito per ogni mio traguardo raggiunto, festeggiando con me il raggiungimento di questo obiettivo. Un sentito ringraziamento va anche ai miei suoceri ed ai miei cognati.

A Beppe e Rina

Per essere stati, più che conoscenti, quasi una seconda famiglia, avermi fatto sentire sempre amato ed essere sempre stati pronti ad aiutarmi ogni volta ne avessi bisogno, sempre con il sorriso, e pronti a condividere una parola o un ricordo con me, laddove se ne presentasse l'occasione. Grazie.

Ai miei amici e colleghi universitari

I quali, più che semplici colleghi, sono stati amici sinceri, con cui lavorare insieme, condividere preoccupazioni, paure, successi, passioni comuni e tanto altro, con i quali, in ogni istante di questa lunga carriera universitaria, hanno reso più piacevole questo viaggio, condiviso insieme, che, grazie anche ai loro suggerimenti ed il loro aiuto, ho finalmente terminato.

Infine, ringraziamento va anche a Michettore, Bianchino e Rossano, per avermi fatto compagnia nelle lunghe giornate di studio, soprattutto quelle estive.